One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

ADV7481 Register Control Manual

INTRODUCTION

This reference manual describes the I²C control registers for the ADV7481. The ADV7481 is an integrated video decoder and combined HDMI®/MHL® receiver. It is targeted at connectivity enabled head units requiring a wired, uncompressed digital audio/video link from smartphones and other consumer electronics devices to support streaming and integration of cloud-based multimedia content and applications into an automotive infotainment system.

The Register Maps section of this reference manual provides detailed register tables for the ADV7481 register maps. The Register Bit Descriptions section provides details about the controls present in each register.

LEGAL TERMS AND CONDITIONS

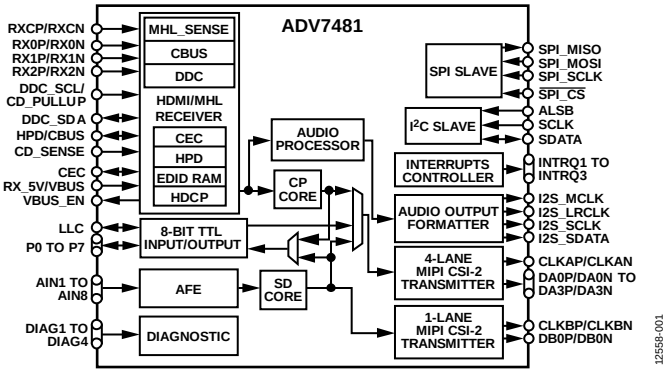

Figure 1. ADV7481 Block Diagram

TABLE OF CONTENTS

REVISION HISTORY

12/14—Revision 0: Initial Version

1 REGISTER MAPS

1.1 IO MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x33	power down controls	rw	-	rx_en	rx_pdn	-	-	xtal_pdn	core_pdn	master_pdn
0x01	0xB6	io_reg_01	rw	pwrdn2b	pwrdnb	prog_xtal_freq[13]	prog_xtal_freq[12]	prog_xtal_freq[11]	prog_xtal_freq[10]	prog_xtal_freq[9]	prog_xtal_freq[8]
0x02	0x00	io_reg_02	rw	prog_xtal_freq[7]	prog_xtal_freq[6]	prog_xtal_freq[5]	prog_xtal_freq[4]	prog_xtal_freq[3]	prog_xtal_freq[2]	prog_xtal_freq[1]	prog_xtal_freq[0]
0x03	0x06	datapath cntrl	rw	avcode_insert_en	cp_v_freq[2]	cp_v_freq[1]	cp_v_freq[0]	cp_op_656_range	cp_data_blank_en	cp_free_run_en	cp_bypass
0x04	0x02	io_reg_04	rw	cp_force_freerun_ch1	-	cp_repl_av_code	-	cp_alt_gamma	cp_alt_data_sat	cp_rgb_out	-
0x05	0x40	vid std	rw	cp_vid_std[7]	cp_vid_std[6]	cp_vid_std[5]	cp_vid_std[4]	cp_vid_std[3]	cp_vid_std[2]	cp_vid_std[1]	cp_vid_std[0]
0x06	0x11	io_reg_06	rw	-	pix_in_reverse	pix_in_input_as_ddr	pix_in_split_avcode	-	-	-	pix_in_keep_avcodes_in_two_ch
0x0A	0x00	video mute control_1	rw	-	vmute	-	-	-	-	-	-
0x0C	0x00	io_reg_0c	rw	llc_dll_en	llc_dll_double	llc_dll_mux	llc_dll_phase[4]	llc_dll_phase[3]	llc_dll_phase[2]	llc_dll_phase[1]	llc_dll_phase[0]
0x0D	0xAA	pad drive strengths	rw	drv_pixel_pads[1]	drv_pixel_pads[0]	drv_audio_pads[1]	drv_audio_pads[0]	dr_str_spi[1]	dr_str_spi[0]	drv_int_i2c_csb_pads[1]	drv_int_i2c_csb_pads[0]
0x0E	0xFF	pad controls	rw	tri_llc	tri_pix	tri_aud	tri_spi	-	pdn_pix	pdn_aud	pdn_spi
0x0F	0x09	pad filter controls	rw	-	-	-	-	pdn_vbus_en	-	xtal_freq_sel[1]	xtal_freq_sel[0]
0x10	0x00	io_reg_10	rw	csi4_en	csi1_en	pix_out_en	sd_thru_pix_out	csi4_in_sel[1]	csi4_in_sel[0]	-	-
0x11	0x08	io_reg_11	rw	-	-	-	-	-	sd_ddr_out	-	-
0x12	0xF0	io_reg_12	rw	cp_inp_color_space[3]	cp_inp_color_space[2]	cp_inp_color_space[1]	cp_inp_color_space[0]	-	cp_out_mode[1]	cp_out_mode[0]	cp_out_10b
0x17	0x80	io_reg_17	rw	br_dith_ccir601_b	br_dith_yuv422_mode	br_dither_mode	rnd_dither_en	br_dither_en	-	-	-
0x18	0x6D	tri 1 slice ctrl	rw	-	diag1_slicers_pwrdn	diag1_bilevel_en	diag1_upper_slice_level[2]	diag1_upper_slice_level[1]	diag1_upper_slice_level[0]	-	-
0x19	0x6D	tri 2 slice ctrl	rw	-	diag2_slicers_pwrdn	diag2_bilevel_en	diag2_upper_slice_level[2]	diag2_upper_slice_level[1]	diag2_upper_slice_level[0]	-	-
0x1A	0x6D	tri 3 slice ctrl	rw	-	diag3_slicers_pwrdn	diag3_bilevel_en	diag3_upper_slice_level[2]	diag3_upper_slice_level[1]	diag3_upper_slice_level[0]	-	-
0x1B	0x6D	tri 4 slice ctrl	rw	-	diag4_slicers_pwrdn	diag4_bilevel_en	diag4_upper_slice_level[2]	diag4_upper_slice_level[1]	diag4_upper_slice_level[0]	-	-
0x1D	0x78	pad controls 1	rw	pdn_int1	pdn_int2	pdn_int3	inv_llc	drv_llc_pad[1]	drv_llc_pad[0]	-	-
0x3F	0x00	int raw status	r	-	-	-	int_cec_st	int_hdmi_st	intrq3_raw	intrq2_raw	intrq_raw
0x40	0x00	int1_configuration	rw	intrq_dur_sel[1]	intrq_dur_sel[0]	-	store_unmasked_irqs	en_umask_raw_intrq	mpu_stim_intrq	intrq_op_sel[1]	intrq_op_sel[0]
0x41	0x20	int2_configuration	rw	intrq2_dur_sel[1]	intrq2_dur_sel[0]	cp_lock_unlock_edge_sel	-	en_umask_raw_intrq2	int2_en	intrq2_op_sel[1]	intrq2_op_sel[0]
0x43	0x00	datapath raw status	r	cp_lock_cp_raw	cp_unlock_cp_raw	vmute_request_hdmi_raw	-	-	-	mpu_stim_intrq_raw	int_sd_raw
0x44	0x00	datapath intstatus	r	cp_lock_cp_st	cp_unlock_cp_st	vmute_request_hdmi_st	-	-	-	mpu_stim_intrq_st	int_sd_st

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x45	0x00	datapath int clr	sc	cp_lock_cp_clr	cp_unlock_cp_clr	vmute_request_hdmi_clr	-	-	-	mpu_stim_intrq_clr	int_sd_clr
0x46	0x00	datapath int2 maskb	rw	cp_lock_cp_mb2	cp_unlock_cp_mb2	vmute_request_hdmi_mb2	-	-	-	mpu_stim_intrq_mb2	int_sd_mb2
0x47	0x00	datapath int maskb	rw	cp_lock_cp_mb1	cp_unlock_cp_mb1	vmute_request_hdmi_mb1	-	-	-	mpu_stim_intrq_mb1	int_sd_mb1
0x49	0x00	cec_status1_raw	r	-	-	cec_rx_rdy2_raw	cec_rx_rdy1_raw	cec_rx_rdy0_raw	cec_tx_retry_timeout_raw	cec_tx_arbitration_lost_raw	cec_tx_ready_raw
0x4A	0x00	cec_status1_int_status	r	-	-	cec_rx_rdy2_st	cec_rx_rdy1_st	cec_rx_rdy0_st	cec_tx_retry_timeout_st	cec_tx_arbitration_lost_st	cec_tx_ready_st
0x4B	0x00	cec_status1_int_clear	sc	-	-	cec_rx_rdy2_clr	cec_rx_rdy1_clr	cec_rx_rdy0_clr	cec_tx_retry_timeout_clr	cec_tx_arbitration_lost_clr	cec_tx_ready_clr
0x4C	0x00	cec_status1_int2_maskb	rw	-	-	cec_rx_rdy2_mb2	cec_rx_rdy1_mb2	cec_rx_rdy0_mb2	cec_tx_retry_timeout_mb2	cec_tx_arbitration_lost_mb2	cec_tx_ready_mb2
0x4D	0x00	cec_status1_int1_maskb	rw	-	-	cec_rx_rdy2_mb1	cec_rx_rdy1_mb1	cec_rx_rdy0_mb1	cec_tx_retry_timeout_mb1	cec_tx_arbitration_lost_mb1	cec_tx_ready_mb1
0x4E	0x00	cec_raw_status2	r	cec_interrupt_byte[7]	cec_interrupt_byte[6]	cec_interrupt_byte[5]	cec_interrupt_byte[4]	cec_interrupt_byte[3]	cec_interrupt_byte[2]	cec_interrupt_byte[1]	cec_interrupt_byte[0]
0x4F	0x00	cec_interrupt_status2	r	cec_interrupt_byte_st[7]	cec_interrupt_byte_st[6]	cec_interrupt_byte_st[5]	cec_interrupt_byte_st[4]	cec_interrupt_byte_st[3]	cec_interrupt_byte_st[2]	cec_interrupt_byte_st[1]	cec_interrupt_byte_st[0]
0x50	0x00	cec_interrupt_clear2	sc	cec_interrupt_byte_clr[7]	cec_interrupt_byte_clr[6]	cec_interrupt_byte_clr[5]	cec_interrupt_byte_clr[4]	cec_interrupt_byte_clr[3]	cec_interrupt_byte_clr[2]	cec_interrupt_byte_clr[1]	cec_interrupt_byte_clr[0]
0x51	0x00	cec_interrupt2_maskb	rw	cec_interrupt_byte_mb2[7]	cec_interrupt_byte_mb2[6]	cec_interrupt_byte_mb2[5]	cec_interrupt_byte_mb2[4]	cec_interrupt_byte_mb2[3]	cec_interrupt_byte_mb2[2]	cec_interrupt_byte_mb2[1]	cec_interrupt_byte_mb2[0]
0x52	0x00	cec_interrupt_maskb	rw	cec_interrupt_byte_mb1[7]	cec_interrupt_byte_mb1[6]	cec_interrupt_byte_mb1[5]	cec_interrupt_byte_mb1[4]	cec_interrupt_byte_mb1[3]	cec_interrupt_byte_mb1[2]	cec_interrupt_byte_mb1[1]	cec_interrupt_byte_mb1[0]
0x53	0x00	tri slice_raw_status	r	tri_slice[7]	tri_slice[6]	tri_slice[5]	tri_slice[4]	tri_slice[3]	tri_slice[2]	tri_slice[1]	tri_slice[0]
0x54	0x00	tri slice_interrupt_status	r	tri_slice_st[7]	tri_slice_st[6]	tri_slice_st[5]	tri_slice_st[4]	tri_slice_st[3]	tri_slice_st[2]	tri_slice_st[1]	tri_slice_st[0]
0x55	0x00	tri slice_interrupt_clear	sc	tri_slice_clr[7]	tri_slice_clr[6]	tri_slice_clr[5]	tri_slice_clr[4]	tri_slice_clr[3]	tri_slice_clr[2]	tri_slice_clr[1]	tri_slice_clr[0]
0x56	0x00	tri slice_interrupt2_maskb	rw	tri_slice_mb2[7]	tri_slice_mb2[6]	tri_slice_mb2[5]	tri_slice_mb2[4]	tri_slice_mb2[3]	tri_slice_mb2[2]	tri_slice_mb2[1]	tri_slice_mb2[0]
0x57	0x00	tri slice_interrupt_maskb	rw	tri_slice_mb1[7]	tri_slice_mb1[6]	tri_slice_mb1[5]	tri_slice_mb1[4]	tri_slice_mb1[3]	tri_slice_mb1[2]	tri_slice_mb1[1]	tri_slice_mb1[0]
0x67	0x00	hdmi lvl raw status 1	r	isrc2_pckt_raw	isrc1_pckt_raw	acp_pckt_raw	vs_info_raw	ms_info_raw	spd_info_raw	audio_info_raw	avi_info_raw
0x68	0x00	hdmi lvl int status 1	r	isrc2_pckt_st	isrc1_pckt_st	acp_pckt_st	vs_info_st	ms_info_st	spd_info_st	audio_info_st	avi_info_st
0x69	0x00	hdmi lvl int clr 1	sc	isrc2_pckt_clr	isrc1_pckt_clr	acp_pckt_clr	vs_info_clr	ms_info_clr	spd_info_clr	audio_info_clr	avi_info_clr

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x6A	0x00	hdmi lvl int2 maskb 1	rw	isrc2_pckt_mb2	isrc1_pckt_mb2	acp_pckt_mb2	vs_info_mb2	ms_info_mb2	spd_info_mb2	audio_info_mb2	avi_info_mb2
0x6B	0x00	hdmi lvl int maskb 1	rw	isrc2_pckt_mb1	isrc1_pckt_mb1	acp_pckt_mb1	vs_info_mb1	ms_info_mb1	spd_info_mb1	audio_info_mb1	avi_info_mb1
0x6C	0x00	hdmi lvl raw status 2	r	cs_data_valid_raw	internal_mute_raw	av_mute_raw	audio_ch_md_raw	hdmi_mode_raw	gen_ctl_pckt_raw	audio_c_pckt_raw	gamut_mdata_raw
0x6D	0x00	hdmi lvl int status 2	r	cs_data_valid_st	internal_mute_st	av_mute_st	audio_ch_md_st	hdmi_mode_st	gen_ctl_pckt_st	audio_c_pckt_st	gamut_mdata_st
0x6E	0x00	hdmi lvl int clr 2	sc	cs_data_valid_clr	internal_mute_clr	av_mute_clr	audio_ch_md_clr	hdmi_mode_clr	gen_ctl_pckt_clr	audio_c_pckt_clr	gamut_mdata_clr
0x6F	0x00	hdmi lvl int2 maskb 2	rw	cs_data_valid_mb2	internal_mute_mb2	av_mute_mb2	audio_ch_md_mb2	hdmi_mode_mb2	gen_ctl_pckt_mb2	audio_c_pckt_mb2	gamut_mdata_mb2
0x70	0x00	hdmi lvl int maskb 2	rw	cs_data_valid_mb1	internal_mute_mb1	av_mute_mb1	audio_ch_md_mb1	hdmi_mode_mb1	gen_ctl_pckt_mb1	audio_c_pckt_mb1	gamut_mdata_mb1
0x71	0x00	hdmi lvl raw status 3	r	tmdspll_lck_a_raw	cable_det_a_raw	hdmi_encrpt_a_raw	-	tmds_clk_a_raw	video_3d_raw	v_locked_raw	de_regen_lck_raw
0x72	0x00	hdmi lvl int status 3	r	tmdspll_lck_a_st	cable_det_a_st	hdmi_encrpt_a_st	-	tmds_clk_a_st	video_3d_st	v_locked_st	de_regen_lck_st
0x73	0x00	hdmi lvl int clr 3	sc	tmdspll_lck_a_clr	cable_det_a_clr	hdmi_encrpt_a_clr	-	tmds_clk_a_clr	video_3d_clr	v_locked_clr	de_regen_lck_clr
0x74	0x00	hdmi lvl int2 maskb 3	rw	tmdspll_lck_a_mb2	cable_det_a_mb2	hdmi_encrpt_a_mb2	-	tmds_clk_a_mb2	video_3d_mb2	v_locked_mb2	de_regen_lck_mb2
0x75	0x00	hdmi lvl int maskb 3	rw	tmdspll_lck_a_mb1	cable_det_a_mb1	hdmi_encrpt_a_mb1	-	tmds_clk_a_mb1	video_3d_mb1	v_locked_mb1	de_regen_lck_mb1
0x80	0x00	hdmi edg raw status 1	r	new_isrc2_pckt_raw	new_isrc1_pckt_raw	new_acp_pckt_raw	new_vs_info_raw	new_ms_info_raw	new_spd_info_raw	new_audio_info_raw	new_avi_info_raw
0x81	0x00	hdmi edg int status 1	r	new_isrc2_pckt_st	new_isrc1_pckt_st	new_acp_pckt_st	new_vs_info_st	new_ms_info_st	new_spd_info_st	new_audio_info_st	new_avi_info_st
0x82	0x00	hdmi edg int clr 1	sc	new_isrc2_pckt_clr	new_isrc1_pckt_clr	new_acp_pckt_clr	new_vs_info_clr	new_ms_info_clr	new_spd_info_clr	new_audio_info_clr	new_avi_info_clr
0x83	0x00	hdmi edg int2 maskb 1	rw	new_isrc2_pckt_mb2	new_isrc1_pckt_mb2	new_acp_pckt_mb2	new_vs_info_mb2	new_ms_info_mb2	new_spd_info_mb2	new_audio_info_mb2	new_avi_info_mb2
0x84	0x00	hdmi edg int maskb 1	rw	new_isrc2_pckt_mb1	new_isrc1_pckt_mb1	new_acp_pckt_mb1	new_vs_info_mb1	new_ms_info_mb1	new_spd_info_mb1	new_audio_info_mb1	new_avi_info_mb1
0x85	0x00	hdmi edg raw status 2	r	fifo_near_ovfl_raw	fifo_underflo_raw	fifo_overflo_raw	cts_pass_thrsh_raw	change_n_raw	packet_error_raw	audio_pckt_err_raw	new_gamut_mdata_raw
0x86	0x00	hdmi edg int status 2	r	fifo_near_ovfl_st	fifo_underflo_st	fifo_overflo_st	cts_pass_thrsh_st	change_n_st	packet_error_st	audio_pckt_err_st	new_gamut_mdata_st
0x87	0x00	hdmi edg int clr 2	sc	fifo_near_ovfl_clr	fifo_underflo_clr	fifo_overflo_clr	cts_pass_thrsh_clr	change_n_clr	packet_error_clr	audio_pckt_err_clr	new_gamut_mdata_clr
0x88	0x00	hdmi edg int2 maskb 2	rw	fifo_near_ovfl_mb2	fifo_underflo_mb2	fifo_overflo_mb2	cts_pass_thrsh_mb2	change_n_mb2	packet_error_mb2	audio_pckt_err_mb2	new_gamut_mdata_mb2
0x89	0x00	hdmi edg int maskb 2	rw	fifo_near_ovfl_mb1	fifo_underflo_mb1	fifo_overflo_mb1	cts_pass_thrsh_mb1	change_n_mb1	packet_error_mb1	audio_pckt_err_mb1	new_gamut_mdata_mb1
0x8A	0x00	hdmi edg raw status 3	r	-	vclk_chng_raw	audio_mode_chng_raw	parity_error_raw	new_samp_rt_raw	audio_flt_line_raw	new_tmds_frq_raw	fifo_near_uflo_raw

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x8B	0x00	hdmi edg status 3	r	-	vclk_chng_st	audio_mode_chng_st	parity_error_st	new_samp_rt_st	audio_flt_line_st	new_tmds_frq_st	fifo_near_uflo_st
0x8C	0x00	hdmi edg int clr 3	sc	-	vclk_chng_clr	audio_mode_chng_clr	parity_error_clr	new_samp_rt_clr	audio_flt_line_clr	new_tmds_frq_clr	fifo_near_uflo_clr
0x8D	0x00	hdmi edg int2 maskb 3	rw	-	vclk_chng_mb2	audio_mode_chng_mb2	parity_error_mb2	new_samp_rt_mb2	audio_flt_line_mb2	new_tmds_frq_mb2	fifo_near_uflo_mb2
0x8E	0x00	hdmi edg int maskb 3	rw	-	vclk_chng_mb1	audio_mode_chng_mb1	parity_error_mb1	new_samp_rt_mb1	audio_flt_line_mb1	new_tmds_frq_mb1	fifo_near_uflo_mb1
0x8F	0x00	hdmi edg raw status 4_1	r	ms_inf_cks_err_raw	spd_inf_cks_err_raw	aud_inf_cks_err_raw	avi_inf_cks_err_raw	aksv_update_a_raw	ri_expired_a_raw	-	vs_inf_cks_err_raw
0x90	0x00	hdmi edg status 4_1	r	ms_inf_cks_err_st	spd_inf_cks_err_st	aud_inf_cks_err_st	avi_inf_cks_err_st	aksv_update_a_st	ri_expired_a_st	-	vs_inf_cks_err_st
0x91	0x00	hdmi edg int clr 4_1	sc	ms_inf_cks_err_clr	spd_inf_cks_err_clr	aud_inf_cks_err_clr	avi_inf_cks_err_clr	aksv_update_a_clr	ri_expired_a_clr	-	vs_inf_cks_err_clr
0x92	0x00	hdmi edg int2 maskb 4_1	rw	ms_inf_cks_err_mb2	spd_inf_cks_err_mb2	aud_inf_cks_err_mb2	avi_inf_cks_err_mb2	aksv_update_a_mb2	ri_expired_a_mb2	-	vs_inf_cks_err_mb2
0x93	0x00	hdmi edg int maskb 4_1	rw	ms_inf_cks_err_mb1	spd_inf_cks_err_mb1	aud_inf_cks_err_mb1	avi_inf_cks_err_mb1	aksv_update_a_mb1	ri_expired_a_mb1	-	vs_inf_cks_err_mb1
0xDF	0x00	chip_rev_id_1	r	rd_info[15]	rd_info[14]	rd_info[13]	rd_info[12]	rd_info[11]	rd_info[10]	rd_info[9]	rd_info[8]
0xE0	0x00	chip_rev_id_2	r	rd_info[7]	rd_info[6]	rd_info[5]	rd_info[4]	rd_info[3]	rd_info[2]	rd_info[1]	rd_info[0]
0xF2	0x01	io_reg_f2	rw	-	-	-	-	-	-	-	read_auto_inc_en
0xF3	0x00	i2c slave addr_1	rw	dpll_slave_addr[6]	dpll_slave_addr[5]	dpll_slave_addr[4]	dpll_slave_addr[3]	dpll_slave_addr[2]	dpll_slave_addr[1]	dpll_slave_addr[0]	-
0xF4	0x00	i2c slave addr_2	rw	cp_slave_addr[7]	cp_slave_addr[6]	cp_slave_addr[5]	cp_slave_addr[4]	cp_slave_addr[3]	cp_slave_addr[2]	cp_slave_addr[1]	-
0xF5	0x00	i2c slave addr_3	rw	hdmi_slave_addr[7]	hdmi_slave_addr[6]	hdmi_slave_addr[5]	hdmi_slave_addr[4]	hdmi_slave_addr[3]	hdmi_slave_addr[2]	hdmi_slave_addr[1]	-
0xF6	0x00	i2c slave addr_4	rw	edid_slave_addr[7]	edid_slave_addr[6]	edid_slave_addr[5]	edid_slave_addr[4]	edid_slave_addr[3]	edid_slave_addr[2]	edid_slave_addr[1]	-
0xF7	0x00	i2c slave addr_5	rw	repeater_slave_addr[6]	repeater_slave_addr[5]	repeater_slave_addr[4]	repeater_slave_addr[3]	repeater_slave_addr[2]	repeater_slave_addr[1]	repeater_slave_addr[0]	-
0xF8	0x00	i2c slave addr_6	rw	infoframe_slave_addr[7]	infoframe_slave_addr[6]	infoframe_slave_addr[5]	infoframe_slave_addr[4]	infoframe_slave_addr[3]	infoframe_slave_addr[2]	infoframe_slave_addr[1]	-
0xF9	0x00	i2c slave addr_7	rw	cbus_slave_addr[6]	cbus_slave_addr[5]	cbus_slave_addr[4]	cbus_slave_addr[3]	cbus_slave_addr[2]	cbus_slave_addr[1]	cbus_slave_addr[0]	-
0xFA	0x00	i2c slave addr_8	rw	cec_slave_addr[7]	cec_slave_addr[6]	cec_slave_addr[5]	cec_slave_addr[4]	cec_slave_addr[3]	cec_slave_addr[2]	cec_slave_addr[1]	-
0xFB	0x00	i2c slave addr_9	rw	sd_core_slave_addr[7]	sd_core_slave_addr[6]	sd_core_slave_addr[5]	sd_core_slave_addr[4]	sd_core_slave_addr[3]	sd_core_slave_addr[2]	sd_core_slave_addr[1]	-
0xFC	0x00	i2c slave addr_10	rw	csi1_tx_slave_addr[7]	csi1_tx_slave_addr[6]	csi1_tx_slave_addr[5]	csi1_tx_slave_addr[4]	csi1_tx_slave_addr[3]	csi1_tx_slave_addr[2]	csi1_tx_slave_addr[1]	-
0xFD	0x00	i2c slave addr_11	rw	csi4_tx_slave_addr[7]	csi4_tx_slave_addr[6]	csi4_tx_slave_addr[5]	csi4_tx_slave_addr[4]	csi4_tx_slave_addr[3]	csi4_tx_slave_addr[2]	csi4_tx_slave_addr[1]	-
0xFF	0x00	io_reg_ff	sc	main_reset	-	-	-	-	-	-	-

1.2 HDMI RX MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x00	hdmi_register_00h	rw	hdcp_a0	hdcp_only_mode	-	-	-	-	-	-
0x01	0x00	hdmi_register_01h	rw	-	-	-	-	-	-	-	term_auto
0x03	0x18	hdmi register_03h	rw	-	i2soutmode[1]	i2soutmode[0]	i2sbitwidth[4]	i2sbitwidth[3]	i2sbitwidth[2]	i2sbitwidth[1]	i2sbitwidth[0]
0x04	0x00	hdmi register_04h	r	-	av_mute	hdcp_keys_read	hdcp_key_error	hdcp_ri_expired	-	tmds_pll_locked	audio_pll_locked
0x05	0x00	hdmi_register_05h	r	hdmi_mode	hdmi_content_encrypted	dvi_hsync_polarity	dvi_vsync_polarity	hdmi_pixel_repetition[3]	hdmi_pixel_repetition[2]	hdmi_pixel_repetition[1]	hdmi_pixel_repetition[0]
0x07	0x00	line width_1	r	vert_filter_locked	audio_channel_mode	de_regen_filter_locked	line_width[12]	line_width[11]	line_width[10]	line_width[9]	line_width[8]
0x08	0x00	line width_2	r	line_width[7]	line_width[6]	line_width[5]	line_width[4]	line_width[3]	line_width[2]	line_width[1]	line_width[0]
0x09	0x00	field0 height_1	r	-	-	-	field0_height[12]	field0_height[11]	field0_height[10]	field0_height[9]	field0_height[8]
0x0A	0x00	field0 height_2	r	field0_height[7]	field0_height[6]	field0_height[5]	field0_height[4]	field0_height[3]	field0_height[2]	field0_height[1]	field0_height[0]
0x0B	0x00	field1 height_1	r	deep_color_mode[1]	deep_color_mode[0]	hdmi_interlaced	field1_height[12]	field1_height[11]	field1_height[10]	field1_height[9]	field1_height[8]
0x0C	0x00	field1 height_2	r	field1_height[7]	field1_height[6]	field1_height[5]	field1_height[4]	field1_height[3]	field1_height[2]	field1_height[1]	field1_height[0]
0x0D	0x04	hdmi_register_0dh	rw	-	-	-	-	freqtolerance[3]	freqtolerance[2]	freqtolerance[1]	freqtolerance[0]
0x0F	0x1F	audio mute speed	rw	man_audio_dl_bypass	audio_delay_line_bypass	-	audio_mute_speed[4]	audio_mute_speed[3]	audio_mute_speed[2]	audio_mute_speed[1]	audio_mute_speed[0]
0x10	0x25	hdmi_register_10h	rw	-	-	cts_change_threshold[5]	cts_change_threshold[4]	cts_change_threshold[3]	cts_change_threshold[2]	cts_change_threshold[1]	cts_change_threshold[0]
0x11	0x7D	audio fifo almost full threshold	rw	-	audio_fifo_almost_full_threshold[6]	audio_fifo_almost_full_threshold[5]	audio_fifo_almost_full_threshold[4]	audio_fifo_almost_full_threshold[3]	audio_fifo_almost_full_threshold[2]	audio_fifo_almost_full_threshold[1]	audio_fifo_almost_full_threshold[0]
0x12	0x02	audio fifo almost empty threshold	rw	-	audio_fifo_almost_empty_threshold[6]	audio_fifo_almost_empty_threshold[5]	audio_fifo_almost_empty_threshold[4]	audio_fifo_almost_empty_threshold[3]	audio_fifo_almost_empty_threshold[2]	audio_fifo_almost_empty_threshold[1]	audio_fifo_almost_empty_threshold[0]
0x13	0x7F	audio coast mask	rw	-	ac_msk_vclk_chng	ac_msk_vpll_unlock	-	ac_msk_new_cts	ac_msk_new_n	-	ac_msk_vclk_det
0x14	0x3F	mute mask 21_16	rw	-	-	mt_msk_comprs_aud	mt_msk_aud_mode_chng	-	-	mt_msk_parity_err	mt_msk_vclk_chng
0x15	0xFF	mute mask 15_8	rw	mt_msk_apll_unlock	mt_msk_vpll_unlock	mt_msk_acr_not_det	-	mt_msk_flatline_det	-	mt_msk_fifo_underlfow	mt_msk_fifo_overflow
0x16	0xFF	mute mask 7_0	rw	mt_msk_avmute	mt_msk_not_hdmimode	mt_msk_new_cts	mt_msk_new_n	mt_msk_chmode_chng	mt_msk_apckt_ec_err	-	mt_msk_vclk_det
0x18	0x00	packets detected	r	-	-	-	-	-	-	-	audio_sample_pckt_det
0x1A	0x80	mute_ctrl	rw	-	ignore_parity_err	-	mute_audio	wait_unmute[2]	wait_unmute[1]	wait_unmute[0]	not_auto_unmute
0x1B	0x18	deepcolor_fifo_debug_1	rw	-	-	-	dcfifo_reset_on_lock	dcfifo_kill_not_locked	dcfifo_kill_dis	-	-
0x1C	0x00	deepcolor_fifo_debug_2	r	-	-	-	-	dcfifo_locked	dcfifo_level[2]	dcfifo_level[1]	dcfifo_level[0]
0x1D	0x00	register_1dh	rw	-	pdn_pkt_processor	up_conversion_mode	-	-	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1E	0x00	total_line_width_1	r	-	-	total_line_width[13]	total_line_width[12]	total_line_width[11]	total_line_width[10]	total_line_width[9]	total_line_width[8]
0x1F	0x00	total_line_width_2	r	total_line_width[7]	total_line_width[6]	total_line_width[5]	total_line_width[4]	total_line_width[3]	total_line_width[2]	total_line_width[1]	total_line_width[0]
0x20	0x00	hsync_front_porch_1	r	-	-	-	hsync_front_porch[12]	hsync_front_porch[11]	hsync_front_porch[10]	hsync_front_porch[9]	hsync_front_porch[8]
0x21	0x00	hsync_front_porch_2	r	hsync_front_porch[7]	hsync_front_porch[6]	hsync_front_porch[5]	hsync_front_porch[4]	hsync_front_porch[3]	hsync_front_porch[2]	hsync_front_porch[1]	hsync_front_porch[0]
0x22	0x00	hsync_pulse_width_1	r	-	-	-	hsync_pulse_width[12]	hsync_pulse_width[11]	hsync_pulse_width[10]	hsync_pulse_width[9]	hsync_pulse_width[8]
0x23	0x00	hsync_pulse_width_2	r	hsync_pulse_width[7]	hsync_pulse_width[6]	hsync_pulse_width[5]	hsync_pulse_width[4]	hsync_pulse_width[3]	hsync_pulse_width[2]	hsync_pulse_width[1]	hsync_pulse_width[0]
0x24	0x00	hsync_back_porch_1	r	-	-	-	hsync_back_porch[12]	hsync_back_porch[11]	hsync_back_porch[10]	hsync_back_porch[9]	hsync_back_porch[8]
0x25	0x00	hsync_back_porch_2	r	hsync_back_porch[7]	hsync_back_porch[6]	hsync_back_porch[5]	hsync_back_porch[4]	hsync_back_porch[3]	hsync_back_porch[2]	hsync_back_porch[1]	hsync_back_porch[0]
0x26	0x00	field0_total_height_1	r	-	-	field0_total_height[13]	field0_total_height[12]	field0_total_height[11]	field0_total_height[10]	field0_total_height[9]	field0_total_height[8]
0x27	0x00	field0_total_height_2	r	field0_total_height[7]	field0_total_height[6]	field0_total_height[5]	field0_total_height[4]	field0_total_height[3]	field0_total_height[2]	field0_total_height[1]	field0_total_height[0]
0x28	0x00	field1_total_height_1	r	-	-	field1_total_height[13]	field1_total_height[12]	field1_total_height[11]	field1_total_height[10]	field1_total_height[9]	field1_total_height[8]
0x29	0x00	field1_total_height_2	r	field1_total_height[7]	field1_total_height[6]	field1_total_height[5]	field1_total_height[4]	field1_total_height[3]	field1_total_height[2]	field1_total_height[1]	field1_total_height[0]
0x2A	0x00	field0_vs_front_porch_1	r	-	-	field0_vs_front_porch[13]	field0_vs_front_porch[12]	field0_vs_front_porch[11]	field0_vs_front_porch[10]	field0_vs_front_porch[9]	field0_vs_front_porch[8]
0x2B	0x00	field0_vs_front_porch_2	r	field0_vs_front_porch[7]	field0_vs_front_porch[6]	field0_vs_front_porch[5]	field0_vs_front_porch[4]	field0_vs_front_porch[3]	field0_vs_front_porch[2]	field0_vs_front_porch[1]	field0_vs_front_porch[0]
0x2C	0x00	field1_vs_front_porch_1	r	-	-	field1_vs_front_porch[13]	field1_vs_front_porch[12]	field1_vs_front_porch[11]	field1_vs_front_porch[10]	field1_vs_front_porch[9]	field1_vs_front_porch[8]
0x2D	0x00	field1_vs_front_porch_2	r	field1_vs_front_porch[7]	field1_vs_front_porch[6]	field1_vs_front_porch[5]	field1_vs_front_porch[4]	field1_vs_front_porch[3]	field1_vs_front_porch[2]	field1_vs_front_porch[1]	field1_vs_front_porch[0]
0x2E	0x00	field0_vs_pulse_width_1	r	-	-	field0_vs_pulse_width[13]	field0_vs_pulse_width[12]	field0_vs_pulse_width[11]	field0_vs_pulse_width[10]	field0_vs_pulse_width[9]	field0_vs_pulse_width[8]
0x2F	0x00	field0_vs_pulse_width_2	r	field0_vs_pulse_width[7]	field0_vs_pulse_width[6]	field0_vs_pulse_width[5]	field0_vs_pulse_width[4]	field0_vs_pulse_width[3]	field0_vs_pulse_width[2]	field0_vs_pulse_width[1]	field0_vs_pulse_width[0]
0x30	0x00	field1_vs_pulse_width_1	r	-	-	field1_vs_pulse_width[13]	field1_vs_pulse_width[12]	field1_vs_pulse_width[11]	field1_vs_pulse_width[10]	field1_vs_pulse_width[9]	field1_vs_pulse_width[8]
0x31	0x00	field1_vs_pulse_width_2	r	field1_vs_pulse_width[7]	field1_vs_pulse_width[6]	field1_vs_pulse_width[5]	field1_vs_pulse_width[4]	field1_vs_pulse_width[3]	field1_vs_pulse_width[2]	field1_vs_pulse_width[1]	field1_vs_pulse_width[0]
0x32	0x00	field0_vs_back_porch_1	r	-	-	field0_vs_back_porch[13]	field0_vs_back_porch[12]	field0_vs_back_porch[11]	field0_vs_back_porch[10]	field0_vs_back_porch[9]	field0_vs_back_porch[8]
0x33	0x00	field0_vs_back_porch_2	r	field0_vs_back_porch[7]	field0_vs_back_porch[6]	field0_vs_back_porch[5]	field0_vs_back_porch[4]	field0_vs_back_porch[3]	field0_vs_back_porch[2]	field0_vs_back_porch[1]	field0_vs_back_porch[0]
0x34	0x00	field1_vs_back_porch_1	r	-	-	field1_vs_back_porch[13]	field1_vs_back_porch[12]	field1_vs_back_porch[11]	field1_vs_back_porch[10]	field1_vs_back_porch[9]	field1_vs_back_porch[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x35	0x00	field1_vs_back_porch_2	r	field1_vs_back_porch[7]	field1_vs_back_porch[6]	field1_vs_back_porch[5]	field1_vs_back_porch[4]	field1_vs_back_porch[3]	field1_vs_back_porch[2]	field1_vs_back_porch[1]	field1_vs_back_porch[0]
0x36	0x00	channel status data_1	r	cs_data[7]	cs_data[6]	cs_data[5]	cs_data[4]	cs_data[3]	cs_data[2]	cs_data[1]	cs_data[0]
0x37	0x00	channel status data_2	r	cs_data[15]	cs_data[14]	cs_data[13]	cs_data[12]	cs_data[11]	cs_data[10]	cs_data[9]	cs_data[8]
0x38	0x00	channel status data_3	r	cs_data[23]	cs_data[22]	cs_data[21]	cs_data[20]	cs_data[19]	cs_data[18]	cs_data[17]	cs_data[16]
0x39	0x00	channel status data_4	r	cs_data[31]	cs_data[30]	cs_data[29]	cs_data[28]	cs_data[27]	cs_data[26]	cs_data[25]	cs_data[24]
0x3A	0x00	channel status data_5	r	cs_data[39]	cs_data[38]	cs_data[37]	cs_data[36]	cs_data[35]	cs_data[34]	cs_data[33]	cs_data[32]
0x40	0x00	register_40h	rw	-	override_deep_color_mode	deep_color_mode_user[1]	deep_color_mode_user[0]	-	-	-	-
0x41	0x40	register_41h	rw	-	-	-	derep_n_override	derep_n[3]	derep_n[2]	derep_n[1]	derep_n[0]
0x47	0x00	register_47h	rw	-	-	-	-	-	qzero_itc_dis	qzero_rgb_full	always_store_inf
0x48	0x00	register_48h	rw	-	dis_cable_det_rst	-	-	-	-	-	-
0x50	0x00	hdmi_register_50	rw	-	-	-	gamut_irq_next_field	-	-	cs_copyright_manual	cs_copyright_value
0x51	0x00	hdmi_reg_51	r	tmdsfreq[8]	tmdsfreq[7]	tmdsfreq[6]	tmdsfreq[5]	tmdsfreq[4]	tmdsfreq[3]	tmdsfreq[2]	tmdsfreq[1]
0x52	0x00	hdmi_reg_52	r	tmdsfreq[0]	tmdsfreq_frac[6]	tmdsfreq_frac[5]	tmdsfreq_frac[4]	tmdsfreq_frac[3]	tmdsfreq_frac[2]	tmdsfreq_frac[1]	tmdsfreq_frac[0]
0x53	0x00	hdmi_colorspace	r	-	-	-	-	hdmi_colorspace[3]	hdmi_colorspace[2]	hdmi_colorspace[1]	hdmi_colorspace[0]
0x56	0x58	filt_5v_det_reg	rw	filt_5v_det_dis	filt_5v_det_timer[6]	filt_5v_det_timer[5]	filt_5v_det_timer[4]	filt_5v_det_timer[3]	filt_5v_det_timer[2]	filt_5v_det_timer[1]	filt_5v_det_timer[0]
0x5A	0x00	register_5a	sc	-	-	-	-	hdcp_rept_edid_reset	dcfifo_recenter	-	force_n_update
0x5B	0x00	cts_n_1	r	cts[19]	cts[18]	cts[17]	cts[16]	cts[15]	cts[14]	cts[13]	cts[12]
0x5C	0x00	cts_n_2	r	cts[11]	cts[10]	cts[9]	cts[8]	cts[7]	cts[6]	cts[5]	cts[4]
0x5D	0x00	cts_n_3	r	cts[3]	cts[2]	cts[1]	cts[0]	n[19]	n[18]	n[17]	n[16]
0x5E	0x00	cts_n_4	r	n[15]	n[14]	n[13]	n[12]	n[11]	n[10]	n[9]	n[8]
0x5F	0x00	cts_n_5	r	n[7]	n[6]	n[5]	n[4]	n[3]	n[2]	n[1]	n[0]
0x6C	0xA3	hdmi_reg_6c	rw	hpa_delay_sel[3]	hpa_delay_sel[2]	hpa_delay_sel[1]	hpa_delay_sel[0]	hpa_ovr_term	hpa_auto_int_edid[1]	hpa_auto_int_edid[0]	hpa_manual
0x6D	0x00	hdmi_reg_6d	rw	i2s_tdm_mode_enable	-	-	-	-	-	-	-
0x73	0x00	ddc pad	rw	ddc_pwrdn[7]	ddc_pwrdn[6]	ddc_pwrdn[5]	ddc_pwrdn[4]	ddc_pwrdn[3]	ddc_pwrdn[2]	ddc_pwrdn[1]	ddc_pwrdn[0]
0x83	0xFF	hdmi_register_02h	rw	-	-	-	-	-	-	-	hdmi_terma_disable
0x89	0x00	eq dynamic enable	rw	-	-	-	-	-	-	-	eq_dyn_en_a
0x8A	0xA3	eq dynamic freq	rw	eq_dyn_freq2[3]	eq_dyn_freq2[2]	eq_dyn_freq2[1]	eq_dyn_freq2[0]	eq_dyn_freq1[3]	eq_dyn_freq1[2]	eq_dyn_freq1[1]	eq_dyn_freq1[0]
0xF8	0x00	hdmi_reg_f8	rw	-	-	-	-	-	-	-	hpa_man_value_port_a
0xF9	0x00	hdmi_reg_f9	rw	-	-	-	-	-	-	-	hpa_tristate_port_a

1.3 HDMI RX REPEATER MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x00	bksv_1	r	bksv[7]	bksv[6]	bksv[5]	bksv[4]	bksv[3]	bksv[2]	bksv[1]	bksv[0]
0x01	0x00	bksv_2	r	bksv[15]	bksv[14]	bksv[13]	bksv[12]	bksv[11]	bksv[10]	bksv[9]	bksv[8]
0x02	0x00	bksv_3	r	bksv[23]	bksv[22]	bksv[21]	bksv[20]	bksv[19]	bksv[18]	bksv[17]	bksv[16]
0x03	0x00	bksv_4	r	bksv[31]	bksv[30]	bksv[29]	bksv[28]	bksv[27]	bksv[26]	bksv[25]	bksv[24]
0x04	0x00	bksv_5	r	bksv[39]	bksv[38]	bksv[37]	bksv[36]	bksv[35]	bksv[34]	bksv[33]	bksv[32]
0x08	0x00	ri_1	r	ri[7]	ri[6]	ri[5]	ri[4]	ri[3]	ri[2]	ri[1]	ri[0]
0x09	0x00	ri_2	r	ri[15]	ri[14]	ri[13]	ri[12]	ri[11]	ri[10]	ri[9]	ri[8]
0x0A	0x00	pj	r	pj[7]	pj[6]	pj[5]	pj[4]	pj[3]	pj[2]	pj[1]	pj[0]
0x10	0x00	aksv_1	rw	aksv[7]	aksv[6]	aksv[5]	aksv[4]	aksv[3]	aksv[2]	aksv[1]	aksv[0]
0x11	0x00	aksv_2	rw	aksv[15]	aksv[14]	aksv[13]	aksv[12]	aksv[11]	aksv[10]	aksv[9]	aksv[8]
0x12	0x00	aksv_3	rw	aksv[23]	aksv[22]	aksv[21]	aksv[20]	aksv[19]	aksv[18]	aksv[17]	aksv[16]
0x13	0x00	aksv_4	rw	aksv[31]	aksv[30]	aksv[29]	aksv[28]	aksv[27]	aksv[26]	aksv[25]	aksv[24]
0x14	0x00	aksv_5	rw	aksv[39]	aksv[38]	aksv[37]	aksv[36]	aksv[35]	aksv[34]	aksv[33]	aksv[32]
0x15	0x00	ainfo	rw	ainfo[7]	ainfo[6]	ainfo[5]	ainfo[4]	ainfo[3]	ainfo[2]	ainfo[1]	ainfo[0]
0x16	0x00	ainfo_rb	r	ainfo_rb[7]	ainfo_rb[6]	ainfo_rb[5]	ainfo_rb[4]	ainfo_rb[3]	ainfo_rb[2]	ainfo_rb[1]	ainfo_rb[0]
0x18	0x00	an_1	rw	an[7]	an[6]	an[5]	an[4]	an[3]	an[2]	an[1]	an[0]
0x19	0x00	an_2	rw	an[15]	an[14]	an[13]	an[12]	an[11]	an[10]	an[9]	an[8]
0x1A	0x00	an_3	rw	an[23]	an[22]	an[21]	an[20]	an[19]	an[18]	an[17]	an[16]
0x1B	0x00	an_4	rw	an[31]	an[30]	an[29]	an[28]	an[27]	an[26]	an[25]	an[24]
0x1C	0x00	an_5	rw	an[39]	an[38]	an[37]	an[36]	an[35]	an[34]	an[33]	an[32]
0x1D	0x00	an_6	rw	an[47]	an[46]	an[45]	an[44]	an[43]	an[42]	an[41]	an[40]
0x1E	0x00	an_7	rw	an[55]	an[54]	an[53]	an[52]	an[51]	an[50]	an[49]	an[48]
0x1F	0x00	an_8	rw	an[63]	an[62]	an[61]	an[60]	an[59]	an[58]	an[57]	an[56]
0x20	0x00	sha_a_1	rw	sha_a[7]	sha_a[6]	sha_a[5]	sha_a[4]	sha_a[3]	sha_a[2]	sha_a[1]	sha_a[0]
0x21	0x00	sha_a_2	rw	sha_a[15]	sha_a[14]	sha_a[13]	sha_a[12]	sha_a[11]	sha_a[10]	sha_a[9]	sha_a[8]
0x22	0x00	sha_a_3	rw	sha_a[23]	sha_a[22]	sha_a[21]	sha_a[20]	sha_a[19]	sha_a[18]	sha_a[17]	sha_a[16]
0x23	0x00	sha_a_4	rw	sha_a[31]	sha_a[30]	sha_a[29]	sha_a[28]	sha_a[27]	sha_a[26]	sha_a[25]	sha_a[24]
0x24	0x00	sha_b_1	rw	sha_b[7]	sha_b[6]	sha_b[5]	sha_b[4]	sha_b[3]	sha_b[2]	sha_b[1]	sha_b[0]
0x25	0x00	sha_b_2	rw	sha_b[15]	sha_b[14]	sha_b[13]	sha_b[12]	sha_b[11]	sha_b[10]	sha_b[9]	sha_b[8]
0x26	0x00	sha_b_3	rw	sha_b[23]	sha_b[22]	sha_b[21]	sha_b[20]	sha_b[19]	sha_b[18]	sha_b[17]	sha_b[16]
0x27	0x00	sha_b_4	rw	sha_b[31]	sha_b[30]	sha_b[29]	sha_b[28]	sha_b[27]	sha_b[26]	sha_b[25]	sha_b[24]
0x28	0x00	sha_c_1	rw	sha_c[7]	sha_c[6]	sha_c[5]	sha_c[4]	sha_c[3]	sha_c[2]	sha_c[1]	sha_c[0]
0x29	0x00	sha_c_2	rw	sha_c[15]	sha_c[14]	sha_c[13]	sha_c[12]	sha_c[11]	sha_c[10]	sha_c[9]	sha_c[8]
0x2A	0x00	sha_c_3	rw	sha_c[23]	sha_c[22]	sha_c[21]	sha_c[20]	sha_c[19]	sha_c[18]	sha_c[17]	sha_c[16]
0x2B	0x00	sha_c_4	rw	sha_c[31]	sha_c[30]	sha_c[29]	sha_c[28]	sha_c[27]	sha_c[26]	sha_c[25]	sha_c[24]
0x2C	0x00	sha_d_1	rw	sha_d[7]	sha_d[6]	sha_d[5]	sha_d[4]	sha_d[3]	sha_d[2]	sha_d[1]	sha_d[0]
0x2D	0x00	sha_d_2	rw	sha_d[15]	sha_d[14]	sha_d[13]	sha_d[12]	sha_d[11]	sha_d[10]	sha_d[9]	sha_d[8]
0x2E	0x00	sha_d_3	rw	sha_d[23]	sha_d[22]	sha_d[21]	sha_d[20]	sha_d[19]	sha_d[18]	sha_d[17]	sha_d[16]
0x2F	0x00	sha_d_4	rw	sha_d[31]	sha_d[30]	sha_d[29]	sha_d[28]	sha_d[27]	sha_d[26]	sha_d[25]	sha_d[24]
0x30	0x00	sha_e_1	rw	sha_e[7]	sha_e[6]	sha_e[5]	sha_e[4]	sha_e[3]	sha_e[2]	sha_e[1]	sha_e[0]
0x31	0x00	sha_e_2	rw	sha_e[15]	sha_e[14]	sha_e[13]	sha_e[12]	sha_e[11]	sha_e[10]	sha_e[9]	sha_e[8]
0x32	0x00	sha_e_3	rw	sha_e[23]	sha_e[22]	sha_e[21]	sha_e[20]	sha_e[19]	sha_e[18]	sha_e[17]	sha_e[16]
0x33	0x00	sha_e_4	rw	sha_e[31]	sha_e[30]	sha_e[29]	sha_e[28]	sha_e[27]	sha_e[26]	sha_e[25]	sha_e[24]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x40	0x83	bcaps	rw	bcaps[7]	bcaps[6]	bcaps[5]	bcaps[4]	bcaps[3]	bcaps[2]	bcaps[1]	bcaps[0]
0x41	0x00	bstatus_1	rw	bstatus[7]	bstatus[6]	bstatus[5]	bstatus[4]	bstatus[3]	bstatus[2]	bstatus[1]	bstatus[0]
0x42	0x00	bstatus_2	rw	bstatus[15]	bstatus[14]	bstatus[13]	bstatus[12]	bstatus[11]	bstatus[10]	bstatus[9]	bstatus[8]
0x69	0x00	repeater_reg_69	rw	-	-	-	-	-	-	-	ksv_list_ready_port_a
0x70	0x00	repeater_reg_70	rw	primary_edid_size[3]	primary_edid_size[2]	primary_edid_size[1]	primary_edid_size[0]	-	-	-	-
0x74	0x00	hdcp edid controls	rw	-	-	-	-	-	-	-	man_edid_a_enable
0x76	0x00	edid debug_2	r	-	-	-	-	-	-	-	edid_a_enabled
0x77	0x00	repeater_reg_77	sc	-	-	-	-	clear_ksv_list	cksum_calc	-	-
0x78	0x00	edid debug_3	sc	-	-	-	-	-	-	-	ksv_list_ready_clr_a
0x7A	0x00	repeater_reg_7a	rw	-	edid_segment_pointer[2]	edid_segment_pointer[1]	edid_segment_pointer[0]	disable_auto_edid	-	edid_sram_space_select[1]	edid_sram_space_select[0]
0x7D	0x00	repeater_reg_7d	rw	-	-	-	-	-	-	-	dual_edid_enable_port_a
0x7E	0x00	repeater_reg_7e	rw	secondary_edid_size[3]	secondary_edid_size[2]	secondary_edid_size[1]	secondary_edid_size[0]	-	-	-	-
0x80	0x00	ksv 0_1	rw	ksv_byte_0[7]	ksv_byte_0[6]	ksv_byte_0[5]	ksv_byte_0[4]	ksv_byte_0[3]	ksv_byte_0[2]	ksv_byte_0[1]	ksv_byte_0[0]
0x81	0x00	ksv 0_2	rw	ksv_byte_1[7]	ksv_byte_1[6]	ksv_byte_1[5]	ksv_byte_1[4]	ksv_byte_1[3]	ksv_byte_1[2]	ksv_byte_1[1]	ksv_byte_1[0]
0x82	0x00	ksv 0_3	rw	ksv_byte_2[7]	ksv_byte_2[6]	ksv_byte_2[5]	ksv_byte_2[4]	ksv_byte_2[3]	ksv_byte_2[2]	ksv_byte_2[1]	ksv_byte_2[0]
0x83	0x00	ksv 0_4	rw	ksv_byte_3[7]	ksv_byte_3[6]	ksv_byte_3[5]	ksv_byte_3[4]	ksv_byte_3[3]	ksv_byte_3[2]	ksv_byte_3[1]	ksv_byte_3[0]
0x84	0x00	ksv 0_5	rw	ksv_byte_4[7]	ksv_byte_4[6]	ksv_byte_4[5]	ksv_byte_4[4]	ksv_byte_4[3]	ksv_byte_4[2]	ksv_byte_4[1]	ksv_byte_4[0]
0x85	0x00	ksv 0_6	rw	ksv_byte_5[7]	ksv_byte_5[6]	ksv_byte_5[5]	ksv_byte_5[4]	ksv_byte_5[3]	ksv_byte_5[2]	ksv_byte_5[1]	ksv_byte_5[0]
0x86	0x00	ksv 0_7	rw	ksv_byte_6[7]	ksv_byte_6[6]	ksv_byte_6[5]	ksv_byte_6[4]	ksv_byte_6[3]	ksv_byte_6[2]	ksv_byte_6[1]	ksv_byte_6[0]
0x87	0x00	ksv 0_8	rw	ksv_byte_7[7]	ksv_byte_7[6]	ksv_byte_7[5]	ksv_byte_7[4]	ksv_byte_7[3]	ksv_byte_7[2]	ksv_byte_7[1]	ksv_byte_7[0]
0x88	0x00	ksv 0_9	rw	ksv_byte_8[7]	ksv_byte_8[6]	ksv_byte_8[5]	ksv_byte_8[4]	ksv_byte_8[3]	ksv_byte_8[2]	ksv_byte_8[1]	ksv_byte_8[0]
0x89	0x00	ksv 0_10	rw	ksv_byte_9[7]	ksv_byte_9[6]	ksv_byte_9[5]	ksv_byte_9[4]	ksv_byte_9[3]	ksv_byte_9[2]	ksv_byte_9[1]	ksv_byte_9[0]
0x8A	0x00	ksv 0_11	rw	ksv_byte_10[7]	ksv_byte_10[6]	ksv_byte_10[5]	ksv_byte_10[4]	ksv_byte_10[3]	ksv_byte_10[2]	ksv_byte_10[1]	ksv_byte_10[0]
0x8B	0x00	ksv 0_12	rw	ksv_byte_11[7]	ksv_byte_11[6]	ksv_byte_11[5]	ksv_byte_11[4]	ksv_byte_11[3]	ksv_byte_11[2]	ksv_byte_11[1]	ksv_byte_11[0]
0x8C	0x00	ksv 0_13	rw	ksv_byte_12[7]	ksv_byte_12[6]	ksv_byte_12[5]	ksv_byte_12[4]	ksv_byte_12[3]	ksv_byte_12[2]	ksv_byte_12[1]	ksv_byte_12[0]
0x8D	0x00	ksv 0_14	rw	ksv_byte_13[7]	ksv_byte_13[6]	ksv_byte_13[5]	ksv_byte_13[4]	ksv_byte_13[3]	ksv_byte_13[2]	ksv_byte_13[1]	ksv_byte_13[0]
0x8E	0x00	ksv 0_15	rw	ksv_byte_14[7]	ksv_byte_14[6]	ksv_byte_14[5]	ksv_byte_14[4]	ksv_byte_14[3]	ksv_byte_14[2]	ksv_byte_14[1]	ksv_byte_14[0]
0x8F	0x00	ksv 0_16	rw	ksv_byte_15[7]	ksv_byte_15[6]	ksv_byte_15[5]	ksv_byte_15[4]	ksv_byte_15[3]	ksv_byte_15[2]	ksv_byte_15[1]	ksv_byte_15[0]
0x90	0x00	ksv 0_17	rw	ksv_byte_16[7]	ksv_byte_16[6]	ksv_byte_16[5]	ksv_byte_16[4]	ksv_byte_16[3]	ksv_byte_16[2]	ksv_byte_16[1]	ksv_byte_16[0]
0x91	0x00	ksv 0_18	rw	ksv_byte_17[7]	ksv_byte_17[6]	ksv_byte_17[5]	ksv_byte_17[4]	ksv_byte_17[3]	ksv_byte_17[2]	ksv_byte_17[1]	ksv_byte_17[0]
0x92	0x00	ksv 0_19	rw	ksv_byte_18[7]	ksv_byte_18[6]	ksv_byte_18[5]	ksv_byte_18[4]	ksv_byte_18[3]	ksv_byte_18[2]	ksv_byte_18[1]	ksv_byte_18[0]
0x93	0x00	ksv 0_20	rw	ksv_byte_19[7]	ksv_byte_19[6]	ksv_byte_19[5]	ksv_byte_19[4]	ksv_byte_19[3]	ksv_byte_19[2]	ksv_byte_19[1]	ksv_byte_19[0]
0x94	0x00	ksv 0_21	rw	ksv_byte_20[7]	ksv_byte_20[6]	ksv_byte_20[5]	ksv_byte_20[4]	ksv_byte_20[3]	ksv_byte_20[2]	ksv_byte_20[1]	ksv_byte_20[0]
0x95	0x00	ksv 0_22	rw	ksv_byte_21[7]	ksv_byte_21[6]	ksv_byte_21[5]	ksv_byte_21[4]	ksv_byte_21[3]	ksv_byte_21[2]	ksv_byte_21[1]	ksv_byte_21[0]
0x96	0x00	ksv 0_23	rw	ksv_byte_22[7]	ksv_byte_22[6]	ksv_byte_22[5]	ksv_byte_22[4]	ksv_byte_22[3]	ksv_byte_22[2]	ksv_byte_22[1]	ksv_byte_22[0]
0x97	0x00	ksv 0_24	rw	ksv_byte_23[7]	ksv_byte_23[6]	ksv_byte_23[5]	ksv_byte_23[4]	ksv_byte_23[3]	ksv_byte_23[2]	ksv_byte_23[1]	ksv_byte_23[0]
0x98	0x00	ksv 0_25	rw	ksv_byte_24[7]	ksv_byte_24[6]	ksv_byte_24[5]	ksv_byte_24[4]	ksv_byte_24[3]	ksv_byte_24[2]	ksv_byte_24[1]	ksv_byte_24[0]
0x99	0x00	ksv 0_26	rw	ksv_byte_25[7]	ksv_byte_25[6]	ksv_byte_25[5]	ksv_byte_25[4]	ksv_byte_25[3]	ksv_byte_25[2]	ksv_byte_25[1]	ksv_byte_25[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x9A	0x00	ksv 0_27	rw	ksv_byte_26[7]	ksv_byte_26[6]	ksv_byte_26[5]	ksv_byte_26[4]	ksv_byte_26[3]	ksv_byte_26[2]	ksv_byte_26[1]	ksv_byte_26[0]
0x9B	0x00	ksv 0_28	rw	ksv_byte_27[7]	ksv_byte_27[6]	ksv_byte_27[5]	ksv_byte_27[4]	ksv_byte_27[3]	ksv_byte_27[2]	ksv_byte_27[1]	ksv_byte_27[0]
0x9C	0x00	ksv 0_29	rw	ksv_byte_28[7]	ksv_byte_28[6]	ksv_byte_28[5]	ksv_byte_28[4]	ksv_byte_28[3]	ksv_byte_28[2]	ksv_byte_28[1]	ksv_byte_28[0]
0x9D	0x00	ksv 0_30	rw	ksv_byte_29[7]	ksv_byte_29[6]	ksv_byte_29[5]	ksv_byte_29[4]	ksv_byte_29[3]	ksv_byte_29[2]	ksv_byte_29[1]	ksv_byte_29[0]
0x9E	0x00	ksv 0_31	rw	ksv_byte_30[7]	ksv_byte_30[6]	ksv_byte_30[5]	ksv_byte_30[4]	ksv_byte_30[3]	ksv_byte_30[2]	ksv_byte_30[1]	ksv_byte_30[0]
0x9F	0x00	ksv 0_32	rw	ksv_byte_31[7]	ksv_byte_31[6]	ksv_byte_31[5]	ksv_byte_31[4]	ksv_byte_31[3]	ksv_byte_31[2]	ksv_byte_31[1]	ksv_byte_31[0]
0xA0	0x00	ksv 0_33	rw	ksv_byte_32[7]	ksv_byte_32[6]	ksv_byte_32[5]	ksv_byte_32[4]	ksv_byte_32[3]	ksv_byte_32[2]	ksv_byte_32[1]	ksv_byte_32[0]
0xA1	0x00	ksv 0_34	rw	ksv_byte_33[7]	ksv_byte_33[6]	ksv_byte_33[5]	ksv_byte_33[4]	ksv_byte_33[3]	ksv_byte_33[2]	ksv_byte_33[1]	ksv_byte_33[0]
0xA2	0x00	ksv 0_35	rw	ksv_byte_34[7]	ksv_byte_34[6]	ksv_byte_34[5]	ksv_byte_34[4]	ksv_byte_34[3]	ksv_byte_34[2]	ksv_byte_34[1]	ksv_byte_34[0]
0xA3	0x00	ksv 0_36	rw	ksv_byte_35[7]	ksv_byte_35[6]	ksv_byte_35[5]	ksv_byte_35[4]	ksv_byte_35[3]	ksv_byte_35[2]	ksv_byte_35[1]	ksv_byte_35[0]
0xA4	0x00	ksv 0_37	rw	ksv_byte_36[7]	ksv_byte_36[6]	ksv_byte_36[5]	ksv_byte_36[4]	ksv_byte_36[3]	ksv_byte_36[2]	ksv_byte_36[1]	ksv_byte_36[0]
0xA5	0x00	ksv 0_38	rw	ksv_byte_37[7]	ksv_byte_37[6]	ksv_byte_37[5]	ksv_byte_37[4]	ksv_byte_37[3]	ksv_byte_37[2]	ksv_byte_37[1]	ksv_byte_37[0]
0xA6	0x00	ksv 0_39	rw	ksv_byte_38[7]	ksv_byte_38[6]	ksv_byte_38[5]	ksv_byte_38[4]	ksv_byte_38[3]	ksv_byte_38[2]	ksv_byte_38[1]	ksv_byte_38[0]
0xA7	0x00	ksv 0_40	rw	ksv_byte_39[7]	ksv_byte_39[6]	ksv_byte_39[5]	ksv_byte_39[4]	ksv_byte_39[3]	ksv_byte_39[2]	ksv_byte_39[1]	ksv_byte_39[0]
0xA8	0x00	ksv 0_41	rw	ksv_byte_40[7]	ksv_byte_40[6]	ksv_byte_40[5]	ksv_byte_40[4]	ksv_byte_40[3]	ksv_byte_40[2]	ksv_byte_40[1]	ksv_byte_40[0]
0xA9	0x00	ksv 0_42	rw	ksv_byte_41[7]	ksv_byte_41[6]	ksv_byte_41[5]	ksv_byte_41[4]	ksv_byte_41[3]	ksv_byte_41[2]	ksv_byte_41[1]	ksv_byte_41[0]
0xAA	0x00	ksv 0_43	rw	ksv_byte_42[7]	ksv_byte_42[6]	ksv_byte_42[5]	ksv_byte_42[4]	ksv_byte_42[3]	ksv_byte_42[2]	ksv_byte_42[1]	ksv_byte_42[0]
0xAB	0x00	ksv 0_44	rw	ksv_byte_43[7]	ksv_byte_43[6]	ksv_byte_43[5]	ksv_byte_43[4]	ksv_byte_43[3]	ksv_byte_43[2]	ksv_byte_43[1]	ksv_byte_43[0]
0xAC	0x00	ksv 0_45	rw	ksv_byte_44[7]	ksv_byte_44[6]	ksv_byte_44[5]	ksv_byte_44[4]	ksv_byte_44[3]	ksv_byte_44[2]	ksv_byte_44[1]	ksv_byte_44[0]
0xAD	0x00	ksv 0_46	rw	ksv_byte_45[7]	ksv_byte_45[6]	ksv_byte_45[5]	ksv_byte_45[4]	ksv_byte_45[3]	ksv_byte_45[2]	ksv_byte_45[1]	ksv_byte_45[0]
0xAE	0x00	ksv 0_47	rw	ksv_byte_46[7]	ksv_byte_46[6]	ksv_byte_46[5]	ksv_byte_46[4]	ksv_byte_46[3]	ksv_byte_46[2]	ksv_byte_46[1]	ksv_byte_46[0]
0xAF	0x00	ksv 0_48	rw	ksv_byte_47[7]	ksv_byte_47[6]	ksv_byte_47[5]	ksv_byte_47[4]	ksv_byte_47[3]	ksv_byte_47[2]	ksv_byte_47[1]	ksv_byte_47[0]
0xB0	0x00	ksv 0_49	rw	ksv_byte_48[7]	ksv_byte_48[6]	ksv_byte_48[5]	ksv_byte_48[4]	ksv_byte_48[3]	ksv_byte_48[2]	ksv_byte_48[1]	ksv_byte_48[0]
0xB1	0x00	ksv 0_50	rw	ksv_byte_49[7]	ksv_byte_49[6]	ksv_byte_49[5]	ksv_byte_49[4]	ksv_byte_49[3]	ksv_byte_49[2]	ksv_byte_49[1]	ksv_byte_49[0]
0xB2	0x00	ksv 0_51	rw	ksv_byte_50[7]	ksv_byte_50[6]	ksv_byte_50[5]	ksv_byte_50[4]	ksv_byte_50[3]	ksv_byte_50[2]	ksv_byte_50[1]	ksv_byte_50[0]
0xB3	0x00	ksv 0_52	rw	ksv_byte_51[7]	ksv_byte_51[6]	ksv_byte_51[5]	ksv_byte_51[4]	ksv_byte_51[3]	ksv_byte_51[2]	ksv_byte_51[1]	ksv_byte_51[0]
0xB4	0x00	ksv 0_53	rw	ksv_byte_52[7]	ksv_byte_52[6]	ksv_byte_52[5]	ksv_byte_52[4]	ksv_byte_52[3]	ksv_byte_52[2]	ksv_byte_52[1]	ksv_byte_52[0]
0xB5	0x00	ksv 0_54	rw	ksv_byte_53[7]	ksv_byte_53[6]	ksv_byte_53[5]	ksv_byte_53[4]	ksv_byte_53[3]	ksv_byte_53[2]	ksv_byte_53[1]	ksv_byte_53[0]
0xB6	0x00	ksv 0_55	rw	ksv_byte_54[7]	ksv_byte_54[6]	ksv_byte_54[5]	ksv_byte_54[4]	ksv_byte_54[3]	ksv_byte_54[2]	ksv_byte_54[1]	ksv_byte_54[0]
0xB7	0x00	ksv 0_56	rw	ksv_byte_55[7]	ksv_byte_55[6]	ksv_byte_55[5]	ksv_byte_55[4]	ksv_byte_55[3]	ksv_byte_55[2]	ksv_byte_55[1]	ksv_byte_55[0]
0xB8	0x00	ksv 0_57	rw	ksv_byte_56[7]	ksv_byte_56[6]	ksv_byte_56[5]	ksv_byte_56[4]	ksv_byte_56[3]	ksv_byte_56[2]	ksv_byte_56[1]	ksv_byte_56[0]
0xB9	0x00	ksv 0_58	rw	ksv_byte_57[7]	ksv_byte_57[6]	ksv_byte_57[5]	ksv_byte_57[4]	ksv_byte_57[3]	ksv_byte_57[2]	ksv_byte_57[1]	ksv_byte_57[0]
0xBA	0x00	ksv 0_59	rw	ksv_byte_58[7]	ksv_byte_58[6]	ksv_byte_58[5]	ksv_byte_58[4]	ksv_byte_58[3]	ksv_byte_58[2]	ksv_byte_58[1]	ksv_byte_58[0]
0xBB	0x00	ksv 0_60	rw	ksv_byte_59[7]	ksv_byte_59[6]	ksv_byte_59[5]	ksv_byte_59[4]	ksv_byte_59[3]	ksv_byte_59[2]	ksv_byte_59[1]	ksv_byte_59[0]
0xBC	0x00	ksv 0_61	rw	ksv_byte_60[7]	ksv_byte_60[6]	ksv_byte_60[5]	ksv_byte_60[4]	ksv_byte_60[3]	ksv_byte_60[2]	ksv_byte_60[1]	ksv_byte_60[0]
0xBD	0x00	ksv 0_62	rw	ksv_byte_61[7]	ksv_byte_61[6]	ksv_byte_61[5]	ksv_byte_61[4]	ksv_byte_61[3]	ksv_byte_61[2]	ksv_byte_61[1]	ksv_byte_61[0]
0xBE	0x00	ksv 0_63	rw	ksv_byte_62[7]	ksv_byte_62[6]	ksv_byte_62[5]	ksv_byte_62[4]	ksv_byte_62[3]	ksv_byte_62[2]	ksv_byte_62[1]	ksv_byte_62[0]
0xBF	0x00	ksv 0_64	rw	ksv_byte_63[7]	ksv_byte_63[6]	ksv_byte_63[5]	ksv_byte_63[4]	ksv_byte_63[3]	ksv_byte_63[2]	ksv_byte_63[1]	ksv_byte_63[0]
0xC0	0x00	ksv 0_65	rw	ksv_byte_64[7]	ksv_byte_64[6]	ksv_byte_64[5]	ksv_byte_64[4]	ksv_byte_64[3]	ksv_byte_64[2]	ksv_byte_64[1]	ksv_byte_64[0]
0xC1	0x00	ksv 0_66	rw	ksv_byte_65[7]	ksv_byte_65[6]	ksv_byte_65[5]	ksv_byte_65[4]	ksv_byte_65[3]	ksv_byte_65[2]	ksv_byte_65[1]	ksv_byte_65[0]
0xC2	0x00	ksv 0_67	rw	ksv_byte_66[7]	ksv_byte_66[6]	ksv_byte_66[5]	ksv_byte_66[4]	ksv_byte_66[3]	ksv_byte_66[2]	ksv_byte_66[1]	ksv_byte_66[0]
0xC3	0x00	ksv 0_68	rw	ksv_byte_67[7]	ksv_byte_67[6]	ksv_byte_67[5]	ksv_byte_67[4]	ksv_byte_67[3]	ksv_byte_67[2]	ksv_byte_67[1]	ksv_byte_67[0]
0xC4	0x00	ksv 0_69	rw	ksv_byte_68[7]	ksv_byte_68[6]	ksv_byte_68[5]	ksv_byte_68[4]	ksv_byte_68[3]	ksv_byte_68[2]	ksv_byte_68[1]	ksv_byte_68[0]
0xC5	0x00	ksv 0_70	rw	ksv_byte_69[7]	ksv_byte_69[6]	ksv_byte_69[5]	ksv_byte_69[4]	ksv_byte_69[3]	ksv_byte_69[2]	ksv_byte_69[1]	ksv_byte_69[0]
0xC6	0x00	ksv 0_71	rw	ksv_byte_70[7]	ksv_byte_70[6]	ksv_byte_70[5]	ksv_byte_70[4]	ksv_byte_70[3]	ksv_byte_70[2]	ksv_byte_70[1]	ksv_byte_70[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xC7	0x00	ksv 0_72	rw	ksv_byte_71[7]	ksv_byte_71[6]	ksv_byte_71[5]	ksv_byte_71[4]	ksv_byte_71[3]	ksv_byte_71[2]	ksv_byte_71[1]	ksv_byte_71[0]
0xC8	0x00	ksv 0_73	rw	ksv_byte_72[7]	ksv_byte_72[6]	ksv_byte_72[5]	ksv_byte_72[4]	ksv_byte_72[3]	ksv_byte_72[2]	ksv_byte_72[1]	ksv_byte_72[0]
0xC9	0x00	ksv 0_74	rw	ksv_byte_73[7]	ksv_byte_73[6]	ksv_byte_73[5]	ksv_byte_73[4]	ksv_byte_73[3]	ksv_byte_73[2]	ksv_byte_73[1]	ksv_byte_73[0]
0xCA	0x00	ksv 0_75	rw	ksv_byte_74[7]	ksv_byte_74[6]	ksv_byte_74[5]	ksv_byte_74[4]	ksv_byte_74[3]	ksv_byte_74[2]	ksv_byte_74[1]	ksv_byte_74[0]
0xCB	0x00	ksv 0_76	rw	ksv_byte_75[7]	ksv_byte_75[6]	ksv_byte_75[5]	ksv_byte_75[4]	ksv_byte_75[3]	ksv_byte_75[2]	ksv_byte_75[1]	ksv_byte_75[0]
0xCC	0x00	ksv 0_77	rw	ksv_byte_76[7]	ksv_byte_76[6]	ksv_byte_76[5]	ksv_byte_76[4]	ksv_byte_76[3]	ksv_byte_76[2]	ksv_byte_76[1]	ksv_byte_76[0]
0xCD	0x00	ksv 0_78	rw	ksv_byte_77[7]	ksv_byte_77[6]	ksv_byte_77[5]	ksv_byte_77[4]	ksv_byte_77[3]	ksv_byte_77[2]	ksv_byte_77[1]	ksv_byte_77[0]
0xCE	0x00	ksv 0_79	rw	ksv_byte_78[7]	ksv_byte_78[6]	ksv_byte_78[5]	ksv_byte_78[4]	ksv_byte_78[3]	ksv_byte_78[2]	ksv_byte_78[1]	ksv_byte_78[0]
0xCF	0x00	ksv 0_80	rw	ksv_byte_79[7]	ksv_byte_79[6]	ksv_byte_79[5]	ksv_byte_79[4]	ksv_byte_79[3]	ksv_byte_79[2]	ksv_byte_79[1]	ksv_byte_79[0]
0xD0	0x00	ksv 0_81	rw	ksv_byte_80[7]	ksv_byte_80[6]	ksv_byte_80[5]	ksv_byte_80[4]	ksv_byte_80[3]	ksv_byte_80[2]	ksv_byte_80[1]	ksv_byte_80[0]
0xD1	0x00	ksv 0_82	rw	ksv_byte_81[7]	ksv_byte_81[6]	ksv_byte_81[5]	ksv_byte_81[4]	ksv_byte_81[3]	ksv_byte_81[2]	ksv_byte_81[1]	ksv_byte_81[0]
0xD2	0x00	ksv 0_83	rw	ksv_byte_82[7]	ksv_byte_82[6]	ksv_byte_82[5]	ksv_byte_82[4]	ksv_byte_82[3]	ksv_byte_82[2]	ksv_byte_82[1]	ksv_byte_82[0]
0xD3	0x00	ksv 0_84	rw	ksv_byte_83[7]	ksv_byte_83[6]	ksv_byte_83[5]	ksv_byte_83[4]	ksv_byte_83[3]	ksv_byte_83[2]	ksv_byte_83[1]	ksv_byte_83[0]
0xD4	0x00	ksv 0_85	rw	ksv_byte_84[7]	ksv_byte_84[6]	ksv_byte_84[5]	ksv_byte_84[4]	ksv_byte_84[3]	ksv_byte_84[2]	ksv_byte_84[1]	ksv_byte_84[0]
0xD5	0x00	ksv 0_86	rw	ksv_byte_85[7]	ksv_byte_85[6]	ksv_byte_85[5]	ksv_byte_85[4]	ksv_byte_85[3]	ksv_byte_85[2]	ksv_byte_85[1]	ksv_byte_85[0]
0xD6	0x00	ksv 0_87	rw	ksv_byte_86[7]	ksv_byte_86[6]	ksv_byte_86[5]	ksv_byte_86[4]	ksv_byte_86[3]	ksv_byte_86[2]	ksv_byte_86[1]	ksv_byte_86[0]
0xD7	0x00	ksv 0_88	rw	ksv_byte_87[7]	ksv_byte_87[6]	ksv_byte_87[5]	ksv_byte_87[4]	ksv_byte_87[3]	ksv_byte_87[2]	ksv_byte_87[1]	ksv_byte_87[0]
0xD8	0x00	ksv 0_89	rw	ksv_byte_88[7]	ksv_byte_88[6]	ksv_byte_88[5]	ksv_byte_88[4]	ksv_byte_88[3]	ksv_byte_88[2]	ksv_byte_88[1]	ksv_byte_88[0]
0xD9	0x00	ksv 0_90	rw	ksv_byte_89[7]	ksv_byte_89[6]	ksv_byte_89[5]	ksv_byte_89[4]	ksv_byte_89[3]	ksv_byte_89[2]	ksv_byte_89[1]	ksv_byte_89[0]
0xDA	0x00	ksv 0_91	rw	ksv_byte_90[7]	ksv_byte_90[6]	ksv_byte_90[5]	ksv_byte_90[4]	ksv_byte_90[3]	ksv_byte_90[2]	ksv_byte_90[1]	ksv_byte_90[0]
0xDB	0x00	ksv 0_92	rw	ksv_byte_91[7]	ksv_byte_91[6]	ksv_byte_91[5]	ksv_byte_91[4]	ksv_byte_91[3]	ksv_byte_91[2]	ksv_byte_91[1]	ksv_byte_91[0]
0xDC	0x00	ksv 0_93	rw	ksv_byte_92[7]	ksv_byte_92[6]	ksv_byte_92[5]	ksv_byte_92[4]	ksv_byte_92[3]	ksv_byte_92[2]	ksv_byte_92[1]	ksv_byte_92[0]
0xDD	0x00	ksv 0_94	rw	ksv_byte_93[7]	ksv_byte_93[6]	ksv_byte_93[5]	ksv_byte_93[4]	ksv_byte_93[3]	ksv_byte_93[2]	ksv_byte_93[1]	ksv_byte_93[0]
0xDE	0x00	ksv 0_95	rw	ksv_byte_94[7]	ksv_byte_94[6]	ksv_byte_94[5]	ksv_byte_94[4]	ksv_byte_94[3]	ksv_byte_94[2]	ksv_byte_94[1]	ksv_byte_94[0]
0xDF	0x00	ksv 0_96	rw	ksv_byte_95[7]	ksv_byte_95[6]	ksv_byte_95[5]	ksv_byte_95[4]	ksv_byte_95[3]	ksv_byte_95[2]	ksv_byte_95[1]	ksv_byte_95[0]
0xE0	0x00	ksv 0_97	rw	ksv_byte_96[7]	ksv_byte_96[6]	ksv_byte_96[5]	ksv_byte_96[4]	ksv_byte_96[3]	ksv_byte_96[2]	ksv_byte_96[1]	ksv_byte_96[0]
0xE1	0x00	ksv 0_98	rw	ksv_byte_97[7]	ksv_byte_97[6]	ksv_byte_97[5]	ksv_byte_97[4]	ksv_byte_97[3]	ksv_byte_97[2]	ksv_byte_97[1]	ksv_byte_97[0]
0xE2	0x00	ksv 0_99	rw	ksv_byte_98[7]	ksv_byte_98[6]	ksv_byte_98[5]	ksv_byte_98[4]	ksv_byte_98[3]	ksv_byte_98[2]	ksv_byte_98[1]	ksv_byte_98[0]
0xE3	0x00	ksv 0_100	rw	ksv_byte_99[7]	ksv_byte_99[6]	ksv_byte_99[5]	ksv_byte_99[4]	ksv_byte_99[3]	ksv_byte_99[2]	ksv_byte_99[1]	ksv_byte_99[0]
0xE4	0x00	ksv 0_101	rw	ksv_byte_100[7]	ksv_byte_100[6]	ksv_byte_100[5]	ksv_byte_100[4]	ksv_byte_100[3]	ksv_byte_100[2]	ksv_byte_100[1]	ksv_byte_100[0]
0xE5	0x00	ksv 0_102	rw	ksv_byte_101[7]	ksv_byte_101[6]	ksv_byte_101[5]	ksv_byte_101[4]	ksv_byte_101[3]	ksv_byte_101[2]	ksv_byte_101[1]	ksv_byte_101[0]
0xE6	0x00	ksv 0_103	rw	ksv_byte_102[7]	ksv_byte_102[6]	ksv_byte_102[5]	ksv_byte_102[4]	ksv_byte_102[3]	ksv_byte_102[2]	ksv_byte_102[1]	ksv_byte_102[0]
0xE7	0x00	ksv 0_104	rw	ksv_byte_103[7]	ksv_byte_103[6]	ksv_byte_103[5]	ksv_byte_103[4]	ksv_byte_103[3]	ksv_byte_103[2]	ksv_byte_103[1]	ksv_byte_103[0]
0xE8	0x00	ksv 0_105	rw	ksv_byte_104[7]	ksv_byte_104[6]	ksv_byte_104[5]	ksv_byte_104[4]	ksv_byte_104[3]	ksv_byte_104[2]	ksv_byte_104[1]	ksv_byte_104[0]
0xE9	0x00	ksv 0_106	rw	ksv_byte_105[7]	ksv_byte_105[6]	ksv_byte_105[5]	ksv_byte_105[4]	ksv_byte_105[3]	ksv_byte_105[2]	ksv_byte_105[1]	ksv_byte_105[0]
0xEA	0x00	ksv 0_107	rw	ksv_byte_106[7]	ksv_byte_106[6]	ksv_byte_106[5]	ksv_byte_106[4]	ksv_byte_106[3]	ksv_byte_106[2]	ksv_byte_106[1]	ksv_byte_106[0]
0xEB	0x00	ksv 0_108	rw	ksv_byte_107[7]	ksv_byte_107[6]	ksv_byte_107[5]	ksv_byte_107[4]	ksv_byte_107[3]	ksv_byte_107[2]	ksv_byte_107[1]	ksv_byte_107[0]
0xEC	0x00	ksv 0_109	rw	ksv_byte_108[7]	ksv_byte_108[6]	ksv_byte_108[5]	ksv_byte_108[4]	ksv_byte_108[3]	ksv_byte_108[2]	ksv_byte_108[1]	ksv_byte_108[0]
0xED	0x00	ksv 0_110	rw	ksv_byte_109[7]	ksv_byte_109[6]	ksv_byte_109[5]	ksv_byte_109[4]	ksv_byte_109[3]	ksv_byte_109[2]	ksv_byte_109[1]	ksv_byte_109[0]
0xEE	0x00	ksv 0_111	rw	ksv_byte_110[7]	ksv_byte_110[6]	ksv_byte_110[5]	ksv_byte_110[4]	ksv_byte_110[3]	ksv_byte_110[2]	ksv_byte_110[1]	ksv_byte_110[0]
0xEF	0x00	ksv 0_112	rw	ksv_byte_111[7]	ksv_byte_111[6]	ksv_byte_111[5]	ksv_byte_111[4]	ksv_byte_111[3]	ksv_byte_111[2]	ksv_byte_111[1]	ksv_byte_111[0]
0xF0	0x00	ksv 0_113	rw	ksv_byte_112[7]	ksv_byte_112[6]	ksv_byte_112[5]	ksv_byte_112[4]	ksv_byte_112[3]	ksv_byte_112[2]	ksv_byte_112[1]	ksv_byte_112[0]
0xF1	0x00	ksv 0_114	rw	ksv_byte_113[7]	ksv_byte_113[6]	ksv_byte_113[5]	ksv_byte_113[4]	ksv_byte_113[3]	ksv_byte_113[2]	ksv_byte_113[1]	ksv_byte_113[0]
0xF2	0x00	ksv 0_115	rw	ksv_byte_114[7]	ksv_byte_114[6]	ksv_byte_114[5]	ksv_byte_114[4]	ksv_byte_114[3]	ksv_byte_114[2]	ksv_byte_114[1]	ksv_byte_114[0]
0xF3	0x00	ksv 0_116	rw	ksv_byte_115[7]	ksv_byte_115[6]	ksv_byte_115[5]	ksv_byte_115[4]	ksv_byte_115[3]	ksv_byte_115[2]	ksv_byte_115[1]	ksv_byte_115[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF4	0x00	ksv 0_117	rw	ksv_byte_116[7]	ksv_byte_116[6]	ksv_byte_116[5]	ksv_byte_116[4]	ksv_byte_116[3]	ksv_byte_116[2]	ksv_byte_116[1]	ksv_byte_116[0]
0xF5	0x00	ksv 0_118	rw	ksv_byte_117[7]	ksv_byte_117[6]	ksv_byte_117[5]	ksv_byte_117[4]	ksv_byte_117[3]	ksv_byte_117[2]	ksv_byte_117[1]	ksv_byte_117[0]
0xF6	0x00	ksv 0_119	rw	ksv_byte_118[7]	ksv_byte_118[6]	ksv_byte_118[5]	ksv_byte_118[4]	ksv_byte_118[3]	ksv_byte_118[2]	ksv_byte_118[1]	ksv_byte_118[0]
0xF7	0x00	ksv 0_120	rw	ksv_byte_119[7]	ksv_byte_119[6]	ksv_byte_119[5]	ksv_byte_119[4]	ksv_byte_119[3]	ksv_byte_119[2]	ksv_byte_119[1]	ksv_byte_119[0]
0xF8	0x00	ksv 0_121	rw	ksv_byte_120[7]	ksv_byte_120[6]	ksv_byte_120[5]	ksv_byte_120[4]	ksv_byte_120[3]	ksv_byte_120[2]	ksv_byte_120[1]	ksv_byte_120[0]
0xF9	0x00	ksv 0_122	rw	ksv_byte_121[7]	ksv_byte_121[6]	ksv_byte_121[5]	ksv_byte_121[4]	ksv_byte_121[3]	ksv_byte_121[2]	ksv_byte_121[1]	ksv_byte_121[0]
0xFA	0x00	ksv 0_123	rw	ksv_byte_122[7]	ksv_byte_122[6]	ksv_byte_122[5]	ksv_byte_122[4]	ksv_byte_122[3]	ksv_byte_122[2]	ksv_byte_122[1]	ksv_byte_122[0]
0xFB	0x00	ksv 0_124	rw	ksv_byte_123[7]	ksv_byte_123[6]	ksv_byte_123[5]	ksv_byte_123[4]	ksv_byte_123[3]	ksv_byte_123[2]	ksv_byte_123[1]	ksv_byte_123[0]
0xFC	0x00	ksv 0_125	rw	ksv_byte_124[7]	ksv_byte_124[6]	ksv_byte_124[5]	ksv_byte_124[4]	ksv_byte_124[3]	ksv_byte_124[2]	ksv_byte_124[1]	ksv_byte_124[0]
0xFD	0x00	ksv 0_126	rw	ksv_byte_125[7]	ksv_byte_125[6]	ksv_byte_125[5]	ksv_byte_125[4]	ksv_byte_125[3]	ksv_byte_125[2]	ksv_byte_125[1]	ksv_byte_125[0]
0xFE	0x00	ksv 0_127	rw	ksv_byte_126[7]	ksv_byte_126[6]	ksv_byte_126[5]	ksv_byte_126[4]	ksv_byte_126[3]	ksv_byte_126[2]	ksv_byte_126[1]	ksv_byte_126[0]
0xFF	0x00	ksv 0_128	rw	ksv_byte_127[7]	ksv_byte_127[6]	ksv_byte_127[5]	ksv_byte_127[4]	ksv_byte_127[3]	ksv_byte_127[2]	ksv_byte_127[1]	ksv_byte_127[0]

1.4 HDMI RX INFOFRAME MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x00	avi_inf_pb_0_1	r	avi_inf_pb[7]	avi_inf_pb[6]	avi_inf_pb[5]	avi_inf_pb[4]	avi_inf_pb[3]	avi_inf_pb[2]	avi_inf_pb[1]	avi_inf_pb[0]
0x01	0x00	avi_inf_pb_0_2	r	avi_inf_pb[15]	avi_inf_pb[14]	avi_inf_pb[13]	avi_inf_pb[12]	avi_inf_pb[11]	avi_inf_pb[10]	avi_inf_pb[9]	avi_inf_pb[8]
0x02	0x00	avi_inf_pb_0_3	r	avi_inf_pb[23]	avi_inf_pb[22]	avi_inf_pb[21]	avi_inf_pb[20]	avi_inf_pb[19]	avi_inf_pb[18]	avi_inf_pb[17]	avi_inf_pb[16]
0x03	0x00	avi_inf_pb_0_4	r	avi_inf_pb[31]	avi_inf_pb[30]	avi_inf_pb[29]	avi_inf_pb[28]	avi_inf_pb[27]	avi_inf_pb[26]	avi_inf_pb[25]	avi_inf_pb[24]
0x04	0x00	avi_inf_pb_0_5	r	avi_inf_pb[39]	avi_inf_pb[38]	avi_inf_pb[37]	avi_inf_pb[36]	avi_inf_pb[35]	avi_inf_pb[34]	avi_inf_pb[33]	avi_inf_pb[32]
0x05	0x00	avi_inf_pb_0_6	r	avi_inf_pb[47]	avi_inf_pb[46]	avi_inf_pb[45]	avi_inf_pb[44]	avi_inf_pb[43]	avi_inf_pb[42]	avi_inf_pb[41]	avi_inf_pb[40]
0x06	0x00	avi_inf_pb_0_7	r	avi_inf_pb[55]	avi_inf_pb[54]	avi_inf_pb[53]	avi_inf_pb[52]	avi_inf_pb[51]	avi_inf_pb[50]	avi_inf_pb[49]	avi_inf_pb[48]
0x07	0x00	avi_inf_pb_0_8	r	avi_inf_pb[63]	avi_inf_pb[62]	avi_inf_pb[61]	avi_inf_pb[60]	avi_inf_pb[59]	avi_inf_pb[58]	avi_inf_pb[57]	avi_inf_pb[56]
0x08	0x00	avi_inf_pb_0_9	r	avi_inf_pb[71]	avi_inf_pb[70]	avi_inf_pb[69]	avi_inf_pb[68]	avi_inf_pb[67]	avi_inf_pb[66]	avi_inf_pb[65]	avi_inf_pb[64]
0x09	0x00	avi_inf_pb_0_10	r	avi_inf_pb[79]	avi_inf_pb[78]	avi_inf_pb[77]	avi_inf_pb[76]	avi_inf_pb[75]	avi_inf_pb[74]	avi_inf_pb[73]	avi_inf_pb[72]
0x0A	0x00	avi_inf_pb_0_11	r	avi_inf_pb[87]	avi_inf_pb[86]	avi_inf_pb[85]	avi_inf_pb[84]	avi_inf_pb[83]	avi_inf_pb[82]	avi_inf_pb[81]	avi_inf_pb[80]
0x0B	0x00	avi_inf_pb_0_12	r	avi_inf_pb[95]	avi_inf_pb[94]	avi_inf_pb[93]	avi_inf_pb[92]	avi_inf_pb[91]	avi_inf_pb[90]	avi_inf_pb[89]	avi_inf_pb[88]
0x0C	0x00	avi_inf_pb_0_13	r	avi_inf_pb[103]	avi_inf_pb[102]	avi_inf_pb[101]	avi_inf_pb[100]	avi_inf_pb[99]	avi_inf_pb[98]	avi_inf_pb[97]	avi_inf_pb[96]
0x0D	0x00	avi_inf_pb_0_14	r	avi_inf_pb[111]	avi_inf_pb[110]	avi_inf_pb[109]	avi_inf_pb[108]	avi_inf_pb[107]	avi_inf_pb[106]	avi_inf_pb[105]	avi_inf_pb[104]
0x0E	0x00	avi_inf_pb_0_15	r	avi_inf_pb[119]	avi_inf_pb[118]	avi_inf_pb[117]	avi_inf_pb[116]	avi_inf_pb[115]	avi_inf_pb[114]	avi_inf_pb[113]	avi_inf_pb[112]
0x0F	0x00	avi_inf_pb_0_16	r	avi_inf_pb[127]	avi_inf_pb[126]	avi_inf_pb[125]	avi_inf_pb[124]	avi_inf_pb[123]	avi_inf_pb[122]	avi_inf_pb[121]	avi_inf_pb[120]
0x10	0x00	avi_inf_pb_0_17	r	avi_inf_pb[135]	avi_inf_pb[134]	avi_inf_pb[133]	avi_inf_pb[132]	avi_inf_pb[131]	avi_inf_pb[130]	avi_inf_pb[129]	avi_inf_pb[128]
0x11	0x00	avi_inf_pb_0_18	r	avi_inf_pb[143]	avi_inf_pb[142]	avi_inf_pb[141]	avi_inf_pb[140]	avi_inf_pb[139]	avi_inf_pb[138]	avi_inf_pb[137]	avi_inf_pb[136]
0x12	0x00	avi_inf_pb_0_19	r	avi_inf_pb[151]	avi_inf_pb[150]	avi_inf_pb[149]	avi_inf_pb[148]	avi_inf_pb[147]	avi_inf_pb[146]	avi_inf_pb[145]	avi_inf_pb[144]
0x13	0x00	avi_inf_pb_0_20	r	avi_inf_pb[159]	avi_inf_pb[158]	avi_inf_pb[157]	avi_inf_pb[156]	avi_inf_pb[155]	avi_inf_pb[154]	avi_inf_pb[153]	avi_inf_pb[152]
0x14	0x00	avi_inf_pb_0_21	r	avi_inf_pb[167]	avi_inf_pb[166]	avi_inf_pb[165]	avi_inf_pb[164]	avi_inf_pb[163]	avi_inf_pb[162]	avi_inf_pb[161]	avi_inf_pb[160]
0x15	0x00	avi_inf_pb_0_22	r	avi_inf_pb[175]	avi_inf_pb[174]	avi_inf_pb[173]	avi_inf_pb[172]	avi_inf_pb[171]	avi_inf_pb[170]	avi_inf_pb[169]	avi_inf_pb[168]
0x16	0x00	avi_inf_pb_0_23	r	avi_inf_pb[183]	avi_inf_pb[182]	avi_inf_pb[181]	avi_inf_pb[180]	avi_inf_pb[179]	avi_inf_pb[178]	avi_inf_pb[177]	avi_inf_pb[176]
0x17	0x00	avi_inf_pb_0_24	r	avi_inf_pb[191]	avi_inf_pb[190]	avi_inf_pb[189]	avi_inf_pb[188]	avi_inf_pb[187]	avi_inf_pb[186]	avi_inf_pb[185]	avi_inf_pb[184]
0x18	0x00	avi_inf_pb_0_25	r	avi_inf_pb[199]	avi_inf_pb[198]	avi_inf_pb[197]	avi_inf_pb[196]	avi_inf_pb[195]	avi_inf_pb[194]	avi_inf_pb[193]	avi_inf_pb[192]
0x19	0x00	avi_inf_pb_0_26	r	avi_inf_pb[207]	avi_inf_pb[206]	avi_inf_pb[205]	avi_inf_pb[204]	avi_inf_pb[203]	avi_inf_pb[202]	avi_inf_pb[201]	avi_inf_pb[200]
0x1A	0x00	avi_inf_pb_0_27	r	avi_inf_pb[215]	avi_inf_pb[214]	avi_inf_pb[213]	avi_inf_pb[212]	avi_inf_pb[211]	avi_inf_pb[210]	avi_inf_pb[209]	avi_inf_pb[208]
0x1B	0x00	avi_inf_pb_0_28	r	avi_inf_pb[223]	avi_inf_pb[222]	avi_inf_pb[221]	avi_inf_pb[220]	avi_inf_pb[219]	avi_inf_pb[218]	avi_inf_pb[217]	avi_inf_pb[216]
0x1C	0x00	aud_inf_pb_0_1	r	aud_inf_pb[7]	aud_inf_pb[6]	aud_inf_pb[5]	aud_inf_pb[4]	aud_inf_pb[3]	aud_inf_pb[2]	aud_inf_pb[1]	aud_inf_pb[0]
0x1D	0x00	aud_inf_pb_0_2	r	aud_inf_pb[15]	aud_inf_pb[14]	aud_inf_pb[13]	aud_inf_pb[12]	aud_inf_pb[11]	aud_inf_pb[10]	aud_inf_pb[9]	aud_inf_pb[8]
0x1E	0x00	aud_inf_pb_0_3	r	aud_inf_pb[23]	aud_inf_pb[22]	aud_inf_pb[21]	aud_inf_pb[20]	aud_inf_pb[19]	aud_inf_pb[18]	aud_inf_pb[17]	aud_inf_pb[16]
0x1F	0x00	aud_inf_pb_0_4	r	aud_inf_pb[31]	aud_inf_pb[30]	aud_inf_pb[29]	aud_inf_pb[28]	aud_inf_pb[27]	aud_inf_pb[26]	aud_inf_pb[25]	aud_inf_pb[24]
0x20	0x00	aud_inf_pb_0_5	r	aud_inf_pb[39]	aud_inf_pb[38]	aud_inf_pb[37]	aud_inf_pb[36]	aud_inf_pb[35]	aud_inf_pb[34]	aud_inf_pb[33]	aud_inf_pb[32]
0x21	0x00	aud_inf_pb_0_6	r	aud_inf_pb[47]	aud_inf_pb[46]	aud_inf_pb[45]	aud_inf_pb[44]	aud_inf_pb[43]	aud_inf_pb[42]	aud_inf_pb[41]	aud_inf_pb[40]
0x22	0x00	aud_inf_pb_0_7	r	aud_inf_pb[55]	aud_inf_pb[54]	aud_inf_pb[53]	aud_inf_pb[52]	aud_inf_pb[51]	aud_inf_pb[50]	aud_inf_pb[49]	aud_inf_pb[48]
0x23	0x00	aud_inf_pb_0_8	r	aud_inf_pb[63]	aud_inf_pb[62]	aud_inf_pb[61]	aud_inf_pb[60]	aud_inf_pb[59]	aud_inf_pb[58]	aud_inf_pb[57]	aud_inf_pb[56]
0x24	0x00	aud_inf_pb_0_9	r	aud_inf_pb[71]	aud_inf_pb[70]	aud_inf_pb[69]	aud_inf_pb[68]	aud_inf_pb[67]	aud_inf_pb[66]	aud_inf_pb[65]	aud_inf_pb[64]
0x25	0x00	aud_inf_pb_0_10	r	aud_inf_pb[79]	aud_inf_pb[78]	aud_inf_pb[77]	aud_inf_pb[76]	aud_inf_pb[75]	aud_inf_pb[74]	aud_inf_pb[73]	aud_inf_pb[72]
0x26	0x00	aud_inf_pb_0_11	r	aud_inf_pb[87]	aud_inf_pb[86]	aud_inf_pb[85]	aud_inf_pb[84]	aud_inf_pb[83]	aud_inf_pb[82]	aud_inf_pb[81]	aud_inf_pb[80]
0x27	0x00	aud_inf_pb_0_12	r	aud_inf_pb[95]	aud_inf_pb[94]	aud_inf_pb[93]	aud_inf_pb[92]	aud_inf_pb[91]	aud_inf_pb[90]	aud_inf_pb[89]	aud_inf_pb[88]
0x28	0x00	aud_inf_pb_0_13	r	aud_inf_pb[103]	aud_inf_pb[102]	aud_inf_pb[101]	aud_inf_pb[100]	aud_inf_pb[99]	aud_inf_pb[98]	aud_inf_pb[97]	aud_inf_pb[96]
0x29	0x00	aud_inf_pb_0_14	r	aud_inf_pb[111]	aud_inf_pb[110]	aud_inf_pb[109]	aud_inf_pb[108]	aud_inf_pb[107]	aud_inf_pb[106]	aud_inf_pb[105]	aud_inf_pb[104]
0x2A	0x00	spd_inf_pb_0_1	r	spd_inf_pb[7]	spd_inf_pb[6]	spd_inf_pb[5]	spd_inf_pb[4]	spd_inf_pb[3]	spd_inf_pb[2]	spd_inf_pb[1]	spd_inf_pb[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x2B	0x00	spd_inf_pb_0_2	r	spd_inf_pb[15]	spd_inf_pb[14]	spd_inf_pb[13]	spd_inf_pb[12]	spd_inf_pb[11]	spd_inf_pb[10]	spd_inf_pb[9]	spd_inf_pb[8]
0x2C	0x00	spd_inf_pb_0_3	r	spd_inf_pb[23]	spd_inf_pb[22]	spd_inf_pb[21]	spd_inf_pb[20]	spd_inf_pb[19]	spd_inf_pb[18]	spd_inf_pb[17]	spd_inf_pb[16]
0x2D	0x00	spd_inf_pb_0_4	r	spd_inf_pb[31]	spd_inf_pb[30]	spd_inf_pb[29]	spd_inf_pb[28]	spd_inf_pb[27]	spd_inf_pb[26]	spd_inf_pb[25]	spd_inf_pb[24]
0x2E	0x00	spd_inf_pb_0_5	r	spd_inf_pb[39]	spd_inf_pb[38]	spd_inf_pb[37]	spd_inf_pb[36]	spd_inf_pb[35]	spd_inf_pb[34]	spd_inf_pb[33]	spd_inf_pb[32]
0x2F	0x00	spd_inf_pb_0_6	r	spd_inf_pb[47]	spd_inf_pb[46]	spd_inf_pb[45]	spd_inf_pb[44]	spd_inf_pb[43]	spd_inf_pb[42]	spd_inf_pb[41]	spd_inf_pb[40]
0x30	0x00	spd_inf_pb_0_7	r	spd_inf_pb[55]	spd_inf_pb[54]	spd_inf_pb[53]	spd_inf_pb[52]	spd_inf_pb[51]	spd_inf_pb[50]	spd_inf_pb[49]	spd_inf_pb[48]
0x31	0x00	spd_inf_pb_0_8	r	spd_inf_pb[63]	spd_inf_pb[62]	spd_inf_pb[61]	spd_inf_pb[60]	spd_inf_pb[59]	spd_inf_pb[58]	spd_inf_pb[57]	spd_inf_pb[56]
0x32	0x00	spd_inf_pb_0_9	r	spd_inf_pb[71]	spd_inf_pb[70]	spd_inf_pb[69]	spd_inf_pb[68]	spd_inf_pb[67]	spd_inf_pb[66]	spd_inf_pb[65]	spd_inf_pb[64]
0x33	0x00	spd_inf_pb_0_10	r	spd_inf_pb[79]	spd_inf_pb[78]	spd_inf_pb[77]	spd_inf_pb[76]	spd_inf_pb[75]	spd_inf_pb[74]	spd_inf_pb[73]	spd_inf_pb[72]
0x34	0x00	spd_inf_pb_0_11	r	spd_inf_pb[87]	spd_inf_pb[86]	spd_inf_pb[85]	spd_inf_pb[84]	spd_inf_pb[83]	spd_inf_pb[82]	spd_inf_pb[81]	spd_inf_pb[80]
0x35	0x00	spd_inf_pb_0_12	r	spd_inf_pb[95]	spd_inf_pb[94]	spd_inf_pb[93]	spd_inf_pb[92]	spd_inf_pb[91]	spd_inf_pb[90]	spd_inf_pb[89]	spd_inf_pb[88]
0x36	0x00	spd_inf_pb_0_13	r	spd_inf_pb[103]	spd_inf_pb[102]	spd_inf_pb[101]	spd_inf_pb[100]	spd_inf_pb[99]	spd_inf_pb[98]	spd_inf_pb[97]	spd_inf_pb[96]
0x37	0x00	spd_inf_pb_0_14	r	spd_inf_pb[111]	spd_inf_pb[110]	spd_inf_pb[109]	spd_inf_pb[108]	spd_inf_pb[107]	spd_inf_pb[106]	spd_inf_pb[105]	spd_inf_pb[104]
0x38	0x00	spd_inf_pb_0_15	r	spd_inf_pb[119]	spd_inf_pb[118]	spd_inf_pb[117]	spd_inf_pb[116]	spd_inf_pb[115]	spd_inf_pb[114]	spd_inf_pb[113]	spd_inf_pb[112]
0x39	0x00	spd_inf_pb_0_16	r	spd_inf_pb[127]	spd_inf_pb[126]	spd_inf_pb[125]	spd_inf_pb[124]	spd_inf_pb[123]	spd_inf_pb[122]	spd_inf_pb[121]	spd_inf_pb[120]
0x3A	0x00	spd_inf_pb_0_17	r	spd_inf_pb[135]	spd_inf_pb[134]	spd_inf_pb[133]	spd_inf_pb[132]	spd_inf_pb[131]	spd_inf_pb[130]	spd_inf_pb[129]	spd_inf_pb[128]
0x3B	0x00	spd_inf_pb_0_18	r	spd_inf_pb[143]	spd_inf_pb[142]	spd_inf_pb[141]	spd_inf_pb[140]	spd_inf_pb[139]	spd_inf_pb[138]	spd_inf_pb[137]	spd_inf_pb[136]
0x3C	0x00	spd_inf_pb_0_19	r	spd_inf_pb[151]	spd_inf_pb[150]	spd_inf_pb[149]	spd_inf_pb[148]	spd_inf_pb[147]	spd_inf_pb[146]	spd_inf_pb[145]	spd_inf_pb[144]
0x3D	0x00	spd_inf_pb_0_20	r	spd_inf_pb[159]	spd_inf_pb[158]	spd_inf_pb[157]	spd_inf_pb[156]	spd_inf_pb[155]	spd_inf_pb[154]	spd_inf_pb[153]	spd_inf_pb[152]
0x3E	0x00	spd_inf_pb_0_21	r	spd_inf_pb[167]	spd_inf_pb[166]	spd_inf_pb[165]	spd_inf_pb[164]	spd_inf_pb[163]	spd_inf_pb[162]	spd_inf_pb[161]	spd_inf_pb[160]
0x3F	0x00	spd_inf_pb_0_22	r	spd_inf_pb[175]	spd_inf_pb[174]	spd_inf_pb[173]	spd_inf_pb[172]	spd_inf_pb[171]	spd_inf_pb[170]	spd_inf_pb[169]	spd_inf_pb[168]
0x40	0x00	spd_inf_pb_0_23	r	spd_inf_pb[183]	spd_inf_pb[182]	spd_inf_pb[181]	spd_inf_pb[180]	spd_inf_pb[179]	spd_inf_pb[178]	spd_inf_pb[177]	spd_inf_pb[176]
0x41	0x00	spd_inf_pb_0_24	r	spd_inf_pb[191]	spd_inf_pb[190]	spd_inf_pb[189]	spd_inf_pb[188]	spd_inf_pb[187]	spd_inf_pb[186]	spd_inf_pb[185]	spd_inf_pb[184]
0x42	0x00	spd_inf_pb_0_25	r	spd_inf_pb[199]	spd_inf_pb[198]	spd_inf_pb[197]	spd_inf_pb[196]	spd_inf_pb[195]	spd_inf_pb[194]	spd_inf_pb[193]	spd_inf_pb[192]
0x43	0x00	spd_inf_pb_0_26	r	spd_inf_pb[207]	spd_inf_pb[206]	spd_inf_pb[205]	spd_inf_pb[204]	spd_inf_pb[203]	spd_inf_pb[202]	spd_inf_pb[201]	spd_inf_pb[200]
0x44	0x00	spd_inf_pb_0_27	r	spd_inf_pb[215]	spd_inf_pb[214]	spd_inf_pb[213]	spd_inf_pb[212]	spd_inf_pb[211]	spd_inf_pb[210]	spd_inf_pb[209]	spd_inf_pb[208]
0x45	0x00	spd_inf_pb_0_28	r	spd_inf_pb[223]	spd_inf_pb[222]	spd_inf_pb[221]	spd_inf_pb[220]	spd_inf_pb[219]	spd_inf_pb[218]	spd_inf_pb[217]	spd_inf_pb[216]
0x46	0x00	ms_inf_pb_0_1	r	ms_inf_pb[7]	ms_inf_pb[6]	ms_inf_pb[5]	ms_inf_pb[4]	ms_inf_pb[3]	ms_inf_pb[2]	ms_inf_pb[1]	ms_inf_pb[0]
0x47	0x00	ms_inf_pb_0_2	r	ms_inf_pb[15]	ms_inf_pb[14]	ms_inf_pb[13]	ms_inf_pb[12]	ms_inf_pb[11]	ms_inf_pb[10]	ms_inf_pb[9]	ms_inf_pb[8]
0x48	0x00	ms_inf_pb_0_3	r	ms_inf_pb[23]	ms_inf_pb[22]	ms_inf_pb[21]	ms_inf_pb[20]	ms_inf_pb[19]	ms_inf_pb[18]	ms_inf_pb[17]	ms_inf_pb[16]
0x49	0x00	ms_inf_pb_0_4	r	ms_inf_pb[31]	ms_inf_pb[30]	ms_inf_pb[29]	ms_inf_pb[28]	ms_inf_pb[27]	ms_inf_pb[26]	ms_inf_pb[25]	ms_inf_pb[24]
0x4A	0x00	ms_inf_pb_0_5	r	ms_inf_pb[39]	ms_inf_pb[38]	ms_inf_pb[37]	ms_inf_pb[36]	ms_inf_pb[35]	ms_inf_pb[34]	ms_inf_pb[33]	ms_inf_pb[32]
0x4B	0x00	ms_inf_pb_0_6	r	ms_inf_pb[47]	ms_inf_pb[46]	ms_inf_pb[45]	ms_inf_pb[44]	ms_inf_pb[43]	ms_inf_pb[42]	ms_inf_pb[41]	ms_inf_pb[40]
0x4C	0x00	ms_inf_pb_0_7	r	ms_inf_pb[55]	ms_inf_pb[54]	ms_inf_pb[53]	ms_inf_pb[52]	ms_inf_pb[51]	ms_inf_pb[50]	ms_inf_pb[49]	ms_inf_pb[48]
0x4D	0x00	ms_inf_pb_0_8	r	ms_inf_pb[63]	ms_inf_pb[62]	ms_inf_pb[61]	ms_inf_pb[60]	ms_inf_pb[59]	ms_inf_pb[58]	ms_inf_pb[57]	ms_inf_pb[56]
0x4E	0x00	ms_inf_pb_0_9	r	ms_inf_pb[71]	ms_inf_pb[70]	ms_inf_pb[69]	ms_inf_pb[68]	ms_inf_pb[67]	ms_inf_pb[66]	ms_inf_pb[65]	ms_inf_pb[64]
0x4F	0x00	ms_inf_pb_0_10	r	ms_inf_pb[79]	ms_inf_pb[78]	ms_inf_pb[77]	ms_inf_pb[76]	ms_inf_pb[75]	ms_inf_pb[74]	ms_inf_pb[73]	ms_inf_pb[72]
0x50	0x00	ms_inf_pb_0_11	r	ms_inf_pb[87]	ms_inf_pb[86]	ms_inf_pb[85]	ms_inf_pb[84]	ms_inf_pb[83]	ms_inf_pb[82]	ms_inf_pb[81]	ms_inf_pb[80]
0x51	0x00	ms_inf_pb_0_12	r	ms_inf_pb[95]	ms_inf_pb[94]	ms_inf_pb[93]	ms_inf_pb[92]	ms_inf_pb[91]	ms_inf_pb[90]	ms_inf_pb[89]	ms_inf_pb[88]
0x52	0x00	ms_inf_pb_0_13	r	ms_inf_pb[103]	ms_inf_pb[102]	ms_inf_pb[101]	ms_inf_pb[100]	ms_inf_pb[99]	ms_inf_pb[98]	ms_inf_pb[97]	ms_inf_pb[96]
0x53	0x00	ms_inf_pb_0_14	r	ms_inf_pb[111]	ms_inf_pb[110]	ms_inf_pb[109]	ms_inf_pb[108]	ms_inf_pb[107]	ms_inf_pb[106]	ms_inf_pb[105]	ms_inf_pb[104]
0x54	0x00	vs_inf_pb_0_1	r	vs_inf_pb[7]	vs_inf_pb[6]	vs_inf_pb[5]	vs_inf_pb[4]	vs_inf_pb[3]	vs_inf_pb[2]	vs_inf_pb[1]	vs_inf_pb[0]
0x55	0x00	vs_inf_pb_0_2	r	vs_inf_pb[15]	vs_inf_pb[14]	vs_inf_pb[13]	vs_inf_pb[12]	vs_inf_pb[11]	vs_inf_pb[10]	vs_inf_pb[9]	vs_inf_pb[8]
0x56	0x00	vs_inf_pb_0_3	r	vs_inf_pb[23]	vs_inf_pb[22]	vs_inf_pb[21]	vs_inf_pb[20]	vs_inf_pb[19]	vs_inf_pb[18]	vs_inf_pb[17]	vs_inf_pb[16]
0x57	0x00	vs_inf_pb_0_4	r	vs_inf_pb[31]	vs_inf_pb[30]	vs_inf_pb[29]	vs_inf_pb[28]	vs_inf_pb[27]	vs_inf_pb[26]	vs_inf_pb[25]	vs_inf_pb[24]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x58	0x00	vs_inf_pb_0_5	r	vs_inf_pb[39]	vs_inf_pb[38]	vs_inf_pb[37]	vs_inf_pb[36]	vs_inf_pb[35]	vs_inf_pb[34]	vs_inf_pb[33]	vs_inf_pb[32]
0x59	0x00	vs_inf_pb_0_6	r	vs_inf_pb[47]	vs_inf_pb[46]	vs_inf_pb[45]	vs_inf_pb[44]	vs_inf_pb[43]	vs_inf_pb[42]	vs_inf_pb[41]	vs_inf_pb[40]
0x5A	0x00	vs_inf_pb_0_7	r	vs_inf_pb[55]	vs_inf_pb[54]	vs_inf_pb[53]	vs_inf_pb[52]	vs_inf_pb[51]	vs_inf_pb[50]	vs_inf_pb[49]	vs_inf_pb[48]
0x5B	0x00	vs_inf_pb_0_8	r	vs_inf_pb[63]	vs_inf_pb[62]	vs_inf_pb[61]	vs_inf_pb[60]	vs_inf_pb[59]	vs_inf_pb[58]	vs_inf_pb[57]	vs_inf_pb[56]
0x5C	0x00	vs_inf_pb_0_9	r	vs_inf_pb[71]	vs_inf_pb[70]	vs_inf_pb[69]	vs_inf_pb[68]	vs_inf_pb[67]	vs_inf_pb[66]	vs_inf_pb[65]	vs_inf_pb[64]
0x5D	0x00	vs_inf_pb_0_10	r	vs_inf_pb[79]	vs_inf_pb[78]	vs_inf_pb[77]	vs_inf_pb[76]	vs_inf_pb[75]	vs_inf_pb[74]	vs_inf_pb[73]	vs_inf_pb[72]
0x5E	0x00	vs_inf_pb_0_11	r	vs_inf_pb[87]	vs_inf_pb[86]	vs_inf_pb[85]	vs_inf_pb[84]	vs_inf_pb[83]	vs_inf_pb[82]	vs_inf_pb[81]	vs_inf_pb[80]
0x5F	0x00	vs_inf_pb_0_12	r	vs_inf_pb[95]	vs_inf_pb[94]	vs_inf_pb[93]	vs_inf_pb[92]	vs_inf_pb[91]	vs_inf_pb[90]	vs_inf_pb[89]	vs_inf_pb[88]
0x60	0x00	vs_inf_pb_0_13	r	vs_inf_pb[103]	vs_inf_pb[102]	vs_inf_pb[101]	vs_inf_pb[100]	vs_inf_pb[99]	vs_inf_pb[98]	vs_inf_pb[97]	vs_inf_pb[96]
0x61	0x00	vs_inf_pb_0_14	r	vs_inf_pb[111]	vs_inf_pb[110]	vs_inf_pb[109]	vs_inf_pb[108]	vs_inf_pb[107]	vs_inf_pb[106]	vs_inf_pb[105]	vs_inf_pb[104]
0x62	0x00	vs_inf_pb_0_15	r	vs_inf_pb[119]	vs_inf_pb[118]	vs_inf_pb[117]	vs_inf_pb[116]	vs_inf_pb[115]	vs_inf_pb[114]	vs_inf_pb[113]	vs_inf_pb[112]
0x63	0x00	vs_inf_pb_0_16	r	vs_inf_pb[127]	vs_inf_pb[126]	vs_inf_pb[125]	vs_inf_pb[124]	vs_inf_pb[123]	vs_inf_pb[122]	vs_inf_pb[121]	vs_inf_pb[120]
0x64	0x00	vs_inf_pb_0_17	r	vs_inf_pb[135]	vs_inf_pb[134]	vs_inf_pb[133]	vs_inf_pb[132]	vs_inf_pb[131]	vs_inf_pb[130]	vs_inf_pb[129]	vs_inf_pb[128]
0x65	0x00	vs_inf_pb_0_18	r	vs_inf_pb[143]	vs_inf_pb[142]	vs_inf_pb[141]	vs_inf_pb[140]	vs_inf_pb[139]	vs_inf_pb[138]	vs_inf_pb[137]	vs_inf_pb[136]
0x66	0x00	vs_inf_pb_0_19	r	vs_inf_pb[151]	vs_inf_pb[150]	vs_inf_pb[149]	vs_inf_pb[148]	vs_inf_pb[147]	vs_inf_pb[146]	vs_inf_pb[145]	vs_inf_pb[144]
0x67	0x00	vs_inf_pb_0_20	r	vs_inf_pb[159]	vs_inf_pb[158]	vs_inf_pb[157]	vs_inf_pb[156]	vs_inf_pb[155]	vs_inf_pb[154]	vs_inf_pb[153]	vs_inf_pb[152]
0x68	0x00	vs_inf_pb_0_21	r	vs_inf_pb[167]	vs_inf_pb[166]	vs_inf_pb[165]	vs_inf_pb[164]	vs_inf_pb[163]	vs_inf_pb[162]	vs_inf_pb[161]	vs_inf_pb[160]
0x69	0x00	vs_inf_pb_0_22	r	vs_inf_pb[175]	vs_inf_pb[174]	vs_inf_pb[173]	vs_inf_pb[172]	vs_inf_pb[171]	vs_inf_pb[170]	vs_inf_pb[169]	vs_inf_pb[168]
0x6A	0x00	vs_inf_pb_0_23	r	vs_inf_pb[183]	vs_inf_pb[182]	vs_inf_pb[181]	vs_inf_pb[180]	vs_inf_pb[179]	vs_inf_pb[178]	vs_inf_pb[177]	vs_inf_pb[176]
0x6B	0x00	vs_inf_pb_0_24	r	vs_inf_pb[191]	vs_inf_pb[190]	vs_inf_pb[189]	vs_inf_pb[188]	vs_inf_pb[187]	vs_inf_pb[186]	vs_inf_pb[185]	vs_inf_pb[184]
0x6C	0x00	vs_inf_pb_0_25	r	vs_inf_pb[199]	vs_inf_pb[198]	vs_inf_pb[197]	vs_inf_pb[196]	vs_inf_pb[195]	vs_inf_pb[194]	vs_inf_pb[193]	vs_inf_pb[192]
0x6D	0x00	vs_inf_pb_0_26	r	vs_inf_pb[207]	vs_inf_pb[206]	vs_inf_pb[205]	vs_inf_pb[204]	vs_inf_pb[203]	vs_inf_pb[202]	vs_inf_pb[201]	vs_inf_pb[200]
0x6E	0x00	vs_inf_pb_0_27	r	vs_inf_pb[215]	vs_inf_pb[214]	vs_inf_pb[213]	vs_inf_pb[212]	vs_inf_pb[211]	vs_inf_pb[210]	vs_inf_pb[209]	vs_inf_pb[208]
0x6F	0x00	vs_inf_pb_0_28	r	vs_inf_pb[223]	vs_inf_pb[222]	vs_inf_pb[221]	vs_inf_pb[220]	vs_inf_pb[219]	vs_inf_pb[218]	vs_inf_pb[217]	vs_inf_pb[216]
0x70	0x00	acp_pb_0_1	r	acp_pb[7]	acp_pb[6]	acp_pb[5]	acp_pb[4]	acp_pb[3]	acp_pb[2]	acp_pb[1]	acp_pb[0]
0x71	0x00	acp_pb_0_2	r	acp_pb[15]	acp_pb[14]	acp_pb[13]	acp_pb[12]	acp_pb[11]	acp_pb[10]	acp_pb[9]	acp_pb[8]
0x72	0x00	acp_pb_0_3	r	acp_pb[23]	acp_pb[22]	acp_pb[21]	acp_pb[20]	acp_pb[19]	acp_pb[18]	acp_pb[17]	acp_pb[16]
0x73	0x00	acp_pb_0_4	r	acp_pb[31]	acp_pb[30]	acp_pb[29]	acp_pb[28]	acp_pb[27]	acp_pb[26]	acp_pb[25]	acp_pb[24]
0x74	0x00	acp_pb_0_5	r	acp_pb[39]	acp_pb[38]	acp_pb[37]	acp_pb[36]	acp_pb[35]	acp_pb[34]	acp_pb[33]	acp_pb[32]
0x75	0x00	acp_pb_0_6	r	acp_pb[47]	acp_pb[46]	acp_pb[45]	acp_pb[44]	acp_pb[43]	acp_pb[42]	acp_pb[41]	acp_pb[40]
0x76	0x00	acp_pb_0_7	r	acp_pb[55]	acp_pb[54]	acp_pb[53]	acp_pb[52]	acp_pb[51]	acp_pb[50]	acp_pb[49]	acp_pb[48]
0x77	0x00	acp_pb_0_8	r	acp_pb[63]	acp_pb[62]	acp_pb[61]	acp_pb[60]	acp_pb[59]	acp_pb[58]	acp_pb[57]	acp_pb[56]
0x78	0x00	acp_pb_0_9	r	acp_pb[71]	acp_pb[70]	acp_pb[69]	acp_pb[68]	acp_pb[67]	acp_pb[66]	acp_pb[65]	acp_pb[64]
0x79	0x00	acp_pb_0_10	r	acp_pb[79]	acp_pb[78]	acp_pb[77]	acp_pb[76]	acp_pb[75]	acp_pb[74]	acp_pb[73]	acp_pb[72]
0x7A	0x00	acp_pb_0_11	r	acp_pb[87]	acp_pb[86]	acp_pb[85]	acp_pb[84]	acp_pb[83]	acp_pb[82]	acp_pb[81]	acp_pb[80]
0x7B	0x00	acp_pb_0_12	r	acp_pb[95]	acp_pb[94]	acp_pb[93]	acp_pb[92]	acp_pb[91]	acp_pb[90]	acp_pb[89]	acp_pb[88]
0x7C	0x00	acp_pb_0_13	r	acp_pb[103]	acp_pb[102]	acp_pb[101]	acp_pb[100]	acp_pb[99]	acp_pb[98]	acp_pb[97]	acp_pb[96]
0x7D	0x00	acp_pb_0_14	r	acp_pb[111]	acp_pb[110]	acp_pb[109]	acp_pb[108]	acp_pb[107]	acp_pb[106]	acp_pb[105]	acp_pb[104]
0x7E	0x00	acp_pb_0_15	r	acp_pb[119]	acp_pb[118]	acp_pb[117]	acp_pb[116]	acp_pb[115]	acp_pb[114]	acp_pb[113]	acp_pb[112]
0x7F	0x00	acp_pb_0_16	r	acp_pb[127]	acp_pb[126]	acp_pb[125]	acp_pb[124]	acp_pb[123]	acp_pb[122]	acp_pb[121]	acp_pb[120]
0x80	0x00	acp_pb_0_17	r	acp_pb[135]	acp_pb[134]	acp_pb[133]	acp_pb[132]	acp_pb[131]	acp_pb[130]	acp_pb[129]	acp_pb[128]
0x81	0x00	acp_pb_0_18	r	acp_pb[143]	acp_pb[142]	acp_pb[141]	acp_pb[140]	acp_pb[139]	acp_pb[138]	acp_pb[137]	acp_pb[136]
0x82	0x00	acp_pb_0_19	r	acp_pb[151]	acp_pb[150]	acp_pb[149]	acp_pb[148]	acp_pb[147]	acp_pb[146]	acp_pb[145]	acp_pb[144]
0x83	0x00	acp_pb_0_20	r	acp_pb[159]	acp_pb[158]	acp_pb[157]	acp_pb[156]	acp_pb[155]	acp_pb[154]	acp_pb[153]	acp_pb[152]
0x84	0x00	acp_pb_0_21	r	acp_pb[167]	acp_pb[166]	acp_pb[165]	acp_pb[164]	acp_pb[163]	acp_pb[162]	acp_pb[161]	acp_pb[160]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x85	0x00	acp_pb_0_22	r	acp_pb[175]	acp_pb[174]	acp_pb[173]	acp_pb[172]	acp_pb[171]	acp_pb[170]	acp_pb[169]	acp_pb[168]
0x86	0x00	acp_pb_0_23	r	acp_pb[183]	acp_pb[182]	acp_pb[181]	acp_pb[180]	acp_pb[179]	acp_pb[178]	acp_pb[177]	acp_pb[176]
0x87	0x00	acp_pb_0_24	r	acp_pb[191]	acp_pb[190]	acp_pb[189]	acp_pb[188]	acp_pb[187]	acp_pb[186]	acp_pb[185]	acp_pb[184]
0x88	0x00	acp_pb_0_25	r	acp_pb[199]	acp_pb[198]	acp_pb[197]	acp_pb[196]	acp_pb[195]	acp_pb[194]	acp_pb[193]	acp_pb[192]
0x89	0x00	acp_pb_0_26	r	acp_pb[207]	acp_pb[206]	acp_pb[205]	acp_pb[204]	acp_pb[203]	acp_pb[202]	acp_pb[201]	acp_pb[200]
0x8A	0x00	acp_pb_0_27	r	acp_pb[215]	acp_pb[214]	acp_pb[213]	acp_pb[212]	acp_pb[211]	acp_pb[210]	acp_pb[209]	acp_pb[208]
0x8B	0x00	acp_pb_0_28	r	acp_pb[223]	acp_pb[222]	acp_pb[221]	acp_pb[220]	acp_pb[219]	acp_pb[218]	acp_pb[217]	acp_pb[216]
0x8C	0x00	isrc1_pb_0_1	r	isrc1_pb[7]	isrc1_pb[6]	isrc1_pb[5]	isrc1_pb[4]	isrc1_pb[3]	isrc1_pb[2]	isrc1_pb[1]	isrc1_pb[0]
0x8D	0x00	isrc1_pb_0_2	r	isrc1_pb[15]	isrc1_pb[14]	isrc1_pb[13]	isrc1_pb[12]	isrc1_pb[11]	isrc1_pb[10]	isrc1_pb[9]	isrc1_pb[8]
0x8E	0x00	isrc1_pb_0_3	r	isrc1_pb[23]	isrc1_pb[22]	isrc1_pb[21]	isrc1_pb[20]	isrc1_pb[19]	isrc1_pb[18]	isrc1_pb[17]	isrc1_pb[16]
0x8F	0x00	isrc1_pb_0_4	r	isrc1_pb[31]	isrc1_pb[30]	isrc1_pb[29]	isrc1_pb[28]	isrc1_pb[27]	isrc1_pb[26]	isrc1_pb[25]	isrc1_pb[24]
0x90	0x00	isrc1_pb_0_5	r	isrc1_pb[39]	isrc1_pb[38]	isrc1_pb[37]	isrc1_pb[36]	isrc1_pb[35]	isrc1_pb[34]	isrc1_pb[33]	isrc1_pb[32]
0x91	0x00	isrc1_pb_0_6	r	isrc1_pb[47]	isrc1_pb[46]	isrc1_pb[45]	isrc1_pb[44]	isrc1_pb[43]	isrc1_pb[42]	isrc1_pb[41]	isrc1_pb[40]
0x92	0x00	isrc1_pb_0_7	r	isrc1_pb[55]	isrc1_pb[54]	isrc1_pb[53]	isrc1_pb[52]	isrc1_pb[51]	isrc1_pb[50]	isrc1_pb[49]	isrc1_pb[48]
0x93	0x00	isrc1_pb_0_8	r	isrc1_pb[63]	isrc1_pb[62]	isrc1_pb[61]	isrc1_pb[60]	isrc1_pb[59]	isrc1_pb[58]	isrc1_pb[57]	isrc1_pb[56]
0x94	0x00	isrc1_pb_0_9	r	isrc1_pb[71]	isrc1_pb[70]	isrc1_pb[69]	isrc1_pb[68]	isrc1_pb[67]	isrc1_pb[66]	isrc1_pb[65]	isrc1_pb[64]
0x95	0x00	isrc1_pb_0_10	r	isrc1_pb[79]	isrc1_pb[78]	isrc1_pb[77]	isrc1_pb[76]	isrc1_pb[75]	isrc1_pb[74]	isrc1_pb[73]	isrc1_pb[72]
0x96	0x00	isrc1_pb_0_11	r	isrc1_pb[87]	isrc1_pb[86]	isrc1_pb[85]	isrc1_pb[84]	isrc1_pb[83]	isrc1_pb[82]	isrc1_pb[81]	isrc1_pb[80]
0x97	0x00	isrc1_pb_0_12	r	isrc1_pb[95]	isrc1_pb[94]	isrc1_pb[93]	isrc1_pb[92]	isrc1_pb[91]	isrc1_pb[90]	isrc1_pb[89]	isrc1_pb[88]
0x98	0x00	isrc1_pb_0_13	r	isrc1_pb[103]	isrc1_pb[102]	isrc1_pb[101]	isrc1_pb[100]	isrc1_pb[99]	isrc1_pb[98]	isrc1_pb[97]	isrc1_pb[96]
0x99	0x00	isrc1_pb_0_14	r	isrc1_pb[111]	isrc1_pb[110]	isrc1_pb[109]	isrc1_pb[108]	isrc1_pb[107]	isrc1_pb[106]	isrc1_pb[105]	isrc1_pb[104]
0x9A	0x00	isrc1_pb_0_15	r	isrc1_pb[119]	isrc1_pb[118]	isrc1_pb[117]	isrc1_pb[116]	isrc1_pb[115]	isrc1_pb[114]	isrc1_pb[113]	isrc1_pb[112]
0x9B	0x00	isrc1_pb_0_16	r	isrc1_pb[127]	isrc1_pb[126]	isrc1_pb[125]	isrc1_pb[124]	isrc1_pb[123]	isrc1_pb[122]	isrc1_pb[121]	isrc1_pb[120]
0x9C	0x00	isrc1_pb_0_17	r	isrc1_pb[135]	isrc1_pb[134]	isrc1_pb[133]	isrc1_pb[132]	isrc1_pb[131]	isrc1_pb[130]	isrc1_pb[129]	isrc1_pb[128]
0x9D	0x00	isrc1_pb_0_18	r	isrc1_pb[143]	isrc1_pb[142]	isrc1_pb[141]	isrc1_pb[140]	isrc1_pb[139]	isrc1_pb[138]	isrc1_pb[137]	isrc1_pb[136]
0x9E	0x00	isrc1_pb_0_19	r	isrc1_pb[151]	isrc1_pb[150]	isrc1_pb[149]	isrc1_pb[148]	isrc1_pb[147]	isrc1_pb[146]	isrc1_pb[145]	isrc1_pb[144]
0x9F	0x00	isrc1_pb_0_20	r	isrc1_pb[159]	isrc1_pb[158]	isrc1_pb[157]	isrc1_pb[156]	isrc1_pb[155]	isrc1_pb[154]	isrc1_pb[153]	isrc1_pb[152]
0xA0	0x00	isrc1_pb_0_21	r	isrc1_pb[167]	isrc1_pb[166]	isrc1_pb[165]	isrc1_pb[164]	isrc1_pb[163]	isrc1_pb[162]	isrc1_pb[161]	isrc1_pb[160]
0xA1	0x00	isrc1_pb_0_22	r	isrc1_pb[175]	isrc1_pb[174]	isrc1_pb[173]	isrc1_pb[172]	isrc1_pb[171]	isrc1_pb[170]	isrc1_pb[169]	isrc1_pb[168]
0xA2	0x00	isrc1_pb_0_23	r	isrc1_pb[183]	isrc1_pb[182]	isrc1_pb[181]	isrc1_pb[180]	isrc1_pb[179]	isrc1_pb[178]	isrc1_pb[177]	isrc1_pb[176]
0xA3	0x00	isrc1_pb_0_24	r	isrc1_pb[191]	isrc1_pb[190]	isrc1_pb[189]	isrc1_pb[188]	isrc1_pb[187]	isrc1_pb[186]	isrc1_pb[185]	isrc1_pb[184]
0xA4	0x00	isrc1_pb_0_25	r	isrc1_pb[199]	isrc1_pb[198]	isrc1_pb[197]	isrc1_pb[196]	isrc1_pb[195]	isrc1_pb[194]	isrc1_pb[193]	isrc1_pb[192]
0xA5	0x00	isrc1_pb_0_26	r	isrc1_pb[207]	isrc1_pb[206]	isrc1_pb[205]	isrc1_pb[204]	isrc1_pb[203]	isrc1_pb[202]	isrc1_pb[201]	isrc1_pb[200]
0xA6	0x00	isrc1_pb_0_27	r	isrc1_pb[215]	isrc1_pb[214]	isrc1_pb[213]	isrc1_pb[212]	isrc1_pb[211]	isrc1_pb[210]	isrc1_pb[209]	isrc1_pb[208]
0xA7	0x00	isrc1_pb_0_28	r	isrc1_pb[223]	isrc1_pb[222]	isrc1_pb[221]	isrc1_pb[220]	isrc1_pb[219]	isrc1_pb[218]	isrc1_pb[217]	isrc1_pb[216]
0xA8	0x00	isrc2_pb_0_1	r	isrc2_pb[7]	isrc2_pb[6]	isrc2_pb[5]	isrc2_pb[4]	isrc2_pb[3]	isrc2_pb[2]	isrc2_pb[1]	isrc2_pb[0]
0xA9	0x00	isrc2_pb_0_2	r	isrc2_pb[15]	isrc2_pb[14]	isrc2_pb[13]	isrc2_pb[12]	isrc2_pb[11]	isrc2_pb[10]	isrc2_pb[9]	isrc2_pb[8]
0xAA	0x00	isrc2_pb_0_3	r	isrc2_pb[23]	isrc2_pb[22]	isrc2_pb[21]	isrc2_pb[20]	isrc2_pb[19]	isrc2_pb[18]	isrc2_pb[17]	isrc2_pb[16]
0xAB	0x00	isrc2_pb_0_4	r	isrc2_pb[31]	isrc2_pb[30]	isrc2_pb[29]	isrc2_pb[28]	isrc2_pb[27]	isrc2_pb[26]	isrc2_pb[25]	isrc2_pb[24]
0xAC	0x00	isrc2_pb_0_5	r	isrc2_pb[39]	isrc2_pb[38]	isrc2_pb[37]	isrc2_pb[36]	isrc2_pb[35]	isrc2_pb[34]	isrc2_pb[33]	isrc2_pb[32]
0xAD	0x00	isrc2_pb_0_6	r	isrc2_pb[47]	isrc2_pb[46]	isrc2_pb[45]	isrc2_pb[44]	isrc2_pb[43]	isrc2_pb[42]	isrc2_pb[41]	isrc2_pb[40]
0xAE	0x00	isrc2_pb_0_7	r	isrc2_pb[55]	isrc2_pb[54]	isrc2_pb[53]	isrc2_pb[52]	isrc2_pb[51]	isrc2_pb[50]	isrc2_pb[49]	isrc2_pb[48]
0xAF	0x00	isrc2_pb_0_8	r	isrc2_pb[63]	isrc2_pb[62]	isrc2_pb[61]	isrc2_pb[60]	isrc2_pb[59]	isrc2_pb[58]	isrc2_pb[57]	isrc2_pb[56]
0xB0	0x00	isrc2_pb_0_9	r	isrc2_pb[71]	isrc2_pb[70]	isrc2_pb[69]	isrc2_pb[68]	isrc2_pb[67]	isrc2_pb[66]	isrc2_pb[65]	isrc2_pb[64]
0xB1	0x00	isrc2_pb_0_10	r	isrc2_pb[79]	isrc2_pb[78]	isrc2_pb[77]	isrc2_pb[76]	isrc2_pb[75]	isrc2_pb[74]	isrc2_pb[73]	isrc2_pb[72]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xB2	0x00	isrc2_pb_0_11	r	isrc2_pb[87]	isrc2_pb[86]	isrc2_pb[85]	isrc2_pb[84]	isrc2_pb[83]	isrc2_pb[82]	isrc2_pb[81]	isrc2_pb[80]
0xB3	0x00	isrc2_pb_0_12	r	isrc2_pb[95]	isrc2_pb[94]	isrc2_pb[93]	isrc2_pb[92]	isrc2_pb[91]	isrc2_pb[90]	isrc2_pb[89]	isrc2_pb[88]
0xB4	0x00	isrc2_pb_0_13	r	isrc2_pb[103]	isrc2_pb[102]	isrc2_pb[101]	isrc2_pb[100]	isrc2_pb[99]	isrc2_pb[98]	isrc2_pb[97]	isrc2_pb[96]
0xB5	0x00	isrc2_pb_0_14	r	isrc2_pb[111]	isrc2_pb[110]	isrc2_pb[109]	isrc2_pb[108]	isrc2_pb[107]	isrc2_pb[106]	isrc2_pb[105]	isrc2_pb[104]
0xB6	0x00	isrc2_pb_0_15	r	isrc2_pb[119]	isrc2_pb[118]	isrc2_pb[117]	isrc2_pb[116]	isrc2_pb[115]	isrc2_pb[114]	isrc2_pb[113]	isrc2_pb[112]
0xB7	0x00	isrc2_pb_0_16	r	isrc2_pb[127]	isrc2_pb[126]	isrc2_pb[125]	isrc2_pb[124]	isrc2_pb[123]	isrc2_pb[122]	isrc2_pb[121]	isrc2_pb[120]
0xB8	0x00	isrc2_pb_0_17	r	isrc2_pb[135]	isrc2_pb[134]	isrc2_pb[133]	isrc2_pb[132]	isrc2_pb[131]	isrc2_pb[130]	isrc2_pb[129]	isrc2_pb[128]
0xB9	0x00	isrc2_pb_0_18	r	isrc2_pb[143]	isrc2_pb[142]	isrc2_pb[141]	isrc2_pb[140]	isrc2_pb[139]	isrc2_pb[138]	isrc2_pb[137]	isrc2_pb[136]
0xBA	0x00	isrc2_pb_0_19	r	isrc2_pb[151]	isrc2_pb[150]	isrc2_pb[149]	isrc2_pb[148]	isrc2_pb[147]	isrc2_pb[146]	isrc2_pb[145]	isrc2_pb[144]
0xBB	0x00	isrc2_pb_0_20	r	isrc2_pb[159]	isrc2_pb[158]	isrc2_pb[157]	isrc2_pb[156]	isrc2_pb[155]	isrc2_pb[154]	isrc2_pb[153]	isrc2_pb[152]
0xBC	0x00	isrc2_pb_0_21	r	isrc2_pb[167]	isrc2_pb[166]	isrc2_pb[165]	isrc2_pb[164]	isrc2_pb[163]	isrc2_pb[162]	isrc2_pb[161]	isrc2_pb[160]
0xBD	0x00	isrc2_pb_0_22	r	isrc2_pb[175]	isrc2_pb[174]	isrc2_pb[173]	isrc2_pb[172]	isrc2_pb[171]	isrc2_pb[170]	isrc2_pb[169]	isrc2_pb[168]
0xBE	0x00	isrc2_pb_0_23	r	isrc2_pb[183]	isrc2_pb[182]	isrc2_pb[181]	isrc2_pb[180]	isrc2_pb[179]	isrc2_pb[178]	isrc2_pb[177]	isrc2_pb[176]
0xBF	0x00	isrc2_pb_0_24	r	isrc2_pb[191]	isrc2_pb[190]	isrc2_pb[189]	isrc2_pb[188]	isrc2_pb[187]	isrc2_pb[186]	isrc2_pb[185]	isrc2_pb[184]
0xC0	0x00	isrc2_pb_0_25	r	isrc2_pb[199]	isrc2_pb[198]	isrc2_pb[197]	isrc2_pb[196]	isrc2_pb[195]	isrc2_pb[194]	isrc2_pb[193]	isrc2_pb[192]
0xC1	0x00	isrc2_pb_0_26	r	isrc2_pb[207]	isrc2_pb[206]	isrc2_pb[205]	isrc2_pb[204]	isrc2_pb[203]	isrc2_pb[202]	isrc2_pb[201]	isrc2_pb[200]
0xC2	0x00	isrc2_pb_0_27	r	isrc2_pb[215]	isrc2_pb[214]	isrc2_pb[213]	isrc2_pb[212]	isrc2_pb[211]	isrc2_pb[210]	isrc2_pb[209]	isrc2_pb[208]
0xC3	0x00	isrc2_pb_0_28	r	isrc2_pb[223]	isrc2_pb[222]	isrc2_pb[221]	isrc2_pb[220]	isrc2_pb[219]	isrc2_pb[218]	isrc2_pb[217]	isrc2_pb[216]
0xC4	0x00	gamut_mdata_pb_0_1	r	gbd[7]	gbd[6]	gbd[5]	gbd[4]	gbd[3]	gbd[2]	gbd[1]	gbd[0]
0xC5	0x00	gamut_mdata_pb_0_2	r	gbd[15]	gbd[14]	gbd[13]	gbd[12]	gbd[11]	gbd[10]	gbd[9]	gbd[8]
0xC6	0x00	gamut_mdata_pb_0_3	r	gbd[23]	gbd[22]	gbd[21]	gbd[20]	gbd[19]	gbd[18]	gbd[17]	gbd[16]
0xC7	0x00	gamut_mdata_pb_0_4	r	gbd[31]	gbd[30]	gbd[29]	gbd[28]	gbd[27]	gbd[26]	gbd[25]	gbd[24]
0xC8	0x00	gamut_mdata_pb_0_5	r	gbd[39]	gbd[38]	gbd[37]	gbd[36]	gbd[35]	gbd[34]	gbd[33]	gbd[32]
0xC9	0x00	gamut_mdata_pb_0_6	r	gbd[47]	gbd[46]	gbd[45]	gbd[44]	gbd[43]	gbd[42]	gbd[41]	gbd[40]
0xCA	0x00	gamut_mdata_pb_0_7	r	gbd[55]	gbd[54]	gbd[53]	gbd[52]	gbd[51]	gbd[50]	gbd[49]	gbd[48]
0xCB	0x00	gamut_mdata_pb_0_8	r	gbd[63]	gbd[62]	gbd[61]	gbd[60]	gbd[59]	gbd[58]	gbd[57]	gbd[56]
0xCC	0x00	gamut_mdata_pb_0_9	r	gbd[71]	gbd[70]	gbd[69]	gbd[68]	gbd[67]	gbd[66]	gbd[65]	gbd[64]
0xCD	0x00	gamut_mdata_pb_0_10	r	gbd[79]	gbd[78]	gbd[77]	gbd[76]	gbd[75]	gbd[74]	gbd[73]	gbd[72]
0xCE	0x00	gamut_mdata_pb_0_11	r	gbd[87]	gbd[86]	gbd[85]	gbd[84]	gbd[83]	gbd[82]	gbd[81]	gbd[80]
0xCF	0x00	gamut_mdata_pb_0_12	r	gbd[95]	gbd[94]	gbd[93]	gbd[92]	gbd[91]	gbd[90]	gbd[89]	gbd[88]
0xD0	0x00	gamut_mdata_pb_0_13	r	gbd[103]	gbd[102]	gbd[101]	gbd[100]	gbd[99]	gbd[98]	gbd[97]	gbd[96]
0xD1	0x00	gamut_mdata_pb_0_14	r	gbd[111]	gbd[110]	gbd[109]	gbd[108]	gbd[107]	gbd[106]	gbd[105]	gbd[104]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xD2	0x00	gamut_mdata_pb_0_15	r	gbd[119]	gbd[118]	gbd[117]	gbd[116]	gbd[115]	gbd[114]	gbd[113]	gbd[112]
0xD3	0x00	gamut_mdata_pb_0_16	r	gbd[127]	gbd[126]	gbd[125]	gbd[124]	gbd[123]	gbd[122]	gbd[121]	gbd[120]
0xD4	0x00	gamut_mdata_pb_0_17	r	gbd[135]	gbd[134]	gbd[133]	gbd[132]	gbd[131]	gbd[130]	gbd[129]	gbd[128]
0xD5	0x00	gamut_mdata_pb_0_18	r	gbd[143]	gbd[142]	gbd[141]	gbd[140]	gbd[139]	gbd[138]	gbd[137]	gbd[136]
0xD6	0x00	gamut_mdata_pb_0_19	r	gbd[151]	gbd[150]	gbd[149]	gbd[148]	gbd[147]	gbd[146]	gbd[145]	gbd[144]
0xD7	0x00	gamut_mdata_pb_0_20	r	gbd[159]	gbd[158]	gbd[157]	gbd[156]	gbd[155]	gbd[154]	gbd[153]	gbd[152]
0xD8	0x00	gamut_mdata_pb_0_21	r	gbd[167]	gbd[166]	gbd[165]	gbd[164]	gbd[163]	gbd[162]	gbd[161]	gbd[160]
0xD9	0x00	gamut_mdata_pb_0_22	r	gbd[175]	gbd[174]	gbd[173]	gbd[172]	gbd[171]	gbd[170]	gbd[169]	gbd[168]
0xDA	0x00	gamut_mdata_pb_0_23	r	gbd[183]	gbd[182]	gbd[181]	gbd[180]	gbd[179]	gbd[178]	gbd[177]	gbd[176]
0xDB	0x00	gamut_mdata_pb_0_24	r	gbd[191]	gbd[190]	gbd[189]	gbd[188]	gbd[187]	gbd[186]	gbd[185]	gbd[184]
0xDC	0x00	gamut_mdata_pb_0_25	r	gbd[199]	gbd[198]	gbd[197]	gbd[196]	gbd[195]	gbd[194]	gbd[193]	gbd[192]
0xDD	0x00	gamut_mdata_pb_0_26	r	gbd[207]	gbd[206]	gbd[205]	gbd[204]	gbd[203]	gbd[202]	gbd[201]	gbd[200]
0xDE	0x00	gamut_mdata_pb_0_27	r	gbd[215]	gbd[214]	gbd[213]	gbd[212]	gbd[211]	gbd[210]	gbd[209]	gbd[208]
0xDF	0x00	gamut_mdata_pb_0_28	r	gbd[223]	gbd[222]	gbd[221]	gbd[220]	gbd[219]	gbd[218]	gbd[217]	gbd[216]
0xE0	0x82	avi_packet_id	rw	avi_packet_id[7]	avi_packet_id[6]	avi_packet_id[5]	avi_packet_id[4]	avi_packet_id[3]	avi_packet_id[2]	avi_packet_id[1]	avi_packet_id[0]
0xE1	0x00	avi_inf_vers	r	avi_inf_vers[7]	avi_inf_vers[6]	avi_inf_vers[5]	avi_inf_vers[4]	avi_inf_vers[3]	avi_inf_vers[2]	avi_inf_vers[1]	avi_inf_vers[0]
0xE2	0x00	avi_inf_len	r	avi_inf_len[7]	avi_inf_len[6]	avi_inf_len[5]	avi_inf_len[4]	avi_inf_len[3]	avi_inf_len[2]	avi_inf_len[1]	avi_inf_len[0]
0xE3	0x84	aud_packet_id	rw	aud_packet_id[7]	aud_packet_id[6]	aud_packet_id[5]	aud_packet_id[4]	aud_packet_id[3]	aud_packet_id[2]	aud_packet_id[1]	aud_packet_id[0]
0xE4	0x00	aud_inf_vers	r	aud_inf_vers[7]	aud_inf_vers[6]	aud_inf_vers[5]	aud_inf_vers[4]	aud_inf_vers[3]	aud_inf_vers[2]	aud_inf_vers[1]	aud_inf_vers[0]
0xE5	0x00	aud_inf_len	r	aud_inf_len[7]	aud_inf_len[6]	aud_inf_len[5]	aud_inf_len[4]	aud_inf_len[3]	aud_inf_len[2]	aud_inf_len[1]	aud_inf_len[0]
0xE6	0x83	spd_packet_id	rw	spd_packet_id[7]	spd_packet_id[6]	spd_packet_id[5]	spd_packet_id[4]	spd_packet_id[3]	spd_packet_id[2]	spd_packet_id[1]	spd_packet_id[0]
0xE7	0x00	spd_inf_vers	r	spd_inf_vers[7]	spd_inf_vers[6]	spd_inf_vers[5]	spd_inf_vers[4]	spd_inf_vers[3]	spd_inf_vers[2]	spd_inf_vers[1]	spd_inf_vers[0]
0xE8	0x00	spd_inf_len	r	spd_inf_len[7]	spd_inf_len[6]	spd_inf_len[5]	spd_inf_len[4]	spd_inf_len[3]	spd_inf_len[2]	spd_inf_len[1]	spd_inf_len[0]
0xE9	0x85	ms_packet_id	rw	ms_packet_id[7]	ms_packet_id[6]	ms_packet_id[5]	ms_packet_id[4]	ms_packet_id[3]	ms_packet_id[2]	ms_packet_id[1]	ms_packet_id[0]
0xEA	0x00	ms_inf_vers	r	ms_inf_vers[7]	ms_inf_vers[6]	ms_inf_vers[5]	ms_inf_vers[4]	ms_inf_vers[3]	ms_inf_vers[2]	ms_inf_vers[1]	ms_inf_vers[0]
0xEB	0x00	ms_inf_len	r	ms_inf_len[7]	ms_inf_len[6]	ms_inf_len[5]	ms_inf_len[4]	ms_inf_len[3]	ms_inf_len[2]	ms_inf_len[1]	ms_inf_len[0]
0xEC	0x81	vs_packet_id	rw	vs_packet_id[7]	vs_packet_id[6]	vs_packet_id[5]	vs_packet_id[4]	vs_packet_id[3]	vs_packet_id[2]	vs_packet_id[1]	vs_packet_id[0]
0xED	0x00	vs_inf_vers	r	vs_inf_vers[7]	vs_inf_vers[6]	vs_inf_vers[5]	vs_inf_vers[4]	vs_inf_vers[3]	vs_inf_vers[2]	vs_inf_vers[1]	vs_inf_vers[0]
0xEE	0x00	vs_inf_len	r	vs_inf_len[7]	vs_inf_len[6]	vs_inf_len[5]	vs_inf_len[4]	vs_inf_len[3]	vs_inf_len[2]	vs_inf_len[1]	vs_inf_len[0]
0xEF	0x04	acp_packet_id	rw	acp_packet_id[7]	acp_packet_id[6]	acp_packet_id[5]	acp_packet_id[4]	acp_packet_id[3]	acp_packet_id[2]	acp_packet_id[1]	acp_packet_id[0]
0xF0	0x00	acp_type	r	acp_type[7]	acp_type[6]	acp_type[5]	acp_type[4]	acp_type[3]	acp_type[2]	acp_type[1]	acp_type[0]
0xF1	0x00	acp_header2	r	acp_header2[7]	acp_header2[6]	acp_header2[5]	acp_header2[4]	acp_header2[3]	acp_header2[2]	acp_header2[1]	acp_header2[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF2	0x05	isrc1_packet_id	rw	isrc1_packet_id[7]	isrc1_packet_id[6]	isrc1_packet_id[5]	isrc1_packet_id[4]	isrc1_packet_id[3]	isrc1_packet_id[2]	isrc1_packet_id[1]	isrc1_packet_id[0]
0xF3	0x00	isrc1_header1	r	isrc1_header1[7]	isrc1_header1[6]	isrc1_header1[5]	isrc1_header1[4]	isrc1_header1[3]	isrc1_header1[2]	isrc1_header1[1]	isrc1_header1[0]
0xF4	0x00	isrc1_header2	r	isrc1_header2[7]	isrc1_header2[6]	isrc1_header2[5]	isrc1_header2[4]	isrc1_header2[3]	isrc1_header2[2]	isrc1_header2[1]	isrc1_header2[0]
0xF5	0x06	isrc2_packet_id	rw	isrc2_packet_id[7]	isrc2_packet_id[6]	isrc2_packet_id[5]	isrc2_packet_id[4]	isrc2_packet_id[3]	isrc2_packet_id[2]	isrc2_packet_id[1]	isrc2_packet_id[0]
0xF6	0x00	isrc2_header1	r	isrc2_header1[7]	isrc2_header1[6]	isrc2_header1[5]	isrc2_header1[4]	isrc2_header1[3]	isrc2_header1[2]	isrc2_header1[1]	isrc2_header1[0]
0xF7	0x00	isrc2_header2	r	isrc2_header2[7]	isrc2_header2[6]	isrc2_header2[5]	isrc2_header2[4]	isrc2_header2[3]	isrc2_header2[2]	isrc2_header2[1]	isrc2_header2[0]
0xF8	0x0A	gamut_packet_id	rw	gamut_packet_id[7]	gamut_packet_id[6]	gamut_packet_id[5]	gamut_packet_id[4]	gamut_packet_id[3]	gamut_packet_id[2]	gamut_packet_id[1]	gamut_packet_id[0]
0xF9	0x00	gamut_header1	r	gamut_header1[7]	gamut_header1[6]	gamut_header1[5]	gamut_header1[4]	gamut_header1[3]	gamut_header1[2]	gamut_header1[1]	gamut_header1[0]
0xFA	0x00	gamut_header2	r	gamut_header2[7]	gamut_header2[6]	gamut_header2[5]	gamut_header2[4]	gamut_header2[3]	gamut_header2[2]	gamut_header2[1]	gamut_header2[0]
0xFD	0x81	infoframe_reg_fd	rw	pkt_cnt_id[7]	pkt_cnt_id[6]	pkt_cnt_id[5]	pkt_cnt_id[4]	pkt_cnt_id[3]	pkt_cnt_id[2]	pkt_cnt_id[1]	pkt_cnt_id[0]
0xFE	0x00	infoframe_reg_fe	rw	-	-	-	en_pkt_cnt_sel	pkt_cnt_sel[3]	pkt_cnt_sel[2]	pkt_cnt_sel[1]	pkt_cnt_sel[0]
0xFF	0x00	infoframe_reg_ff	r	-	-	-	-	rb_pkt_cnt[3]	rb_pkt_cnt[2]	rb_pkt_cnt[1]	rb_pkt_cnt[0]

1.5 CBUS MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x00	requester_tx_enable	sc	req_tx_clear_buffers	req_rx_clear_buffers	req_tx_buffer_size[4]	req_tx_buffer_size[3]	req_tx_buffer_size[2]	req_tx_buffer_size[1]	req_tx_buffer_size[0]	req_tx_en
0x01	0x00	responder_tx_enable	sc	resp_tx_clear_buffers	resp_rx_clear_buffers	resp_tx_buffer_size[4]	resp_tx_buffer_size[3]	resp_tx_buffer_size[2]	resp_tx_buffer_size[1]	resp_tx_buffer_size[0]	resp_tx_en
0x02	0x00	msc_status	r	-	-	-	-	edid_ready	msc_state[2]	msc_state[1]	msc_state[0]
0x03	0x08	cbus_reg_03	rw	enable_timeout_counters	-	-	-	-	-	-	-
0x05	0x60	cbus_reg_05	rw	-	-	clear_buffers_on_abort	-	-	-	-	-
0x06	0x00	cbus_reg_06	r	-	-	-	-	-	-	-	rb_mhl_mode
0x07	0x03	spi_config	rw	-	-	-	-	-	-	spi_mode[1]	spi_mode[0]
0x0F	0x0B	int_config	rw	-	mhl_interrupt_en	-	-	intrq3_op_sel[1]	intrq3_op_sel[0]	intrq3_dur_sel[1]	intrq3_dur_sel[0]
0x10	0x00	int_event_status_0	r	msc_got_abort_st	msc_got_nack_st	msc_state_req_rx_st	msc_state_req_tx_st	msc_state_resp_tx_st	msc_resp_rx_size_match_st	msc_state_resp_rx_st	msc_state_idle_st
0x11	0x00	int_event_status_1	r	msc_resp_rx_packet_received_st	msc_resp_tx_packet_sent_st	msc_req_rx_packet_received_st	msc_req_tx_packet_sent_st	nretry_nacks_follow_st	nretry_nacks_init_st	-	-
0x12	0x00	int_level_status	r	-	-	-	-	-	-	mhl_electrical_discovery_st	mhl_mode_detected_st
0x13	0x00	int_event_clear_0	sc	msc_got_abort_clr	msc_got_nack_clr	msc_state_req_rx_clr	msc_state_req_tx_clr	msc_state_resp_tx_clr	msc_resp_rx_size_match_clr	msc_state_resp_rx_clr	msc_state_idle_clr
0x14	0x00	int_event_clear_1	sc	msc_resp_rx_packet_received_clr	msc_resp_tx_packet_sent_clr	msc_req_rx_packet_received_clr	msc_req_tx_packet_sent_clr	nretry_nacks_follow_clr	nretry_nacks_init_clr	-	-
0x15	0x00	int_level_clear	sc	-	-	-	-	-	-	mhl_electrical_discovery_clr	mhl_mode_detected_clr
0x16	0x00	int_event_mask_0	rw	msc_got_abort_mb	msc_got_nack_mb	msc_state_req_rx_mb	msc_state_req_tx_mb	msc_state_resp_tx_mb	msc_resp_rx_size_match_mb	msc_state_resp_rx_mb	msc_state_idle_mb
0x17	0x00	int_event_mask_1	rw	msc_resp_rx_packet_received_mb	msc_resp_tx_packet_sent_mb	msc_req_rx_packet_received_mb	msc_req_tx_packet_sent_mb	nretry_nacks_follow_mb	nretry_nacks_init_mb	-	-
0x18	0x00	int_level_mask	rw	-	-	-	-	-	-	mhl_electrical_discovery_mb	mhl_mode_detected_mb
0x1B	0x00	ddc_error_code	r	ddc_error_code[7]	ddc_error_code[6]	ddc_error_code[5]	ddc_error_code[4]	ddc_error_code[3]	ddc_error_code[2]	ddc_error_code[1]	ddc_error_code[0]
0x1C	0x00	link_mode_status	rw	-	-	-	link_mode_muted	link_mode_path_en	link_mode_clk_mode[2]	link_mode_clk_mode[1]	link_mode_clk_mode[0]
0x25	0x00	cbus_reg_25	rw	-	-	-	-	-	-	standby_impl_sel[1]	standby_impl_sel[0]
0x26	0x00	cbus_reg_26	sc	-	-	-	-	-	-	send_abort	restart_discovery
0x27	0x02	cbus_reg_27	rw	sink_standby	float_cbus	-	-	manual_vbus_value	manual_vbus_enable	disable_wake_pulses_sink1	disable_wake_pulses_stby
0x28	0x00	cbus_reg_28	r	rb_sequence_state[2]	rb_sequence_state[1]	rb_sequence_state[0]	-	-	-	-	-
0x29	0x00	cbus_reg_29	r	-	-	-	-	-	-	-	rb_sink_standby_statuts
0x30	0x00	req_rx_detect_buf_size	r	-	-	-	req_rx_detect_buffer_size[4]	req_rx_detect_buffer_size[3]	req_rx_detect_buffer_size[2]	req_rx_detect_buffer_size[1]	req_rx_detect_buffer_size[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x31	0x00	req_tx_latched_buff_size	r	-	-	-	req_tx_latched_buffer_size[4]	req_tx_latched_buffer_size[3]	req_tx_latched_buffer_size[2]	req_tx_latched_buffer_size[1]	req_tx_latched_buffer_size[0]
0x32	0x00	resp_rx_detect_buff_size	r	-	-	-	resp_rx_detect_buffer_size[4]	resp_rx_detect_buffer_size[3]	resp_rx_detect_buffer_size[2]	resp_rx_detect_buffer_size[1]	resp_rx_detect_buffer_size[0]
0x33	0x00	resp_tx_latched_buff_size	r	-	-	-	resp_tx_latched_buffer_size[4]	resp_tx_latched_buffer_size[3]	resp_tx_latched_buffer_size[2]	resp_tx_latched_buffer_size[1]	resp_tx_latched_buffer_size[0]
0x34	0x1F	req_exp_rx_buff_size_1	sc	-	-	-	req_expected_rx_buff_size[4]	req_expected_rx_buff_size[3]	req_expected_rx_buff_size[2]	req_expected_rx_buff_size[1]	req_expected_rx_buff_size[0]
0x35	0x1F	resp_exp_rx_buff_size_1	sc	-	-	-	resp_expected_rx_buff_size[4]	resp_expected_rx_buff_size[3]	resp_expected_rx_buff_size[2]	resp_expected_rx_buff_size[1]	resp_expected_rx_buff_size[0]
0x36	0x1F	req_exp_rx_buff_size_2	r	-	-	-	req_expected_rx_buff_size_latched[4]	req_expected_rx_buff_size_latched[3]	req_expected_rx_buff_size_latched[2]	req_expected_rx_buff_size_latched[1]	req_expected_rx_buff_size_latched[0]
0x37	0x1F	resp_exp_rx_buff_size_2	r	-	-	-	resp_expected_rx_buff_size_latched[4]	resp_expected_rx_buff_size_latched[3]	resp_expected_rx_buff_size_latched[2]	resp_expected_rx_buff_size_latched[1]	resp_expected_rx_buff_size_latched[0]
0x40	0x00	req_tx_cmd_data_flags_1	rw	req_tx_cmd_data b_flags[7]	req_tx_cmd_data b_flags[6]	req_tx_cmd_data b_flags[5]	req_tx_cmd_data b_flags[4]	req_tx_cmd_data b_flags[3]	req_tx_cmd_data b_flags[2]	req_tx_cmd_data b_flags[1]	req_tx_cmd_data b_flags[0]
0x41	0x00	req_tx_cmd_data_flags_2	rw	req_tx_cmd_data b_flags[15]	req_tx_cmd_data b_flags[14]	req_tx_cmd_data b_flags[13]	req_tx_cmd_data b_flags[12]	req_tx_cmd_data b_flags[11]	req_tx_cmd_data b_flags[10]	req_tx_cmd_data b_flags[9]	req_tx_cmd_data b_flags[8]
0x42	0x00	req_tx_cmd_data_flags_3	rw	req_tx_cmd_data b_flags[23]	req_tx_cmd_data b_flags[22]	req_tx_cmd_data b_flags[21]	req_tx_cmd_data b_flags[20]	req_tx_cmd_data b_flags[19]	req_tx_cmd_data b_flags[18]	req_tx_cmd_data b_flags[17]	req_tx_cmd_data b_flags[16]
0x43	0x00	req_tx_cmd_data_flags_4	rw	req_tx_cmd_data b_flags[31]	req_tx_cmd_data b_flags[30]	req_tx_cmd_data b_flags[29]	req_tx_cmd_data b_flags[28]	req_tx_cmd_data b_flags[27]	req_tx_cmd_data b_flags[26]	req_tx_cmd_data b_flags[25]	req_tx_cmd_data b_flags[24]
0x50	0x00	req_rx_cmd_data_flags_1	r	req_rx_cmd_data b_flags[7]	req_rx_cmd_data b_flags[6]	req_rx_cmd_data b_flags[5]	req_rx_cmd_data b_flags[4]	req_rx_cmd_data b_flags[3]	req_rx_cmd_data b_flags[2]	req_rx_cmd_data b_flags[1]	req_rx_cmd_data b_flags[0]
0x51	0x00	req_rx_cmd_data_flags_2	r	req_rx_cmd_data b_flags[15]	req_rx_cmd_data b_flags[14]	req_rx_cmd_data b_flags[13]	req_rx_cmd_data b_flags[12]	req_rx_cmd_data b_flags[11]	req_rx_cmd_data b_flags[10]	req_rx_cmd_data b_flags[9]	req_rx_cmd_data b_flags[8]
0x52	0x00	req_rx_cmd_data_flags_3	r	req_rx_cmd_data b_flags[23]	req_rx_cmd_data b_flags[22]	req_rx_cmd_data b_flags[21]	req_rx_cmd_data b_flags[20]	req_rx_cmd_data b_flags[19]	req_rx_cmd_data b_flags[18]	req_rx_cmd_data b_flags[17]	req_rx_cmd_data b_flags[16]
0x53	0x00	req_rx_cmd_data_flags_4	r	req_rx_cmd_data b_flags[31]	req_rx_cmd_data b_flags[30]	req_rx_cmd_data b_flags[29]	req_rx_cmd_data b_flags[28]	req_rx_cmd_data b_flags[27]	req_rx_cmd_data b_flags[26]	req_rx_cmd_data b_flags[25]	req_rx_cmd_data b_flags[24]
0x60	0x00	resp_tx_cmd_data_flags_1	rw	resp_tx_cmd_data b_flags[7]	resp_tx_cmd_data b_flags[6]	resp_tx_cmd_data b_flags[5]	resp_tx_cmd_data b_flags[4]	resp_tx_cmd_data b_flags[3]	resp_tx_cmd_data b_flags[2]	resp_tx_cmd_data b_flags[1]	resp_tx_cmd_data b_flags[0]
0x61	0x00	resp_tx_cmd_data_flags_2	rw	resp_tx_cmd_data b_flags[15]	resp_tx_cmd_data b_flags[14]	resp_tx_cmd_data b_flags[13]	resp_tx_cmd_data b_flags[12]	resp_tx_cmd_data b_flags[11]	resp_tx_cmd_data b_flags[10]	resp_tx_cmd_data b_flags[9]	resp_tx_cmd_data b_flags[8]
0x62	0x00	resp_tx_cmd_data_flags_3	rw	resp_tx_cmd_data b_flags[23]	resp_tx_cmd_data b_flags[22]	resp_tx_cmd_data b_flags[21]	resp_tx_cmd_data b_flags[20]	resp_tx_cmd_data b_flags[19]	resp_tx_cmd_data b_flags[18]	resp_tx_cmd_data b_flags[17]	resp_tx_cmd_data b_flags[16]
0x63	0x00	resp_tx_cmd_data_flags_4	rw	resp_tx_cmd_data b_flags[31]	resp_tx_cmd_data b_flags[30]	resp_tx_cmd_data b_flags[29]	resp_tx_cmd_data b_flags[28]	resp_tx_cmd_data b_flags[27]	resp_tx_cmd_data b_flags[26]	resp_tx_cmd_data b_flags[25]	resp_tx_cmd_data b_flags[24]
0x70	0x00	resp_rx_cmd_data_flags_1	r	resp_rx_cmd_data b_flags[7]	resp_rx_cmd_data b_flags[6]	resp_rx_cmd_data b_flags[5]	resp_rx_cmd_data b_flags[4]	resp_rx_cmd_data b_flags[3]	resp_rx_cmd_data b_flags[2]	resp_rx_cmd_data b_flags[1]	resp_rx_cmd_data b_flags[0]
0x71	0x00	resp_rx_cmd_data_flags_2	r	resp_rx_cmd_data b_flags[15]	resp_rx_cmd_data b_flags[14]	resp_rx_cmd_data b_flags[13]	resp_rx_cmd_data b_flags[12]	resp_rx_cmd_data b_flags[11]	resp_rx_cmd_data b_flags[10]	resp_rx_cmd_data b_flags[9]	resp_rx_cmd_data b_flags[8]
0x72	0x00	resp_rx_cmd_data_flags_3	r	resp_rx_cmd_data b_flags[23]	resp_rx_cmd_data b_flags[22]	resp_rx_cmd_data b_flags[21]	resp_rx_cmd_data b_flags[20]	resp_rx_cmd_data b_flags[19]	resp_rx_cmd_data b_flags[18]	resp_rx_cmd_data b_flags[17]	resp_rx_cmd_data b_flags[16]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x73	0x00	resp_rx_cmd_data_flags_4	r	resp_rx_cmd_data b_flags[31]	resp_rx_cmd_data b_flags[30]	resp_rx_cmd_data b_flags[29]	resp_rx_cmd_data b_flags[28]	resp_rx_cmd_data b_flags[27]	resp_rx_cmd_data b_flags[26]	resp_rx_cmd_data b_flags[25]	resp_rx_cmd_data b_flags[24]
0x80	0x00	req_tx_byte_0	rw	req_tx_byte_0[7]	req_tx_byte_0[6]	req_tx_byte_0[5]	req_tx_byte_0[4]	req_tx_byte_0[3]	req_tx_byte_0[2]	req_tx_byte_0[1]	req_tx_byte_0[0]
0x81	0x00	req_tx_byte_1	rw	req_tx_byte_1[7]	req_tx_byte_1[6]	req_tx_byte_1[5]	req_tx_byte_1[4]	req_tx_byte_1[3]	req_tx_byte_1[2]	req_tx_byte_1[1]	req_tx_byte_1[0]
0x82	0x00	req_tx_byte_2	rw	req_tx_byte_2[7]	req_tx_byte_2[6]	req_tx_byte_2[5]	req_tx_byte_2[4]	req_tx_byte_2[3]	req_tx_byte_2[2]	req_tx_byte_2[1]	req_tx_byte_2[0]
0x83	0x00	req_tx_byte_3	rw	req_tx_byte_3[7]	req_tx_byte_3[6]	req_tx_byte_3[5]	req_tx_byte_3[4]	req_tx_byte_3[3]	req_tx_byte_3[2]	req_tx_byte_3[1]	req_tx_byte_3[0]
0x84	0x00	req_tx_byte_4	rw	req_tx_byte_4[7]	req_tx_byte_4[6]	req_tx_byte_4[5]	req_tx_byte_4[4]	req_tx_byte_4[3]	req_tx_byte_4[2]	req_tx_byte_4[1]	req_tx_byte_4[0]
0x85	0x00	req_tx_byte_5	rw	req_tx_byte_5[7]	req_tx_byte_5[6]	req_tx_byte_5[5]	req_tx_byte_5[4]	req_tx_byte_5[3]	req_tx_byte_5[2]	req_tx_byte_5[1]	req_tx_byte_5[0]
0x86	0x00	req_tx_byte_6	rw	req_tx_byte_6[7]	req_tx_byte_6[6]	req_tx_byte_6[5]	req_tx_byte_6[4]	req_tx_byte_6[3]	req_tx_byte_6[2]	req_tx_byte_6[1]	req_tx_byte_6[0]
0x87	0x00	req_tx_byte_7	rw	req_tx_byte_7[7]	req_tx_byte_7[6]	req_tx_byte_7[5]	req_tx_byte_7[4]	req_tx_byte_7[3]	req_tx_byte_7[2]	req_tx_byte_7[1]	req_tx_byte_7[0]
0x88	0x00	req_tx_byte_8	rw	req_tx_byte_8[7]	req_tx_byte_8[6]	req_tx_byte_8[5]	req_tx_byte_8[4]	req_tx_byte_8[3]	req_tx_byte_8[2]	req_tx_byte_8[1]	req_tx_byte_8[0]
0x89	0x00	req_tx_byte_9	rw	req_tx_byte_9[7]	req_tx_byte_9[6]	req_tx_byte_9[5]	req_tx_byte_9[4]	req_tx_byte_9[3]	req_tx_byte_9[2]	req_tx_byte_9[1]	req_tx_byte_9[0]
0x8A	0x00	req_tx_byte_10	rw	req_tx_byte_10[7]	req_tx_byte_10[6]	req_tx_byte_10[5]	req_tx_byte_10[4]	req_tx_byte_10[3]	req_tx_byte_10[2]	req_tx_byte_10[1]	req_tx_byte_10[0]
0x8B	0x00	req_tx_byte_11	rw	req_tx_byte_11[7]	req_tx_byte_11[6]	req_tx_byte_11[5]	req_tx_byte_11[4]	req_tx_byte_11[3]	req_tx_byte_11[2]	req_tx_byte_11[1]	req_tx_byte_11[0]
0x8C	0x00	req_tx_byte_12	rw	req_tx_byte_12[7]	req_tx_byte_12[6]	req_tx_byte_12[5]	req_tx_byte_12[4]	req_tx_byte_12[3]	req_tx_byte_12[2]	req_tx_byte_12[1]	req_tx_byte_12[0]
0x8D	0x00	req_tx_byte_13	rw	req_tx_byte_13[7]	req_tx_byte_13[6]	req_tx_byte_13[5]	req_tx_byte_13[4]	req_tx_byte_13[3]	req_tx_byte_13[2]	req_tx_byte_13[1]	req_tx_byte_13[0]
0x8E	0x00	req_tx_byte_14	rw	req_tx_byte_14[7]	req_tx_byte_14[6]	req_tx_byte_14[5]	req_tx_byte_14[4]	req_tx_byte_14[3]	req_tx_byte_14[2]	req_tx_byte_14[1]	req_tx_byte_14[0]
0x8F	0x00	req_tx_byte_15	rw	req_tx_byte_15[7]	req_tx_byte_15[6]	req_tx_byte_15[5]	req_tx_byte_15[4]	req_tx_byte_15[3]	req_tx_byte_15[2]	req_tx_byte_15[1]	req_tx_byte_15[0]
0x90	0x00	req_tx_byte_16	rw	req_tx_byte_16[7]	req_tx_byte_16[6]	req_tx_byte_16[5]	req_tx_byte_16[4]	req_tx_byte_16[3]	req_tx_byte_16[2]	req_tx_byte_16[1]	req_tx_byte_16[0]
0x91	0x00	req_tx_byte_17	rw	req_tx_byte_17[7]	req_tx_byte_17[6]	req_tx_byte_17[5]	req_tx_byte_17[4]	req_tx_byte_17[3]	req_tx_byte_17[2]	req_tx_byte_17[1]	req_tx_byte_17[0]
0x92	0x00	req_tx_byte_18	rw	req_tx_byte_18[7]	req_tx_byte_18[6]	req_tx_byte_18[5]	req_tx_byte_18[4]	req_tx_byte_18[3]	req_tx_byte_18[2]	req_tx_byte_18[1]	req_tx_byte_18[0]
0x93	0x00	req_tx_byte_19	rw	req_tx_byte_19[7]	req_tx_byte_19[6]	req_tx_byte_19[5]	req_tx_byte_19[4]	req_tx_byte_19[3]	req_tx_byte_19[2]	req_tx_byte_19[1]	req_tx_byte_19[0]
0x94	0x00	req_tx_byte_20	rw	req_tx_byte_20[7]	req_tx_byte_20[6]	req_tx_byte_20[5]	req_tx_byte_20[4]	req_tx_byte_20[3]	req_tx_byte_20[2]	req_tx_byte_20[1]	req_tx_byte_20[0]
0x95	0x00	req_tx_byte_21	rw	req_tx_byte_21[7]	req_tx_byte_21[6]	req_tx_byte_21[5]	req_tx_byte_21[4]	req_tx_byte_21[3]	req_tx_byte_21[2]	req_tx_byte_21[1]	req_tx_byte_21[0]
0x96	0x00	req_tx_byte_22	rw	req_tx_byte_22[7]	req_tx_byte_22[6]	req_tx_byte_22[5]	req_tx_byte_22[4]	req_tx_byte_22[3]	req_tx_byte_22[2]	req_tx_byte_22[1]	req_tx_byte_22[0]
0x97	0x00	req_tx_byte_23	rw	req_tx_byte_23[7]	req_tx_byte_23[6]	req_tx_byte_23[5]	req_tx_byte_23[4]	req_tx_byte_23[3]	req_tx_byte_23[2]	req_tx_byte_23[1]	req_tx_byte_23[0]
0x98	0x00	req_tx_byte_24	rw	req_tx_byte_24[7]	req_tx_byte_24[6]	req_tx_byte_24[5]	req_tx_byte_24[4]	req_tx_byte_24[3]	req_tx_byte_24[2]	req_tx_byte_24[1]	req_tx_byte_24[0]
0x99	0x00	req_tx_byte_25	rw	req_tx_byte_25[7]	req_tx_byte_25[6]	req_tx_byte_25[5]	req_tx_byte_25[4]	req_tx_byte_25[3]	req_tx_byte_25[2]	req_tx_byte_25[1]	req_tx_byte_25[0]
0x9A	0x00	req_tx_byte_26	rw	req_tx_byte_26[7]	req_tx_byte_26[6]	req_tx_byte_26[5]	req_tx_byte_26[4]	req_tx_byte_26[3]	req_tx_byte_26[2]	req_tx_byte_26[1]	req_tx_byte_26[0]
0x9B	0x00	req_tx_byte_27	rw	req_tx_byte_27[7]	req_tx_byte_27[6]	req_tx_byte_27[5]	req_tx_byte_27[4]	req_tx_byte_27[3]	req_tx_byte_27[2]	req_tx_byte_27[1]	req_tx_byte_27[0]
0x9C	0x00	req_tx_byte_28	rw	req_tx_byte_28[7]	req_tx_byte_28[6]	req_tx_byte_28[5]	req_tx_byte_28[4]	req_tx_byte_28[3]	req_tx_byte_28[2]	req_tx_byte_28[1]	req_tx_byte_28[0]
0x9D	0x00	req_tx_byte_29	rw	req_tx_byte_29[7]	req_tx_byte_29[6]	req_tx_byte_29[5]	req_tx_byte_29[4]	req_tx_byte_29[3]	req_tx_byte_29[2]	req_tx_byte_29[1]	req_tx_byte_29[0]
0x9E	0x00	req_tx_byte_30	rw	req_tx_byte_30[7]	req_tx_byte_30[6]	req_tx_byte_30[5]	req_tx_byte_30[4]	req_tx_byte_30[3]	req_tx_byte_30[2]	req_tx_byte_30[1]	req_tx_byte_30[0]
0x9F	0x00	req_tx_byte_31	rw	req_tx_byte_31[7]	req_tx_byte_31[6]	req_tx_byte_31[5]	req_tx_byte_31[4]	req_tx_byte_31[3]	req_tx_byte_31[2]	req_tx_byte_31[1]	req_tx_byte_31[0]
0xA0	0x00	req_rx_byte_0	r	req_rx_byte_0[7]	req_rx_byte_0[6]	req_rx_byte_0[5]	req_rx_byte_0[4]	req_rx_byte_0[3]	req_rx_byte_0[2]	req_rx_byte_0[1]	req_rx_byte_0[0]
0xA1	0x00	req_rx_byte_1	r	req_rx_byte_1[7]	req_rx_byte_1[6]	req_rx_byte_1[5]	req_rx_byte_1[4]	req_rx_byte_1[3]	req_rx_byte_1[2]	req_rx_byte_1[1]	req_rx_byte_1[0]
0xA2	0x00	req_rx_byte_2	r	req_rx_byte_2[7]	req_rx_byte_2[6]	req_rx_byte_2[5]	req_rx_byte_2[4]	req_rx_byte_2[3]	req_rx_byte_2[2]	req_rx_byte_2[1]	req_rx_byte_2[0]
0xA3	0x00	req_rx_byte_3	r	req_rx_byte_3[7]	req_rx_byte_3[6]	req_rx_byte_3[5]	req_rx_byte_3[4]	req_rx_byte_3[3]	req_rx_byte_3[2]	req_rx_byte_3[1]	req_rx_byte_3[0]
0xA4	0x00	req_rx_byte_4	r	req_rx_byte_4[7]	req_rx_byte_4[6]	req_rx_byte_4[5]	req_rx_byte_4[4]	req_rx_byte_4[3]	req_rx_byte_4[2]	req_rx_byte_4[1]	req_rx_byte_4[0]
0xA5	0x00	req_rx_byte_5	r	req_rx_byte_5[7]	req_rx_byte_5[6]	req_rx_byte_5[5]	req_rx_byte_5[4]	req_rx_byte_5[3]	req_rx_byte_5[2]	req_rx_byte_5[1]	req_rx_byte_5[0]
0xA6	0x00	req_rx_byte_6	r	req_rx_byte_6[7]	req_rx_byte_6[6]	req_rx_byte_6[5]	req_rx_byte_6[4]	req_rx_byte_6[3]	req_rx_byte_6[2]	req_rx_byte_6[1]	req_rx_byte_6[0]
0xA7	0x00	req_rx_byte_7	r	req_rx_byte_7[7]	req_rx_byte_7[6]	req_rx_byte_7[5]	req_rx_byte_7[4]	req_rx_byte_7[3]	req_rx_byte_7[2]	req_rx_byte_7[1]	req_rx_byte_7[0]
0xA8	0x00	req_rx_byte_8	r	req_rx_byte_8[7]	req_rx_byte_8[6]	req_rx_byte_8[5]	req_rx_byte_8[4]	req_rx_byte_8[3]	req_rx_byte_8[2]	req_rx_byte_8[1]	req_rx_byte_8[0]
0xA9	0x00	req_rx_byte_9	r	req_rx_byte_9[7]	req_rx_byte_9[6]	req_rx_byte_9[5]	req_rx_byte_9[4]	req_rx_byte_9[3]	req_rx_byte_9[2]	req_rx_byte_9[1]	req_rx_byte_9[0]
0xAA	0x00	req_rx_byte_10	r	req_rx_byte_10[7]	req_rx_byte_10[6]	req_rx_byte_10[5]	req_rx_byte_10[4]	req_rx_byte_10[3]	req_rx_byte_10[2]	req_rx_byte_10[1]	req_rx_byte_10[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xAB	0x00	req_rx_byte_11	r	req_rx_byte_11[7]	req_rx_byte_11[6]	req_rx_byte_11[5]	req_rx_byte_11[4]	req_rx_byte_11[3]	req_rx_byte_11[2]	req_rx_byte_11[1]	req_rx_byte_11[0]
0xAC	0x00	req_rx_byte_12	r	req_rx_byte_12[7]	req_rx_byte_12[6]	req_rx_byte_12[5]	req_rx_byte_12[4]	req_rx_byte_12[3]	req_rx_byte_12[2]	req_rx_byte_12[1]	req_rx_byte_12[0]
0xAD	0x00	req_rx_byte_13	r	req_rx_byte_13[7]	req_rx_byte_13[6]	req_rx_byte_13[5]	req_rx_byte_13[4]	req_rx_byte_13[3]	req_rx_byte_13[2]	req_rx_byte_13[1]	req_rx_byte_13[0]
0xAE	0x00	req_rx_byte_14	r	req_rx_byte_14[7]	req_rx_byte_14[6]	req_rx_byte_14[5]	req_rx_byte_14[4]	req_rx_byte_14[3]	req_rx_byte_14[2]	req_rx_byte_14[1]	req_rx_byte_14[0]
0xAF	0x00	req_rx_byte_15	r	req_rx_byte_15[7]	req_rx_byte_15[6]	req_rx_byte_15[5]	req_rx_byte_15[4]	req_rx_byte_15[3]	req_rx_byte_15[2]	req_rx_byte_15[1]	req_rx_byte_15[0]
0xB0	0x00	req_rx_byte_16	r	req_rx_byte_16[7]	req_rx_byte_16[6]	req_rx_byte_16[5]	req_rx_byte_16[4]	req_rx_byte_16[3]	req_rx_byte_16[2]	req_rx_byte_16[1]	req_rx_byte_16[0]
0xB1	0x00	req_rx_byte_17	r	req_rx_byte_17[7]	req_rx_byte_17[6]	req_rx_byte_17[5]	req_rx_byte_17[4]	req_rx_byte_17[3]	req_rx_byte_17[2]	req_rx_byte_17[1]	req_rx_byte_17[0]
0xB2	0x00	req_rx_byte_18	r	req_rx_byte_18[7]	req_rx_byte_18[6]	req_rx_byte_18[5]	req_rx_byte_18[4]	req_rx_byte_18[3]	req_rx_byte_18[2]	req_rx_byte_18[1]	req_rx_byte_18[0]
0xB3	0x00	req_rx_byte_19	r	req_rx_byte_19[7]	req_rx_byte_19[6]	req_rx_byte_19[5]	req_rx_byte_19[4]	req_rx_byte_19[3]	req_rx_byte_19[2]	req_rx_byte_19[1]	req_rx_byte_19[0]
0xB4	0x00	req_rx_byte_20	r	req_rx_byte_20[7]	req_rx_byte_20[6]	req_rx_byte_20[5]	req_rx_byte_20[4]	req_rx_byte_20[3]	req_rx_byte_20[2]	req_rx_byte_20[1]	req_rx_byte_20[0]
0xB5	0x00	req_rx_byte_21	r	req_rx_byte_21[7]	req_rx_byte_21[6]	req_rx_byte_21[5]	req_rx_byte_21[4]	req_rx_byte_21[3]	req_rx_byte_21[2]	req_rx_byte_21[1]	req_rx_byte_21[0]
0xB6	0x00	req_rx_byte_22	r	req_rx_byte_22[7]	req_rx_byte_22[6]	req_rx_byte_22[5]	req_rx_byte_22[4]	req_rx_byte_22[3]	req_rx_byte_22[2]	req_rx_byte_22[1]	req_rx_byte_22[0]
0xB7	0x00	req_rx_byte_23	r	req_rx_byte_23[7]	req_rx_byte_23[6]	req_rx_byte_23[5]	req_rx_byte_23[4]	req_rx_byte_23[3]	req_rx_byte_23[2]	req_rx_byte_23[1]	req_rx_byte_23[0]
0xB8	0x00	req_rx_byte_24	r	req_rx_byte_24[7]	req_rx_byte_24[6]	req_rx_byte_24[5]	req_rx_byte_24[4]	req_rx_byte_24[3]	req_rx_byte_24[2]	req_rx_byte_24[1]	req_rx_byte_24[0]
0xB9	0x00	req_rx_byte_25	r	req_rx_byte_25[7]	req_rx_byte_25[6]	req_rx_byte_25[5]	req_rx_byte_25[4]	req_rx_byte_25[3]	req_rx_byte_25[2]	req_rx_byte_25[1]	req_rx_byte_25[0]
0xBA	0x00	req_rx_byte_26	r	req_rx_byte_26[7]	req_rx_byte_26[6]	req_rx_byte_26[5]	req_rx_byte_26[4]	req_rx_byte_26[3]	req_rx_byte_26[2]	req_rx_byte_26[1]	req_rx_byte_26[0]
0xBB	0x00	req_rx_byte_27	r	req_rx_byte_27[7]	req_rx_byte_27[6]	req_rx_byte_27[5]	req_rx_byte_27[4]	req_rx_byte_27[3]	req_rx_byte_27[2]	req_rx_byte_27[1]	req_rx_byte_27[0]
0xBC	0x00	req_rx_byte_28	r	req_rx_byte_28[7]	req_rx_byte_28[6]	req_rx_byte_28[5]	req_rx_byte_28[4]	req_rx_byte_28[3]	req_rx_byte_28[2]	req_rx_byte_28[1]	req_rx_byte_28[0]
0xBD	0x00	req_rx_byte_29	r	req_rx_byte_29[7]	req_rx_byte_29[6]	req_rx_byte_29[5]	req_rx_byte_29[4]	req_rx_byte_29[3]	req_rx_byte_29[2]	req_rx_byte_29[1]	req_rx_byte_29[0]
0xBE	0x00	req_rx_byte_30	r	req_rx_byte_30[7]	req_rx_byte_30[6]	req_rx_byte_30[5]	req_rx_byte_30[4]	req_rx_byte_30[3]	req_rx_byte_30[2]	req_rx_byte_30[1]	req_rx_byte_30[0]
0xBF	0x00	req_rx_byte_31	r	req_rx_byte_31[7]	req_rx_byte_31[6]	req_rx_byte_31[5]	req_rx_byte_31[4]	req_rx_byte_31[3]	req_rx_byte_31[2]	req_rx_byte_31[1]	req_rx_byte_31[0]
0xC0	0x00	resp_tx_byte_0	rw	resp_tx_byte_0[7]	resp_tx_byte_0[6]	resp_tx_byte_0[5]	resp_tx_byte_0[4]	resp_tx_byte_0[3]	resp_tx_byte_0[2]	resp_tx_byte_0[1]	resp_tx_byte_0[0]
0xC1	0x00	resp_tx_byte_1	rw	resp_tx_byte_1[7]	resp_tx_byte_1[6]	resp_tx_byte_1[5]	resp_tx_byte_1[4]	resp_tx_byte_1[3]	resp_tx_byte_1[2]	resp_tx_byte_1[1]	resp_tx_byte_1[0]
0xC2	0x00	resp_tx_byte_2	rw	resp_tx_byte_2[7]	resp_tx_byte_2[6]	resp_tx_byte_2[5]	resp_tx_byte_2[4]	resp_tx_byte_2[3]	resp_tx_byte_2[2]	resp_tx_byte_2[1]	resp_tx_byte_2[0]
0xC3	0x00	resp_tx_byte_3	rw	resp_tx_byte_3[7]	resp_tx_byte_3[6]	resp_tx_byte_3[5]	resp_tx_byte_3[4]	resp_tx_byte_3[3]	resp_tx_byte_3[2]	resp_tx_byte_3[1]	resp_tx_byte_3[0]
0xC4	0x00	resp_tx_byte_4	rw	resp_tx_byte_4[7]	resp_tx_byte_4[6]	resp_tx_byte_4[5]	resp_tx_byte_4[4]	resp_tx_byte_4[3]	resp_tx_byte_4[2]	resp_tx_byte_4[1]	resp_tx_byte_4[0]
0xC5	0x00	resp_tx_byte_5	rw	resp_tx_byte_5[7]	resp_tx_byte_5[6]	resp_tx_byte_5[5]	resp_tx_byte_5[4]	resp_tx_byte_5[3]	resp_tx_byte_5[2]	resp_tx_byte_5[1]	resp_tx_byte_5[0]
0xC6	0x00	resp_tx_byte_6	rw	resp_tx_byte_6[7]	resp_tx_byte_6[6]	resp_tx_byte_6[5]	resp_tx_byte_6[4]	resp_tx_byte_6[3]	resp_tx_byte_6[2]	resp_tx_byte_6[1]	resp_tx_byte_6[0]
0xC7	0x00	resp_tx_byte_7	rw	resp_tx_byte_7[7]	resp_tx_byte_7[6]	resp_tx_byte_7[5]	resp_tx_byte_7[4]	resp_tx_byte_7[3]	resp_tx_byte_7[2]	resp_tx_byte_7[1]	resp_tx_byte_7[0]
0xC8	0x00	resp_tx_byte_8	rw	resp_tx_byte_8[7]	resp_tx_byte_8[6]	resp_tx_byte_8[5]	resp_tx_byte_8[4]	resp_tx_byte_8[3]	resp_tx_byte_8[2]	resp_tx_byte_8[1]	resp_tx_byte_8[0]
0xC9	0x00	resp_tx_byte_9	rw	resp_tx_byte_9[7]	resp_tx_byte_9[6]	resp_tx_byte_9[5]	resp_tx_byte_9[4]	resp_tx_byte_9[3]	resp_tx_byte_9[2]	resp_tx_byte_9[1]	resp_tx_byte_9[0]
0xCA	0x00	resp_tx_byte_10	rw	resp_tx_byte_10[7]	resp_tx_byte_10[6]	resp_tx_byte_10[5]	resp_tx_byte_10[4]	resp_tx_byte_10[3]	resp_tx_byte_10[2]	resp_tx_byte_10[1]	resp_tx_byte_10[0]
0xCB	0x00	resp_tx_byte_11	rw	resp_tx_byte_11[7]	resp_tx_byte_11[6]	resp_tx_byte_11[5]	resp_tx_byte_11[4]	resp_tx_byte_11[3]	resp_tx_byte_11[2]	resp_tx_byte_11[1]	resp_tx_byte_11[0]
0xCC	0x00	resp_tx_byte_12	rw	resp_tx_byte_12[7]	resp_tx_byte_12[6]	resp_tx_byte_12[5]	resp_tx_byte_12[4]	resp_tx_byte_12[3]	resp_tx_byte_12[2]	resp_tx_byte_12[1]	resp_tx_byte_12[0]
0xCD	0x00	resp_tx_byte_13	rw	resp_tx_byte_13[7]	resp_tx_byte_13[6]	resp_tx_byte_13[5]	resp_tx_byte_13[4]	resp_tx_byte_13[3]	resp_tx_byte_13[2]	resp_tx_byte_13[1]	resp_tx_byte_13[0]
0xCE	0x00	resp_tx_byte_14	rw	resp_tx_byte_14[7]	resp_tx_byte_14[6]	resp_tx_byte_14[5]	resp_tx_byte_14[4]	resp_tx_byte_14[3]	resp_tx_byte_14[2]	resp_tx_byte_14[1]	resp_tx_byte_14[0]
0xCF	0x00	resp_tx_byte_15	rw	resp_tx_byte_15[7]	resp_tx_byte_15[6]	resp_tx_byte_15[5]	resp_tx_byte_15[4]	resp_tx_byte_15[3]	resp_tx_byte_15[2]	resp_tx_byte_15[1]	resp_tx_byte_15[0]
0xD0	0x00	resp_tx_byte_16	rw	resp_tx_byte_16[7]	resp_tx_byte_16[6]	resp_tx_byte_16[5]	resp_tx_byte_16[4]	resp_tx_byte_16[3]	resp_tx_byte_16[2]	resp_tx_byte_16[1]	resp_tx_byte_16[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xD1	0x00	resp_tx_byte_17	rw	resp_tx_byte_17[7]	resp_tx_byte_17[6]	resp_tx_byte_17[5]	resp_tx_byte_17[4]	resp_tx_byte_17[3]	resp_tx_byte_17[2]	resp_tx_byte_17[1]	resp_tx_byte_17[0]
0xD2	0x00	resp_tx_byte_18	rw	resp_tx_byte_18[7]	resp_tx_byte_18[6]	resp_tx_byte_18[5]	resp_tx_byte_18[4]	resp_tx_byte_18[3]	resp_tx_byte_18[2]	resp_tx_byte_18[1]	resp_tx_byte_18[0]
0xD3	0x00	resp_tx_byte_19	rw	resp_tx_byte_19[7]	resp_tx_byte_19[6]	resp_tx_byte_19[5]	resp_tx_byte_19[4]	resp_tx_byte_19[3]	resp_tx_byte_19[2]	resp_tx_byte_19[1]	resp_tx_byte_19[0]
0xD4	0x00	resp_tx_byte_20	rw	resp_tx_byte_20[7]	resp_tx_byte_20[6]	resp_tx_byte_20[5]	resp_tx_byte_20[4]	resp_tx_byte_20[3]	resp_tx_byte_20[2]	resp_tx_byte_20[1]	resp_tx_byte_20[0]
0xD5	0x00	resp_tx_byte_21	rw	resp_tx_byte_21[7]	resp_tx_byte_21[6]	resp_tx_byte_21[5]	resp_tx_byte_21[4]	resp_tx_byte_21[3]	resp_tx_byte_21[2]	resp_tx_byte_21[1]	resp_tx_byte_21[0]
0xD6	0x00	resp_tx_byte_22	rw	resp_tx_byte_22[7]	resp_tx_byte_22[6]	resp_tx_byte_22[5]	resp_tx_byte_22[4]	resp_tx_byte_22[3]	resp_tx_byte_22[2]	resp_tx_byte_22[1]	resp_tx_byte_22[0]
0xD7	0x00	resp_tx_byte_23	rw	resp_tx_byte_23[7]	resp_tx_byte_23[6]	resp_tx_byte_23[5]	resp_tx_byte_23[4]	resp_tx_byte_23[3]	resp_tx_byte_23[2]	resp_tx_byte_23[1]	resp_tx_byte_23[0]
0xD8	0x00	resp_tx_byte_24	rw	resp_tx_byte_24[7]	resp_tx_byte_24[6]	resp_tx_byte_24[5]	resp_tx_byte_24[4]	resp_tx_byte_24[3]	resp_tx_byte_24[2]	resp_tx_byte_24[1]	resp_tx_byte_24[0]
0xD9	0x00	resp_tx_byte_25	rw	resp_tx_byte_25[7]	resp_tx_byte_25[6]	resp_tx_byte_25[5]	resp_tx_byte_25[4]	resp_tx_byte_25[3]	resp_tx_byte_25[2]	resp_tx_byte_25[1]	resp_tx_byte_25[0]
0xDA	0x00	resp_tx_byte_26	rw	resp_tx_byte_26[7]	resp_tx_byte_26[6]	resp_tx_byte_26[5]	resp_tx_byte_26[4]	resp_tx_byte_26[3]	resp_tx_byte_26[2]	resp_tx_byte_26[1]	resp_tx_byte_26[0]
0xDB	0x00	resp_tx_byte_27	rw	resp_tx_byte_27[7]	resp_tx_byte_27[6]	resp_tx_byte_27[5]	resp_tx_byte_27[4]	resp_tx_byte_27[3]	resp_tx_byte_27[2]	resp_tx_byte_27[1]	resp_tx_byte_27[0]
0xDC	0x00	resp_tx_byte_28	rw	resp_tx_byte_28[7]	resp_tx_byte_28[6]	resp_tx_byte_28[5]	resp_tx_byte_28[4]	resp_tx_byte_28[3]	resp_tx_byte_28[2]	resp_tx_byte_28[1]	resp_tx_byte_28[0]
0xDD	0x00	resp_tx_byte_29	rw	resp_tx_byte_29[7]	resp_tx_byte_29[6]	resp_tx_byte_29[5]	resp_tx_byte_29[4]	resp_tx_byte_29[3]	resp_tx_byte_29[2]	resp_tx_byte_29[1]	resp_tx_byte_29[0]
0xDE	0x00	resp_tx_byte_30	rw	resp_tx_byte_30[7]	resp_tx_byte_30[6]	resp_tx_byte_30[5]	resp_tx_byte_30[4]	resp_tx_byte_30[3]	resp_tx_byte_30[2]	resp_tx_byte_30[1]	resp_tx_byte_30[0]
0xDF	0x00	resp_tx_byte_31	rw	resp_tx_byte_31[7]	resp_tx_byte_31[6]	resp_tx_byte_31[5]	resp_tx_byte_31[4]	resp_tx_byte_31[3]	resp_tx_byte_31[2]	resp_tx_byte_31[1]	resp_tx_byte_31[0]
0xE0	0x00	resp_rx_byte_0	r	resp_rx_byte_0[7]	resp_rx_byte_0[6]	resp_rx_byte_0[5]	resp_rx_byte_0[4]	resp_rx_byte_0[3]	resp_rx_byte_0[2]	resp_rx_byte_0[1]	resp_rx_byte_0[0]
0xE1	0x00	resp_rx_byte_1	r	resp_rx_byte_1[7]	resp_rx_byte_1[6]	resp_rx_byte_1[5]	resp_rx_byte_1[4]	resp_rx_byte_1[3]	resp_rx_byte_1[2]	resp_rx_byte_1[1]	resp_rx_byte_1[0]
0xE2	0x00	resp_rx_byte_2	r	resp_rx_byte_2[7]	resp_rx_byte_2[6]	resp_rx_byte_2[5]	resp_rx_byte_2[4]	resp_rx_byte_2[3]	resp_rx_byte_2[2]	resp_rx_byte_2[1]	resp_rx_byte_2[0]
0xE3	0x00	resp_rx_byte_3	r	resp_rx_byte_3[7]	resp_rx_byte_3[6]	resp_rx_byte_3[5]	resp_rx_byte_3[4]	resp_rx_byte_3[3]	resp_rx_byte_3[2]	resp_rx_byte_3[1]	resp_rx_byte_3[0]
0xE4	0x00	resp_rx_byte_4	r	resp_rx_byte_4[7]	resp_rx_byte_4[6]	resp_rx_byte_4[5]	resp_rx_byte_4[4]	resp_rx_byte_4[3]	resp_rx_byte_4[2]	resp_rx_byte_4[1]	resp_rx_byte_4[0]
0xE5	0x00	resp_rx_byte_5	r	resp_rx_byte_5[7]	resp_rx_byte_5[6]	resp_rx_byte_5[5]	resp_rx_byte_5[4]	resp_rx_byte_5[3]	resp_rx_byte_5[2]	resp_rx_byte_5[1]	resp_rx_byte_5[0]
0xE6	0x00	resp_rx_byte_6	r	resp_rx_byte_6[7]	resp_rx_byte_6[6]	resp_rx_byte_6[5]	resp_rx_byte_6[4]	resp_rx_byte_6[3]	resp_rx_byte_6[2]	resp_rx_byte_6[1]	resp_rx_byte_6[0]
0xE7	0x00	resp_rx_byte_7	r	resp_rx_byte_7[7]	resp_rx_byte_7[6]	resp_rx_byte_7[5]	resp_rx_byte_7[4]	resp_rx_byte_7[3]	resp_rx_byte_7[2]	resp_rx_byte_7[1]	resp_rx_byte_7[0]
0xE8	0x00	resp_rx_byte_8	r	resp_rx_byte_8[7]	resp_rx_byte_8[6]	resp_rx_byte_8[5]	resp_rx_byte_8[4]	resp_rx_byte_8[3]	resp_rx_byte_8[2]	resp_rx_byte_8[1]	resp_rx_byte_8[0]
0xE9	0x00	resp_rx_byte_9	r	resp_rx_byte_9[7]	resp_rx_byte_9[6]	resp_rx_byte_9[5]	resp_rx_byte_9[4]	resp_rx_byte_9[3]	resp_rx_byte_9[2]	resp_rx_byte_9[1]	resp_rx_byte_9[0]
0xEA	0x00	resp_rx_byte_10	r	resp_rx_byte_10[7]	resp_rx_byte_10[6]	resp_rx_byte_10[5]	resp_rx_byte_10[4]	resp_rx_byte_10[3]	resp_rx_byte_10[2]	resp_rx_byte_10[1]	resp_rx_byte_10[0]
0xEB	0x00	resp_rx_byte_11	r	resp_rx_byte_11[7]	resp_rx_byte_11[6]	resp_rx_byte_11[5]	resp_rx_byte_11[4]	resp_rx_byte_11[3]	resp_rx_byte_11[2]	resp_rx_byte_11[1]	resp_rx_byte_11[0]
0xEC	0x00	resp_rx_byte_12	r	resp_rx_byte_12[7]	resp_rx_byte_12[6]	resp_rx_byte_12[5]	resp_rx_byte_12[4]	resp_rx_byte_12[3]	resp_rx_byte_12[2]	resp_rx_byte_12[1]	resp_rx_byte_12[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xED	0x00	resp_rx_byte_13	r	resp_rx_byte_13[7]	resp_rx_byte_13[6]	resp_rx_byte_13[5]	resp_rx_byte_13[4]	resp_rx_byte_13[3]	resp_rx_byte_13[2]	resp_rx_byte_13[1]	resp_rx_byte_13[0]
0xEE	0x00	resp_rx_byte_14	r	resp_rx_byte_14[7]	resp_rx_byte_14[6]	resp_rx_byte_14[5]	resp_rx_byte_14[4]	resp_rx_byte_14[3]	resp_rx_byte_14[2]	resp_rx_byte_14[1]	resp_rx_byte_14[0]
0xEF	0x00	resp_rx_byte_15	r	resp_rx_byte_15[7]	resp_rx_byte_15[6]	resp_rx_byte_15[5]	resp_rx_byte_15[4]	resp_rx_byte_15[3]	resp_rx_byte_15[2]	resp_rx_byte_15[1]	resp_rx_byte_15[0]
0xF0	0x00	resp_rx_byte_16	r	resp_rx_byte_16[7]	resp_rx_byte_16[6]	resp_rx_byte_16[5]	resp_rx_byte_16[4]	resp_rx_byte_16[3]	resp_rx_byte_16[2]	resp_rx_byte_16[1]	resp_rx_byte_16[0]
0xF1	0x00	resp_rx_byte_17	r	resp_rx_byte_17[7]	resp_rx_byte_17[6]	resp_rx_byte_17[5]	resp_rx_byte_17[4]	resp_rx_byte_17[3]	resp_rx_byte_17[2]	resp_rx_byte_17[1]	resp_rx_byte_17[0]
0xF2	0x00	resp_rx_byte_18	r	resp_rx_byte_18[7]	resp_rx_byte_18[6]	resp_rx_byte_18[5]	resp_rx_byte_18[4]	resp_rx_byte_18[3]	resp_rx_byte_18[2]	resp_rx_byte_18[1]	resp_rx_byte_18[0]
0xF3	0x00	resp_rx_byte_19	r	resp_rx_byte_19[7]	resp_rx_byte_19[6]	resp_rx_byte_19[5]	resp_rx_byte_19[4]	resp_rx_byte_19[3]	resp_rx_byte_19[2]	resp_rx_byte_19[1]	resp_rx_byte_19[0]
0xF4	0x00	resp_rx_byte_20	r	resp_rx_byte_20[7]	resp_rx_byte_20[6]	resp_rx_byte_20[5]	resp_rx_byte_20[4]	resp_rx_byte_20[3]	resp_rx_byte_20[2]	resp_rx_byte_20[1]	resp_rx_byte_20[0]
0xF5	0x00	resp_rx_byte_21	r	resp_rx_byte_21[7]	resp_rx_byte_21[6]	resp_rx_byte_21[5]	resp_rx_byte_21[4]	resp_rx_byte_21[3]	resp_rx_byte_21[2]	resp_rx_byte_21[1]	resp_rx_byte_21[0]
0xF6	0x00	resp_rx_byte_22	r	resp_rx_byte_22[7]	resp_rx_byte_22[6]	resp_rx_byte_22[5]	resp_rx_byte_22[4]	resp_rx_byte_22[3]	resp_rx_byte_22[2]	resp_rx_byte_22[1]	resp_rx_byte_22[0]
0xF7	0x00	resp_rx_byte_23	r	resp_rx_byte_23[7]	resp_rx_byte_23[6]	resp_rx_byte_23[5]	resp_rx_byte_23[4]	resp_rx_byte_23[3]	resp_rx_byte_23[2]	resp_rx_byte_23[1]	resp_rx_byte_23[0]
0xF8	0x00	resp_rx_byte_24	r	resp_rx_byte_24[7]	resp_rx_byte_24[6]	resp_rx_byte_24[5]	resp_rx_byte_24[4]	resp_rx_byte_24[3]	resp_rx_byte_24[2]	resp_rx_byte_24[1]	resp_rx_byte_24[0]
0xF9	0x00	resp_rx_byte_25	r	resp_rx_byte_25[7]	resp_rx_byte_25[6]	resp_rx_byte_25[5]	resp_rx_byte_25[4]	resp_rx_byte_25[3]	resp_rx_byte_25[2]	resp_rx_byte_25[1]	resp_rx_byte_25[0]
0xFA	0x00	resp_rx_byte_26	r	resp_rx_byte_26[7]	resp_rx_byte_26[6]	resp_rx_byte_26[5]	resp_rx_byte_26[4]	resp_rx_byte_26[3]	resp_rx_byte_26[2]	resp_rx_byte_26[1]	resp_rx_byte_26[0]
0xFB	0x00	resp_rx_byte_27	r	resp_rx_byte_27[7]	resp_rx_byte_27[6]	resp_rx_byte_27[5]	resp_rx_byte_27[4]	resp_rx_byte_27[3]	resp_rx_byte_27[2]	resp_rx_byte_27[1]	resp_rx_byte_27[0]
0xFC	0x00	resp_rx_byte_28	r	resp_rx_byte_28[7]	resp_rx_byte_28[6]	resp_rx_byte_28[5]	resp_rx_byte_28[4]	resp_rx_byte_28[3]	resp_rx_byte_28[2]	resp_rx_byte_28[1]	resp_rx_byte_28[0]
0xFD	0x00	resp_rx_byte_29	r	resp_rx_byte_29[7]	resp_rx_byte_29[6]	resp_rx_byte_29[5]	resp_rx_byte_29[4]	resp_rx_byte_29[3]	resp_rx_byte_29[2]	resp_rx_byte_29[1]	resp_rx_byte_29[0]
0xFE	0x00	resp_rx_byte_30	r	resp_rx_byte_30[7]	resp_rx_byte_30[6]	resp_rx_byte_30[5]	resp_rx_byte_30[4]	resp_rx_byte_30[3]	resp_rx_byte_30[2]	resp_rx_byte_30[1]	resp_rx_byte_30[0]
0xFF	0x00	resp_rx_byte_31	r	resp_rx_byte_31[7]	resp_rx_byte_31[6]	resp_rx_byte_31[5]	resp_rx_byte_31[4]	resp_rx_byte_31[3]	resp_rx_byte_31[2]	resp_rx_byte_31[1]	resp_rx_byte_31[0]

1.6 SDP MAIN MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x0E	user_map_rw_reg_00	rw	-	-	-	insel[4]	insel[3]	insel[2]	insel[1]	insel[0]
0x01	0xC8	user_map_rw_reg_01	rw	-	enhspll	betacam	-	envsproc	sqpe	-	-
0x02	0x04	user_map_rw_reg_02	rw	vid_sel[3]	vid_sel[2]	vid_sel[1]	vid_sel[0]	-	-	-	-
0x03	0x4C	user_map_rw_reg_03	rw	vbi_en	-	-	-	-	-	-	-
0x04	0x35	user_map_rw_reg_04	rw	bt656_4	-	-	-	-	bl_c_vbi	-	range
0x07	0x7F	user_map_rw_reg_07	rw	ad_sec525_en	ad_secam_en	ad_n443_en	ad_p60_en	ad_paln_en	ad_palm_en	ad_ntsc_en	ad_pal_en
0x08	0x80	user_map_rw_reg_08	rw	contrast[7]	contrast[6]	contrast[5]	contrast[4]	contrast[3]	contrast[2]	contrast[1]	contrast[0]
0x0A	0x00	user_map_rw_reg_0a	rw	brightness[7]	brightness[6]	brightness[5]	brightness[4]	brightness[3]	brightness[2]	brightness[1]	brightness[0]
0x0B	0x00	user_map_rw_reg_0b	rw	hue[7]	hue[6]	hue[5]	hue[4]	hue[3]	hue[2]	hue[1]	hue[0]
0x0C	0x36	user_map_rw_reg_0c	rw	def_y[5]	def_y[4]	def_y[3]	def_y[2]	def_y[1]	def_y[0]	def_val_auto_en	def_val_en
0x0D	0x7C	user_map_rw_reg_0d	rw	def_c[7]	def_c[6]	def_c[5]	def_c[4]	def_c[3]	def_c[2]	def_c[1]	def_c[0]
0x0E	0x00	user_map_rw_reg_0e	rw	-	sub_usr_en[1]	sub_usr_en[0]	-	-	r_only_maps_sel[2]	r_only_maps_sel[1]	r_only_maps_sel[0]
0x0F	0x20	user_map_rw_reg_0f	rw	res	-	pwrdn	-	-	-	-	-
0x10	0x00	user_map_rw_reg_10	sc	-	traq	-	-	-	-	-	-
0x14	0x10	user_map_rw_reg_14	rw	-	-	-	cclen	-	free_run_pat_sel[2]	free_run_pat_sel[1]	free_run_pat_sel[0]
0x15	0x00	user_map_rw_reg_15	rw	-	dct[1]	dct[0]	-	-	-	-	-
0x17	0x01	user_map_rw_reg_17	rw	csfm[2]	csfm[1]	csfm[0]	ysfm[4]	ysfm[3]	ysfm[2]	ysfm[1]	ysfm[0]
0x18	0x93	user_map_rw_reg_18	rw	wysfmovr	-	-	wysfm[4]	wysfm[3]	wysfm[2]	wysfm[1]	wysfm[0]
0x19	0xF1	user_map_rw_reg_19	rw	-	-	-	-	nsfsel[1]	nsfsel[0]	psfsel[1]	psfsel[0]
0x27	0x58	user_map_rw_reg_27	rw	swpc	auto_pdc_en	cta[2]	cta[1]	cta[0]	-	lta[1]	lta[0]
0x2B	0xC1	user_map_rw_reg_2b	rw	-	cke	-	-	-	-	-	pw_upd
0x2C	0x22	user_map_rw_reg_2c	rw	-	lagc[2]	lagc[1]	lagc[0]	-	-	cagc[1]	cagc[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x2D	0xC4	user_map_rw_reg_2d	rw	cagt[1]	cagt[0]	-	-	cmg[11]	cmg[10]	cmg[9]	cmg[8]
0x2E	0x00	user_map_rw_reg_2e	rw	cmg[7]	cmg[6]	cmg[5]	cmg[4]	cmg[3]	cmg[2]	cmg[1]	cmg[0]
0x2F	0xC0	user_map_rw_reg_2f	rw	lagt[1]	lagt[0]	-	-	lmg[11]	lmg[10]	lmg[9]	lmg[8]
0x30	0x00	user_map_rw_reg_30	rw	lmg[7]	lmg[6]	lmg[5]	lmg[4]	lmg[3]	lmg[2]	lmg[1]	lmg[0]
0x31	0x02	user_map_rw_reg_31	rw	-	-	-	newavmode	-	-	-	-
0x38	0x80	user_map_rw_reg_38	rw	ctapsn[1]	ctapsn[0]	ccmn[2]	ccmn[1]	ccmn[0]	ycmn[2]	ycmn[1]	ycmn[0]
0x39	0xC0	user_map_rw_reg_39	rw	ctapsp[1]	ctapsp[0]	ccmp[2]	ccmp[1]	ccmp[0]	ycmp[2]	ycmp[1]	ycmp[0]
0x3A	0x00	user_map_rw_reg_3a	rw	-	-	-	-	pdn_adc0	pdn_adc1	pdn_adc2	adc_pdn_override
0x3D	0x22	user_map_rw_reg_3d	rw	-	ckillthr[2]	ckillthr[1]	ckillthr[0]	-	-	-	-
0x48	0x00	user_map_rw_reg_48	rw	gdecel[15]	gdecel[14]	gdecel[13]	gdecel[12]	gdecel[11]	gdecel[10]	gdecel[9]	gdecel[8]
0x49	0x00	user_map_rw_reg_49	rw	gdecel[7]	gdecel[6]	gdecel[5]	gdecel[4]	gdecel[3]	gdecel[2]	gdecel[1]	gdecel[0]
0x4A	0x00	user_map_rw_reg_4a	rw	gdecol[15]	gdecol[14]	gdecol[13]	gdecol[12]	gdecol[11]	gdecol[10]	gdecol[9]	gdecol[8]
0x4B	0x00	user_map_rw_reg_4b	rw	gdecol[7]	gdecol[6]	gdecol[5]	gdecol[4]	gdecol[3]	gdecol[2]	gdecol[1]	gdecol[0]
0x4C	0x00	user_map_rw_reg_4c	rw	-	-	-	-	gde_sel_old_adf	-	-	gdecad
0x4D	0xEF	user_map_rw_reg_4d	rw	-	-	dnr_en	-	cti_ab[1]	cti_ab[0]	cti_ab_en	cti_en
0x4E	0x08	user_map_rw_reg_4e	rw	cti_c_th[7]	cti_c_th[6]	cti_c_th[5]	cti_c_th[4]	cti_c_th[3]	cti_c_th[2]	cti_c_th[1]	cti_c_th[0]
0x50	0x08	user_map_rw_reg_50	rw	dnr_th[7]	dnr_th[6]	dnr_th[5]	dnr_th[4]	dnr_th[3]	dnr_th[2]	dnr_th[1]	dnr_th[0]
0x51	0x24	user_map_rw_reg_51	rw	fscle	srls	col[2]	col[1]	col[0]	cil[2]	cil[1]	cil[0]
0x60	0x10	user_map_rw_reg_60	rw	-	-	-	-	adc0n_sw[3]	adc0n_sw[2]	adc0n_sw[1]	adc0n_sw[0]
0xC3	0x00	user_map_rw_reg_c3	rw	adc1_sw[3]	adc1_sw[2]	adc1_sw[1]	adc1_sw[0]	adc0_sw[3]	adc0_sw[2]	adc0_sw[1]	adc0_sw[0]
0xC4	0x00	user_map_rw_reg_c4	rw	adc_sw_man	-	-	-	adc2_sw[3]	adc2_sw[2]	adc2_sw[1]	adc2_sw[0]
0xDC	0xAC	user_map_rw_reg_dc	rw	-	-	-	lb_th[4]	lb_th[3]	lb_th[2]	lb_th[1]	lb_th[0]
0xDD	0x4C	user_map_rw_reg_dd	rw	lb_sl[3]	lb_sl[2]	lb_sl[1]	lb_sl[0]	lb_el[3]	lb_el[2]	lb_el[1]	lb_el[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE1	0x80	user_map_rw_reg_e1	rw	sd_off_u[7]	sd_off_u[6]	sd_off_u[5]	sd_off_u[4]	sd_off_u[3]	sd_off_u[2]	sd_off_u[1]	sd_off_u[0]
0xE2	0x80	user_map_rw_reg_e2	rw	sd_off_v[7]	sd_off_v[6]	sd_off_v[5]	sd_off_v[4]	sd_off_v[3]	sd_off_v[2]	sd_off_v[1]	sd_off_v[0]
0xE3	0x80	user_map_rw_reg_e3	rw	sd_sat_u[7]	sd_sat_u[6]	sd_sat_u[5]	sd_sat_u[4]	sd_sat_u[3]	sd_sat_u[2]	sd_sat_u[1]	sd_sat_u[0]
0xE4	0x80	user_map_rw_reg_e4	rw	sd_sat_v[7]	sd_sat_v[6]	sd_sat_v[5]	sd_sat_v[4]	sd_sat_v[3]	sd_sat_v[2]	sd_sat_v[1]	sd_sat_v[0]
0xE5	0x25	user_map_rw_reg_e5	rw	nvbegdelo	nvbegdele	nvbegsign	nvbeg[4]	nvbeg[3]	nvbeg[2]	nvbeg[1]	nvbeg[0]
0xE6	0x04	user_map_rw_reg_e6	rw	nvenddelo	nvenddele	nvendsign	nvend[4]	nvend[3]	nvend[2]	nvend[1]	nvend[0]
0xE7	0x63	user_map_rw_reg_e7	rw	nftogdelo	nftogdele	nftogsign	nftog[4]	nftog[3]	nftog[2]	nftog[1]	nftog[0]
0xE8	0x65	user_map_rw_reg_e8	rw	pvbegdelo	pvbegdele	pvbegsign	pvbeg[4]	pvbeg[3]	pvbeg[2]	pvbeg[1]	pvbeg[0]
0xE9	0x14	user_map_rw_reg_e9	rw	pvenddelo	pvenddele	pvendsign	pvend[4]	pvend[3]	pvend[2]	pvend[1]	pvend[0]
0xEA	0x63	user_map_rw_reg_ea	rw	pftogdelo	pftogdele	pftogsign	pftog[4]	pftog[3]	pftog[2]	pftog[1]	pftog[0]
0xEB	0x55	user_map_rw_reg_eb	rw	nvbiolcm[1]	nvbiolcm[0]	nvbielcm[1]	nvbielcm[0]	pvbiolcm[1]	pvbiolcm[0]	pvbielcm[1]	pvbielcm[0]
0xEC	0x55	user_map_rw_reg_ec	rw	nvbioccm[1]	nvbioccm[0]	nvbieccm[1]	nvbieccm[0]	pvbioccm[1]	pvbioccm[0]	pvbieccm[1]	pvbieccm[0]
0xF3	0x00	user_map_rw_reg_f3	rw	-	-	-	aa_filt_man_ovr	aa_filt_en[3]	aa_filt_en[2]	aa_filt_en[1]	aa_filt_en[0]
0xF8	0x00	user_map_rw_reg_f8	rw	-	-	-	-	-	iffiltsel[2]	iffiltsel[1]	iffiltsel[0]
0xF9	0x03	user_map_rw_reg_f9	rw	-	-	-	-	vs_coast_mode[1]	vs_coast_mode[0]	extend_vs_min_freq	extend_vs_max_freq
0xFB	0x40	user_map_rw_reg_fb	rw	peaking_gain[7]	peaking_gain[6]	peaking_gain[5]	peaking_gain[4]	peaking_gain[3]	peaking_gain[2]	peaking_gain[1]	peaking_gain[0]
0xFC	0x04	user_map_rw_reg_fc	rw	dnr_th_2[7]	dnr_th_2[6]	dnr_th_2[5]	dnr_th_2[4]	dnr_th_2[3]	dnr_th_2[2]	dnr_th_2[1]	dnr_th_2[0]

1.7 SDP R/O MAIN MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x0E	0x00	user_map_r_reg_0e	r	-	sub_usr_en_rb[1]	sub_usr_en_rb[0]	-	-	r_only_maps_sel_rb[2]	r_only_maps_sel_rb[1]	r_only_maps_sel_rb[0]
0x10	0x00	user_map_r_reg_10	r	col_kill	ad_result[2]	ad_result[1]	ad_result[0]	follow_pw	fsc_lock	lost_lock	in_lock
0x12	0x00	user_map_r_reg_12	r	-	-	fsc nstd	ll nstd	mv agc det	mv ps det	mvcs t3	mvcs det
0x13	0x00	user_map_r_reg_13	r	pal sw lock	interlace	std fld len	free_run_act	-	sd_op_50hz_rb	-	inst_hlock
0x90	0x00	user_map_r_reg_90	r	-	-	-	-	cgmsd	edtvd	ccapd_rb	wssd
0x91	0x00	user_map_r_reg_91	r	wss1[7]	wss1[6]	wss1[5]	wss1[4]	wss1[3]	wss1[2]	wss1[1]	wss1[0]
0x92	0x00	user_map_r_reg_92	r	wss2[7]	wss2[6]	wss2[5]	wss2[4]	wss2[3]	wss2[2]	wss2[1]	wss2[0]
0x93	0x00	user_map_r_reg_93	r	edtv1[7]	edtv1[6]	edtv1[5]	edtv1[4]	edtv1[3]	edtv1[2]	edtv1[1]	edtv1[0]
0x94	0x00	user_map_r_reg_94	r	edtv2[7]	edtv2[6]	edtv2[5]	edtv2[4]	edtv2[3]	edtv2[2]	edtv2[1]	edtv2[0]
0x95	0x00	user_map_r_reg_95	r	edtv3[7]	edtv3[6]	edtv3[5]	edtv3[4]	edtv3[3]	edtv3[2]	edtv3[1]	edtv3[0]
0x96	0x00	user_map_r_reg_96	r	cgms1[7]	cgms1[6]	cgms1[5]	cgms1[4]	cgms1[3]	cgms1[2]	cgms1[1]	cgms1[0]
0x97	0x00	user_map_r_reg_97	r	cgms2[7]	cgms2[6]	cgms2[5]	cgms2[4]	cgms2[3]	cgms2[2]	cgms2[1]	cgms2[0]
0x98	0x00	user_map_r_reg_98	r	cgms3[7]	cgms3[6]	cgms3[5]	cgms3[4]	cgms3[3]	cgms3[2]	cgms3[1]	cgms3[0]
0x99	0x00	user_map_r_reg_99	r	ccap1[7]	ccap1[6]	ccap1[5]	ccap1[4]	ccap1[3]	ccap1[2]	ccap1[1]	ccap1[0]
0x9A	0x00	user_map_r_reg_9a	r	ccap2[7]	ccap2[6]	ccap2[5]	ccap2[4]	ccap2[3]	ccap2[2]	ccap2[1]	ccap2[0]
0x9B	0x00	user_map_r_reg_9b	r	lb_lct[7]	lb_lct[6]	lb_lct[5]	lb_lct[4]	lb_lct[3]	lb_lct[2]	lb_lct[1]	lb_lct[0]
0x9C	0x00	user_map_r_reg_9c	r	lb_lcm[7]	lb_lcm[6]	lb_lcm[5]	lb_lcm[4]	lb_lcm[3]	lb_lcm[2]	lb_lcm[1]	lb_lcm[0]
0x9D	0x00	user_map_r_reg_9d	r	lb_lcb[7]	lb_lcb[6]	lb_lcb[5]	lb_lcb[4]	lb_lcb[3]	lb_lcb[2]	lb_lcb[1]	lb_lcb[0]
0xDE	0x00	user_map_r_reg_de	r	-	-	-	-	st_noise_vld	st_noise[10]	st_noise[9]	st_noise[8]
0xDF	0x00	user_map_r_reg_df	r	st_noise[7]	st_noise[6]	st_noise[5]	st_noise[4]	st_noise[3]	st_noise[2]	st_noise[1]	st_noise[0]

1.8 SDP MAP 1

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x40	0x10	user_sub_map_1_rw_reg_40	rw	-	-	mv_intrq_sel[1]	mv_intrq_sel[0]	-	mpu_stim_intrq	-	-
0x43	0x00	user_sub_map_1_rw_reg_43	sc	-	mv_ps_cs_clr	sd_fr_chng_clr	-	-	-	sd_unlock_clr	sd_lock_clr
0x44	0x00	user_sub_map_1_rw_reg_44	rw	-	mv_ps_cs_mskb	sd_fr_chng_mskb	-	-	-	sd_unlock_mskb	sd_lock_mskb
0x47	0x00	user_sub_map_1_rw_reg_47	sc	mpu_stim_intrq_clr	-	chx_min_max_intrq_clr	sd_field_chngd_clr	-	-	-	ccapd_clr
0x48	0x00	user_sub_map_1_rw_reg_48	rw	mpu_stim_intrq_mskb	-	chx_min_max_intrq_mskb	sd_field_chngd_mskb	-	-	-	ccapd_mskb
0x4B	0x00	user_sub_map_1_rw_reg_4b	sc	-	-	pal_sw_lk_chng_clr	scm_lock_chng_clr	sd_ad_chng_clr	sd_h_lock_chng_clr	sd_v_lock_chng_clr	sd_op_chng_clr
0x4C	0x00	user_sub_map_1_rw_reg_4c	rw	-	-	pal_sw_lk_chng_mskb	scm_lock_chng_mskb	sd_ad_chng_mskb	sd_h_lock_chng_mskb	sd_v_lock_chng_mskb	sd_op_chng_mskb
0x4F	0x00	user_sub_map_1_rw_reg_4f	sc	-	vdp_vitc_clr	-	vdp_pdc_vps_utc_chng_clr	-	vdp_cgms_wss_chngd_clr	-	vdp_ccapd_clr
0x50	0x00	user_sub_map_1_rw_reg_50	rw	-	vdp_vitc_mskb	-	vdp_pdc_vps_utc_chng_mskb	-	vdp_cgms_wss_chngd_mskb	-	vdp_ccapd_mskb
0x61	0x10	user_sub_map_1_rw_reg_61	rw	-	-	-	auto_detect_gs_type	-	-	-	-
0x62	0x15	user_sub_map_1_rw_reg_62	rw	adf_enable	adf_mode[1]	adf_mode[0]	adf_did[4]	adf_did[3]	adf_did[2]	adf_did[1]	adf_did[0]
0x63	0x2A	user_sub_map_1_rw_reg_63	rw	duplicate_adf	-	adf_sdid[5]	adf_sdid[4]	adf_sdid[3]	adf_sdid[2]	adf_sdid[1]	adf_sdid[0]
0x64	0x00	user_sub_map_1_rw_reg_64	rw	man_line_pgm	-	-	-	vbi_data_p318[3]	vbi_data_p318[2]	vbi_data_p318[1]	vbi_data_p318[0]
0x65	0x00	user_sub_map_1_rw_reg_65	rw	vbi_data_p6_n23[3]	vbi_data_p6_n23[2]	vbi_data_p6_n23[1]	vbi_data_p6_n23[0]	vbi_data_p319_n286[3]	vbi_data_p319_n286[2]	vbi_data_p319_n286[1]	vbi_data_p319_n286[0]
0x66	0x00	user_sub_map_1_rw_reg_66	rw	vbi_data_p7_n24[3]	vbi_data_p7_n24[2]	vbi_data_p7_n24[1]	vbi_data_p7_n24[0]	vbi_data_p320_n287[3]	vbi_data_p320_n287[2]	vbi_data_p320_n287[1]	vbi_data_p320_n287[0]
0x67	0x00	user_sub_map_1_rw_reg_67	rw	vbi_data_p8_n25[3]	vbi_data_p8_n25[2]	vbi_data_p8_n25[1]	vbi_data_p8_n25[0]	vbi_data_p321_n288[3]	vbi_data_p321_n288[2]	vbi_data_p321_n288[1]	vbi_data_p321_n288[0]
0x68	0x00	user_sub_map_1_rw_reg_68	rw	vbi_data_p9[3]	vbi_data_p9[2]	vbi_data_p9[1]	vbi_data_p9[0]	vbi_data_p322[3]	vbi_data_p322[2]	vbi_data_p322[1]	vbi_data_p322[0]
0x69	0x00	user_sub_map_1_rw_reg_69	rw	vbi_data_p10[3]	vbi_data_p10[2]	vbi_data_p10[1]	vbi_data_p10[0]	vbi_data_p323[3]	vbi_data_p323[2]	vbi_data_p323[1]	vbi_data_p323[0]
0x6A	0x00	user_sub_map_1_rw_reg_6a	rw	vbi_data_p11[3]	vbi_data_p11[2]	vbi_data_p11[1]	vbi_data_p11[0]	vbi_data_p324_n272[3]	vbi_data_p324_n272[2]	vbi_data_p324_n272[1]	vbi_data_p324_n272[0]
0x6B	0x00	user_sub_map_1_rw_reg_6b	rw	vbi_data_p12_n10[3]	vbi_data_p12_n10[2]	vbi_data_p12_n10[1]	vbi_data_p12_n10[0]	vbi_data_p325_n273[3]	vbi_data_p325_n273[2]	vbi_data_p325_n273[1]	vbi_data_p325_n273[0]
0x6C	0x00	user_sub_map_1_rw_reg_6c	rw	vbi_data_p13_n11[3]	vbi_data_p13_n11[2]	vbi_data_p13_n11[1]	vbi_data_p13_n11[0]	vbi_data_p326_n274[3]	vbi_data_p326_n274[2]	vbi_data_p326_n274[1]	vbi_data_p326_n274[0]
0x6D	0x00	user_sub_map_1_rw_reg_6d	rw	vbi_data_p14_n12[3]	vbi_data_p14_n12[2]	vbi_data_p14_n12[1]	vbi_data_p14_n12[0]	vbi_data_p327_n275[3]	vbi_data_p327_n275[2]	vbi_data_p327_n275[1]	vbi_data_p327_n275[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x6E	0x00	user_sub_map_1_rw_reg_6e	rw	vbi_data_p15_n13[3]	vbi_data_p15_n13[2]	vbi_data_p15_n13[1]	vbi_data_p15_n13[0]	vbi_data_p328_n276[3]	vbi_data_p328_n276[2]	vbi_data_p328_n276[1]	vbi_data_p328_n276[0]
0x6F	0x00	user_sub_map_1_rw_reg_6f	rw	vbi_data_p16_n14[3]	vbi_data_p16_n14[2]	vbi_data_p16_n14[1]	vbi_data_p16_n14[0]	vbi_data_p329_n277[3]	vbi_data_p329_n277[2]	vbi_data_p329_n277[1]	vbi_data_p329_n277[0]
0x70	0x00	user_sub_map_1_rw_reg_70	rw	vbi_data_p17_n15[3]	vbi_data_p17_n15[2]	vbi_data_p17_n15[1]	vbi_data_p17_n15[0]	vbi_data_p330_n278[3]	vbi_data_p330_n278[2]	vbi_data_p330_n278[1]	vbi_data_p330_n278[0]
0x71	0x00	user_sub_map_1_rw_reg_71	rw	vbi_data_p18_n16[3]	vbi_data_p18_n16[2]	vbi_data_p18_n16[1]	vbi_data_p18_n16[0]	vbi_data_p331_n279[3]	vbi_data_p331_n279[2]	vbi_data_p331_n279[1]	vbi_data_p331_n279[0]
0x72	0x00	user_sub_map_1_rw_reg_72	rw	vbi_data_p19_n17[3]	vbi_data_p19_n17[2]	vbi_data_p19_n17[1]	vbi_data_p19_n17[0]	vbi_data_p332_n280[3]	vbi_data_p332_n280[2]	vbi_data_p332_n280[1]	vbi_data_p332_n280[0]
0x73	0x00	user_sub_map_1_rw_reg_73	rw	vbi_data_p20_n18[3]	vbi_data_p20_n18[2]	vbi_data_p20_n18[1]	vbi_data_p20_n18[0]	vbi_data_p333_n281[3]	vbi_data_p333_n281[2]	vbi_data_p333_n281[1]	vbi_data_p333_n281[0]
0x74	0x00	user_sub_map_1_rw_reg_74	rw	vbi_data_p21_n19[3]	vbi_data_p21_n19[2]	vbi_data_p21_n19[1]	vbi_data_p21_n19[0]	vbi_data_p334_n282[3]	vbi_data_p334_n282[2]	vbi_data_p334_n282[1]	vbi_data_p334_n282[0]
0x75	0x00	user_sub_map_1_rw_reg_75	rw	vbi_data_p22_n20[3]	vbi_data_p22_n20[2]	vbi_data_p22_n20[1]	vbi_data_p22_n20[0]	vbi_data_p335_n283[3]	vbi_data_p335_n283[2]	vbi_data_p335_n283[1]	vbi_data_p335_n283[0]
0x76	0x00	user_sub_map_1_rw_reg_76	rw	vbi_data_p23_n21[3]	vbi_data_p23_n21[2]	vbi_data_p23_n21[1]	vbi_data_p23_n21[0]	vbi_data_p336_n284[3]	vbi_data_p336_n284[2]	vbi_data_p336_n284[1]	vbi_data_p336_n284[0]
0x77	0x00	user_sub_map_1_rw_reg_77	rw	vbi_data_p24_n22[3]	vbi_data_p24_n22[2]	vbi_data_p24_n22[1]	vbi_data_p24_n22[0]	vbi_data_p337_n285[3]	vbi_data_p337_n285[2]	vbi_data_p337_n285[1]	vbi_data_p337_n285[0]
0x78	0x00	user_sub_map_1_rw_reg_78	sc	-	vitc_clr	-	gs_pdc_vps_utc_clr	-	cgms_wss_clr	-	cc_clr
0x9C	0x30	user_sub_map_1_rw_reg_9c	rw	-	-	-	wss_cgms_cb_change	-	-	-	-

1.9 SDP R/O MAP 1

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x0E	0x00	user_sub_map_1_r_reg_0e	r	-	sub_usr_en_rb[1]	sub_usr_en_rb[0]	-	-	r_only_maps_sel_rb[2]	r_only_maps_sel_rb[1]	r_only_maps_sel_rb[0]
0x42	0x00	user_sub_map_1_r_reg_42	r	-	mv_ps_cs_q	sd_fr_chng_q	-	-	-	sd_unlock_q	sd_lock_q
0x45	0x00	user_sub_map_1_r_reg_45	r	mpu_stim_intrq	-	chx_min_max_intrq	even_field	-	-	-	ccapd
0x46	0x00	user_sub_map_1_r_reg_46	r	mpu_stim_intrq_q	-	chx_min_max_intrq_q	sd_field_chngd_q	-	-	-	ccapd_q
0x49	0x00	user_sub_map_1_r_reg_49	r	-	-	-	scm_lock	-	sd_h_lock	sd_v_lock	sd_op_50hz
0x4A	0x00	user_sub_map_1_r_reg_4a	r	-	-	pal_sw_lk_chng_q	scm_lock_chng_q	sd_ad_chng_q	sd_h_lock_chng_q	sd_v_lock_chng_q	sd_op_chng_q
0x4E	0x00	user_sub_map_1_r_reg_4e	r	-	vdp_vitc_q	-	vdp_pdc_vps_utc_chng_q	-	vdp_cgms_wss_chngd_q	-	vdp_ccapd_q
0x51	0x00	user_sub_map_1_r_reg_51	r	-	-	y channel min violation	y channel max violation	cb channel min violation	cb channel max violation	cr channel min violation	cr channel max violation
0x53	0x00	user_sub_map_1_r_reg_53	r	-	-	-	-	diag_tri2_l1	diag_tri2_l0	diag_tri1_l1	diag_tri1_l0
0x78	0x00	user_sub_map_1_r_reg_78	r	-	-	-	-	-	cgms_wss_avl	cc_even_field	cc_avl
0x79	0x00	user_sub_map_1_r_reg_79	r	vdp_ccap_data_0[7]	vdp_ccap_data_0[6]	vdp_ccap_data_0[5]	vdp_ccap_data_0[4]	vdp_ccap_data_0[3]	vdp_ccap_data_0[2]	vdp_ccap_data_0[1]	vdp_ccap_data_0[0]
0x7A	0x00	user_sub_map_1_r_reg_7a	r	vdp_ccap_data_1[7]	vdp_ccap_data_1[6]	vdp_ccap_data_1[5]	vdp_ccap_data_1[4]	vdp_ccap_data_1[3]	vdp_ccap_data_1[2]	vdp_ccap_data_1[1]	vdp_ccap_data_1[0]
0x7D	0x00	user_sub_map_1_r_reg_7d	r	-	-	-	-	vdp_cgms_wss_data_crc[5]	vdp_cgms_wss_data_crc[4]	vdp_cgms_wss_data_crc[3]	vdp_cgms_wss_data_crc[2]
0x7E	0x00	user_sub_map_1_r_reg_7e	r	vdp_cgms_wss_data_crc[1]	vdp_cgms_wss_data_crc[0]	vdp_cgms_wss_data[13]	vdp_cgms_wss_data[12]	vdp_cgms_wss_data[11]	vdp_cgms_wss_data[10]	vdp_cgms_wss_data[9]	vdp_cgms_wss_data[8]
0x7F	0x00	user_sub_map_1_r_reg_7f	r	vdp_cgms_wss_data[7]	vdp_cgms_wss_data[6]	vdp_cgms_wss_data[5]	vdp_cgms_wss_data[4]	vdp_cgms_wss_data[3]	vdp_cgms_wss_data[2]	vdp_cgms_wss_data[1]	vdp_cgms_wss_data[0]
0x84	0x00	user_sub_map_1_r_reg_84	r	vps_pdc_utc_byte_0[7]	vps_pdc_utc_byte_0[6]	vps_pdc_utc_byte_0[5]	vps_pdc_utc_byte_0[4]	vps_pdc_utc_byte_0[3]	vps_pdc_utc_byte_0[2]	vps_pdc_utc_byte_0[1]	vps_pdc_utc_byte_0[0]
0x85	0x00	user_sub_map_1_r_reg_85	r	vps_pdc_utc_byte_1[7]	vps_pdc_utc_byte_1[6]	vps_pdc_utc_byte_1[5]	vps_pdc_utc_byte_1[4]	vps_pdc_utc_byte_1[3]	vps_pdc_utc_byte_1[2]	vps_pdc_utc_byte_1[1]	vps_pdc_utc_byte_1[0]
0x86	0x00	user_sub_map_1_r_reg_86	r	vps_pdc_utc_byte_2[7]	vps_pdc_utc_byte_2[6]	vps_pdc_utc_byte_2[5]	vps_pdc_utc_byte_2[4]	vps_pdc_utc_byte_2[3]	vps_pdc_utc_byte_2[2]	vps_pdc_utc_byte_2[1]	vps_pdc_utc_byte_2[0]
0x87	0x00	user_sub_map_1_r_reg_87	r	vps_pdc_utc_byte_3[7]	vps_pdc_utc_byte_3[6]	vps_pdc_utc_byte_3[5]	vps_pdc_utc_byte_3[4]	vps_pdc_utc_byte_3[3]	vps_pdc_utc_byte_3[2]	vps_pdc_utc_byte_3[1]	vps_pdc_utc_byte_3[0]
0x88	0x00	user_sub_map_1_r_reg_88	r	vps_pdc_utc_byte_4[7]	vps_pdc_utc_byte_4[6]	vps_pdc_utc_byte_4[5]	vps_pdc_utc_byte_4[4]	vps_pdc_utc_byte_4[3]	vps_pdc_utc_byte_4[2]	vps_pdc_utc_byte_4[1]	vps_pdc_utc_byte_4[0]
0x89	0x00	user_sub_map_1_r_reg_89	r	vps_pdc_utc_byte_5[7]	vps_pdc_utc_byte_5[6]	vps_pdc_utc_byte_5[5]	vps_pdc_utc_byte_5[4]	vps_pdc_utc_byte_5[3]	vps_pdc_utc_byte_5[2]	vps_pdc_utc_byte_5[1]	vps_pdc_utc_byte_5[0]
0x8A	0x00	user_sub_map_1_r_reg_8a	r	vps_pdc_utc_byte_6[7]	vps_pdc_utc_byte_6[6]	vps_pdc_utc_byte_6[5]	vps_pdc_utc_byte_6[4]	vps_pdc_utc_byte_6[3]	vps_pdc_utc_byte_6[2]	vps_pdc_utc_byte_6[1]	vps_pdc_utc_byte_6[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x8B	0x00	user_sub_map_1_r_reg_8b	r	vps_pdc_utc_byte_7[7]	vps_pdc_utc_byte_7[6]	vps_pdc_utc_byte_7[5]	vps_pdc_utc_byte_7[4]	vps_pdc_utc_byte_7[3]	vps_pdc_utc_byte_7[2]	vps_pdc_utc_byte_7[1]	vps_pdc_utc_byte_7[0]
0x8C	0x00	user_sub_map_1_r_reg_8c	r	vps_pdc_utc_byte_8[7]	vps_pdc_utc_byte_8[6]	vps_pdc_utc_byte_8[5]	vps_pdc_utc_byte_8[4]	vps_pdc_utc_byte_8[3]	vps_pdc_utc_byte_8[2]	vps_pdc_utc_byte_8[1]	vps_pdc_utc_byte_8[0]
0x8D	0x00	user_sub_map_1_r_reg_8d	r	vps_pdc_utc_byte_9[7]	vps_pdc_utc_byte_9[6]	vps_pdc_utc_byte_9[5]	vps_pdc_utc_byte_9[4]	vps_pdc_utc_byte_9[3]	vps_pdc_utc_byte_9[2]	vps_pdc_utc_byte_9[1]	vps_pdc_utc_byte_9[0]
0x8E	0x00	user_sub_map_1_r_reg_8e	r	vps_pdc_utc_byte_10[7]	vps_pdc_utc_byte_10[6]	vps_pdc_utc_byte_10[5]	vps_pdc_utc_byte_10[4]	vps_pdc_utc_byte_10[3]	vps_pdc_utc_byte_10[2]	vps_pdc_utc_byte_10[1]	vps_pdc_utc_byte_10[0]
0x8F	0x00	user_sub_map_1_r_reg_8f	r	vps_pdc_utc_byte_11[7]	vps_pdc_utc_byte_11[6]	vps_pdc_utc_byte_11[5]	vps_pdc_utc_byte_11[4]	vps_pdc_utc_byte_11[3]	vps_pdc_utc_byte_11[2]	vps_pdc_utc_byte_11[1]	vps_pdc_utc_byte_11[0]
0x90	0x00	user_sub_map_1_r_reg_90	r	vps_pdc_utc_byte_12[7]	vps_pdc_utc_byte_12[6]	vps_pdc_utc_byte_12[5]	vps_pdc_utc_byte_12[4]	vps_pdc_utc_byte_12[3]	vps_pdc_utc_byte_12[2]	vps_pdc_utc_byte_12[1]	vps_pdc_utc_byte_12[0]
0x91	0x00	user_sub_map_1_r_reg_91	r	vps_biphase_error_count[7]	vps_biphase_error_count[6]	vps_biphase_error_count[5]	vps_biphase_error_count[4]	vps_biphase_error_count[3]	vps_biphase_error_count[2]	vps_biphase_error_count[1]	vps_biphase_error_count[0]
0x92	0x00	user_sub_map_1_r_reg_92	r	vitc_data_1[7]	vitc_data_1[6]	vitc_data_1[5]	vitc_data_1[4]	vitc_data_1[3]	vitc_data_1[2]	vitc_data_1[1]	vitc_data_1[0]
0x93	0x00	user_sub_map_1_r_reg_93	r	vitc_data_2[7]	vitc_data_2[6]	vitc_data_2[5]	vitc_data_2[4]	vitc_data_2[3]	vitc_data_2[2]	vitc_data_2[1]	vitc_data_2[0]
0x94	0x00	user_sub_map_1_r_reg_94	r	vitc_data_3[7]	vitc_data_3[6]	vitc_data_3[5]	vitc_data_3[4]	vitc_data_3[3]	vitc_data_3[2]	vitc_data_3[1]	vitc_data_3[0]
0x95	0x00	user_sub_map_1_r_reg_95	r	vitc_data_4[7]	vitc_data_4[6]	vitc_data_4[5]	vitc_data_4[4]	vitc_data_4[3]	vitc_data_4[2]	vitc_data_4[1]	vitc_data_4[0]
0x96	0x00	user_sub_map_1_r_reg_96	r	vitc_data_5[7]	vitc_data_5[6]	vitc_data_5[5]	vitc_data_5[4]	vitc_data_5[3]	vitc_data_5[2]	vitc_data_5[1]	vitc_data_5[0]
0x97	0x00	user_sub_map_1_r_reg_97	r	vitc_data_6[7]	vitc_data_6[6]	vitc_data_6[5]	vitc_data_6[4]	vitc_data_6[3]	vitc_data_6[2]	vitc_data_6[1]	vitc_data_6[0]
0x98	0x00	user_sub_map_1_r_reg_98	r	vitc_data_7[7]	vitc_data_7[6]	vitc_data_7[5]	vitc_data_7[4]	vitc_data_7[3]	vitc_data_7[2]	vitc_data_7[1]	vitc_data_7[0]
0x99	0x00	user_sub_map_1_r_reg_99	r	vitc_data_8[7]	vitc_data_8[6]	vitc_data_8[5]	vitc_data_8[4]	vitc_data_8[3]	vitc_data_8[2]	vitc_data_8[1]	vitc_data_8[0]
0x9A	0x00	user_sub_map_1_r_reg_9a	r	vitc_data_9[7]	vitc_data_9[6]	vitc_data_9[5]	vitc_data_9[4]	vitc_data_9[3]	vitc_data_9[2]	vitc_data_9[1]	vitc_data_9[0]
0x9B	0x00	user_sub_map_1_r_reg_9b	r	vitc_crc[7]	vitc_crc[6]	vitc_crc[5]	vitc_crc[4]	vitc_crc[3]	vitc_crc[2]	vitc_crc[1]	vitc_crc[0]

1.10 SDP MAP 2

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x80	0x00	user_sub_map_2_rw_reg_80	rw	ace_enable	-	-	-	-	-	-	-
0x83	0x0D	user_sub_map_2_rw_reg_83	rw	-	-	-	ace_luma_gain[4]	ace_luma_gain[3]	ace_luma_gain[2]	ace_luma_gain[1]	ace_luma_gain[0]
0x84	0x88	user_sub_map_2_rw_reg_84	rw	ace_chroma_max[3]	ace_chroma_max[2]	ace_chroma_max[1]	ace_chroma_max[0]	ace_chroma_gain[3]	ace_chroma_gain[2]	ace_chroma_gain[1]	ace_chroma_gain[0]
0x85	0xF8	user_sub_map_2_rw_reg_85	rw	ace_response_speed[3]	ace_response_speed[2]	ace_response_speed[1]	ace_response_speed[0]	ace_gamma_gain[3]	ace_gamma_gain[2]	ace_gamma_gain[1]	ace_gamma_gain[0]
0x92	0x06	user_sub_map_2_rw_reg_92	rw	-	-	-	-	-	-	-	sdp_br_dither_mode
0xD9	0x00	user_sub_map_2_rw_reg_d9	rw	min_thresh_y[7]	min_thresh_y[6]	min_thresh_y[5]	min_thresh_y[4]	min_thresh_y[3]	min_thresh_y[2]	min_thresh_y[1]	min_thresh_y[0]
0xDA	0xFF	user_sub_map_2_rw_reg_da	rw	max_thresh_y[7]	max_thresh_y[6]	max_thresh_y[5]	max_thresh_y[4]	max_thresh_y[3]	max_thresh_y[2]	max_thresh_y[1]	max_thresh_y[0]
0xDB	0x00	user_sub_map_2_rw_reg_db	rw	min_thresh_c[7]	min_thresh_c[6]	min_thresh_c[5]	min_thresh_c[4]	min_thresh_c[3]	min_thresh_c[2]	min_thresh_c[1]	min_thresh_c[0]
0xDC	0xFF	user_sub_map_2_rw_reg_dc	rw	max_thresh_c[7]	max_thresh_c[6]	max_thresh_c[5]	max_thresh_c[4]	max_thresh_c[3]	max_thresh_c[2]	max_thresh_c[1]	max_thresh_c[0]
0xDD	0xCC	user_sub_map_2_rw_reg_dd	rw	min_samples_allowed_y[3]	min_samples_allowed_y[2]	min_samples_allowed_y[1]	min_samples_allowed_y[0]	max_samples_allowed_y[3]	max_samples_allowed_y[2]	max_samples_allowed_y[1]	max_samples_allowed_y[0]
0xDE	0xCC	user_sub_map_2_rw_reg_de	rw	min_samples_allowed_c[3]	min_samples_allowed_c[2]	min_samples_allowed_c[1]	min_samples_allowed_c[0]	max_samples_allowed_c[3]	max_samples_allowed_c[2]	max_samples_allowed_c[1]	max_samples_allowed_c[0]
0xE0	0x00	user_sub_map_2_rw_reg_e0	rw	-	-	-	-	-	-	-	fl_enable
0xE1	0x11	user_sub_map_2_rw_reg_e1	rw	line_start[8]	line_start[7]	line_start[6]	line_start[5]	line_start[4]	line_start[3]	line_start[2]	line_start[1]
0xE2	0x88	user_sub_map_2_rw_reg_e2	rw	line_end[8]	line_end[7]	line_end[6]	line_end[5]	line_end[4]	line_end[3]	line_end[2]	line_end[1]
0xE3	0x1B	user_sub_map_2_rw_reg_e3	rw	sample_start[9]	sample_start[8]	sample_start[7]	sample_start[6]	sample_start[5]	sample_start[4]	sample_start[3]	sample_start[2]
0xE4	0xD7	user_sub_map_2_rw_reg_e4	rw	sample_end[9]	sample_end[8]	sample_end[7]	sample_end[6]	sample_end[5]	sample_end[4]	sample_end[3]	sample_end[2]
0xE5	0x23	user_sub_map_2_rw_reg_e5	rw	sample_end[1]	sample_end[0]	sample_start[1]	sample_start[0]	-	-	line_end[0]	line_start[0]
0xE6	0x10	user_sub_map_2_rw_reg_e6	rw	-	-	-	y_avg_time_const[2]	y_avg_time_const[1]	y_avg_time_const[0]	y_avg_filt_en	-
0xE7	0x00	user_sub_map_2_rw_reg_e7	sc	-	-	-	-	-	-	-	capture_value

1.11 SDP R/O MAP 2

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x0E	0x00	user_sub_map_2_r_reg_0e	r	-	sub_usr_en_rb[1]	sub_usr_en_rb[0]	-	-	r_only_maps_sel_rb[2]	r_only_maps_sel_rb[1]	r_only_maps_sel_rb[0]
0xE7	0x00	user_sub_map_2_r_reg_e7	r	rb_y_average_data[9]	rb_y_average_data[8]	rb_y_average_data[7]	rb_y_average_data[6]	rb_y_average_data[5]	rb_y_average_data[4]	rb_y_average_data[3]	rb_y_average_data[2]
0xE8	0x00	user_sub_map_2_r_reg_e8	r	-	-	-	-	-	-	rb_y_average_data[1]	rb_y_average_data[0]

1.12 DPLL MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xB5	0x01	mclk fs	rw	-	-	-	-	-	mclk_fs_n[2]	mclk_fs_n[1]	mclk_fs_n[0]

1.13 CP MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x0B	0x00	csc_coeff_rb_1	r	rb_csc_scale[1]	rb_csc_scale[0]	-	rb_a4[12]	rb_a4[11]	rb_a4[10]	rb_a4[9]	rb_a4[8]
0x0C	0x00	csc_coeff_rb_2	r	rb_a4[7]	rb_a4[6]	rb_a4[5]	rb_a4[4]	rb_a4[3]	rb_a4[2]	rb_a4[1]	rb_a4[0]
0x0D	0x00	csc_coeff_rb_3	r	-	rb_a3[12]	rb_a3[11]	rb_a3[10]	rb_a3[9]	rb_a3[8]	rb_a3[7]	rb_a3[6]
0x0E	0x00	csc_coeff_rb_4	r	rb_a3[5]	rb_a3[4]	rb_a3[3]	rb_a3[2]	rb_a3[1]	rb_a3[0]	rb_a2[12]	rb_a2[11]
0x0F	0x00	csc_coeff_rb_5	r	rb_a2[10]	rb_a2[9]	rb_a2[8]	rb_a2[7]	rb_a2[6]	rb_a2[5]	rb_a2[4]	rb_a2[3]
0x10	0x00	csc_coeff_rb_6	r	rb_a2[2]	rb_a2[1]	rb_a2[0]	rb_a1[12]	rb_a1[11]	rb_a1[10]	rb_a1[9]	rb_a1[8]
0x11	0x00	csc_coeff_rb_7	r	rb_a1[7]	rb_a1[6]	rb_a1[5]	rb_a1[4]	rb_a1[3]	rb_a1[2]	rb_a1[1]	rb_a1[0]
0x12	0x00	csc_coeff_rb_8	r	-	-	-	rb_b4[12]	rb_b4[11]	rb_b4[10]	rb_b4[9]	rb_b4[8]
0x13	0x00	csc_coeff_rb_9	r	rb_b4[7]	rb_b4[6]	rb_b4[5]	rb_b4[4]	rb_b4[3]	rb_b4[2]	rb_b4[1]	rb_b4[0]
0x14	0x00	csc_coeff_rb_10	r	-	rb_b3[12]	rb_b3[11]	rb_b3[10]	rb_b3[9]	rb_b3[8]	rb_b3[7]	rb_b3[6]
0x15	0x00	csc_coeff_rb_11	r	rb_b3[5]	rb_b3[4]	rb_b3[3]	rb_b3[2]	rb_b3[1]	rb_b3[0]	rb_b2[12]	rb_b2[11]
0x16	0x00	csc_coeff_rb_12	r	rb_b2[10]	rb_b2[9]	rb_b2[8]	rb_b2[7]	rb_b2[6]	rb_b2[5]	rb_b2[4]	rb_b2[3]
0x17	0x00	csc_coeff_rb_13	r	rb_b2[2]	rb_b2[1]	rb_b2[0]	rb_b1[12]	rb_b1[11]	rb_b1[10]	rb_b1[9]	rb_b1[8]
0x18	0x00	csc_coeff_rb_14	r	rb_b1[7]	rb_b1[6]	rb_b1[5]	rb_b1[4]	rb_b1[3]	rb_b1[2]	rb_b1[1]	rb_b1[0]
0x19	0x00	csc_coeff_rb_15	r	-	-	-	rb_c4[12]	rb_c4[11]	rb_c4[10]	rb_c4[9]	rb_c4[8]
0x1A	0x00	csc_coeff_rb_16	r	rb_c4[7]	rb_c4[6]	rb_c4[5]	rb_c4[4]	rb_c4[3]	rb_c4[2]	rb_c4[1]	rb_c4[0]
0x1B	0x00	csc_coeff_rb_17	r	-	rb_c3[12]	rb_c3[11]	rb_c3[10]	rb_c3[9]	rb_c3[8]	rb_c3[7]	rb_c3[6]
0x1C	0x00	csc_coeff_rb_18	r	rb_c3[5]	rb_c3[4]	rb_c3[3]	rb_c3[2]	rb_c3[1]	rb_c3[0]	rb_c2[12]	rb_c2[11]
0x1D	0x00	csc_coeff_rb_19	r	rb_c2[10]	rb_c2[9]	rb_c2[8]	rb_c2[7]	rb_c2[6]	rb_c2[5]	rb_c2[4]	rb_c2[3]
0x1E	0x00	csc_coeff_rb_20	r	rb_c2[2]	rb_c2[1]	rb_c2[0]	rb_c1[12]	rb_c1[11]	rb_c1[10]	rb_c1[9]	rb_c1[8]
0x1F	0x00	csc_coeff_rb_21	r	rb_c1[7]	rb_c1[6]	rb_c1[5]	rb_c1[4]	rb_c1[3]	rb_c1[2]	rb_c1[1]	rb_c1[0]
0x30	0x00	de_pos_adj_1	rw	de_v_start_r[3]	de_v_start_r[2]	de_v_start_r[1]	de_v_start_r[0]	de_v_end_r[3]	de_v_end_r[2]	de_v_end_r[1]	de_v_end_r[0]
0x31	0x00	de_pos_adj_2	rw	de_v_start_even_r[3]	de_v_start_even_r[2]	de_v_start_even_r[1]	de_v_start_even_r[0]	de_v_end_even_r[3]	de_v_end_even_r[2]	de_v_end_even_r[1]	de_v_end_even_r[0]
0x37	0x00	int_pat_gen_1	rw	cp_int_pat_gen_en	cp_border_pix_en	cp_border_width	-	-	cp_pat_gen_sel[2]	cp_pat_gen_sel[1]	cp_pat_gen_sel[0]
0x3A	0x80	contrast_cntrl	rw	cp_contrast[7]	cp_contrast[6]	cp_contrast[5]	cp_contrast[4]	cp_contrast[3]	cp_contrast[2]	cp_contrast[1]	cp_contrast[0]
0x3B	0x80	saturation_cntrl	rw	cp_saturation[7]	cp_saturation[6]	cp_saturation[5]	cp_saturation[4]	cp_saturation[3]	cp_saturation[2]	cp_saturation[1]	cp_saturation[0]
0x3C	0x00	brightness_cntrl	rw	cp_brightness[7]	cp_brightness[6]	cp_brightness[5]	cp_brightness[4]	cp_brightness[3]	cp_brightness[2]	cp_brightness[1]	cp_brightness[0]
0x3D	0x00	hue_cntrl	rw	cp_hue[7]	cp_hue[6]	cp_hue[5]	cp_hue[4]	cp_hue[3]	cp_hue[2]	cp_hue[1]	cp_hue[0]
0x3E	0x00	vid_adj_0	rw	vid_adj_en	-	cp_uv_align_sel[1]	cp_uv_align_sel[0]	cp_uv_dval_inv	-	alt_sat_uv_man	alt_sat_uv
0x52	0x40	csc_coeffs_1	rw	csc_scale[1]	csc_scale[0]	-	a4[12]	a4[11]	a4[10]	a4[9]	a4[8]
0x53	0x00	csc_coeffs_2	rw	a4[7]	a4[6]	a4[5]	a4[4]	a4[3]	a4[2]	a4[1]	a4[0]
0x54	0x00	csc_coeffs_3	rw	-	a3[12]	a3[11]	a3[10]	a3[9]	a3[8]	a3[7]	a3[6]
0x55	0x00	csc_coeffs_4	rw	a3[5]	a3[4]	a3[3]	a3[2]	a3[1]	a3[0]	a2[12]	a2[11]
0x56	0x00	csc_coeffs_5	rw	a2[10]	a2[9]	a2[8]	a2[7]	a2[6]	a2[5]	a2[4]	a2[3]
0x57	0x08	csc_coeffs_6	rw	a2[2]	a2[1]	a2[0]	a1[12]	a1[11]	a1[10]	a1[9]	a1[8]
0x58	0x00	csc_coeffs_7	rw	a1[7]	a1[6]	a1[5]	a1[4]	a1[3]	a1[2]	a1[1]	a1[0]
0x59	0x00	csc_coeffs_8	rw	-	-	-	b4[12]	b4[11]	b4[10]	b4[9]	b4[8]
0x5A	0x00	csc_coeffs_9	rw	b4[7]	b4[6]	b4[5]	b4[4]	b4[3]	b4[2]	b4[1]	b4[0]
0x5B	0x00	csc_coeffs_10	rw	-	b3[12]	b3[11]	b3[10]	b3[9]	b3[8]	b3[7]	b3[6]
0x5C	0x01	csc_coeffs_11	rw	b3[5]	b3[4]	b3[3]	b3[2]	b3[1]	b3[0]	b2[12]	b2[11]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x5D	0x00	csc_coeffs_12	rw	b2[10]	b2[9]	b2[8]	b2[7]	b2[6]	b2[5]	b2[4]	b2[3]
0x5E	0x00	csc_coeffs_13	rw	b2[2]	b2[1]	b2[0]	b1[12]	b1[11]	b1[10]	b1[9]	b1[8]
0x5F	0x00	csc_coeffs_14	rw	b1[7]	b1[6]	b1[5]	b1[4]	b1[3]	b1[2]	b1[1]	b1[0]
0x60	0x00	csc_coeffs_15	rw	-	-	-	c4[12]	c4[11]	c4[10]	c4[9]	c4[8]
0x61	0x00	csc_coeffs_16	rw	c4[7]	c4[6]	c4[5]	c4[4]	c4[3]	c4[2]	c4[1]	c4[0]
0x62	0x20	csc_coeffs_17	rw	-	c3[12]	c3[11]	c3[10]	c3[9]	c3[8]	c3[7]	c3[6]
0x63	0x00	csc_coeffs_18	rw	c3[5]	c3[4]	c3[3]	c3[2]	c3[1]	c3[0]	c2[12]	c2[11]
0x64	0x00	csc_coeffs_19	rw	c2[10]	c2[9]	c2[8]	c2[7]	c2[6]	c2[5]	c2[4]	c2[3]
0x65	0x00	csc_coeffs_20	rw	c2[2]	c2[1]	c2[0]	c1[12]	c1[11]	c1[10]	c1[9]	c1[8]
0x66	0x00	csc_coeffs_21	rw	c1[7]	c1[6]	c1[5]	c1[4]	c1[3]	c1[2]	c1[1]	c1[0]
0x68	0xF0	csc_decim_cntrl	rw	csc_coeff_sel[3]	csc_coeff_sel[2]	csc_coeff_sel[1]	csc_coeff_sel[0]	cp_chroma_low_en	-	-	-
0x69	0x04	vid_adj_1	rw	-	man_cp_decim_en	-	man_cp_csc_en	-	eia_861_compliance	-	-
0x77	0xFF	offset_cntrl_1	rw	cp_prec[1]	cp_prec[0]	-	-	-	-	-	-
0x7B	0x05	avcode_cntrl	rw	av_inv_f	av_inv_v	-	-	-	av_pos_sel	-	-
0x7C	0xC0	sync_cntrl_1	rw	cp_inv_hs	cp_inv_vs	-	-	start_hs[9]	start_hs[8]	end_hs[9]	end_hs[8]
0x7D	0x00	sync_cntrl_2	rw	end_hs[7]	end_hs[6]	end_hs[5]	end_hs[4]	end_hs[3]	end_hs[2]	end_hs[1]	end_hs[0]
0x7E	0x00	sync_cntrl_3	rw	start_hs[7]	start_hs[6]	start_hs[5]	start_hs[4]	start_hs[3]	start_hs[2]	start_hs[1]	start_hs[0]
0x7F	0x00	sync_cntrl_4	rw	start_vs[3]	start_vs[2]	start_vs[1]	start_vs[0]	end_vs[3]	end_vs[2]	end_vs[1]	end_vs[0]
0x87	0x00	de_pos_adj_4	rw	-	-	de_v_start_even[5]	de_v_start_even[4]	de_v_start_even[3]	de_v_start_even[2]	de_v_start_even[1]	de_v_start_even[0]
0x88	0x00	de_pos_adj_5	rw	-	-	de_v_end_even[5]	de_v_end_even[4]	de_v_end_even[3]	de_v_end_even[2]	de_v_end_even[1]	de_v_end_even[0]
0x89	0x00	sync_cntrl_6	rw	start_vs_even[3]	start_vs_even[2]	start_vs_even[1]	start_vs_even[0]	end_vs_even[3]	end_vs_even[2]	end_vs_even[1]	end_vs_even[0]
0x8B	0x40	de_pos_adj_6	rw	-	-	-	-	de_h_start[9]	de_h_start[8]	de_h_end[9]	de_h_end[8]
0x8C	0x00	de_pos_adj_7	rw	de_h_end[7]	de_h_end[6]	de_h_end[5]	de_h_end[4]	de_h_end[3]	de_h_end[2]	de_h_end[1]	de_h_end[0]
0x8D	0x00	de_pos_adj_8	rw	de_h_start[7]	de_h_start[6]	de_h_start[5]	de_h_start[4]	de_h_start[3]	de_h_start[2]	de_h_start[1]	de_h_start[0]
0x91	0x40	vid_adj_2	rw	-	interlaced	-	interlaced_3d	-	-	-	-
0x98	0x00	de_pos_adj_9	rw	-	-	de_v_start[5]	de_v_start[4]	de_v_start[3]	de_v_start[2]	de_v_start[1]	de_v_start[0]
0x99	0x00	de_pos_adj_10	rw	-	-	de_v_end[5]	de_v_end[4]	de_v_end[3]	de_v_end[2]	de_v_end[1]	de_v_end[0]
0xBA	0x01	hdmi_cp_cntrl_1	rw	-	-	-	-	-	-	hdmi_frun_mode	hdmi_frun_en
0xBE	0x04	dly_adj	rw	-	-	-	-	-	-	hcount_align_adj[4]	hcount_align_adj[3]
0xBF	0x12	fr_color_sel_1	rw	hcount_align_adj[2]	hcount_align_adj[1]	hcount_align_adj[0]	-	-	cp_def_col_man_val	cp_def_col_auto	cp_force_freerun
0xC0	0x00	fr_color_sel_2	rw	def_col_cha[7]	def_col_cha[6]	def_col_cha[5]	def_col_cha[4]	def_col_cha[3]	def_col_cha[2]	def_col_cha[1]	def_col_cha[0]
0xC1	0x00	fr_color_sel_3	rw	def_col_chb[7]	def_col_chb[6]	def_col_chb[5]	def_col_chb[4]	def_col_chb[3]	def_col_chb[2]	def_col_chb[1]	def_col_chb[0]
0xC2	0x00	fr_color_sel_4	rw	def_col_chc[7]	def_col_chc[6]	def_col_chc[5]	def_col_chc[4]	def_col_chc[3]	def_col_chc[2]	def_col_chc[1]	def_col_chc[0]
0xC9	0x2C	clmp_pos_cntrl_4	rw	-	-	-	-	-	swap_split_av	-	dis_auto_param_buff
0xE0	0x00	status_0_1	r	-	hdmi_cp_autoparm_locked	-	-	-	-	-	-
0xF3	0xD4	sync_det_cntrl_ch1_4	rw	-	-	ch1_fl_fr_threshold[2]	ch1_fl_fr_threshold[1]	ch1_fl_fr_threshold[0]	ch1_f_run_thr[2]	ch1_f_run_thr[1]	ch1_f_run_thr[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF4	0x00	csc_coeff_sel_rb	r	csc_coeff_sel_rb[3]	csc_coeff_sel_rb[2]	csc_coeff_sel_rb[1]	csc_coeff_sel_rb[0]	-	-	-	-
0xFF	0x00	cp_reg_ff	r	-	-	-	cp_free_run	-	-	-	-

1.14 CEC MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x00	cec_reg_00	rw	cec_tx_frame_header[7]	cec_tx_frame_header[6]	cec_tx_frame_header[5]	cec_tx_frame_header[4]	cec_tx_frame_header[3]	cec_tx_frame_header[2]	cec_tx_frame_header[1]	cec_tx_frame_header[0]
0x01	0x00	cec_reg_01	rw	cec_tx_frame_data0[7]	cec_tx_frame_data0[6]	cec_tx_frame_data0[5]	cec_tx_frame_data0[4]	cec_tx_frame_data0[3]	cec_tx_frame_data0[2]	cec_tx_frame_data0[1]	cec_tx_frame_data0[0]
0x02	0x00	cec_reg_02	rw	cec_tx_frame_data1[7]	cec_tx_frame_data1[6]	cec_tx_frame_data1[5]	cec_tx_frame_data1[4]	cec_tx_frame_data1[3]	cec_tx_frame_data1[2]	cec_tx_frame_data1[1]	cec_tx_frame_data1[0]
0x03	0x00	cec_reg_03	rw	cec_tx_frame_data2[7]	cec_tx_frame_data2[6]	cec_tx_frame_data2[5]	cec_tx_frame_data2[4]	cec_tx_frame_data2[3]	cec_tx_frame_data2[2]	cec_tx_frame_data2[1]	cec_tx_frame_data2[0]
0x04	0x00	cec_reg_04	rw	cec_tx_frame_data3[7]	cec_tx_frame_data3[6]	cec_tx_frame_data3[5]	cec_tx_frame_data3[4]	cec_tx_frame_data3[3]	cec_tx_frame_data3[2]	cec_tx_frame_data3[1]	cec_tx_frame_data3[0]
0x05	0x00	cec_reg_05	rw	cec_tx_frame_data4[7]	cec_tx_frame_data4[6]	cec_tx_frame_data4[5]	cec_tx_frame_data4[4]	cec_tx_frame_data4[3]	cec_tx_frame_data4[2]	cec_tx_frame_data4[1]	cec_tx_frame_data4[0]
0x06	0x00	cec_reg_06	rw	cec_tx_frame_data5[7]	cec_tx_frame_data5[6]	cec_tx_frame_data5[5]	cec_tx_frame_data5[4]	cec_tx_frame_data5[3]	cec_tx_frame_data5[2]	cec_tx_frame_data5[1]	cec_tx_frame_data5[0]
0x07	0x00	cec_reg_07	rw	cec_tx_frame_data6[7]	cec_tx_frame_data6[6]	cec_tx_frame_data6[5]	cec_tx_frame_data6[4]	cec_tx_frame_data6[3]	cec_tx_frame_data6[2]	cec_tx_frame_data6[1]	cec_tx_frame_data6[0]
0x08	0x00	cec_reg_08	rw	cec_tx_frame_data7[7]	cec_tx_frame_data7[6]	cec_tx_frame_data7[5]	cec_tx_frame_data7[4]	cec_tx_frame_data7[3]	cec_tx_frame_data7[2]	cec_tx_frame_data7[1]	cec_tx_frame_data7[0]
0x09	0x00	cec_reg_09	rw	cec_tx_frame_data8[7]	cec_tx_frame_data8[6]	cec_tx_frame_data8[5]	cec_tx_frame_data8[4]	cec_tx_frame_data8[3]	cec_tx_frame_data8[2]	cec_tx_frame_data8[1]	cec_tx_frame_data8[0]
0x0A	0x00	cec_reg_0a	rw	cec_tx_frame_data9[7]	cec_tx_frame_data9[6]	cec_tx_frame_data9[5]	cec_tx_frame_data9[4]	cec_tx_frame_data9[3]	cec_tx_frame_data9[2]	cec_tx_frame_data9[1]	cec_tx_frame_data9[0]
0x0B	0x00	cec_reg_0b	rw	cec_tx_frame_data10[7]	cec_tx_frame_data10[6]	cec_tx_frame_data10[5]	cec_tx_frame_data10[4]	cec_tx_frame_data10[3]	cec_tx_frame_data10[2]	cec_tx_frame_data10[1]	cec_tx_frame_data10[0]
0x0C	0x00	cec_reg_0c	rw	cec_tx_frame_data11[7]	cec_tx_frame_data11[6]	cec_tx_frame_data11[5]	cec_tx_frame_data11[4]	cec_tx_frame_data11[3]	cec_tx_frame_data11[2]	cec_tx_frame_data11[1]	cec_tx_frame_data11[0]
0x0D	0x00	cec_reg_0d	rw	cec_tx_frame_data12[7]	cec_tx_frame_data12[6]	cec_tx_frame_data12[5]	cec_tx_frame_data12[4]	cec_tx_frame_data12[3]	cec_tx_frame_data12[2]	cec_tx_frame_data12[1]	cec_tx_frame_data12[0]
0x0E	0x00	cec_reg_0e	rw	cec_tx_frame_data13[7]	cec_tx_frame_data13[6]	cec_tx_frame_data13[5]	cec_tx_frame_data13[4]	cec_tx_frame_data13[3]	cec_tx_frame_data13[2]	cec_tx_frame_data13[1]	cec_tx_frame_data13[0]
0x0F	0x00	cec_reg_0f	rw	cec_tx_frame_data14[7]	cec_tx_frame_data14[6]	cec_tx_frame_data14[5]	cec_tx_frame_data14[4]	cec_tx_frame_data14[3]	cec_tx_frame_data14[2]	cec_tx_frame_data14[1]	cec_tx_frame_data14[0]
0x10	0x00	cec_reg_10	rw	-	-	-	cec_tx_frame_length[4]	cec_tx_frame_length[3]	cec_tx_frame_length[2]	cec_tx_frame_length[1]	cec_tx_frame_length[0]
0x11	0x00	cec_reg_11	rw	-	-	-	-	-	-	-	cec_tx_enable
0x12	0x13	cec_reg_12	rw	-	cec_tx_retry[2]	cec_tx_retry[1]	cec_tx_retry[0]	cec_retry_sft[3]	cec_retry_sft[2]	cec_retry_sft[1]	cec_retry_sft[0]
0x13	0x57	cec_reg_13	rw	cec_tx_sft[3]	cec_tx_sft[2]	cec_tx_sft[1]	cec_tx_sft[0]	cec_tx_sft[3]	cec_tx_sft[2]	cec_tx_sft[1]	cec_tx_sft[0]
0x14	0x00	cec_reg_14	r	cec_tx_lowdrive_counter[3]	cec_tx_lowdrive_counter[2]	cec_tx_lowdrive_counter[1]	cec_tx_lowdrive_counter[0]	cec_tx_nack_counter[3]	cec_tx_nack_counter[2]	cec_tx_nack_counter[1]	cec_tx_nack_counter[0]
0x15	0x00	cec_reg_15	r	cec_buf0_rx_frame_header[7]	cec_buf0_rx_frame_header[6]	cec_buf0_rx_frame_header[5]	cec_buf0_rx_frame_header[4]	cec_buf0_rx_frame_header[3]	cec_buf0_rx_frame_header[2]	cec_buf0_rx_frame_header[1]	cec_buf0_rx_frame_header[0]
0x16	0x00	cec_reg_16	r	cec_buf0_rx_frame_data0[7]	cec_buf0_rx_frame_data0[6]	cec_buf0_rx_frame_data0[5]	cec_buf0_rx_frame_data0[4]	cec_buf0_rx_frame_data0[3]	cec_buf0_rx_frame_data0[2]	cec_buf0_rx_frame_data0[1]	cec_buf0_rx_frame_data0[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x17	0x00	cec_reg_17	r	cec_buf0_rx_frame_data1[7]	cec_buf0_rx_frame_data1[6]	cec_buf0_rx_frame_data1[5]	cec_buf0_rx_frame_data1[4]	cec_buf0_rx_frame_data1[3]	cec_buf0_rx_frame_data1[2]	cec_buf0_rx_frame_data1[1]	cec_buf0_rx_frame_data1[0]
0x18	0x00	cec_reg_18	r	cec_buf0_rx_frame_data2[7]	cec_buf0_rx_frame_data2[6]	cec_buf0_rx_frame_data2[5]	cec_buf0_rx_frame_data2[4]	cec_buf0_rx_frame_data2[3]	cec_buf0_rx_frame_data2[2]	cec_buf0_rx_frame_data2[1]	cec_buf0_rx_frame_data2[0]
0x19	0x00	cec_reg_19	r	cec_buf0_rx_frame_data3[7]	cec_buf0_rx_frame_data3[6]	cec_buf0_rx_frame_data3[5]	cec_buf0_rx_frame_data3[4]	cec_buf0_rx_frame_data3[3]	cec_buf0_rx_frame_data3[2]	cec_buf0_rx_frame_data3[1]	cec_buf0_rx_frame_data3[0]
0x1A	0x00	cec_reg_1a	r	cec_buf0_rx_frame_data4[7]	cec_buf0_rx_frame_data4[6]	cec_buf0_rx_frame_data4[5]	cec_buf0_rx_frame_data4[4]	cec_buf0_rx_frame_data4[3]	cec_buf0_rx_frame_data4[2]	cec_buf0_rx_frame_data4[1]	cec_buf0_rx_frame_data4[0]
0x1B	0x00	cec_reg_1b	r	cec_buf0_rx_frame_data5[7]	cec_buf0_rx_frame_data5[6]	cec_buf0_rx_frame_data5[5]	cec_buf0_rx_frame_data5[4]	cec_buf0_rx_frame_data5[3]	cec_buf0_rx_frame_data5[2]	cec_buf0_rx_frame_data5[1]	cec_buf0_rx_frame_data5[0]
0x1C	0x00	cec_reg_1c	r	cec_buf0_rx_frame_data6[7]	cec_buf0_rx_frame_data6[6]	cec_buf0_rx_frame_data6[5]	cec_buf0_rx_frame_data6[4]	cec_buf0_rx_frame_data6[3]	cec_buf0_rx_frame_data6[2]	cec_buf0_rx_frame_data6[1]	cec_buf0_rx_frame_data6[0]
0x1D	0x00	cec_reg_1d	r	cec_buf0_rx_frame_data7[7]	cec_buf0_rx_frame_data7[6]	cec_buf0_rx_frame_data7[5]	cec_buf0_rx_frame_data7[4]	cec_buf0_rx_frame_data7[3]	cec_buf0_rx_frame_data7[2]	cec_buf0_rx_frame_data7[1]	cec_buf0_rx_frame_data7[0]
0x1E	0x00	cec_reg_1e	r	cec_buf0_rx_frame_data8[7]	cec_buf0_rx_frame_data8[6]	cec_buf0_rx_frame_data8[5]	cec_buf0_rx_frame_data8[4]	cec_buf0_rx_frame_data8[3]	cec_buf0_rx_frame_data8[2]	cec_buf0_rx_frame_data8[1]	cec_buf0_rx_frame_data8[0]
0x1F	0x00	cec_reg_1f	r	cec_buf0_rx_frame_data9[7]	cec_buf0_rx_frame_data9[6]	cec_buf0_rx_frame_data9[5]	cec_buf0_rx_frame_data9[4]	cec_buf0_rx_frame_data9[3]	cec_buf0_rx_frame_data9[2]	cec_buf0_rx_frame_data9[1]	cec_buf0_rx_frame_data9[0]
0x20	0x00	cec_reg_20	r	cec_buf0_rx_frame_data10[7]	cec_buf0_rx_frame_data10[6]	cec_buf0_rx_frame_data10[5]	cec_buf0_rx_frame_data10[4]	cec_buf0_rx_frame_data10[3]	cec_buf0_rx_frame_data10[2]	cec_buf0_rx_frame_data10[1]	cec_buf0_rx_frame_data10[0]
0x21	0x00	cec_reg_21	r	cec_buf0_rx_frame_data11[7]	cec_buf0_rx_frame_data11[6]	cec_buf0_rx_frame_data11[5]	cec_buf0_rx_frame_data11[4]	cec_buf0_rx_frame_data11[3]	cec_buf0_rx_frame_data11[2]	cec_buf0_rx_frame_data11[1]	cec_buf0_rx_frame_data11[0]
0x22	0x00	cec_reg_22	r	cec_buf0_rx_frame_data12[7]	cec_buf0_rx_frame_data12[6]	cec_buf0_rx_frame_data12[5]	cec_buf0_rx_frame_data12[4]	cec_buf0_rx_frame_data12[3]	cec_buf0_rx_frame_data12[2]	cec_buf0_rx_frame_data12[1]	cec_buf0_rx_frame_data12[0]
0x23	0x00	cec_reg_23	r	cec_buf0_rx_frame_data13[7]	cec_buf0_rx_frame_data13[6]	cec_buf0_rx_frame_data13[5]	cec_buf0_rx_frame_data13[4]	cec_buf0_rx_frame_data13[3]	cec_buf0_rx_frame_data13[2]	cec_buf0_rx_frame_data13[1]	cec_buf0_rx_frame_data13[0]
0x24	0x00	cec_reg_24	r	cec_buf0_rx_frame_data14[7]	cec_buf0_rx_frame_data14[6]	cec_buf0_rx_frame_data14[5]	cec_buf0_rx_frame_data14[4]	cec_buf0_rx_frame_data14[3]	cec_buf0_rx_frame_data14[2]	cec_buf0_rx_frame_data14[1]	cec_buf0_rx_frame_data14[0]
0x25	0x00	cec_reg_25	r	-	-	-	cec_buf0_rx_frame_length[4]	cec_buf0_rx_frame_length[3]	cec_buf0_rx_frame_length[2]	cec_buf0_rx_frame_length[1]	cec_buf0_rx_frame_length[0]
0x27	0x10	cec_reg_27	rw	-	cec_logical_address_mask[2]	cec_logical_address_mask[1]	cec_logical_address_mask[0]	cec_error_report_mode	cec_error_det_mode	cec_force_nack	cec_force_ignore
0x28	0xFF	cec_reg_28	rw	cec_logical_address1[3]	cec_logical_address1[2]	cec_logical_address1[1]	cec_logical_address1[0]	cec_logical_address0[3]	cec_logical_address0[2]	cec_logical_address0[1]	cec_logical_address0[0]
0x29	0x0F	cec_reg_29	rw	-	-	-	-	cec_logical_address2[3]	cec_logical_address2[2]	cec_logical_address2[1]	cec_logical_address2[0]
0x2A	0x3E	cec_reg_2a	rw	-	-	-	-	-	-	-	cec_power_up
0x2B	0x07	cec_reg_2b	rw	-	-	cec_glitch_filter_ctrl[5]	cec_glitch_filter_ctrl[4]	cec_glitch_filter_ctrl[3]	cec_glitch_filter_ctrl[2]	cec_glitch_filter_ctrl[1]	cec_glitch_filter_ctrl[0]
0x2C	0x00	cec_reg_2c	sc	-	-	-	-	cec_clr_rx_rdy2	cec_clr_rx_rdy1	cec_clr_rx_rdy0	cec_soft_reset
0x53	0x00	cec_reg_53	r	-	-	cec_buf2_timestamp[1]	cec_buf2_timestamp[0]	cec_buf1_timestamp[1]	cec_buf1_timestamp[0]	cec_buf0_timestamp[1]	cec_buf0_timestamp[0]
0x54	0x00	cec_reg_54	r	cec_buf1_rx_frame_header[7]	cec_buf1_rx_frame_header[6]	cec_buf1_rx_frame_header[5]	cec_buf1_rx_frame_header[4]	cec_buf1_rx_frame_header[3]	cec_buf1_rx_frame_header[2]	cec_buf1_rx_frame_header[1]	cec_buf1_rx_frame_header[0]
0x55	0x00	cec_reg_55	r	cec_buf1_rx_frame_data0[7]	cec_buf1_rx_frame_data0[6]	cec_buf1_rx_frame_data0[5]	cec_buf1_rx_frame_data0[4]	cec_buf1_rx_frame_data0[3]	cec_buf1_rx_frame_data0[2]	cec_buf1_rx_frame_data0[1]	cec_buf1_rx_frame_data0[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x56	0x00	cec_reg_56	r	cec_buf1_rx_frame_data1[7]	cec_buf1_rx_frame_data1[6]	cec_buf1_rx_frame_data1[5]	cec_buf1_rx_frame_data1[4]	cec_buf1_rx_frame_data1[3]	cec_buf1_rx_frame_data1[2]	cec_buf1_rx_frame_data1[1]	cec_buf1_rx_frame_data1[0]
0x57	0x00	cec_reg_57	r	cec_buf1_rx_frame_data2[7]	cec_buf1_rx_frame_data2[6]	cec_buf1_rx_frame_data2[5]	cec_buf1_rx_frame_data2[4]	cec_buf1_rx_frame_data2[3]	cec_buf1_rx_frame_data2[2]	cec_buf1_rx_frame_data2[1]	cec_buf1_rx_frame_data2[0]
0x58	0x00	cec_reg_58	r	cec_buf1_rx_frame_data3[7]	cec_buf1_rx_frame_data3[6]	cec_buf1_rx_frame_data3[5]	cec_buf1_rx_frame_data3[4]	cec_buf1_rx_frame_data3[3]	cec_buf1_rx_frame_data3[2]	cec_buf1_rx_frame_data3[1]	cec_buf1_rx_frame_data3[0]
0x59	0x00	cec_reg_59	r	cec_buf1_rx_frame_data4[7]	cec_buf1_rx_frame_data4[6]	cec_buf1_rx_frame_data4[5]	cec_buf1_rx_frame_data4[4]	cec_buf1_rx_frame_data4[3]	cec_buf1_rx_frame_data4[2]	cec_buf1_rx_frame_data4[1]	cec_buf1_rx_frame_data4[0]
0x5A	0x00	cec_reg_5a	r	cec_buf1_rx_frame_data5[7]	cec_buf1_rx_frame_data5[6]	cec_buf1_rx_frame_data5[5]	cec_buf1_rx_frame_data5[4]	cec_buf1_rx_frame_data5[3]	cec_buf1_rx_frame_data5[2]	cec_buf1_rx_frame_data5[1]	cec_buf1_rx_frame_data5[0]
0x5B	0x00	cec_reg_5b	r	cec_buf1_rx_frame_data6[7]	cec_buf1_rx_frame_data6[6]	cec_buf1_rx_frame_data6[5]	cec_buf1_rx_frame_data6[4]	cec_buf1_rx_frame_data6[3]	cec_buf1_rx_frame_data6[2]	cec_buf1_rx_frame_data6[1]	cec_buf1_rx_frame_data6[0]
0x5C	0x00	cec_reg_5c	r	cec_buf1_rx_frame_data7[7]	cec_buf1_rx_frame_data7[6]	cec_buf1_rx_frame_data7[5]	cec_buf1_rx_frame_data7[4]	cec_buf1_rx_frame_data7[3]	cec_buf1_rx_frame_data7[2]	cec_buf1_rx_frame_data7[1]	cec_buf1_rx_frame_data7[0]
0x5D	0x00	cec_reg_5d	r	cec_buf1_rx_frame_data8[7]	cec_buf1_rx_frame_data8[6]	cec_buf1_rx_frame_data8[5]	cec_buf1_rx_frame_data8[4]	cec_buf1_rx_frame_data8[3]	cec_buf1_rx_frame_data8[2]	cec_buf1_rx_frame_data8[1]	cec_buf1_rx_frame_data8[0]
0x5E	0x00	cec_reg_5e	r	cec_buf1_rx_frame_data9[7]	cec_buf1_rx_frame_data9[6]	cec_buf1_rx_frame_data9[5]	cec_buf1_rx_frame_data9[4]	cec_buf1_rx_frame_data9[3]	cec_buf1_rx_frame_data9[2]	cec_buf1_rx_frame_data9[1]	cec_buf1_rx_frame_data9[0]
0x5F	0x00	cec_reg_5f	r	cec_buf1_rx_frame_data10[7]	cec_buf1_rx_frame_data10[6]	cec_buf1_rx_frame_data10[5]	cec_buf1_rx_frame_data10[4]	cec_buf1_rx_frame_data10[3]	cec_buf1_rx_frame_data10[2]	cec_buf1_rx_frame_data10[1]	cec_buf1_rx_frame_data10[0]
0x60	0x00	cec_reg_60	r	cec_buf1_rx_frame_data11[7]	cec_buf1_rx_frame_data11[6]	cec_buf1_rx_frame_data11[5]	cec_buf1_rx_frame_data11[4]	cec_buf1_rx_frame_data11[3]	cec_buf1_rx_frame_data11[2]	cec_buf1_rx_frame_data11[1]	cec_buf1_rx_frame_data11[0]
0x61	0x00	cec_reg_61	r	cec_buf1_rx_frame_data12[7]	cec_buf1_rx_frame_data12[6]	cec_buf1_rx_frame_data12[5]	cec_buf1_rx_frame_data12[4]	cec_buf1_rx_frame_data12[3]	cec_buf1_rx_frame_data12[2]	cec_buf1_rx_frame_data12[1]	cec_buf1_rx_frame_data12[0]
0x62	0x00	cec_reg_62	r	cec_buf1_rx_frame_data13[7]	cec_buf1_rx_frame_data13[6]	cec_buf1_rx_frame_data13[5]	cec_buf1_rx_frame_data13[4]	cec_buf1_rx_frame_data13[3]	cec_buf1_rx_frame_data13[2]	cec_buf1_rx_frame_data13[1]	cec_buf1_rx_frame_data13[0]
0x63	0x00	cec_reg_63	r	cec_buf1_rx_frame_data14[7]	cec_buf1_rx_frame_data14[6]	cec_buf1_rx_frame_data14[5]	cec_buf1_rx_frame_data14[4]	cec_buf1_rx_frame_data14[3]	cec_buf1_rx_frame_data14[2]	cec_buf1_rx_frame_data14[1]	cec_buf1_rx_frame_data14[0]
0x64	0x00	cec_reg_64	r	-	-	-	cec_buf1_rx_frame_length[4]	cec_buf1_rx_frame_length[3]	cec_buf1_rx_frame_length[2]	cec_buf1_rx_frame_length[1]	cec_buf1_rx_frame_length[0]
0x65	0x00	cec_reg_65	r	cec_buf2_rx_frame_header[7]	cec_buf2_rx_frame_header[6]	cec_buf2_rx_frame_header[5]	cec_buf2_rx_frame_header[4]	cec_buf2_rx_frame_header[3]	cec_buf2_rx_frame_header[2]	cec_buf2_rx_frame_header[1]	cec_buf2_rx_frame_header[0]
0x66	0x00	cec_reg_66	r	cec_buf2_rx_frame_data0[7]	cec_buf2_rx_frame_data0[6]	cec_buf2_rx_frame_data0[5]	cec_buf2_rx_frame_data0[4]	cec_buf2_rx_frame_data0[3]	cec_buf2_rx_frame_data0[2]	cec_buf2_rx_frame_data0[1]	cec_buf2_rx_frame_data0[0]
0x67	0x00	cec_reg_67	r	cec_buf2_rx_frame_data1[7]	cec_buf2_rx_frame_data1[6]	cec_buf2_rx_frame_data1[5]	cec_buf2_rx_frame_data1[4]	cec_buf2_rx_frame_data1[3]	cec_buf2_rx_frame_data1[2]	cec_buf2_rx_frame_data1[1]	cec_buf2_rx_frame_data1[0]
0x68	0x00	cec_reg_68	r	cec_buf2_rx_frame_data2[7]	cec_buf2_rx_frame_data2[6]	cec_buf2_rx_frame_data2[5]	cec_buf2_rx_frame_data2[4]	cec_buf2_rx_frame_data2[3]	cec_buf2_rx_frame_data2[2]	cec_buf2_rx_frame_data2[1]	cec_buf2_rx_frame_data2[0]
0x69	0x00	cec_reg_69	r	cec_buf2_rx_frame_data3[7]	cec_buf2_rx_frame_data3[6]	cec_buf2_rx_frame_data3[5]	cec_buf2_rx_frame_data3[4]	cec_buf2_rx_frame_data3[3]	cec_buf2_rx_frame_data3[2]	cec_buf2_rx_frame_data3[1]	cec_buf2_rx_frame_data3[0]
0x6A	0x00	cec_reg_6a	r	cec_buf2_rx_frame_data4[7]	cec_buf2_rx_frame_data4[6]	cec_buf2_rx_frame_data4[5]	cec_buf2_rx_frame_data4[4]	cec_buf2_rx_frame_data4[3]	cec_buf2_rx_frame_data4[2]	cec_buf2_rx_frame_data4[1]	cec_buf2_rx_frame_data4[0]
0x6B	0x00	cec_reg_6b	r	cec_buf2_rx_frame_data5[7]	cec_buf2_rx_frame_data5[6]	cec_buf2_rx_frame_data5[5]	cec_buf2_rx_frame_data5[4]	cec_buf2_rx_frame_data5[3]	cec_buf2_rx_frame_data5[2]	cec_buf2_rx_frame_data5[1]	cec_buf2_rx_frame_data5[0]
0x6C	0x00	cec_reg_6c	r	cec_buf2_rx_frame_data6[7]	cec_buf2_rx_frame_data6[6]	cec_buf2_rx_frame_data6[5]	cec_buf2_rx_frame_data6[4]	cec_buf2_rx_frame_data6[3]	cec_buf2_rx_frame_data6[2]	cec_buf2_rx_frame_data6[1]	cec_buf2_rx_frame_data6[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x6D	0x00	cec_reg_6d	r	cec_buf2_rx_frame_data7[7]	cec_buf2_rx_frame_data7[6]	cec_buf2_rx_frame_data7[5]	cec_buf2_rx_frame_data7[4]	cec_buf2_rx_frame_data7[3]	cec_buf2_rx_frame_data7[2]	cec_buf2_rx_frame_data7[1]	cec_buf2_rx_frame_data7[0]
0x6E	0x00	cec_reg_6e	r	cec_buf2_rx_frame_data8[7]	cec_buf2_rx_frame_data8[6]	cec_buf2_rx_frame_data8[5]	cec_buf2_rx_frame_data8[4]	cec_buf2_rx_frame_data8[3]	cec_buf2_rx_frame_data8[2]	cec_buf2_rx_frame_data8[1]	cec_buf2_rx_frame_data8[0]
0x6F	0x00	cec_reg_6f	r	cec_buf2_rx_frame_data9[7]	cec_buf2_rx_frame_data9[6]	cec_buf2_rx_frame_data9[5]	cec_buf2_rx_frame_data9[4]	cec_buf2_rx_frame_data9[3]	cec_buf2_rx_frame_data9[2]	cec_buf2_rx_frame_data9[1]	cec_buf2_rx_frame_data9[0]
0x70	0x00	cec_reg_70	r	cec_buf2_rx_frame_data10[7]	cec_buf2_rx_frame_data10[6]	cec_buf2_rx_frame_data10[5]	cec_buf2_rx_frame_data10[4]	cec_buf2_rx_frame_data10[3]	cec_buf2_rx_frame_data10[2]	cec_buf2_rx_frame_data10[1]	cec_buf2_rx_frame_data10[0]
0x71	0x00	cec_reg_71	r	cec_buf2_rx_frame_data11[7]	cec_buf2_rx_frame_data11[6]	cec_buf2_rx_frame_data11[5]	cec_buf2_rx_frame_data11[4]	cec_buf2_rx_frame_data11[3]	cec_buf2_rx_frame_data11[2]	cec_buf2_rx_frame_data11[1]	cec_buf2_rx_frame_data11[0]
0x72	0x00	cec_reg_72	r	cec_buf2_rx_frame_data12[7]	cec_buf2_rx_frame_data12[6]	cec_buf2_rx_frame_data12[5]	cec_buf2_rx_frame_data12[4]	cec_buf2_rx_frame_data12[3]	cec_buf2_rx_frame_data12[2]	cec_buf2_rx_frame_data12[1]	cec_buf2_rx_frame_data12[0]
0x73	0x00	cec_reg_73	r	cec_buf2_rx_frame_data13[7]	cec_buf2_rx_frame_data13[6]	cec_buf2_rx_frame_data13[5]	cec_buf2_rx_frame_data13[4]	cec_buf2_rx_frame_data13[3]	cec_buf2_rx_frame_data13[2]	cec_buf2_rx_frame_data13[1]	cec_buf2_rx_frame_data13[0]
0x74	0x00	cec_reg_74	r	cec_buf2_rx_frame_data14[7]	cec_buf2_rx_frame_data14[6]	cec_buf2_rx_frame_data14[5]	cec_buf2_rx_frame_data14[4]	cec_buf2_rx_frame_data14[3]	cec_buf2_rx_frame_data14[2]	cec_buf2_rx_frame_data14[1]	cec_buf2_rx_frame_data14[0]
0x75	0x00	cec_reg_75	r	-	-	-	cec_buf2_rx_frame_length[4]	cec_buf2_rx_frame_length[3]	cec_buf2_rx_frame_length[2]	cec_buf2_rx_frame_length[1]	cec_buf2_rx_frame_length[0]
0x76	0x00	cec_reg_76	r	-	-	-	-	-	cec_rx_rdy2	cec_rx_rdy1	cec_rx_rdy0
0x77	0x00	cec_reg_77	rw	-	-	-	-	-	-	-	cec_use_all_bufs
0x78	0x6D	cec_reg_78	rw	cec_wake_opcode0[7]	cec_wake_opcode0[6]	cec_wake_opcode0[5]	cec_wake_opcode0[4]	cec_wake_opcode0[3]	cec_wake_opcode0[2]	cec_wake_opcode0[1]	cec_wake_opcode0[0]
0x79	0x8F	cec_reg_79	rw	cec_wake_opcode1[7]	cec_wake_opcode1[6]	cec_wake_opcode1[5]	cec_wake_opcode1[4]	cec_wake_opcode1[3]	cec_wake_opcode1[2]	cec_wake_opcode1[1]	cec_wake_opcode1[0]
0x7A	0x82	cec_reg_7a	rw	cec_wake_opcode2[7]	cec_wake_opcode2[6]	cec_wake_opcode2[5]	cec_wake_opcode2[4]	cec_wake_opcode2[3]	cec_wake_opcode2[2]	cec_wake_opcode2[1]	cec_wake_opcode2[0]
0x7B	0x04	cec_reg_7b	rw	cec_wake_opcode3[7]	cec_wake_opcode3[6]	cec_wake_opcode3[5]	cec_wake_opcode3[4]	cec_wake_opcode3[3]	cec_wake_opcode3[2]	cec_wake_opcode3[1]	cec_wake_opcode3[0]
0x7C	0x0D	cec_reg_7c	rw	cec_wake_opcode4[7]	cec_wake_opcode4[6]	cec_wake_opcode4[5]	cec_wake_opcode4[4]	cec_wake_opcode4[3]	cec_wake_opcode4[2]	cec_wake_opcode4[1]	cec_wake_opcode4[0]
0x7D	0x70	cec_reg_7d	rw	cec_wake_opcode5[7]	cec_wake_opcode5[6]	cec_wake_opcode5[5]	cec_wake_opcode5[4]	cec_wake_opcode5[3]	cec_wake_opcode5[2]	cec_wake_opcode5[1]	cec_wake_opcode5[0]
0x7E	0x42	cec_reg_7e	rw	cec_wake_opcode6[7]	cec_wake_opcode6[6]	cec_wake_opcode6[5]	cec_wake_opcode6[4]	cec_wake_opcode6[3]	cec_wake_opcode6[2]	cec_wake_opcode6[1]	cec_wake_opcode6[0]
0x7F	0x41	cec_reg_7f	rw	cec_wake_opcode7[7]	cec_wake_opcode7[6]	cec_wake_opcode7[5]	cec_wake_opcode7[4]	cec_wake_opcode7[3]	cec_wake_opcode7[2]	cec_wake_opcode7[1]	cec_wake_opcode7[0]

1.15 CSI-TXA MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x81	csi_tx_top_reg_00	rw	csitx_pwrdn	-	en_autocalc_dphy_params	-	-	num_lanes[2]	num_lanes[1]	num_lanes[0]
0x0D	0x00	csi_tx_top_reg_0d	rw	vc_ref[1]	vc_ref[0]	-	-	-	-	-	-
0x1E	0x00	csi_tx_top_reg_1e	rw	interpret_fs_as_ls	-	-	-	-	-	-	-
0x1F	0x00	csi_tx_top_reg_1f	rw	framenumber_interlaced	-	-	-	-	-	-	-
0x20	0x00	csi_tx_top_reg_20	rw	linenumber_incr_interlaced	-	-	-	-	-	-	-
0x21	0x03	csi_tx_top_reg_21	rw	linenumber1_f1_interlaced[7]	linenumber1_f1_interlaced[6]	linenumber1_f1_interlaced[5]	linenumber1_f1_interlaced[4]	linenumber1_f1_interlaced[3]	linenumber1_f1_interlaced[2]	linenumber1_f1_interlaced[1]	linenumber1_f1_interlaced[0]
0x22	0x02	csi_tx_top_reg_22	rw	linenumber1_f2_interlaced[7]	linenumber1_f2_interlaced[6]	linenumber1_f2_interlaced[5]	linenumber1_f2_interlaced[4]	linenumber1_f2_interlaced[3]	linenumber1_f2_interlaced[2]	linenumber1_f2_interlaced[1]	linenumber1_f2_interlaced[0]
0x26	0x00	csi_tx_top_reg_26	rw	-	-	esc_mode_en_clk	esc_xshutdown_clk	-	-	-	-
0x27	0x00	csi_tx_top_reg_27	rw	esc_mode_en_d3	esc_xshutdown_d3	esc_mode_en_d2	esc_xshutdown_d2	esc_mode_en_d1	esc_xshutdown_d1	esc_mode_en_d0	esc_xshutdown_d0
0x2B	0x1E	csi_tx_top_reg_2b	rw	esc_byte[7]	esc_byte[6]	esc_byte[5]	esc_byte[4]	esc_byte[3]	esc_byte[2]	esc_byte[1]	esc_byte[0]
0x70	0x10	csi_tx_top_reg_70	rw	-	-	-	-	f_bit_inv_pol	-	-	-
0x7E	0x00	csi_tx_top_reg_7e	rw	en_man_fmt	man_fmt[2]	man_fmt[1]	man_fmt[0]	en_man_bpp	man_bpp[2]	man_bpp[1]	man_bpp[0]
0x9C	0x10	csi_tx_top_reg_9c	rw	-	lane1_num[2]	lane1_num[1]	lane1_num[0]	-	lane0_num[2]	lane0_num[1]	lane0_num[0]
0x9D	0x32	csi_tx_top_reg_9d	rw	-	lane3_num[2]	lane3_num[1]	lane3_num[0]	-	lane2_num[2]	lane2_num[1]	lane2_num[0]
0x9E	0x04	csi_tx_top_reg_9e	rw	-	-	-	-	-	laneclk_num[2]	laneclk_num[1]	laneclk_num[0]
0xC1	0x3B	csi_tx_top_reg_c1	rw	-	-	clkln_is_clock_lane	clkln_zero_clk_lane	clkln_lane_is_master	-	-	-
0xC4	0x0A	csi_tx_top_reg_c4	rw	-	-	d0ln_is_clock_lane	d0ln_zero_clk_lane	d0ln_lane_is_master	-	-	-
0xC7	0x02	csi_tx_top_reg_c7	rw	-	-	d1ln_is_clock_lane	d1ln_zero_clk_lane	d1ln_lane_is_master	-	-	-
0xCA	0x02	csi_tx_top_reg_ca	rw	-	-	d2ln_is_clock_lane	d2ln_zero_clk_lane	d2ln_lane_is_master	-	-	-
0xCD	0x02	csi_tx_top_reg_cd	rw	-	-	d3ln_is_clock_lane	d3ln_zero_clk_lane	d3ln_lane_is_master	-	-	-
0xDA	0x00	csi_tx_top_reg_da	rw	-	-	-	-	-	mipi_pll_lock_flag	mipi_pll_clk_det	mipi_pll_en
0xF0	0x01	csi_tx_top_reg_f0	rw	-	-	-	-	-	-	-	dphy_pwdn

1.16 CSI-TXB MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x00	0x81	csi_tx_top_reg_00	rw	csitx_pwrdn	-	en_autocalc_dphy_params	-	-	-	-	-
0x0D	0x00	csi_tx_top_reg_0d	rw	vc_ref[1]	vc_ref[0]	-	-	-	-	-	-
0x1E	0x00	csi_tx_top_reg_1e	rw	interpret_fs_as_ls	-	-	-	-	-	-	-
0x1F	0x00	csi_tx_top_reg_1f	rw	framenumber_interlaced	-	-	-	-	-	-	-
0x20	0x00	csi_tx_top_reg_20	rw	linenumber_incr_interlaced	-	-	-	-	-	-	-
0x21	0x03	csi_tx_top_reg_21	rw	linenumber1_f1_interlaced[7]	linenumber1_f1_interlaced[6]	linenumber1_f1_interlaced[5]	linenumber1_f1_interlaced[4]	linenumber1_f1_interlaced[3]	linenumber1_f1_interlaced[2]	linenumber1_f1_interlaced[1]	linenumber1_f1_interlaced[0]
0x22	0x02	csi_tx_top_reg_22	rw	linenumber1_f2_interlaced[7]	linenumber1_f2_interlaced[6]	linenumber1_f2_interlaced[5]	linenumber1_f2_interlaced[4]	linenumber1_f2_interlaced[3]	linenumber1_f2_interlaced[2]	linenumber1_f2_interlaced[1]	linenumber1_f2_interlaced[0]
0x26	0x00	csi_tx_top_reg_26	rw	-	-	esc_mode_en_clk	esc_xshutdown_clk	-	-	-	-
0x27	0x00	csi_tx_top_reg_27	rw	esc_mode_en_d3	esc_xshutdown_d3	esc_mode_en_d2	esc_xshutdown_d2	esc_mode_en_d1	esc_xshutdown_d1	esc_mode_en_d0	esc_xshutdown_d0
0x2B	0x1E	csi_tx_top_reg_2b	rw	esc_byte[7]	esc_byte[6]	esc_byte[5]	esc_byte[4]	esc_byte[3]	esc_byte[2]	esc_byte[1]	esc_byte[0]
0x70	0x10	csi_tx_top_reg_70	rw	-	-	-	-	f_bit_inv_pol	-	-	-
0xDA	0x00	csi_tx_top_reg_da	rw	-	-	-	-	-	mipi_pll_lock_flag	mipi_pll_clk_det	mipi_pll_en
0xF0	0x01	csi_tx_top_reg_f0	rw	-	-	-	-	-	-	-	dphy_pwdn

2 REGISTER BIT DESCRIPTIONS

2.1 IO MAP

Reg	Bits	Description	
RX_EN			R/W
0x00	00110011	This bit is used to enable the HDMI/MHL Rx. 0 = Disable HDMI/MHL Rx 1 = Enable HDMI/MHL Rx	
RX_PDN			R/W
0x00	00110011	This bit is used to power down the HDMI/MHL Rx digital processor. 0 = Power up HDMI/MHL Rx 1 = Power down HDMI/MHL Rx	
XTAL_PDN			R/W
0x00	00110011	This bit powers down the crystal amplifier. 0 = Power up XTAL amplifier 1 = Power down XTAL amplifier	
CORE_PDN			R/W
0x00	00110011	This bit is the Mode 1 power-down control. In this power-down mode, all blocks are powered down except for the I²C slave and the CEC (to monitor wakeup interrupts). 0 = Power down depending on individual power-down bits 1 = Power down all blocks except I²C slave, CEC, and Interrupt Generator	
MASTER_PDN			R/W
0x00	00110011	This bit is the Mode 0 Power-Down control. In this power-down mode, all blocks except for the I²C slave, CEC, and INTRQ1 are powered down. 0 = Power down depending on individual power-down bits 1 = Power down all blocks except I²C slave, CEC and INTRQ1	
PWRDN2B			R/W
0x01	10110110	Control to enable the power up via CEC when the device is in power down mode. To enable CEC wake-up, set this bit to 1 (default). In power up mode, this bit must be set to 0 for normal CEC and interrupt operations. 0 = CEC wake-up disabled in power-down mode. CEC and interrupt functions operate normally in power-up mode. 1 = CEC wake-up enabled in power-down mode (default)	
PWRDNB			R/W
0x01	10110110	Control to enable the power up via CEC when the device is in power down mode. To enable CEC wake up, set this to 0 (default). In power up mode, this bit must be set to 1 for normal CEC and interrupt operations. 0 = CEC wake up enabled in power-down mode (default) 1 = CEC wake up disabled in power-down mode. CEC and interrupt functions operate normally in power-up mode.	
PROG_XTAL_FREQ[13:0]			R/W
0x01 0x02	10110110 00000000	This control can be used to program the exact XTAL frequency. It is in 5.9 format.	
AVCODE_INSERT_EN			R/W
0x03	00000110	This control is used to insert AV codes into the CP data stream. 0 = Do not insert AV codes into the CP data stream 1 = Insert AV codes into the CP data stream	
CP_V_FREQ[2:0]			R/W
0x03	00000110	This signal selects the vertical frequency for the video through the CP. 000 = 60 Hz 001 = 50 Hz 010 = 30 Hz 011 = 25 Hz 100 = 24 Hz All Others = Reserved	
CP_OP_656_RANGE			R/W
0x03	00000110	This bit enables clipping to 16-to-235 for the Y component and to 16-to-240 in the CP. 0 = No clipping in CP 1 = Enable clipping to 16-to-235/240 range in CP	

Reg	Bits	Description	
CP_DATA_BLANK_EN			R/W
0x03	00000110	This bit enables data-blanking in CP during horizontal and vertical blanking regions. 0 = Disable data blanking in CP 1 = Enable data blanking in CP	
CP_FREE_RUN_EN			R/W
0x03	00000110	This bit enables free run mode in CP. 0 = Disable CP free run mode 1 = Enable CP free run mode	
CP_BYPASS			R/W
0x03	00000110	This bit is used to bypass the CP block. CP is used for video timing adjustment and to support free run with buffered parameters. 0 = Enable CP 1 = Bypass CP	
CP_FORCE_FREERUN_CH1			R/W
0x04	00000010	This control can be used to force the CP core to free run. 0 = CP free runs automatically 1 = Force CP to free run	
CP_REPL_AV_CODE			R/W
0x04	00000010	This control can be used to decide if the AV codes are duplicated on all the data channels, or if they are spread across them. 0 = AV codes are not replicated. AV codes are spread across the data channels. 1 = AV codes are replicated. Complete AV codes are output on all the data channels.	
CP_ALT_GAMMA			R/W
0x04	00000010	This control is used to select the type of YPbPr color space conversion. It is to be used in conjunction with CP_INP_COLOR_SPACE[3:0] and CP_RGB_OUT. If CP_ALT_GAMMA is set to 1 and CP_RGB_OUT is set to 0, a color space conversion is applied to convert from 601 to 709 or 709 to 601. It is valid only if CP_RGB_OUT is set to 0. 0 = No conversion 1 = Apply YUV601 to YUV709 conversion if input is YUV601, apply YUV709 to YUV601 conversion if input is YUV709	
CP_ALT_DATA_SAT			R/W
0x04	00000010	This control is used to disable the data saturator that limits the output range independently of CP_OP_656_RANGE. It is used to support extended data range modes. 0 = Data saturator enabled or disabled according to CP_OP_656_RANGE setting. 1 = Reverses CP_OP_656_RANGE decision to enable or disable the data saturator	
CP_RGB_OUT			R/W
0x04	00000010	This control is used to select the output color space and the correct digital blank level and offsets on the RGB or YPrPb outputs. It is used in conjunction with CP_INP_COLOR_SPACE[3:0] and CP_ALT_GAMMA to select the applied CSC. 0 = YPbPr color space output 1 = RGB color space output	

Reg	Bits	Description	
CP_VID_STD[7:0]			R/W
0x05	01000000	This register selects the video standard for the video through CP. If not in buffered free run, the video-standard parameters are used to generate the free run video. Buffered free-run happens when the input was removed after having been connected. Non-buffered free run happens when no input was ever connected, or when the device is free running with DIS_AUTO_PARAM_BUFF (CP Map, Register 0xC9[0]) is set to 1. 0x40 = 480i (720 × 480 active resolution) 0x41 = 576i (720 × 576 active resolution) 0x4A = 480p (720 × 480 active resolution) 0x4B = 576p (720 × 576 active resolution) 0x53 = 720p (1280 × 720 active resolution) 0x54 = 1080i (1920 × 1080 active resolution) 0x5E = 1080p (1920 × 1080 active resolution) 0x80 = SVGA 56 (800 × 600 active resolution, 56 Hz frame rate) 0x81 = SVGA 60 (800 × 600 active resolution, 60 Hz frame rate) 0x82 = SVGA 72 (800 × 600 active resolution, 72 Hz frame rate) 0x83 = SVGA 75 (800 × 600 active resolution, 75 Hz frame rate) 0x84 = SVGA 85 (800 × 600 active resolution, 85 Hz frame rate) 0x85 = SXGA 60 (1280 × 1024 active resolution, 60 Hz frame rate) 0x86 = SXGA 75 (1280 × 1024 active resolution, 75 Hz frame rate) 0x88 = VGA 60 (640 × 480 active resolution, 60 Hz frame rate) 0x89 = VGA 72 (640 × 480 active resolution, 72 Hz frame rate) 0x8A = VGA 75 (640 × 480 active resolution, 75 Hz frame rate) 0x8B = VGA 85 (640 × 480 active resolution, 85 Hz frame rate) 0x8C = XGA 60 (1024 × 768 active resolution, 60 Hz frame rate) 0x8D = XGA 70 (1024 × 768 active resolution, 70 Hz frame rate) 0x8E = XGA 75 (1024 × 768 active resolution, 75 Hz frame rate) 0x8F = XGA 85 (1024 × 768 active resolution, 85 Hz frame rate) 0x96 = UXGA 60 (1600 × 1200 active resolution, 60 Hz frame rate)	
PIX_IN_REVERSE			R/W
0x06	00010001	This control is used to reverse the 8-bit digital input interface. 0 = Do not reverse the input 8-bit bus 1 = Reverse the input 8-bit bus	
PIX_IN_INPUT_AS_DDR			R/W
0x06	00010001	This control is used to configure the 8-bit digital input interface in DDR or SDR mode. 0 = 8-bit digital input interface is in SDR mode 1 = 8-bit digital input interface is in DDR mode	
PIX_IN_SPLIT_AVCODE			R/W
0x06	00010001	This control is used to indicate if the AV codes in the 8-bit digital input interface are split across the Y and Cr/Cb channels, or replicated on the Y and Cr/Cb channels. 0 = AV codes are repeated on both Y and Cr/Cb channels. 0xFF 0xFF 0x00 0x00 0x00 0x00 0xXY 0xXY AV codes are received on the 8-bit digital input interface. 1 = AV codes are split between the Y and Cr/Cb channels. 0xFF 0x00 0x00 0xXY AV codes are received on the 8-bit digital input interface.	
PIX_IN_KEEP_AVCODES_IN_TWO_CH			R/W
0x06	00010001	This control is used to keep the AV codes between the 8-bit digital input interface and the MIPI Tx block. The MIPI Tx block always needs the AV codes at its input; therefore, this bit must be set to 1. 0 = AV codes removed when the data from the 8-bit digital input interface is passed to the MIPI transmitter 1 = AV codes maintained when the data from the 8-bit digital input interface is passed to the MIPI transmitter	
VMUTE			R/W
0x0A	00000000	This control can be used to manually force a Video Mute. 0 = Do not force a Video Mute 1 = Manually force a Video mute	
LLC_DLL_EN			R/W
0x0C	00000000	This control is used to enable the delay locked loop (DLL) for the LLC output pixel clock. 0 = Disable LLC DLL 1 = Enable LLC DLL	
LLC_DLL_DOUBLE			R/W
0x0C	00000000	This control is used to double the LLC output pixel clock. 0 = Do not double LLC pixel clock 1 = Double LLC pixel clock	

Reg	Bits	Description	
LLC_DLL_MUX			R/W
0x0C	00<u>0</u>00000	This control is used to bypass the DLL for the LLC output pixel clock. 0 = Bypass LLC DLL 1 = Do not bypass LLC DLL	
LLC_DLL_PHASE[4:0]			R/W
0x0C	000<u>00000</u>	This control is used to adjust the LLC DLL phase in increments of 1/32 of a clock period. 0x00 = No phase shift is applied; the phase is set to its default value 0x01 = the phase is incremented by 1 × (1/32) of a clock period. … 0x1F = the phase is incremented by 31 × (1/32) of a clock period.	
DRV_PIXEL_PADS[1:0]			R/W
0x0D	<u>10</u>101010	This signal is used to set the drive strength for the 8-bit digital output pads. 00 = Not used 01 = Min drive strength 10 = Mid drive strength 11 = Max drive strength	
DRV_AUDIO_PADS[1:0]			R/W
0x0D	10<u>10</u>1010	This signal is used to set the drive strength for the Audio pads. 00 = Not used 01 = Min drive strength 10 = Mid drive strength 11 = Max drive strength	
DR_STR_SPI[1:0]			R/W
0x0D	1010<u>10</u>10	This signal is used to set the drive strength for the SPI pads. 00 = Not used 01 = Min drive strength 10 = Mid drive strength 11 = Max drive strength	
DRV_INT_I2C_CSB_PADS[1:0]			R/W
0x0D	101010<u>10</u>	This signal is used to set the drive strength for the INT and I^2C pads. 00 = Not used 01 = Min drive strength 10 = Mid drive strength 11 = Max drive strength	
TRI_LLC			R/W
0x0E	<u>1</u>1111111	This control can be used to tristate the LLC output 0 = Do not tristate LLC output pad 1 = Tristate LLC output pad	
TRI_PIX			R/W
0x0E	1<u>1</u>111111	This control can be used to tristate the Pixel output port pins 0 = Do not tristate pixel output pins 1 = Tristate pixel output pins	
TRI_AUD			R/W
0x0E	11<u>1</u>11111	This control can be used to tristate the audio output port pins 0 = Do not tristate audio output pins 1 = Tristate audio output pins	
TRI_SPI			R/W
0x0E	111<u>1</u>1111	This control can be used to tristate the SPI MISO pin output 0 = Do not tristate SPI MISO output 1 = Tristate SPI MISO output	
PDN_PIX			R/W
0x0E	11111<u>1</u>11	This control can be used to power down the pixel IO pads 0 = Power up pixel IO port pads 1 = Power down pixel IO port pads	
PDN_AUD			R/W
0x0E	111111<u>1</u>1	This control can be used to power down the audio pads 0 = Power up audio output pads 1 = Power down audio output pads	

Reg	Bits	Description	
PDN_SPI			R/W
0x0E	1111111<u>1</u>	This control can be used to power down the SPI pads 0 = Power up SPI pads 1 = Power down SPI pads	
PDN_VBUS_EN			R/W
0x0F	0000<u>1</u>001	This control can be used to power down the VBUS_EN pad 0 = Power up VBUS_EN pad 1 = Power down VBUS_EN pad	
XTAL_FREQ_SEL[1:0]			R/W
0x0F	00001<u>01</u>	This control can be used to specify the crystal frequency used. 00 = 27 MHz 01 = 28.63636 MHz 10 = 24.576 MHz 11 = Crystal frequency specified by PROG_XTAL_FREQ	
CSI4_EN			R/W
0x10	<u>0</u>0000000	This control is used to enable the 4-lane MIPI CSI transmitter. 0 = Disable CSI 4 lane Tx 1 = Enable CSI 4 lane Tx	
CSI1_EN			R/W
0x10	0<u>0</u>000000	This control is used to enable the 1-lane MIPI CSI transmitter. 0 = Disable CSI 1 lane Tx 1 = Enable CSI 1 lane Tx	
PIX_OUT_EN			R/W
0x10	000<u>0</u>0000	This control is used to enable the 8-bit digital output port. 0 = Disable pixel output port 1 = Enable pixel output port	
SD_THRU_PIX_OUT			R/W
0x10	0000<u>0</u>000	This control is used to select if the output of the SD core or the output of the CP is routed to the 8-bit digital output port. 0 = Output of CP core is routed to pixel output port 1 = Output of SD core is routed to pixel output port	
CSI4_IN_SEL[1:0]			R/W
0x10	000000<u>00</u>0	This control is used to select the signal routed to the 4-lane MIPI CSI transmitter. 00 = Output of CP core is routed to MIPI CSI 4-lane Tx 01 = 8-bit digital input pixel port routed to MIPI CSI 4-lane Tx 1x = Output of SD core routed to MIPI CSI 4-lane Tx	
SD_DDR_OUT			R/W
0x11	0000<u>1</u>000	This control is used to select SDR or DDR output mode when the output of the SD core is routed to the 8-bit digital output port. 0 = 8-bit digital output port in SDR mode for data from SD core 1 = 8-bit digital output port in DDR mode for data from SD core	
CP_INP_COLOR_SPACE[3:0]			R/W
0x12	<u>1111</u>0000	This control is used to set the color space of the input video. It is to be used in conjunction with CP_ALT_GAMMA and CP_RGB_OUT to configure the color space converter. A value of 4'b1111 selects automatic setting of the input color space base on the primary mode and video standard settings. Settings 1000 to 1110 are undefined. 0000 = Forces RGB (range 16 to 235) input 0001 = Forces RGB (range 0 to 255) input 0010 = Forces YCrCb input (601 color space) (range of 16 to 235) 0011 = Forces YCrCb input (709 color space) (range of 16 to 235) 0100 = Forces XVYCC 601 0101 = Forces XVYCC 709 0110 = Forces YCrCb input (601 color space) (range of 0 to 255) 0111 = Forces YCrCb input (709 color space) (range of 0 to 255) 1111 = In HDMI/MHL mode, input color space depends on color space reported by HDMI/MHL block.	

Reg	Bits	Description	
CP_OUT_MODE[1:0]			R/W
0x12	11110000	This control is used to select SDR 444, SDR 422 2ch, SDR 422 1ch or DDR 422 output mode for the output of the CP core. SDR444 and SDR422 2ch (Y and C separate and nominal pixel rate) can only be used when the output of the CP core is routed to the 4-lane MIPI CSI transmitter. SDR 422 1ch (Y and C interleaved over a single channel and pixel clock doubled) and DDR 422 only apply when the output of the CP core is routed to the 8-bit digital output port. However, when SDR 422 1ch or DDR 422 is selected, video can be output on both the 8-bit digital output port and the 4-lane MIPI CSI transmitter. In this situation, the output of the CP core is routed to the 4-lane MIPI CSI transmitter as 16-bit YCrCb SDR 4:2:2. 00 = SDR 444 (4-lane MIPI CSI output only) 01 = SDR 422 2ch (4-lane MIPI CSI output only) 10 = SDR 422 1ch (only applies to the 8-bit digital output port) 11 = DDR 422 (only applies to the 8-bit digital output port)	
CP_OUT_10B			R/W
0x12	11110000	This control is used to select the bit width at the output of the CP core. 0 = CP out is 8-bit (4-lane MIPI CSI output and 8-bit digital output port) 1 = CP out is 10-bit (4-lane MIPI CSI output only)	
BR_DITH_CCIR601_B			R/W
0x17	10000000	This control can be used to specify the output range when dither is applied. 0 = Saturate Luma and Chroma values to 235 and 240, respectively 1 = Saturate both Luma and Chroma values to 254	
BR_DITH_YUV422_MODE			R/W
0x17	10000000	This control can be used to define the format of the input to the dither block. 0 = Input format to dither block is YUV444/RGB444 1 = Input format to dither block is YUV422	
BR_DITHER_MODE			R/W
0x17	10000000	This control is used to select the bit reduction dither mode. 0 = No bit reduction (8-bit mode) 1 = 8-bit to 6-bit reduction (6-bit mode)	
RND_DITHER_EN			R/W
0x17	10000000	This bit is used to control the randomization of dither. 0 = No randomization of dither 1 = Random data added to dither	
BR_DITHER_EN			R/W
0x17	10000000	This control is used to select if dither or simple rounding is applied in 8-bit to 6-bit reduction mode. 0 = Dithering disabled 1 = Dithering enabled	
DIAG1_SLICERS_PWRDN			R/W
0x18	01101101	This control can be used to power up the DIAG1 level slicers. 0 = Power up the DIAG1 slicers 1 = Power down the DIAG1 slicers	
DIAG1_BILEVEL_EN			R/W
0x18	01101101	This control can be used to enable the upper level slicer for DIAG1. 0 = Enable the upper level slicer for DIAG1 1 = Reserved	
DIAG1_UPPER_SLICE_LEVEL[2:0]			R/W
0x18	01101101	This control can be used to set the upper slice level for DIAG1. 110 = Sets the DIAG1 upper slice level to 975 mV 111 = Sets the DIAG1 upper slice level to 1.125 V	
DIAG2_SLICERS_PWRDN			R/W
0x19	01101101	This control can be used to power up the DIAG2 level slicers. 0 = Power up the DIAG2 slicers 1 = Power down the DIAG2 slicers	
DIAG2_BILEVEL_EN			R/W
0x19	01101101	This control can be used to enable the upper level slicer for DIAG2. 0 = Enable the upper level slicer for DIAG2 1 = Reserved	

Reg	Bits	Description	
DIAG2_UPPER_SLICE_LEVEL[2:0]			R/W
0x19	011011<u>01</u>	This control can be used to set the upper slice level for DIAG2. 110 = Sets the DIAG2 upper slice level to 975 mV 111 = Sets the DIAG2 upper slice level to 1.125 V	
DIAG3_SLICERS_PWRDN			R/W
0x1A	0<u>1</u>101101	This control can be used to power up the DIAG3 level slicers. 0 = Power up the DIAG3 slicers 1 = Power down the DIAG3 slicers	
DIAG3_BILEVEL_EN			R/W
0x1A	01<u>1</u>01101	This control can be used to enable the upper level slicer for DIAG3. 0 = Enable the upper level slicer for DIAG3 1 = Reserved	
DIAG3_UPPER_SLICE_LEVEL[2:0]			R/W
0x1A	011011<u>01</u>	This control can be used to set the upper slice level for DIAG3. 110 = Sets the DIAG3 upper slice level to 975 mV 111 = Sets the DIAG3 upper slice level to 1.125 V	
DIAG4_SLICERS_PWRDN			R/W
0x1B	0<u>1</u>101101	This control can be used to power up the DIAG4 level slicers. 0 = Power up the DIAG4 slicers 1 = Power down the DIAG4 slicers	
DIAG4_BILEVEL_EN			R/W
0x1B	01<u>1</u>01101	This control can be used to enable the upper level slicer for DIAG4. 0 = Enable the upper level slicer for DIAG4 1 = Reserved	
DIAG4_UPPER_SLICE_LEVEL[2:0]			R/W
0x1B	011011<u>01</u>	This control can be used to set the upper slice level for DIAG4. 110 = Sets the DIAG4 upper slice level to 975 mV 111 = Sets the DIAG4 upper slice level to 1.125 V	
PDN_INT1			R/W
0x1D	<u>0</u>1111000	This control is used to power down the interrupt INTRQ1 pad. 0 = Do not power down INTRQ1 pad 1 = Power down INTRQ1 pad	
PDN_INT2			R/W
0x1D	0<u>1</u>111000	This control is used to power down the interrupt INTRQ2 pad. 0 = Do not power down INTRQ2 pad 1 = Power down INTRQ2 pad	
PDN_INT3			R/W
0x1D	01<u>1</u>11000	This control is used to power down the interrupt INTRQ3 pad. 0 = Do not power down INTRQ3 pad 1 = Power down INTRQ3 pad	
INV_LLC			R/W
0x1D	011<u>1</u>1000	This control is used to invert the polarity of the LLC output clock. 0 = LLC output polarity not inverted 1 = LLC output polarity inverted	
DRV_LLC_PAD[1:0]			R/W
0x1D	0111<u>10</u>00	This signal is used to set the drive strength for the LLC output pad. 00 = Not used 01 = Min drive strength 10 = Mid drive strength 11 = Max drive strength	
INT_CEC_ST			R
0x3F	0000<u>0</u>000	This bit indicates that a CEC interrupt has occurred. 0 = CEC interrupt has not occurred 1 = CEC interrupt has occurred	

Reg	Bits	Description
INT_HDMI_ST		R
0x3F	0000**0**000	This bit indicates that an HDMI/MHL interrupt has occurred. 0 = HDMI/MHL interrupt has not occurred 1 = HDMI/MHL interrupt has occurred
INTRQ3_RAW		R
0x3F	00000**0**00	This bit indicates the status of the interrupt signal on the INTRQ3 interrupt pin. If an interrupt event that has been enabled for the INTRQ3 pin has occurred, this bit is set to 1. Interrupts for INTRQ3 are set via the Interrupt 3 mask bits. This bit remains set to 1 until all status for interrupts enabled on INTRQ3 are cleared. 0 = No interrupt on INTRQ3 1 = An interrupt event for INTRQ3 has occurred
INTRQ2_RAW		R
0x3F	000000**0**0	This bit indicates the status of the interrupt signal on the INTRQ2 interrupt pin. If an interrupt event that has been enabled for the INTRQ2 pin has occurred, this bit is set to 1. Interrupts for INTRQ2 are set via the Interrupt 2 mask bits. This bit remains set to 1 until all status for interrupts enabled on INTRQ2 are cleared. 0 = No interrupt on INTRQ2 1 = An interrupt event for INTRQ2 has occurred
INTRQ_RAW		R
0x3F	0000000**0**	This bit indicates the status of the interrupt signal on the INTRQ1 interrupt pin. If an interrupt event that has been enabled for the INTRQ1 pin has occurred, this bit is set to 1. Interrupts for INTRQ1 are set via the Interrupt 1 mask bits. This bit remains set to 1 until all status for interrupts enabled on INTRQ1 are cleared. 0 = No interrupt on INTRQ1 1 = An interrupt event for INTRQ1 has occurred
INTRQ_DUR_SEL[1:0]		R/W
0x40	**00**000000	This signal selects the interrupt signal duration for the interrupt signal on INTRQ1. 00 = 4 XTAL periods 01 = 16 XTAL periods 10 = 64 XTAL periods 11 = Active until cleared
STORE_UNMASKED_IRQS		R/W
0x40	000**0**0000	This bit allows the HDMI/MHL status flags for any HDMI/MHL interrupt to be triggered regardless of whether the mask bits are set. This bit allows a HDMI/MHL interrupt to trigger and allows this interrupt to be read back through the corresponding status bit without triggering an interrupt on the interrupt pin. The status is stored until the clear bit is used to clear the status register and allows another interrupt to occur. 0 = Does not allow x_ST flag of any HDMI/MHL interrupt to be set independently of mask bits 1 = Allows x_ST flag of any HDMI/MHL interrupt to be set independently of mask bits
EN_UMASK_RAW_INTRQ		R/W
0x40	0000**0**000	This bit outputs the internal raw interrupt signal on the INTRQ1 interrupt. 0 = Does not output raw interrupt signal on INTRQ1 1 = Outputs raw interrupt signal on INTRQ1
MPU_STIM_INTRQ		R/W
0x40	00000**0**00	Manual interrupt set control. Use this feature for test purposes only. Note that the appropriate mask bit must be set to generate an interrupt at the pin. 0 = Disables manual interrupt mode 1 = Enables manual interrupt mode
INTRQ_OP_SEL[1:0]		R/W
0x40	000000**00**	This signal selects the interrupt signal configuration for INTRQ1. 00 = Open drain 01 = Drives low when active 10 = Drives high when active 11 = Disabled
INTRQ2_DUR_SEL[1:0]		R/W
0x41	**00**100000	This signal selects the interrupt signal duration for the interrupt signal on INTRQ2. 00 = 4 XTAL periods 01 = 16 XTAL periods 10 = 64 XTAL periods 11 = Active until cleared

Reg	Bits	Description	
CP_LOCK_UNLOCK_EDGE_SEL			R/W
0x41	00100000	This bit configures the functionality of the CP_LOCK, UNLOCK interrupts. 0 = Generate interrupt for a low to high change in CP_LOCK, UNLOCK status 1 = Generate interrupt for a low to high or a high to low change in CP_LOCK, UNLOCK status	
EN_UMASK_RAW_INTRQ2			R/W
0x41	00100000	This bit outputs the internal raw interrupt signal on the INTRQ2 interrupt pin. 0 = Does not output raw interrupt signal on INTRQ2 1 = Outputs raw interrupt on INTRQ2	
INT2_EN			R/W
0x41	00100000	This bit enables the INTRQ2 interrupt. 0 = Disable INTRQ2 1 = Enable INTRQ2	
INTRQ2_OP_SEL[1:0]			R/W
0x41	00100000	This bit selects the interrupt signal configuration for INTRQ2. 00 = Open drain 01 = Drives low when active 10 = Drives high when active 11 = Disabled	
CP_LOCK_CP_RAW			R
0x43	00000000	This bit indicates that the CP input has changed from an unlocked state to a locked state. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this bit to be functional. 0 = No change 1 = CP input changed from an unlocked state to a locked state	
CP_UNLOCK_CP_RAW			R
0x43	00000000	This bit indicates that the CP input has changed from a locked state to an unlocked state. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this bit to be functional. 0 = No change 1 = CP input has changed from a locked state to an unlocked state	
VMUTE_REQUEST_HDMI_RAW			R
0x43	00000000	This bit indicates the raw status of the VMUTE_REQUEST_HDMI signal. This bit is set if a condition in the HDMI/MHL input signal caused the video to be muted 0 = VMUTE_REQUEST_HDMI is not valid. HDMI/MHL input video is not muted 1 = VMUTE_REQUEST_HDMI is valid. HDMI/MHL input video is muted	
MPU_STIM_INTRQ_RAW			R
0x43	00000000	This bit indicates the raw status of the manual forced interrupt signal. 0 = Manual forced interrupt not applied 1 = Manual forced interrupt applied	
INT_SD_RAW			R
0x43	00000000	Raw status signal indicating that an SDP interrupt has occurred 0 = No change 1 = SDP interrupt has occurred	
CP_LOCK_CP_ST			R
0x44	00000000	This bit indicates that the status of the CP Locked interrupt has changed from an unlocked state to a locked state and has triggered an interrupt. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this bit to be functional. 0 = No change: an interrupt has not been generated 1 = CP input has changed from an unlocked state to a locked state	
CP_UNLOCK_CP_ST			R
0x44	00000000	This bit indicates that the status of the CP Unlocked interrupt has changed from a locked state to an unlocked state and has triggered an interrupt. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this bit to be functional. 0 = No change: an interrupt has not been generated 1 = CP input changed from a locked state to an unlocked state	
VMUTE_REQUEST_HDMI_ST			R
0x44	00000000	This bit indicates the latched signal status of the VMUTE_REQUEST_HDMI interrupt signal. Once set, this bit remains high until the interrupt has been cleared via VMUTE_REQUEST_HDMI_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No change: an interrupt has not been generated 1 = VMUTE_REQUEST_HDMI interrupt has occurred	

Reg	Bits	Description	
MPU_STIM_INTRQ_ST			R
0x44	0000000**0**	This bit indicates the latched signal status of the manual forced interrupt signal. Once set, this bit remains high until the interrupt has been cleared via MPU_STIM_INTRQ_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Forced manual interrupt event has not occurred 1 = Forced manual interrupt event has occurred	
INT_SD_ST			R
0x44	0000000**0**	This bit indicates that the status of the INT_SD_RAW interrupt has changed. Once set, this bit remains high until the interrupt has been cleared via INT_SD_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No change: an interrupt has not been generated 1 = SDP_INT_RAW has changed	
CP_LOCK_CP_CLR			SC
0x45	**0**0000000	This bit is used to clear the CP_LOCK_CP_ST interrupt. 0 = Does not clear 1 = Clears CP_LOCK_CP_ST	
CP_UNLOCK_CP_CLR			SC
0x45	0**0**000000	This bit is used to clear the CP_UNLOCK_CP_ST interrupt. 0 = Does not clear 1 = Clears CP_UNLOCK_CP_ST	
VMUTE_REQUEST_HDMI_CLR			SC
0x45	00**0**00000	This bit is used to clear the VMUTE request interrupt signal, VMUTE_REQUEST_HDMI_ST. 0 = Does not clear VMUTE_REQUEST_HDMI_ST 1 = Clears VMUTE_REQUEST_HDMI_ST	
MPU_STIM_INTRQ_CLR			SC
0x45	000000**0**0	This bit is used to clear the manual forced interrupt, MPU_STIM_INTRQ_ST. 0 = Does not clear MPU_STIM_INT_ST bit 1 = Clears MPU_STIM_INT_ST bit	
INT_SD_CLR			SC
0x45	0000000**0**	This bit is used to clear the SD interrupt, INT_SD_ST. 0 = Does not clear INT_SD_ST 1 = Clears INT_SD_ST	
CP_LOCK_CP_MB2			R/W
0x46	**0**0000000	This bit is used to unmask the CP_LOCK_CP_ST interrupt on INTRQ2. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this interrupt to be functional. 0 = Disables CP_LOCK_CP_ST interrupt for INTRQ2 1 = Enables CP_LOCK_CP_ST interrupt for INTRQ2	
CP_UNLOCK_CP_MB2			R/W
0x46	0**0**000000	This bit is used to unmask the CP_UNLOCK_CP_ST interrupt on INTRQ2. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this interrupt to be functional. 0 = Disables CP_UNLOCK_CP_ST interrupt for INTRQ2 1 = Enables CP_UNLOCK_CP_ST interrupt for INTRQ2	
VMUTE_REQUEST_HDMI_MB2			R/W
0x46	00**0**00000	This bit is the INTRQ2 interrupt mask for the VMUTE_REQUEST_HDMI interrupt. When set, the VMUTE_REQUEST_HDMI triggers the INTRQ2 interrupt and VMUTE_REQUEST_HDMI_ST indicates the interrupt status. 0 = Disables VMUTE_REQUEST_HDMI interrupt for INTRQ2 1 = Enables VMUTE_REQUEST_HDMI interrupt for INTRQ2	
MPU_STIM_INTRQ_MB2			R/W
0x46	000000**0**0	This bit is the INTRQ2 interrupt mask for the manual forced interrupt signal. When set, the manual forced interrupt triggers the INTRQ2 interrupt and MPU_STIM_INTRQ_ST indicates the interrupt status. 0 = Disables manual forced interrupt for INTRQ2 1 = Enables manual forced interrupt for INTRQ2	
INT_SD_MB2			R/W
0x46	0000000**0**	This bit is used to unmask the INT_SD_ST interrupt on INTRQ2. When set, the SDP interrupt triggers the INTRQ2 interrupt and INT_SD_ST indicates the interrupt status. 0 = Disables INT_SD_ST interrupt for INTRQ2 1 = Enables INT_SD_ST interrupt for INTRQ2	

Reg	Bits	Description	
CP_LOCK_CP_MB1			R/W
0x47	00000000	This bit is used to unmask the CP_LOCK_CP_ST interrupt on INTRQ1. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this interrupt to be functional. 0 = Disables CP_LOCK_CP_ST interrupt for INTRQ1 1 = Enables CP_LOCK_CP_ST interrupt for INTRQ1	
CP_UNLOCK_CP_MB1			R/W
0x47	00000000	This bit is used to unmask the CP_UNLOCK_CP_ST interrupt on INTRQ1. CP free run mode must be enabled (IO Map Register 0x03[1] set to 1) for this interrupt to be functional. 0 = Disables CP_UNLOCK_CP_ST interrupt for INTRQ1 1 = Enables CP_UNLOCK_CP_ST interrupt for INTRQ1	
VMUTE_REQUEST_HDMI_MB1			R/W
0x47	00000000	This bit is the INTRQ1 interrupt mask for the VMUTE_REQUEST_HDMI interrupt. When set, the VMUTE_REQUEST_HDMI interrupt triggers the INTRQ1 interrupt and VMUTE_REQUEST_HDMI_ST indicates the interrupt status. 0 = Disables VMUTE_REQUEST_HDMI interrupt for INTRQ1 1 = Enables VMUTE_REQUEST_HDMI interrupt for INTRQ1	
MPU_STIM_INTRQ_MB1			R/W
0x47	00000000	This bit is the INTRQ1 interrupt mask for the manual forced interrupt signal. When set, the manual forced interrupt triggers the INTRQ1 interrupt and MPU_STIM_INTRQ_ST indicates the interrupt status. 0 = Disables manual forced interrupt for INTRQ1 1 = Enables manual forced interrupt for INTRQ1	
INT_SD_MB1			R/W
0x47	00000000	This bit is used to unmask the INT_SD_ST interrupt on INTRQ1. When set, the SDP interrupt triggers the INTRQ1 interrupt and INT_SD_ST indicates the interrupt status. 0 = Disables INT_SD_ST interrupt for INTRQ1 1 = Enables INT_SD_ST interrupt for INTRQ1	
CEC_RX_RDY2_RAW			R
0x49	00000000	Raw status of CEC Receiver Buffer 2 Ready signal. When set to 1, it indicates that a CEC frame has been received and is waiting to be read in Receiver Frame Buffer 2. 0 = No change 1 = CEC Rx Buffer 2 has received a complete message that is ready to be read by the host	
CEC_RX_RDY1_RAW			R
0x49	00000000	Raw status of CEC Receiver Buffer 1 Ready signal. When set to 1, it indicates that a CEC frame has been received and is waiting to be read in Receiver Frame Buffer 1. 0 = No change 1 = CEC Rx buffer 1 has received a complete message that is ready to be read by the host	
CEC_RX_RDY0_RAW			R
0x49	00000000	Raw status of CEC Receiver Buffer 0 Ready signal. When set to 1, it indicates that a CEC frame has been received and is waiting to be read in Receiver Frame Buffer 0. 0 = No change 1 = CEC Rx Buffer 0 has received a complete message that is ready to be read by the host	
CEC_TX_RETRY_TIMEOUT_RAW			R
0x49	00000000	Raw status of CEC Transmitter retry timeout signal. 0 = No change 1 = CEC Tx has retried to send the current message by the no. of times specified by CEC_TX_RETRY but it was unsuccessful every time	
CEC_TX_ARBITRATION_LOST_RAW			R
0x49	00000000	Raw status of CEC transmitter arbitration lost signal. 0 = No change 1 = CEC Tx has lost arbitration to another Tx	
CEC_TX_READY_RAW			R
0x49	00000000	Raw status of CEC transmitter message sent signal. This bit goes high whenever the Tx has successfully sent a message. 0 = No change 1 = CEC Tx has successfully sent the last outgoing message	

Reg	Bits	Description	
CEC_RX_RDY2_ST			R
0x4A	00**0**00000	Latched status of CEC_RX_RDY2_RAW signal. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. When a message has been received in Buffer 2, this bit is set. Once set, this bit remains high until the interrupt has been cleared via CEC_RX_RDY2_CLR. 0 = No change 1 = New CEC message received in Buffer 2	
CEC_RX_RDY1_ST			R
0x4A	000**0**0000	Latched status of CEC_RX_RDY1_RAW signal. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. When a message has been received in Buffer 1, this bit is set. Once set, this bit remains high until the interrupt has been cleared via CEC_RX_RDY1_CLR. 0 = No change 1 = New CEC message received in Buffer 1	
CEC_RX_RDY0_ST			R
0x4A	0000**0**000	Latched status of CEC_RX_RDY0_RAW signal. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. When a message has been received in Buffer 0, this bit is set. Once set, this bit remains high until the interrupt has been cleared via CEC_RX_RDY0_CLR. 0 = No change 1 = New CEC message received in Buffer 0	
CEC_TX_RETRY_TIMEOUT_ST			R
0x4A	00000**0**00	Latched status of CEC_TX_RETRY_TIMEOUT_RAW signal. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. If the CEC Tx fails to send the current message within the number of retry attempts specified by CEC_TX_RETRY, this bit is set. Once set, this bit remains high until the interrupt has been cleared via CEC_TX_RETRY_TIMEOUT_CLR. 0 = No change 1 = CEC Tx has tried but failed to resend the current message for the number of times specified by CEC_TX_RETRY	
CEC_TX_ARBITRATION_LOST_ST			R
0x4A	000000**0**0	Latched status of CEC_TX_ARBITRATION_LOST_RAW signal. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. If the CEC Tx loses arbitration while trying to send a message, this bit is set. Once set, this bit remains high until the interrupt has been cleared via CEC_TX_ARBITRATION_LOST_CLR. 0 = No change 1 = The CEC Tx has lost arbitration to another Tx	
CEC_TX_READY_ST			R
0x4A	0000000**0**	Latched status of CEC_TX_READY_RAW signal. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. When the CEC Tx successfully sends the current message, this bit is set. Once set, this bit remains high until the interrupt has been cleared via CEC_TX_READY_CLR. 0 = No change 1 = Message transmitted successfully	
CEC_RX_RDY2_CLR			SC
0x4B	00**0**00000	Clear bit for CEC Receiver Buffer 2 Ready interrupt. 0 = Does not clear CEC_RX_RDY2_ST 1 = Clears CEC_RX_RDY2_ST	
CEC_RX_RDY1_CLR			SC
0x4B	000**0**0000	Clear bit for CEC Receiver Buffer 1 Ready interrupt. 0 = Does not clear CEC_RX_RDY1_ST 1 = Clears CEC_RX_RDY1_ST	
CEC_RX_RDY0_CLR			SC
0x4B	0000**0**000	Clear bit for CEC Receiver Buffer 0 Ready interrupt. 0 = Does not clear CEC_RX_RDY0_ST 1 = Clears CEC_RX_RDY0_ST	
CEC_TX_RETRY_TIMEOUT_CLR			SC
0x4B	00000**0**00	Clear bit for CEC Transmitter Retry Timeout interrupt. 0 = Does not clear CEC_TX_RETRY_TIMEOUT_ST 1 = Clears CEC_TX_RETRY_TIMEOUT_ST	
CEC_TX_ARBITRATION_LOST_CLR			SC
0x4B	000000**0**0	Clear bit for CEC Transmitter Arbitration Lost interrupt. 0 = Does not clear CEC_TX_ARBITRATION_LOST_ST 1 = Clears CEC_TX_ARBITRATION_LOST_ST	

Reg	Bits	Description	
CEC_TX_READY_CLR			SC
0x4B	0000000**0**	Clear bit for CEC Transmitter Ready interrupt. 0 = Does not clear CEC_TX_READY_ST 1 = Clears CEC_TX_READY_ST	
CEC_RX_RDY2_MB2			R/W
0x4C	00**0**00000	INTRQ2 interrupt mask for CEC Receiver Buffer 2 Ready interrupt. When set, the CEC Receiver Buffer 2 Ready interrupt triggers the INTRQ2 interrupt and CEC_RX_RDY2_ST indicates the interrupt status. 0 = Disables CEC Receiver Buffer 2 Ready interrupt on INTRQ2 1 = Enables CEC Receiver Buffer 2 Ready interrupt on INTRQ2	
CEC_RX_RDY1_MB2			R/W
0x4C	000**0**0000	INTRQ2 interrupt mask for CEC Receiver Buffer 1 Ready interrupt. When set, the CEC Receiver Buffer 1 Ready interrupt triggers the INTRQ2 interrupt and CEC_RX_RDY1_ST indicates the interrupt status. 0 = Disables CEC Receiver Buffer 1 Ready interrupt on INTRQ2 1 = Enables CEC Receiver Buffer 1 Ready interrupt on INTRQ2	
CEC_RX_RDY0_MB2			R/W
0x4C	0000**0**000	INTRQ2 interrupt mask for CEC Receiver Buffer 0 Ready interrupt. When set, the CEC Receiver Buffer 0 Ready interrupt triggers the INTRQ2 interrupt and CEC_RX_RDY0_ST indicates the interrupt status. 0 = Disables CEC Receiver Buffer 0 Ready interrupt on INTRQ2 1 = Enables CEC Receiver Buffer 0 Ready interrupt on INTRQ2	
CEC_TX_RETRY_TIMEOUT_MB2			R/W
0x4C	00000**0**00	INTRQ2 interrupt mask for CEC Transmitter Retry Timeout interrupt. When set, the CEC Transmitter Retry Timeout interrupt triggers the INTRQ2 interrupt and CEC_TX_RETRY_TIMEOUT_ST indicates the interrupt status. 0 = Disables CEC Receiver Transmitter Timeout Retry interrupt on INTRQ2 1 = Enables CEC Receiver Transmitter Timeout Retry interrupt on INTRQ2	
CEC_TX_ARBITRATION_LOST_MB2			R/W
0x4C	000000**0**0	INTRQ2 interrupt mask for CEC Transmitter Arbitration Lost interrupt. When set, the CEC Transmitter Arbitration Lost interrupt triggers the INTRQ2 interrupt and CEC_TX_ARBITRATION_LOST_ST indicates the interrupt status. 0 = Disables CEC Receiver Transmitter Arbitration Lost interrupt on INTRQ2 1 = Enables CEC Receiver Transmitter Arbitration Lost interrupt on INTRQ2	
CEC_TX_READY_MB2			R/W
0x4C	0000000**0**	INTRQ2 interrupt mask for CEC Transmitter Ready interrupt. When set, the CEC Transmitter Ready interrupt triggers the INTRQ2 interrupt and CEC_TX_RDY_ST indicates the interrupt status. 0 = Disables CEC Receiver Transmitter Ready interrupt on INTRQ2 1 = Enables CEC Receiver Transmitter Ready interrupt on INTRQ2	
CEC_RX_RDY2_MB1			R/W
0x4D	00**0**00000	INTRQ1 interrupt mask for CEC Receiver Buffer 2 Ready interrupt. When set, the CEC Receiver Buffer 2 Ready interrupt triggers the INTRQ1 interrupt and CEC_RX_RDY2_ST indicates the interrupt status. 0 = Disables CEC Receiver Buffer 2 Ready interrupt on INTRQ1 1 = Enables CEC Receiver Buffer 2 Ready interrupt on INTRQ1	
CEC_RX_RDY1_MB1			R/W
0x4D	000**0**0000	INTRQ1 interrupt mask for CEC Receiver Buffer 1 Ready interrupt. When set, the CEC Receiver Buffer 1 Ready interrupt triggers the INTRQ1 interrupt and CEC_RX_RDY1_ST indicates the interrupt status. 0 = Disables CEC Receiver Buffer 1 Ready interrupt on INTRQ1 1 = Enables CEC Receiver Buffer 1 Ready interrupt on INTRQ1	
CEC_RX_RDY0_MB1			R/W
0x4D	0000**0**000	INTRQ1 interrupt mask for CEC Receiver Buffer 0 Ready interrupt. When set, the CEC Receiver Buffer 0 Ready interrupt triggers the INTRQ1 interrupt and CEC_RX_RDY0_ST indicates the interrupt status. 0 = Disables CEC Receiver Buffer 0 Ready interrupt on INTRQ1 1 = Enables CEC Receiver Buffer 0 Ready interrupt on INTRQ1	
CEC_TX_RETRY_TIMEOUT_MB1			R/W
0x4D	00000**0**00	INTRQ1 interrupt mask for CEC Transmitter Retry Timeout interrupt. When set, the CEC Transmitter Retry Timeout interrupt triggers the INTRQ1 interrupt and CEC_TX_RETRY_TIMEOUT_ST indicates the interrupt status. 0 = Disables CEC Receiver Transmitter Timeout Retry interrupt on INTRQ1 1 = Enables CEC Receiver Transmitter Timeout Retry interrupt on INTRQ1	

Reg	Bits	Description	
CEC_TX_ARBITRATION_LOST_MB1			R/W
0x4D	0000000<u>0</u>	INTRQ1 interrupt mask for CEC Transmitter Arbitration Lost interrupt. When set, the CEC Transmitter Arbitration Lost interrupt triggers the INTRQ1 interrupt and CEC_TX_ARBITRATION_LOST_ST indicates the interrupt status. 0 = Disables CEC Receiver Transmitter Arbitration Lost interrupt on INTRQ1 1 = Enables CEC Receiver Transmitter Arbitration Lost interrupt on INTRQ1	
CEC_TX_READY_MB1			R/W
0x4D	0000000<u>0</u>	INTRQ1 interrupt mask for CEC Transmitter Ready interrupt. When set, the CEC Transmitter Ready interrupt triggers the INTRQ1 interrupt and CEC_TX_RDY_ST indicates the interrupt status. 0 = Disables CEC Receiver Transmitter Ready interrupt on INTRQ1 1 = Enables CEC Receiver Transmitter Ready interrupt on INTRQ1	
CEC_INTERRUPT_BYTE[7:0]			R
0x4E	<u>00000000</u>	One of the 8 preprogrammed commands received. 00 = No change 01 = Opcode 1 received 02 = Opcode 2 received 04 = Opcode 3 received 08 = Opcode 4 received 10 = Opcode 5 received 20 = Opcode 6 received 40 = Opcode 7 received 80 = Opcode 8 received	
CEC_INTERRUPT_BYTE_ST[7:0]			R
0x4F	<u>00000000</u>	0 = No change 1 = one of the 8 opcodes received	
CEC_INTERRUPT_BYTE_CLR[7:0]			SC
0x50	<u>00000000</u>	0 = Does not clear 1 = Clears CEC_INTERRUPT_BYTE_ST	
CEC_INTERRUPT_BYTE_MB2[7:0]			R/W
0x51	<u>00000000</u>	INTRQ2 interrupt mask for CEC Interrupt Byte interrupt. When set, the CEC Interrupt Byte interrupt triggers the INTRQ2 interrupt and CEC_INTERRUPT_BYTE_ST indicates the interrupt status 0 = Disables CEC Interrupt Byte interrupt on INTRQ2 1 = Enables CEC Interrupt Byte interrupt on INTRQ2	
CEC_INTERRUPT_BYTE_MB1[7:0]			R/W
0x52	<u>00000000</u>	INTRQ1 interrupt mask for CEC Interrupt Byte interrupt. When set, the CEC Interrupt Byte interrupt triggers the INTRQ1 interrupt and CEC_INTERRUPT_BYTE_ST indicates the interrupt status 0 = Disables CEC Interrupt Byte interrupt on INTRQ1 1 = Enables CEC Interrupt Byte interrupt on INTRQ1	
TRI_SLICE[7:0]			R
0x53	<u>00000000</u>	Raw status of interrupts corresponding to the diagnostic pins. Bit 7 corresponds to DIAG1, bit 5 to DIAG2, bit 3 to DIAG3 and bit 1 to DIAG4. 00000000 = All voltages at diagnostic pins are lower than their respective slicers xxxxxx1x = DIAG4 pin voltage is higher than upper level slicer voltage xxxx1xxx = DIAG3 pin voltage is higher than upper level slicer voltage xx1xxxxx = DIAG2 pin voltage is higher than upper level slicer voltage 1xxxxxxx = DIAG1 pin voltage is higher than upper level slicer voltage	
TRI_SLICE_ST[7:0]			R
0x54	<u>00000000</u>	Latched status of interrupts corresponding to the diagnostic pins. The bits in this register are set if the corresponding interrupts have been unmasked, and the corresponding diagnostic slicers have flagged an interrupt. Once they have been set, they remain high until they have been cleared. 00000000 = All voltages at diagnostic pins are lower than their respective slicers xxxxxx1x = DIAG4 pin voltage is higher than upper level slicer voltage xxxx1xxx = DIAG3 pin voltage is higher than upper level slicer voltage xx1xxxxx = DIAG2 pin voltage is higher than upper level slicer voltage 1xxxxxxx = DIAG1 pin voltage is higher than upper level slicer voltage	
TRI_SLICE_CLR[7:0]			SC
0x55	<u>00000000</u>	Clear bits for the diagnostic slicer interrupts 00000000 = Do not clear any interrupt xxxxxx1x = Clear DIAG4 upper slicer interrupt xxxx1xxx = Clear DIAG3 upper slicer interrupt xx1xxxxx = Clear DIAG2 upper slicer interrupt 1xxxxxxx = Clear DIAG1 upper slicer interrupt	

Reg	Bits	Description	
TRI_SLICE_MB2[7:0]			R/W
0x56	00000000	INTRQ2 interrupt masks for the diagnostic slicer interrupts. 00000000 = Disable all diagnostic slicer interrupts for INTRQ2 xxxxxx1x = Enable DIAG4 upper slicer interrupt for INTRQ2 xxxx1xxx = Enable DIAG3 upper slicer interrupt for INTRQ2 xx1xxxxx = Enable DIAG2 upper slicer interrupt for INTRQ2 1xxxxxxx = Enable DIAG1 upper slicer interrupt for INTRQ2	
TRI_SLICE_MB1[7:0]			R/W
0x57	00000000	INTRQ1 interrupt masks for the diagnostic slicer interrupts. 00000000 = Disable all diagnostic slicer interrupts for INTRQ1 xxxxxx1x = Enable DIAG4 upper slicer interrupt for INTRQ1 xxxx1xxx = Enable DIAG3 upper slicer interrupt for INTRQ1 xx1xxxxx = Enable DIAG2 upper slicer interrupt for INTRQ1 1xxxxxxx = Enable DIAG1 upper slicer interrupt for INTRQ1	
ISRC2_PCKT_RAW			R
0x67	00000000	Raw status signal of International Standard Recording Code 2 (ISRC2) Packet detection signal. 0 = No ISRC2 packets received since the last HDMI/MHL packet detection reset. 1 = ISRC2 packets have been received. This bit resets to zero after an HDMI/MHL packet detection reset or upon writing to ISRC2_PACKET_ID.	
ISRC1_PCKT_RAW			R
0x67	00000000	Raw status signal of International Standard Recording Code 1 (ISRC1) Packet detection signal. 0 = No ISRC1 packets received since the last HDMI/MHL packet detection reset. 1 = ISRC1 packets have been received. This bit resets to zero after an HDMI/MHL packet detection reset or upon writing to ISRC1_PACKET_ID.	
ACP_PCKT_RAW			R
0x67	00000000	Raw status signal of audio content protection (ACP) packet detection signal. 0 = No ACP packet received within the last 600 ms or since the last HDMI/MHL packet detection reset. 1 = ACP packets have been received within the last 600 ms. This bit resets to zero after an HDMI/MHL packet detection reset or upon writing to ACP_PACKET_ID.	
VS_INFO_RAW			R
0x67	00000000	Raw status signal of vendor specific InfoFrame detection signal. 0 = No new vendor specific InfoFrame has been received since the last HDMI/MHL packet detection reset. 1 = A new vendor specific InfoFrame has been received. This bit resets to zero after an HDMI/MHL packet detection reset or upon writing to VS_PACKET_ID.	
MS_INFO_RAW			R
0x67	00000000	Raw status signal of MPEG Source InfoFrame detection signal. 0 = No source product description InfoFrame received within the last three VS signals or since the last HDMI/MHL packet detection reset. 1 = MPEG Source InfoFrame received. This bit resets to zero after an HDMI/MHL packet detection reset or upon writing to MS_PACKET_ID.	
SPD_INFO_RAW			R
0x67	00000000	Raw status of source product description (SPD) InfoFrame detected signal. 0 = No source product description InfoFrame received since the last HDMI/MHL packet detection reset. 1 = Source product description InfoFrame received. This bit resets to zero after an HDMI/MHL packet detection reset or upon writing to SPD_PACKET_ID.	
AUDIO_INFO_RAW			R
0x67	00000000	Raw status of audio InfoFrame detected signal. 0 = No AVI InfoFrame has been received within the last three VS signals or since the last HDMI/MHL packet detection reset. 1 = An Audio InfoFrame has been received within the last three VS signals. This bit resets to zero on the fourth VS leading edge following an Audio InfoFrame, after an HDMI/MHL packet detection reset or upon writing to AUD_PACKET_ID.	
AVI_INFO_RAW			R
0x67	00000000	Raw status of AVI InfoFrame detected signal. This bit is set to one when an AVI InfoFrame is received and is reset to zero if no AVI InfoFrame is received for more than 7 VS signals (on the eighth VS leading edge following the last received AVI InfoFrame), after an HDMI/MHL packet detection reset or upon writing to AVI_PACKET_ID. 0 = No AVI InfoFrame has been received within the last seven VS signals or since the last HDMI/MHL packet detection reset. 1 = An AVI InfoFrame has been received within the last seven VS signals.	

Reg	Bits	Description	
ISRC2_PCKT_ST			R
0x68	0xxxxxxx	Latched status of ISRC2 Packet detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via ISRC2_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No interrupt generated from this register 1 = ISRC2_PCKT_RAW has changed. Interrupt has been generated.	
ISRC1_PCKT_ST			R
0x68	x0xxxxxx	Latched status of ISRC1 Packet detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via ISRC1_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No interrupt generated from this register 1 = ISRC1_PCKT_RAW has changed. Interrupt has been generated.	
ACP_PCKT_ST			R
0x68	xx0xxxxx	Latched status of ACP packet detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via ACP_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No interrupt generated from this register 1 = ACP_PCKT_RAW has changed. Interrupt has been generated.	
VS_INFO_ST			R
0x68	xxx0xxxx	Latched status of Vendor Specific InfoFrame detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via VS_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No interrupt generated from this register 1 = VS_INFO_RAW has changed. Interrupt has been generated.	
MS_INFO_ST			R
0x68	xxxx0xxx	Latched status of MPEG Source InfoFrame detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via MS_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No interrupt generated from this register 1 = MS_INFO_RAW has changed. Interrupt has been generated.	
SPD_INFO_ST			R
0x68	xxxxx0xx	Latched status of SPD InfoFrame detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via SPD_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No interrupt generated from this register 1 = SPD_INFO_RAW has changed. Interrupt has been generated.	
AUDIO_INFO_ST			R
0x68	xxxxxx0x	Latched status of Audio InfoFrame detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via AUDIO_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No interrupt generated from this register 1 = AUDIO_INFO_RAW has changed. Interrupt has been generated.	
AVI_INFO_ST			R
0x68	xxxxxxx0	Latched status of AVI_INFO_RAW signal. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. Once set, this bit remains high until the interrupt has been cleared via AVI_INFO_CLR. 0 = AVI_INFO_RAW has not changed state 1 = AVI_INFO_RAW has changed state	
ISRC2_PCKT_CLR			SC
0x69	0xxxxxxx	Clear bit for ISRC2 Packet detection interrupt signal. 0 = Does not clear ISRC2_PCKT_ST 1 = Clears ISRC2_PCKT_ST	
ISRC1_PCKT_CLR			SC
0x69	x0xxxxxx	Clear bit for ISRC1 Packet detection interrupt signal. 0 = Does not clear ISRC1_INFO_ST 1 = Clears ISRC1_INFO_ST	
ACP_PCKT_CLR			SC
0x69	xx0xxxxx	Clear bit for ACP packet detected interrupt signal. 0 = Does not clear ACP_INFO_ST 1 = Clears ACP_INFO_ST	
VS_INFO_CLR			SC
0x69	xxx0xxxx	Clear bit for Vendor Specific InfoFrame interrupt signal. 0 = Does not clear VS_INFO_ST 1 = Clears VS_INFO_ST	

Reg	Bits	Description	
MS_INFO_CLR			SC
0x69	0000**0**000	Clear bit for MPEG Source InfoFrame interrupt signal. 0 = Does not clear MS_INFO_ST 1 = Clears MS_INFO_ST	
SPD_INFO_CLR			SC
0x69	00000**0**00	Clear bit for SPD InfoFrame interrupt signal. 0 = Does not clear SPD_INFO_ST 1 = Clears SPD_INFO_ST	
AUDIO_INFO_CLR			SC
0x69	000000**0**0	Clear bit for Audio InfoFrame interrupt signal. 0 = Does not clear AUDIO_INFO_ST 1 = Clears AUDIO_INFO_ST	
AVI_INFO_CLR			SC
0x69	0000000**0**	Clear bit for AVI_INFO_RAW and AVI_INFO_ST bits. 0 = No function 1 = Clear AVI_INFO_RAW and AVI_INFO_ST	
ISRC2_PCKT_MB2			R/W
0x6A	**0**0000000	INTRQ2 interrupt mask for ISRC2 Packet detection interrupt. When set, the ISRC2 Packet detection interrupt triggers the INTRQ2 interrupt and ISRC2_INFO_ST indicates the interrupt status. 0 = Disables ISRC2 InfoFrame detection interrupt for INTRQ2 1 = Enables ISRC2 InfoFrame detection interrupt for INTRQ2	
ISRC1_PCKT_MB2			R/W
0x6A	0**0**000000	INTRQ2 interrupt mask for ISRC1 Packet detection interrupt. When set, the ISRC1 Packet detection interrupt triggers the INTRQ2 interrupt and ISRC1_INFO_ST indicates the interrupt status. 0 = Disables ISRC1 InfoFrame detection interrupt for INTRQ2 1 = Enables ISRC1 InfoFrame detection interrupt for INTRQ2	
ACP_PCKT_MB2			R/W
0x6A	00**0**00000	INTRQ2 interrupt mask for ACP packet detection interrupt. When set, the ACP InfoFrame detection interrupt triggers the INTRQ2 interrupt and ACP_INFO_ST indicates the interrupt status. 0 = Disables Audio Content Protection InfoFrame detection interrupt for INTRQ2 1 = Enables Audio Content Protection InfoFrame detection interrupt for INTRQ2	
VS_INFO_MB2			R/W
0x6A	000**0**0000	INTRQ2 interrupt mask for Vendor Specific InfoFrame detection interrupt. When set, the Vendor Specific InfoFrame detection interrupt triggers the INTRQ2 interrupt and VS_INFO_ST indicates the interrupt status. 0 = Disables Vendor Specific InfoFrame detection interrupt for INTRQ2 1 = Enables Vendor Specific InfoFrame detection interrupt for INTRQ2	
MS_INFO_MB2			R/W
0x6A	0000**0**000	INTRQ2 interrupt mask for MPEG source InfoFrame detection interrupt. When set, the MPEG Source InfoFrame detection interrupt triggers the INTRQ2 interrupt and MS_INFO_ST indicates the interrupt status. 0 = Disables MPEG source Info frame detection interrupt for INTRQ2 1 = Enables MPEG source Info frame detection interrupt for INTRQ2	
SPD_INFO_MB2			R/W
0x6A	00000**0**00	INTRQ2 interrupt mask for SPD InfoFrame detection interrupt. When set, the SPD InfoFrame detection interrupt triggers the INTRQ2 interrupt and SPD_INFO_ST indicates the interrupt status. 0 = Disables SPD Info frame detection interrupt for INTRQ2 1 = Enables SPD Info frame detection interrupt for INTRQ2	
AUDIO_INFO_MB2			R/W
0x6A	000000**0**0	INTRQ2 interrupt mask for Audio InfoFrame detection interrupt. When set, the Audio InfoFrame detection interrupt triggers the INTRQ2 interrupt and AVI_INFO_ST indicates the interrupt status. 0 = Disables AUDIO Info frame detection interrupt for INTRQ2 1 = Enables AUDIO Info frame detection interrupt for INTRQ2	
AVI_INFO_MB2			R/W
0x6A	0000000**0**	INTRQ2 interrupt mask for AVI InfoFrame detection interrupt. When set, an AVI InfoFrame detection event causes AVI_INFO_ST to be set and an interrupt is generated on INTRQ2. 0 = Disables AVI Info frame detection interrupt for INTRQ2 1 = Enables AVI Info frame detection interrupt for INTRQ2	

Reg	Bits	Description	
ISRC2_PCKT_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for ISRC2 InfoFrame detection interrupt. When set, the ISRC2 InfoFrame detection interrupt triggers the INTRQ1 interrupt and ISRC2_INFO_ST indicates the interrupt status. 0 = Disables ISRC2 Packet detection interrupt for INTRQ1 1 = Enables ISRC2 Packet detection interrupt for INTRQ1	
ISRC1_PCKT_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for ISRC1 InfoFrame detection interrupt. When set, the ISRC1 InfoFrame detection interrupt triggers the INTRQ1 interrupt and ISRC1_INFO_ST indicates the interrupt status. 0 = Disables ISRC1 InfoFrame detection interrupt for INTRQ1 1 = Enables ISRC1 InfoFrame detection interrupt for INTRQ1	
ACP_PCKT_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for ACP packet detection interrupt. When set, the ACP packet detection interrupt triggers the INTRQ1 interrupt and ACP_INFO_ST indicates the interrupt status. 0 = Disables ACP InfoFrame detection interrupt for INTRQ1 1 = Enables ACP InfoFrame detection interrupt for INTRQ1	
VS_INFO_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for Vendor Specific InfoFrame detection interrupt. When set, the Vendor Specific InfoFrame detection interrupt triggers the INTRQ1 interrupt and VS_INFO_ST indicates the interrupt status. 0 = Disables Vendor Specific InfoFrame detection interrupt for INTRQ1 1 = Enables Vendor Specific InfoFrame detection interrupt for INTRQ1	
MS_INFO_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for MPEG source InfoFrame detection interrupt. When set, the MPEG source InfoFrame detection interrupt triggers the INTRQ1 interrupt and MS_INFO_ST indicates the interrupt status. 0 = Disables MPEG source InfoFrame detection interrupt for INTRQ1 1 = Enables MPEG source InfoFrame detection interrupt for INTRQ1	
SPD_INFO_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for SPD InfoFrame detection interrupt. When set, the SPD InfoFrame detection interrupt triggers the INTRQ1 interrupt and SPD_INFO_ST indicates the interrupt status. 0 = Disables SPD InfoFrame detection interrupt for INTRQ1 1 = Enables SPD InfoFrame detection interrupt for INTRQ1	
AUDIO_INFO_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for Audio InfoFrame detection interrupt. When set, the Audio InfoFrame detection interrupt triggers the INTRQ1 interrupt and AVI_INFO_ST indicates the interrupt status. 0 = Disables AUDIO InfoFrame detection interrupt for INTRQ1 1 = Enables AUDIO InfoFrame detection interrupt for INTRQ1	
AVI_INFO_MB1			R/W
0x6B	00000000	INTRQ1 interrupt mask for AVI InfoFrame detection interrupt. When set an AVI InfoFrame detection event causes AVI_INFO_ST to be set and an interrupt is generated on INTRQ1. 0 = Disables AVI InfoFrame detection interrupt for INTRQ1 1 = Enables AVI InfoFrame detection interrupt for INTRQ1	
CS_DATA_VALID_RAW			R
0x6C	00000000	Raw status signal of Channel Status Data Valid signal. 0 = Channel status data is not valid 1 = Channel status data is valid	
INTERNAL_MUTE_RAW			R
0x6C	00000000	Raw status signal of Internal Mute signal. 0 = Audio is not muted 1 = Audio is muted	
AV_MUTE_RAW			R
0x6C	00000000	Raw status signal of AV Mute detection signal. 0 = No AV mute raw received since last HDMI/MHL reset condition 1 = AV mute received	
AUDIO_CH_MD_RAW			R
0x6C	00000000	Raw status signal indicating the layout value of the audio packets has changed 0 = The layout value of the audio packets has not changed 1 = The layout value of the audio packets has changed	

Reg	Bits	Description	
HDMI_MODE_RAW			R
0x6C	0000**0**000	Raw status signal of HDMI Mode signal. 0 = DVI is being received 1 = HDMI or MHL is being received	
GEN_CTL_PCKT_RAW			R
0x6C	00000**0**00	Raw status signal of General Control Packet detection signal. 0 = No general control packets received since the last HDMI/MHL reset condition 1 = General control packets received	
AUDIO_C_PCKT_RAW			R
0x6C	000000**0**0	Raw status signal of Audio Clock Regeneration Packet detection signal. 0 = No audio clock regeneration packets received since the last HDMI/MHL reset condition 1 = Audio clock regeneration packets received	
GAMUT_MDATA_RAW			R
0x6C	0000000**0**	Raw status signal of Gamut Metadata Packet detection signal. 0 = No Gamut Metadata packet has been received in the last video frame or since the last HDMI/MHL packet detection reset. 1 = A Gamut Metadata packet has been received in the last video frame. This bit resets to zero after an HDMI/MHL packet detection reset or upon writing to GAMUT_PACKET_ID.	
CS_DATA_VALID_ST			R
0x6D	**0**0000000	Latched status of Channel Status Data Valid interrupt signal. Once set, this bit remains high until the interrupt has been cleared via CS_DATA_VALID_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = CS_DATA_VALID_RAW has not changed. An interrupt has not been generated. 1 = CS_DATA_VALID_RAW has changed. An interrupt has been generated.	
INTERNAL_MUTE_ST			R
0x6D	0**0**000000	Latched status of Internal Mute interrupt signal. Once set, this bit remains high until the interrupt has been cleared via INTERNAL_MUTE_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = INTERNAL_MUTE_RAW has not changed. An interrupt has not been generated. 1 = INTERNAL_MUTE_RAW has changed. An interrupt has been generated.	
AV_MUTE_ST			R
0x6D	00**0**00000	Latched status of AV Mute detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via AV_MUTE_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = AV_MUTE_RAW has not changed. An interrupt has not been generated. 1 = AV_MUTE_RAW has changed. An interrupt has been generated.	
AUDIO_CH_MD_ST			R
0x6D	000**0**0000	Latched status of Audio Channel mode interrupt signal. Once set, this bit remains high until the interrupt has been cleared via AUDIO_CH_MD_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = AUDIO_CH_MD_RAW has not changed. An interrupt has not been generated. 1 = AUDIO_MODE_CHNG_RAW has changed. An interrupt has been generated.	
HDMI_MODE_ST			R
0x6D	0000**0**000	Latched status of HDMI Mode interrupt signal. Once set, this bit remains high until the interrupt has been cleared via HDMI_MODE_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = HDMI_MODE_RAW has not changed. An interrupt has not been generated. 1 = HDMI_MODE_RAW has changed. An interrupt has been generated.	
GEN_CTL_PCKT_ST			R
0x6D	00000**0**00	Latched status of General Control Packet interrupt signal. Once set, this bit remains high until the interrupt has been cleared via GEN_CTL_PCKT_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = GEN_CTL_PCKT_RAW has not changed. Interrupt has not been generated from this register. 1 = GEN_CTL_PCKT_RAW has changed. Interrupt has been generated from this register.	
AUDIO_C_PCKT_ST			R
0x6D	000000**0**0	Latched status of Audio Clock Regeneration Packet interrupt signal. Once set, this bit remains high until the interrupt has been cleared via AUDIO_PCKT_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = AUDIO_C_PCKT_RAW has not changed. Interrupt has not been generated from this register 1 = AUDIO_C_PCKT_RAW has changed. Interrupt has been generated from this register.	

Reg	Bits	Description	
GAMUT_MDATA_ST			R
0x6D	0000000**0**	Latched status of Gamut Metadata Packet detected interrupt signal. Once set, this bit remains high until the interrupt has been cleared via GAMUT_MDATA_PCKT_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = GAMUT_MDATA_RAW has not changed. Interrupt has not been generated from this register 1 = GAMUT_MDATA_RAW has changed. Interrupt has been generated from this register.	
CS_DATA_VALID_CLR			SC
0x6E	**0**0000000	Clear bit for Channel Status Data Valid interrupt signal. 0 = Does not clear 1 = Clears CS_DATA_VALID_ST	
INTERNAL_MUTE_CLR			SC
0x6E	0**0**000000	Clear bit for Internal Mute interrupt signal. 0 = Does not clear INTERNAL_MUTE_ST 1 = Clears INTERNAL_MUTE_ST	
AV_MUTE_CLR			SC
0x6E	00**0**00000	Clear bit for AV Mute Detected interrupt signal. 0 = Does not clear AV_MUTE_ST 1 = Clears AV_MUTE_ST	
AUDIO_CH_MD_CLR			SC
0x6E	0000**0**000	Clear bit for Audio Channel mode interrupt signal. 0 = Does not clear AUDIO_CH_MD_ST 1 = Clears AUDIO_CH_MD_ST	
HDMI_MODE_CLR			SC
0x6E	0000**0**000	Clear bit for HDMI Mode interrupt signal. 0 = Does not clear HDMI_MODE_ST 1 = Clears HDMI_MODE_ST	
GEN_CTL_PCKT_CLR			SC
0x6E	00000**0**00	Clear bit for General Control Packet detection interrupt signal. 0 = Does not clear GEN_CTL_PCKT_ST 1 = Clears GEN_CTL_PCKT_ST	
AUDIO_C_PCKT_CLR			SC
0x6E	000000**0**0	Clear bit for Audio Clock Regeneration Packet detection interrupt signal. 0 = Does not clear AUDIO_C_PCKT_ST 1 = Clears AUDIO_C_PCKT_ST	
GAMUT_MDATA_CLR			SC
0x6E	0000000**0**	Clear bit for Gamut Metadata Packet detection interrupt signal. 0 = Does not clear GAMUT_MDATA_ST 1 = Clears GAMUT_MDATA_ST	
CS_DATA_VALID_MB2			R/W
0x6F	**0**0000000	INTRQ2 interrupt mask for Channel Status Data Valid interrupt. When set, the Channel Status Data Valid interrupt triggers the INTRQ2 interrupt and CS_DATA_VALID_ST indicates the interrupt status. 0 = Disables Channel Status Data Valid interrupt for INTRQ2 1 = Enables Channel Status Data Valid interrupt for INTRQ2	
INTERNAL_MUTE_MB2			R/W
0x6F	0**0**000000	INTRQ2 interrupt mask for Internal Mute interrupt. When set, the Internal Mute interrupt triggers the INTRQ2 interrupt and INTERNAL_MUTE_ST indicates the interrupt status. 0 = Disables Internal Mute interrupt for INTRQ2 1 = Enables Internal Mute interrupt for INTRQ2	
AV_MUTE_MB2			R/W
0x6F	00**0**00000	INTRQ2 interrupt mask for AV Mute detected interrupt. When set, the AV Mute detected interrupt triggers the INTRQ2 interrupt and AV_MUTE_ST indicates the interrupt status. 0 = Disables AV Mute detected interrupt for INTRQ2 1 = Enables AV Mute detected interrupt for INTRQ2	

Reg	Bits	Description	
AUDIO_CH_MD_MB2			R/W
0x6F	0000**0**0000	INTRQ2 interrupt mask for Audio Channel mode interrupt. When set, the Audio Channel mode interrupt triggers the INTRQ2 interrupt and AUDIO_CH_MD_ST indicates the interrupt status. 0 = Disables Audio Channel Mode interrupt for INTRQ2 1 = Enables Audio Channel Mode interrupt for INTRQ2	
HDMI_MODE_MB2			R/W
0x6F	0000**0**000	INTRQ2 interrupt mask for HDMI Mode interrupt. When set, the HDMI Mode interrupt triggers the INTRQ2 interrupt and HDMI_MODE_ST indicates the interrupt status. 0 = Disables HDMI Mode interrupt for INTRQ2 1 = Enables HDMI Mode interrupt for INTRQ2	
GEN_CTL_PCKT_MB2			R/W
0x6F	00000**0**00	INTRQ2 interrupt mask for General Control Packet detection interrupt. When set, the General Control Packet detection interrupt triggers the INTRQ2 interrupt and AUDIO_C_PCKT_ST indicates the interrupt status. 0 = Disables General Control Packet detection interrupt for INTRQ2 1 = Enables General Control Packet detection interrupt for INTRQ2	
AUDIO_C_PCKT_MB2			R/W
0x6F	000000**0**0	INTRQ2 interrupt mask for Audio Clock Regeneration Packet detection interrupt. When set, the Audio Clock Regeneration Packet detection interrupt triggers the INTRQ2 interrupt and AUDIO_C_PCKT_ST indicates the interrupt status. 0 = Disables Audio Clock Regeneration Packet detection interrupt for INTRQ2 1 = Enables Audio Clock Regeneration Packet detection interrupt for INTRQ2	
GAMUT_MDATA_MB2			R/W
0x6F	0000000**0**	INTRQ2 interrupt mask for Gamut Metadata Packet detection interrupt. When set, the Gamut Metadata Packet detection interrupt triggers the INTRQ2 interrupt and GAMUT_MDATA_ST indicates the interrupt status. 0 = Disables Gamut Metadata Packet detection interrupt for INTRQ2 1 = Enables Gamut Metadata Packet detection interrupt for INTRQ2	
CS_DATA_VALID_MB1			R/W
0x70	**0**0000000	INTRQ1 interrupt mask for Channel Status Data Valid interrupt. When set, the Channel Status Data Valid interrupt triggers the INTRQ1 interrupt and CS_DATA_VALID_ST indicates the interrupt status. 0 = Disables Channel Status Data Valid interrupt for INTRQ1 1 = Enables Channel Status Data Valid interrupt for INTRQ1	
INTERNAL_MUTE_MB1			R/W
0x70	0**0**000000	INTRQ1 interrupt mask for Internal Mute interrupt. When set, the Internal Mute interrupt triggers the INTRQ1 interrupt and INTERNAL_MUTE_ST indicates the interrupt status. 0 = Disables AV Mute detected interrupt for INTRQ1 1 = Enables AV Mute detected interrupt for INTRQ1	
AV_MUTE_MB1			R/W
0x70	00**0**00000	INTRQ1 interrupt mask for AV Mute detected interrupt. When set, the AV Mute detected interrupt triggers the INTRQ1 interrupt and AV_MUTE_ST indicates the interrupt status. 0 = Disables AV Mute detected interrupt for INTRQ1 1 = Enables AV Mute detected interrupt for INTRQ1	
AUDIO_CH_MD_MB1			R/W
0x70	0000**0**0000	INTRQ1 interrupt mask for Audio Channel mode interrupt. When set, the Audio Channel mode interrupt triggers the INTRQ1 interrupt and AUDIO_CH_MD_ST indicates the interrupt status. 0 = Disables Audio Channel Mode interrupt for INTRQ1 1 = Enables Audio Channel Mode interrupt for INTRQ1	
HDMI_MODE_MB1			R/W
0x70	0000**0**000	INTRQ1 interrupt mask for HDMI Mode detection interrupt. When set, the HDMI Mode interrupt triggers the INTRQ1 interrupt and HDMI_MODE_ST indicates the interrupt status. 0 = Disables HDMI Mode interrupt for INTRQ1 1 = Enables HDMI Mode interrupt for INTRQ1	
GEN_CTL_PCKT_MB1			R/W
0x70	00000**0**00	INTRQ1 interrupt mask for General Control Packet detection interrupt. When set, the General Control Packet detection interrupt triggers the INTRQ1 interrupt and GEN_CTL_PCKT_ST indicates the interrupt status. 0 = Disables General Control Packet detection interrupt for INTRQ1 1 = Enables General Control Packet detection interrupt for INTRQ1	

Reg	Bits	Description	
AUDIO_C_PCKT_MB1			R/W
0x70	0000000**0**	INTRQ1 interrupt mask for Audio Clock Regeneration Packet detection interrupt. When set, the Audio Clock Regeneration Packet detection interrupt triggers the INTRQ1 interrupt and AUDIO_C_PCKT_ST indicates the interrupt status. 0 = Disables Audio Clock Regeneration Packet detection interrupt for INTRQ1 1 = Enables Audio Clock Regeneration Packet detection interrupt for INTRQ1	
GAMUT_MDATA_MB1			R/W
0x70	0000000**0**	INTRQ1 interrupt mask for Gamut Metadata Packet detection interrupt. When set, the Gamut Metadata Packet detection interrupt triggers the INTRQ1 interrupt and GAMUT_MDATA_ST indicates the interrupt status. 0 = Disables Gamut Metadata Packet detection interrupt for INTRQ1 1 = Enables Gamut Metadata Packet detection interrupt for INTRQ1	
TMDSPLL_LCK_A_RAW			R
0x71	**0**0000000	A readback to indicate the raw status of the port A TMDS PLL lock signal. 0 = TMDS PLL on Port A is not locked 1 = TMDS PLL on Port A is locked to the incoming clock	
CABLE_DET_A_RAW			R
0x71	0**0**000000	Raw status of Port A 5 V cable detection signal in HDMI input mode. 0 = No cable detected on Port A 1 = Cable detected on Port A (High level on RXA_5V)	
HDMI_ENCRPT_A_RAW			R
0x71	00**0**00000	Raw status of Port A Encryption detection signal. 0 = Current frame in port A is not encrypted 1 = Current frame in port A is encrypted	
TMDS_CLK_A_RAW			R
0x71	0000**0**000	Raw status of Port A TMDS Clock detection signal. 0 = No TMDS clock detected on port A 1 = TMDS clock detected on port A	
VIDEO_3D_RAW			R
0x71	00000**0**00	Raw status of the Video 3D signal. 0 = Video 3D not detected 1 = Video 3D detected	
V_LOCKED_RAW			R
0x71	000000**0**0	Raw status of the Vertical Filter Locked signal. 0 = Vertical filter has not locked and vertical sync parameters are not valid 1 = Vertical filter has locked and vertical sync parameters are valid	
DE_REGEN_LCK_RAW			R
0x71	0000000**0**	Raw status of the DE regeneration lock signal. 0 = DE regeneration block has not been locked 1 = DE regeneration block has been locked to the incoming DE signal	
TMDSPLL_LCK_A_ST			R
0x72	**0**0000000	Latched status of Port A TMDS PLL Lock interrupt signal. Once set, this bit remains high until the interrupt has been cleared via TMDSPLL_LCK_A_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = TMDSPLL_LCK_A_RAW has not changed. An interrupt has not been generated. 1 = TMDSPLL_LCK_A_RAW has changed. An interrupt has been generated.	
CABLE_DET_A_ST			R
0x72	0**0**000000	Latched status for Port A 5 V cable detection interrupt signal. Once set, this bit remains high until the interrupt has been cleared via CABLE_DET_A_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = CABLE_DET_A_RAW has not changed. Interrupt has not been generated from this register. 1 = CABLE_DET_A_RAW has changed. Interrupt has been generated from this register.	
HDMI_ENCRPT_A_ST			R
0x72	00**0**00000	Latched status for Port A Encryption detection interrupt signal. Once set, this bit remains high until the interrupt has been cleared via HDMI_ENCRPT_A_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = HDMI_ENCRPT_A_RAW has not changed. An interrupt has not been generated. 1 = HDMI_ENCRPT_A_RAW has changed. An interrupt has been generated.	

Reg	Bits	Description	
TMDS_CLK_A_ST			R
0x72	0000**0**000	Latched status of Port A TMDS Clock Detection interrupt signal. Once set, this bit remains high until the interrupt has been cleared via TMDS_CLK_A_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = TMDS_CLK_A_RAW has not changed. An interrupt has not been generated. 1 = TMDS_CLK_A_RAW has changed. An interrupt has been generated.	
VIDEO_3D_ST			R
0x72	00000**0**00	Latched status for the Video 3D interrupt. Once set, this bit remains high until the interrupt has been cleared via VIDEO_3D_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit 0 = VIDEO_3D_RAW has not changed. An interrupt has not been generated. 1 = VIDEO_3D_RAW has changed. An interrupt has been generated.	
V_LOCKED_ST			R
0x72	000000**0**0	Latched status for the Vertical Filter Locked interrupt. Once set, this bit remains high until the interrupt has been cleared via V_LOCKED_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit 0 = V_LOCKED_RAW has not changed. An interrupt has not been generated. 1 = V_LOCKED_RAW has changed. An interrupt has been generated.	
DE_REGEN_LCK_ST			R
0x72	0000000**0**	Latched status for DE Regeneration Lock interrupt signal. Once set, this bit remains high until the interrupt has been cleared via DE_REGEN_LCK_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit 0 = DE_REGEN_LCK_RAW has not changed. An interrupt has not been generated. 1 = DE_REGEN_LCK_RAW has changed. An interrupt has been generated.	
TMDSPLL_LCK_A_CLR			SC
0x73	**0**0000000	Clear bit for Port A TMDS PLL Lock interrupt signal. 0 = Does not clear TMDSPLL_LCK_A_ST 1 = Clears TMDSPLL_LCK_A_ST	
CABLE_DET_A_CLR			SC
0x73	0**0**000000	Clear bit for Port A 5 V cable detection interrupt signal. 0 = Does not clear 1 = Clears CABLE_DET_A_ST	
HDMI_ENCRPT_A_CLR			SC
0x73	00**0**00000	Clear bit for Port A Encryption detection interrupt signal. 0 = Does not clear HDMI_ENCRPT_A_ST 1 = Clears HDMI_ENCRPT_A_ST	
TMDS_CLK_A_CLR			SC
0x73	0000**0**000	Clear bit for Port A TMDS Clock Detection interrupt signal. 0 = Does not clear TMDS_CLK_A_ST 1 = Clears TMDS_CLK_A_ST	
VIDEO_3D_CLR			SC
0x73	00000**0**00	Clear bit for Video 3D Interrupt 0 = Does not clear VIDEO_3D_ST 1 = Clears VIDEO_3D_ST	
V_LOCKED_CLR			SC
0x73	000000**0**0	Clear bit for Vertical Filter Locked Interrupt 0 = Does not clear V_LOCKED_ST 1 = Clears V_LOCKED_ST	
DE_REGEN_LCK_CLR			SC
0x73	0000000**0**	Clear bit for DE Regeneration Lock interrupt signal. 0 = Does not clear DE_REGEN_LCK_ST 1 = Clears DE_REGEN_LCK_ST	
TMDSPLL_LCK_A_MB2			R/W
0x74	**0**0000000	INTRQ2 interrupt mask for Port A TMDS PLL Lock interrupt. When set, the Port A TMDS PLL Lock interrupt triggers the INTRQ2 interrupt and TMDSPLL_LCK_A_ST indicates the interrupt status. 0 = Disables Port A TMDSPLL Lock interrupt for INTRQ2 1 = Enables Port A TMDSPLL Lock interrupt for INTRQ2	

Reg	Bits	Description
CABLE_DET_A_MB2		R/W
0x74	00000000	INTRQ2 interrupt mask for Port A 5 V cable detection interrupt. When set, the Port A 5 V cable detection interrupt triggers the INTRQ2 interrupt and CABLE_DET_A_ST indicates the interrupt status. 0 = Disables Port A 5 V Cable Detection interrupt for INTRQ2 1 = Enables Port A 5 V Cable Detection interrupt for INTRQ2
HDMI_ENCRPT_A_MB2		R/W
0x74	00000000	INTRQ2 interrupt mask for Port A Encryption detection interrupt. When set, the Port A Encryption detection interrupt triggers the INTRQ2 interrupt and HDMI_ENCRPT_A_ST indicates the interrupt status. 0 = Disables Port A HDMI Encryption detection interrupt for INTRQ2 1 = Enables Port A HDMI Encryption detection interrupt for INTRQ2
TMDS_CLK_A_MB2		R/W
0x74	00000000	INTRQ2 interrupt mask for Port A TMDS Clock detection interrupt. When set, the Port A TMDS Clock detection interrupt triggers the INTRQ2 interrupt and TMDS_CLK_A_ST indicates the interrupt status. 0 = Disables Port A TMDS Clock Detection interrupt for INTRQ2 1 = Enables Port A TMDS Clock Detection interrupt for INTRQ2
VIDEO_3D_MB2		R/W
0x74	00000000	INTRQ2 interrupt mask for Video 3D interrupt. When set, the Video 3D interrupt triggers the INTRQ2 interrupt and VIDEO_3D_ST indicates the interrupt status. 0 = Disables Video 3D interrupt on INTRQ2 1 = Enables Video 3D interrupt on INTRQ2
V_LOCKED_MB2		R/W
0x74	00000000	INTRQ2 interrupt mask for Vertical Filter Locked interrupt. When set, the Vertical Filter Locked interrupt triggers the INTRQ2 interrupt and V_LOCKED_ST indicates the interrupt status. 0 = Disables Vertical Filter Lock interrupt on INTRQ2 1 = Enables Vertical Filter Lock interrupt on INTRQ2
DE_REGEN_LCK_MB2		R/W
0x74	00000000	INTRQ2 interrupt mask for DE Regeneration Lock interrupt. When set, the DE Regeneration Lock interrupt triggers the INTRQ2 interrupt and DE_REGEN_LCK_ST indicates the interrupt status. 0 = Disables DE Regeneration Lock interrupt on INTRQ2 1 = Enables DE Regeneration Lock interrupt on INTRQ2
TMDSPLL_LCK_A_MB1		R/W
0x75	00000000	INTRQ1 interrupt mask for Port A TMDS PLL Lock interrupt. When set, the Port A TMDS PLL Lock interrupt triggers the INTRQ1 interrupt and TMDSPLL_LCK_A_ST indicates the interrupt status. 0 = Disables Port A TMDSPLL Lock interrupt for INTRQ1 1 = Enables Port A TMDSPLL Lock interrupt for INTRQ1
CABLE_DET_A_MB1		R/W
0x75	00000000	INTRQ1 interrupt mask for Port A 5 V cable detection interrupt. When set, the Port A 5 V cable detection interrupt triggers the INTRQ1 interrupt and CABLE_DET_A_ST indicates the interrupt status. 0 = Disables Port A 5 V Cable Detection interrupt for INTRQ1 1 = Enables Port A 5 V Cable Detection interrupt for INTRQ1
HDMI_ENCRPT_A_MB1		R/W
0x75	00000000	INTRQ1 interrupt mask for Port A Encryption detection interrupt. When set, the Port A Encryption detection interrupt triggers the INTRQ1 interrupt and HDMI_ENCRPT_A_ST indicates the interrupt status. 0 = Disables Port A HDMI Encryption detection interrupt for INTRQ1 1 = Enables Port A HDMI Encryption detection interrupt for INTRQ1
TMDS_CLK_A_MB1		R/W
0x75	00000000	INTRQ1 interrupt mask for Port A TMDS Clock detection interrupt. When set, the Port A TMDS Clock detection interrupt triggers the INTRQ1 interrupt and TMDS_CLK_A_ST indicates the interrupt status. 0 = Disables Port A TMDS Clock Detection interrupt for INTRQ1 1 = Enables Port A TMDS Clock Detection interrupt for INTRQ1
VIDEO_3D_MB1		R/W
0x75	00000000	INTRQ1 interrupt mask for Video 3D interrupt. When set, the Video 3D interrupt triggers the INTRQ1 interrupt and VIDEO_3D_ST indicates the interrupt status. 0 = Disables Video 3D interrupt on INTRQ1 1 = Enables Video 3D interrupt on INTRQ1

Reg	Bits	Description	
V_LOCKED_MB1			R/W
0x75	0000000**0**	INTRQ1 interrupt mask for Vertical Filter Locked interrupt. When set, the Vertical Filter Locked interrupt triggers the INTRQ1 interrupt and V_LOCKED_ST indicates the interrupt status. 0 = Disables Vertical Filter Lock interrupt on INTRQ1 1 = Enables Vertical Filter Lock interrupt on INTRQ1	
DE_REGEN_LCK_MB1			R/W
0x75	0000000**0**	INTRQ1 interrupt mask for DE Regeneration Lock interrupt. When set, the DE Regeneration Lock interrupt triggers the INTRQ1 interrupt and DE_REGEN_LCK_ST indicates the interrupt status. 0 = Disables DE Regeneration Lock interrupt on INTRQ1 1 = Enables DE Regeneration Lock interrupt on INTRQ1	
NEW_ISRC2_PCKT_RAW			R
0x80	**0**0000000	Status of the New ISRC2 interrupt signal. When set to 1, it indicates that an ISRC2 packet has been received with new contents. Once set, this bit remains high until it is cleared via NEW_ISRC2_PCKT_CLR. 0 = No new ISRC2 packet received 1 = ISRC2 packet with new content received	
NEW_ISRC1_PCKT_RAW			R
0x80	0**0**000000	Status of the New ISRC1 interrupt signal. When set to 1, it indicates that an ISRC1 packet has been received with new contents. Once set, this bit remains high until it is cleared via NEW_ISRC1_PCKT_CLR. 0 = No new ISRC1 packet received 1 = ISRC1 packet with new content received	
NEW_ACP_PCKT_RAW			R
0x80	00**0**00000	Status of the New ACP Packet interrupt signal. When set to 1, it indicates that an ACP packet has been received with new contents. Once set, this bit remains high until it is cleared via NEW_ACP_PCKT_CLR. 0 = No new ACP packet received 1 = ACP packet with new content received	
NEW_VS_INFO_RAW			R
0x80	000**0**0000	Status of the new vendor specific InfoFrame interrupt signal. When set to 1, it indicates that a Vendor Specific InfoFrame has been received with new contents. Once set, this bit remains high until it is cleared via NEW_VS_INFO_CLR. 0 = No new vendor specific packet received 1 = vendor specific packet with new content received	
NEW_MS_INFO_RAW			R
0x80	0000**0**000	Status of the new MPEG source InfoFrame interrupt signal. When set to 1, it indicates that an MPEG source InfoFrame has been received with new contents. Once set, this bit remains high until it is cleared via NEW_MS_INFO_CLR. 0 = No new MPEG source InfoFrame received 1 = MPEG source InfoFrame with new content received	
NEW_SPD_INFO_RAW			R
0x80	00000**0**00	Status of the new source product descriptor packet interrupt signal. When set to 1, it indicates that a Source Product Descriptor packet has been received with new contents. Once set, this bit remains high until it is cleared via NEW_SPD_INFO_CLR. 0 = No new SPD InfoFrame received 1 = SPD InfoFrame with new content received	
NEW_AUDIO_INFO_RAW			R
0x80	0000000**0**	Status of the New Audio InfoFrame interrupt signal. When set to 1, it indicates that an Audio InfoFrame has been received with new contents. Once set, this bit remains high until it is cleared via NEW_AUDIO_INFO_CLR. 0 = No new audio InfoFrame received 1 = Audio InfoFrame with new content received	
NEW_AVI_INFO_RAW			R
0x80	0000000**0**	Status of the New AVI InfoFrame interrupt signal. When set to 1, it indicates that an AVI InfoFrame has been received with new contents. Once set, this bit remains high until the interrupt has been cleared via NEW_AVI_INFO_CLR. 0 = No new AVI InfoFrame received 1 = AVI InfoFrame with new content received	
NEW_ISRC2_PCKT_ST			R
0x81	**0**0000000	Latched status for the New ISRC2 Packet interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_ISRC2_PCKT_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new ISRC2 packet received. An interrupt has not been generated. 1 = ISRC2 packet with new content received. An interrupt has been generated.	

Reg	Bits	Description	
NEW_ISRC1_PCKT_ST			R
0x81	0**0**000000	Latched status for the New ISRC1 Packet interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_ISRC1_PCKT_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new ISRC1 packet received. An interrupt has not been generated. 1 = ISRC1 packet with new content received. An interrupt has been generated.	
NEW_ACP_PCKT_ST			R
0x81	00**0**00000	Latched status for the new ACP Packet interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_ACP_PCKT_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new ACP packet received. An interrupt has not been generated. 1 = ACP packet with new content received. An interrupt has been generated.	
NEW_VS_INFO_ST			R
0x81	000**0**0000	Latched status for the new vendor specific InfoFrame interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_VS_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new vendor specific packet received. An interrupt has not been generated. 1 = vendor specific packet with new content received. An interrupt has been generated.	
NEW_MS_INFO_ST			R
0x81	0000**0**000	Latched status for the new MPEG source InfoFrame interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_MS_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new MPEG Source InfoFrame received. Interrupt has not been generated. 1 = MPEG Source InfoFrame with new content received. Interrupt has been generated.	
NEW_SPD_INFO_ST			R
0x81	00000**0**00	Latched status for the new source product descriptor InfoFrame interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_SPD_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new SPD InfoFrame received. Interrupt has not been generated. 1 = SPD InfoFrame with new content received. Interrupt has been generated.	
NEW_AUDIO_INFO_ST			R
0x81	000000**0**0	Latched status for the New Audio InfoFrame interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_AUDIO_INFO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new Audio InfoFrame received. Interrupt has not been generated. 1 = Audio InfoFrame with new content received. Interrupt has been generated.	
NEW_AVI_INFO_ST			R
0x81	0000000**0**	Latched status for the NEW_AVI_INFO_RAW. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. Once set, this bit remains high until the interrupt has been cleared via NEW_AVI_INFO_CLR. 0 = NEW_AVI_INFO_RAW has not changed state 1 = NEW_AVI_INFO_RAW has changed state	
NEW_ISRC2_PCKT_CLR			SC
0x82	**0**0000000	Clear bit for NEW_ISRC2_PCKT_RAW and NEW_ISRC2_PCKT_ST bits. 0 = No function 1 = Clear NEW_ISRC2_PCKT_RAW and NEW_ISRC2_PCKT_ST	
NEW_ISRC1_PCKT_CLR			SC
0x82	0**0**000000	Clear bit for NEW_ISRC1_PCKT_RAW and NEW_ISCR1_PCKT_ST bits. 0 = No function 1 = Clear NEW_ISRC1_PCKT_RAW and NEW_ISRC1_PCKT_ST	
NEW_ACP_PCKT_CLR			SC
0x82	00**0**00000	Clear bit for NEW_ACP_PCKT_RAW and NEW_ACP_PCKT_ST bits. 0 = No function 1 = Clear NEW_ACP_PCKT_RAW and NEW_ACP_PCKT_ST	
NEW_VS_INFO_CLR			SC
0x82	000**0**0000	Clear bit for NEW_VS_INFO_RAW and NEW_VS_INFO_ST bits. 0 = No function 1 = Clear NEW_VS_INFO_RAW and NEW_VS_INFO_ST	
NEW_MS_INFO_CLR			SC
0x82	0000**0**000	Clear bit for NEW_MS_INFO_RAW and NEW_MS_INFO_ST bits. 0 = No function 1 = Clear NEW_MS_INFO_RAW and NEW_MS_INFO_ST	

Reg	Bits	Description	
NEW_SPD_INFO_CLR			SC
0x82	000000<u>0</u>00	Clear bit for NEW_SPD_INFO_RAW and NEW_SPD_INFO_ST bits. 0 = No function 1 = Clear NEW_SPD_INFO_RAW and NEW_SPD_INFO_ST	
NEW_AUDIO_INFO_CLR			SC
0x82	0000000<u>0</u>0	Clear bit for NEW_AUDIO_INFO_RAW and NEW_AUDIO_INFO_ST bits. 0 = No function 1 = Clear NEW_AUDIO_INFO_RAW and NEW_AUDIO_INFO_ST	
NEW_AVI_INFO_CLR			SC
0x82	00000000<u>0</u>	Clear bit for NEW_AVI_INFO_RAW and NEW_AVI_INFO_ST bits. 0 = No function 1 = Clear NEW_AVI_INFO_RAW and NEW_AVI_INFO_ST	
NEW_ISRC2_PCKT_MB2			R/W
0x83	<u>0</u>0000000	INTRQ2 interrupt mask for New ISRC2 Packet interrupt. When set, the New ISRC2 interrupt triggers the INTRQ2 interrupt and NEW_ISRC2_ST indicates the interrupt status. 0 = Disables New ISRC2 Packet interrupt for INTRQ2 1 = Enables New ISRC2 Packet interrupt for INTRQ2	
NEW_ISRC1_PCKT_MB2			R/W
0x83	0<u>0</u>000000	INTRQ2 interrupt mask for New ISRC1 Packet interrupt. When set, the New ISRC2 interrupt triggers the INTRQ2 interrupt and NEW_ISRC1_ST indicates the interrupt status. 0 = Disables New ISRC1 Packet interrupt for INTRQ2 1 = Enables New ISRC1 Packet interrupt for INTRQ2	
NEW_ACP_PCKT_MB2			R/W
0x83	00<u>0</u>00000	INTRQ2 interrupt mask for new ACP packet interrupt. When set, the new ACP interrupt triggers the INTRQ2 interrupt and NEW_ACP_ST indicates the interrupt status. 0 = Disables new ACP Packet interrupt for INTRQ2 1 = Enables new ACP Packet interrupt for INTRQ2	
NEW_VS_INFO_MB2			R/W
0x83	000<u>0</u>0000	INTRQ2 interrupt mask for New Vendor Specific InfoFrame interrupt. When set, the New Vendor Specific InfoFrame interrupt triggers the INTRQ2 interrupt and NEW_VS_INFO_ST indicates the interrupt status. 0 = Disables new vendor specific InfoFrame interrupt for INTRQ2 1 = Enables new vendor specific InfoFrame interrupt for INTRQ2	
NEW_MS_INFO_MB2			R/W
0x83	0000<u>0</u>000	INTRQ2 interrupt mask for new MPEG source InfoFrame interrupt. When set, the New MPEG Source InfoFrame interrupt triggers the INTRQ2 interrupt and NEW_SPD_INFO_ST indicates the interrupt status. 0 = Disables new MPEG source (MS) InfoFrame interrupt for INTRQ2 1 = Enables new MS InfoFrame interrupt for INTRQ2	
NEW_SPD_INFO_MB2			R/W
0x83	000000<u>0</u>00	INTRQ2 interrupt mask for New Source Product Descriptor InfoFrame interrupt. When set, the New Source Product Descriptor InfoFrame interrupt triggers the INTRQ2 interrupt and NEW_SPD_INFO_ST indicates the interrupt status. 0 = Disables New SPD InfoFrame interrupt for INTRQ2 1 = Enables New SPD InfoFrame interrupt for INTRQ2	
NEW_AUDIO_INFO_MB2			R/W
0x83	0000000<u>0</u>0	INTRQ2 interrupt mask for New Audio InfoFrame interrupt. When set, the New Audio InfoFrame interrupt triggers the INTRQ2 interrupt and NEW_AUDIO_INFO_ST indicates the interrupt status. 0 = Disables New Audio InfoFrame interrupt for INTRQ2 1 = Enables New Audio InfoFrame interrupt for INTRQ2	
NEW_AVI_INFO_MB2			R/W
0x83	00000000<u>0</u>	INTRQ2 interrupt mask for New AVI InfoFrame detection interrupt. When set a new AVI InfoFrame detection event causes NEW_AVI_INFO_ST to be set and an interrupt is generated on INTRQ2. 0 = Disables New SPD InfoFrame interrupt for INTRQ2 1 = Enables New SPD InfoFrame interrupt for INTRQ2	
NEW_ISRC2_PCKT_MB1			R/W
0x84	<u>0</u>0000000	INTRQ1 interrupt mask for New ISRC2 Packet interrupt. When set, the New ISRC2 interrupt triggers the INTRQ1 interrupt and NEW_ISRC2_ST indicates the interrupt status. 0 = Disables New ISRC2 Packet interrupt for INTRQ1 1 = Enables New ISRC2 Packet interrupt for INTRQ1	

Reg	Bits	Description	
NEW_ISRC1_PCKT_MB1			R/W
0x84	0<u>0</u>000000	INTRQ1 interrupt mask for New ISRC1 Packet interrupt. When set, the New ISRC2 interrupt triggers the INTRQ1 interrupt and NEW_ISRC1_ST indicates the interrupt status. 0 = Disables New ISRC1 Packet interrupt for INTRQ1 1 = Enables New ISRC1 Packet interrupt for INTRQ1	
NEW_ACP_PCKT_MB1			R/W
0x84	00<u>0</u>00000	INTRQ1 interrupt mask for New ACP Packet interrupt. When set, the New ACP interrupt triggers the INTRQ1 interrupt and NEW_ACP_ST indicates the interrupt status. 0 = Disables New ACP Packet interrupt for INTRQ1 1 = Enables New ACP Packet interrupt for INTRQ1	
NEW_VS_INFO_MB1			R/W
0x84	000<u>0</u>0000	INTRQ1 interrupt mask for New Vendor Specific InfoFrame interrupt. When set, the New Vendor Specific InfoFrame interrupt triggers the INTRQ1 interrupt and NEW_VS_INFO_ST indicates the interrupt status. 0 = Disables New vendor specific InfoFrame interrupt for INTRQ1 1 = Enables New vendor specific InfoFrame interrupt for INTRQ1	
NEW_MS_INFO_MB1			R/W
0x84	0000<u>0</u>000	INTRQ1 interrupt mask for New MPEG Source InfoFrame interrupt. When set, the New MPEG Source InfoFrame interrupt triggers the INTRQ1 interrupt and NEW_SPD_INFO_ST indicates the interrupt status. 0 = Disables New MS InfoFrame interrupt for INTRQ1 1 = Enables New MS InfoFrame interrupt for INTRQ1	
NEW_SPD_INFO_MB1			R/W
0x84	00000<u>0</u>00	INTRQ1 interrupt mask for New Source Product Descriptor InfoFrame interrupt. When set, the New Source Product Descriptor InfoFrame interrupt triggers the INTRQ1 interrupt and NEW_SPD_INFO_ST indicates the interrupt status. 0 = Disables New SPD InfoFrame interrupt for INTRQ1 1 = Enables New SPD InfoFrame interrupt for INTRQ1	
NEW_AUDIO_INFO_MB1			R/W
0x84	000000<u>0</u>0	INTRQ1 interrupt mask for New Audio InfoFrame interrupt. When set, the New Audio InfoFrame interrupt triggers the INTRQ1 interrupt and NEW_AUDIO_INFO_ST indicates the interrupt status. 0 = Disables New Audio InfoFrame interrupt for INTRQ1 1 = Enables New Audio InfoFrame interrupt for INTRQ1	
NEW_AVI_INFO_MB1			R/W
0x84	0000000<u>0</u>	INTRQ1 interrupt mask for New AVI InfoFrame detection interrupt. When set a new AVI InfoFrame detection event causes NEW_AVI_INFO_ST to be set and an interrupt is generated on INTRQ1. 0 = Disable new AVI InfoFrame interrupt for INTRQ1 1 = Enable new AVI InfoFrame interrupt for INTRQ1	
FIFO_NEAR_OVFL_RAW			R
0x85	<u>0</u>0000000	Status of Audio FIFO Near Overflow interrupt signal. When set to 1, it indicates the Audio FIFO is near overflow as the number FIFO registers containing stereo data is greater or equal to value set in AUDIO_FIFO_ALMOST_FULL_THRESHOLD. Once set, this bit remains high until it is cleared via FIFO_NEAR_OVFL_CLR. 0 = Audio FIFO has not reached the high threshold defined in AUDIO_FIFO_ALMOST_FULL_THRESHOLD[5:0] 1 = Audio FIFO has reached high the threshold defined in AUDIO_FIFO_ALMOST_FULL_THRESHOLD[5:0]	
FIFO_UNDERFLO_RAW			R
0x85	0<u>0</u>000000	Status of Audio FIFO Underflow interrupt signal. When set to 1, it indicates the Audio FIFO read pointer has reached the write pointer causing the audio FIFO to underflow. Once set, this bit remains high until it is cleared via AUDIO_FIFO_UNDERFLO_CLR. 0 = Audio FIFO has not underflowed 1 = Audio FIFO has underflowed	
FIFO_OVERFLO_RAW			R
0x85	00<u>0</u>00000	Status of Audio FIFO Overflow interrupt signal. When set to 1, it indicates Audio FIFO write pointer has reached the read pointer causing the audio FIFO to overflow. Once set, this bit remains high until it is cleared via AUDIO_FIFO_OVERFLO_CLR. 0 = Audio FIFO has not overflowed 1 = Audio FIFO has overflowed	
CTS_PASS_THRSH_RAW			R
0x85	000<u>0</u>0000	Status of the ACR CTS value exceed threshold interrupt signal. When set to 1, it indicates the CTS Value of the ACR packets has exceeded the threshold set by CTS_CHANGE_THRESHOLD. Once set, this bit remains high until it is cleared via CTS_PASS_THRSH_CLR. 0 = Audio clock regeneration CTS value has not passed the threshold 1 = Audio clock regeneration CTS value has changed more than threshold	

Reg	Bits	Description
CHANGE_N_RAW		R
0x85	0000<u>0</u>000	Status of the ACR N Value changed interrupt signal. When set to 1, it indicates the N Value of the ACR packets has changed. Once set, this bit remains high until it is cleared via CHANGE_N_CLR. 0 = Audio clock regeneration N value has not changed 1 = Audio clock regeneration N value has changed
PACKET_ERROR_RAW		R
0x85	00000<u>0</u>00	Status of the Packet Error interrupt signal. When set to 1, it indicates that a packet has been received with an uncorrectable EEC error in either the header or body. Once set, this bit remains high until it is cleared via PACKET_ERROR_CLR. 0 = No uncorrectable error detected in packet header 1 = Uncorrectable error detected in an unknown packet (error in packet header)
AUDIO_PCKT_ERR_RAW		R
0x85	000000<u>0</u>0	Status of the Audio Packet Error interrupt signal. When set to 1, it indicates that an Audio packet has been received with an uncorrectable error. Once set, this bit remains high until it is cleared via AUDIO_PCKT_ERR_CLR. 0 = No uncorrectable error detected in audio packets 1 = Uncorrectable error detected in an audio packet
NEW_GAMUT_MDATA_RAW		R
0x85	0000000<u>0</u>	Status of the New Gamut Metadata Packet interrupt signal. When set to 1, it indicates that a Gamut Metadata packet has been received with new contents. Once set, this bit remains high until it is cleared via NEW_GAMUT_MDATA_CLR. 0 = No new Gamut metadata packet received or no change has taken place 1 = New Gamut metadata packet received that triggered this interrupt
FIFO_NEAR_OVFL_ST		R
0x86	<u>0</u>0000000	Latched status for the Audio FIFO Near Overflow interrupt. Once set, this bit remains high until the interrupt has been cleared via FIFO_OVFL_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio FIFO has not reached high threshold 1 = Audio FIFO has reached high threshold
FIFO_UNDERFLO_ST		R
0x86	0<u>0</u>000000	Latched status for the Audio FIFO Underflow interrupt. Once set, this bit remains high until the interrupt has been cleared via FIFO_UNDERFLO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio FIFO has not underflowed 1 = Audio FIFO has underflowed
FIFO_OVERFLO_ST		R
0x86	00<u>0</u>00000	Latched status for the Audio FIFO Overflow interrupt. Once set, this bit remains high until the interrupt has been cleared via FIFO_OVERFLO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio FIFO has not overflowed 1 = Audio FIFO has overflowed
CTS_PASS_THRSH_ST		R
0x86	000<u>0</u>0000	Latched status for the ACR CTS Value Exceed Threshold interrupt. Once set, this bit remains high until the interrupt has been cleared via CTS_PASS_THRSH_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio clock regeneration CTS value has not passed the threshold 1 = Audio clock regeneration CTS value has exceeded the threshold
CHANGE_N_ST		R
0x86	0000<u>0</u>000	Latched status for the ACR N Value Changed interrupt. Once set, this bit remains high until the interrupt has been cleared via CHANGE_N_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio clock regeneration N value has not changed 1 = Audio clock regeneration N value has changed
PACKET_ERROR_ST		R
0x86	00000<u>0</u>00	Latched status for the Packet Error interrupt. Once set, this bit remains high until the interrupt has been cleared via PACKET_ERROR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No uncorrectable error detected in packet header. An interrupt has not been generated. 1 = Uncorrectable error detected in an unknown packet (in packet header). An interrupt has been generated.
AUDIO_PCKT_ERR_ST		R
0x86	000000<u>0</u>0	Latched status for the Audio Packet Error interrupt. Once set, this bit remains high until the interrupt has been cleared via AUDIO_PCKT_ERR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No uncorrectable error detected in audio packets. An interrupt has not been generated. 1 = Uncorrectable error detected in an audio packet. An interrupt has been generated.

Reg	Bits	Description	
NEW_GAMUT_MDATA_ST			R
0x86	0000000**0**	Latched status for the New Gamut Metadata Packet interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_GAMUT_MDATA_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No new Gamut metadata packet received or no change has taken place. An interrupt has not been generated. 1 = New Gamut metadata packet received. An interrupt has been generated.	
FIFO_NEAR_OVFL_CLR			SC
0x87	**0**0000000	Clear bit for the Audio FIFO Near Overflow interrupt. 0 = Does not clear 1 = Clears FIFO_NEAR_OVERL_ST	
FIFO_UNDERFLO_CLR			SC
0x87	0**0**000000	Clear bit for the Audio FIFO Underflow interrupt. 0 = Does not clear FIFO_UNDERFLO_ST 1 = Clears FIFO_UNDERFLO_ST	
FIFO_OVERFLO_CLR			SC
0x87	00**0**00000	Clear bit for the Audio FIFO Overflow interrupt. 0 = Does not clear FIFO_OVERFLO_ST 1 = Clears FIFO_OVERFLO_ST	
CTS_PASS_THRSH_CLR			SC
0x87	000**0**0000	Clear bit for ACR CTS Value Exceed Threshold interrupt. 0 = Does not clear 1 = Clears CTS_PASS_THRSH_ST	
CHANGE_N_CLR			SC
0x87	0000**0**000	Clear bit for ACR N Value Changed interrupt. 0 = Does not clear CHANGE_N_ST 1 = Clears CHANGE_N_ST	
PACKET_ERROR_CLR			SC
0x87	00000**0**00	Clear bit for Packet Error interrupt. 0 = Does not clear PACKET_ERROR_ST 1 = Clears PACKET_ERROR_ST	
AUDIO_PCKT_ERR_CLR			SC
0x87	000000**0**0	Clear bit for Audio Packet Error interrupt. 0 = Does not clear AUDIO_PCKT_ERR_ST 1 = Clears AUDIO_PCKT_ERR_ST	
NEW_GAMUT_MDATA_CLR			SC
0x87	0000000**0**	Clear bit for New Gamut Metadata Packet interrupt. 0 = Does not clear NEW_GAMUT_MDATA_ST 1 = Clears NEW_GAMUT_MDATA_ST	
FIFO_NEAR_OVFL_MB2			R/W
0x88	**0**0000000	INTRQ2 interrupt mask for Audio FIFO Near Overflow interrupt. When set, the Audio FIFO Near Overflow interrupt triggers the INTRQ2 interrupt and FIFO_NEAR_OVFL_ST indicates the interrupt status. 0 = Disable Audio FIFO Near Overflow interrupt on INTRQ2 1 = Enable Audio FIFO Near Overflow interrupt on INTRQ2	
FIFO_UNDERFLO_MB2			R/W
0x88	0**0**000000	INTRQ2 interrupt mask for Audio FIFO Underflow interrupt. When set, the Audio FIFO Underflow interrupt triggers the INTRQ2 interrupt and FIFO_UNDERFLO_ST indicates the interrupt status. 0 = Disable Audio FIFO Underflow interrupt on INTRQ2 1 = Enable Audio FIFO Underflow interrupt on INTRQ2	
FIFO_OVERFLO_MB2			R/W
0x88	00**0**00000	INTRQ2 interrupt mask for Audio FIFO Overflow interrupt. When set, the Audio FIFO Overflow interrupt triggers the INTRQ2 interrupt and FIFO_OVERFLO_ST indicates the interrupt status. 0 = Disable Audio FIFO Overflow interrupt on INTRQ2 1 = Enable Audio FIFO Overflow interrupt on INTRQ2	

Reg	Bits	Description	
CTS_PASS_THRSH_MB2			R/W
0x88	0000**0**0000	INTRQ2 interrupt mask for ACR CTS Value Exceed Threshold interrupt. When set, the ACR CTS Value Exceed Threshold interrupt triggers the INTRQ2 interrupt and CTS_PASS_THRSH_ST indicates the interrupt status. 0 = Disable ACR CTS Value Exceeded Threshold interrupt on INTRQ2 1 = Enable ACR CTS Value Exceeded Threshold interrupt on INTRQ2	
CHANGE_N_MB2			R/W
0x88	0000**0**000	INTRQ2 interrupt mask for ACR N Value changed interrupt. When set, the ACR N Value changed interrupt triggers the INTRQ2 interrupt and CHANGE_N_ST indicates the interrupt status. 0 = Disables ACR N Value Changed interrupt for INTRQ2 1 = Enables ACR N Value Changed interrupt for INTRQ2	
PACKET_ERROR_MB2			R/W
0x88	00000**0**00	INTRQ2 interrupt mask for Packet Error interrupt. When set, the Audio Packet Error interrupt triggers the INTRQ2 interrupt and PACKET_ERROR_ST indicates the interrupt status. 0 = Disables Packet Error interrupt for INTRQ2 1 = Enables Packet Error interrupt for INTRQ2	
AUDIO_PCKT_ERR_MB2			R/W
0x88	000000**0**0	INTRQ2 interrupt mask for Audio Packet Error interrupt. When set, the Audio Packet Error interrupt triggers the INTRQ2 interrupt and AUDIO_PCKT_ERR_ST indicates the interrupt status. 0 = Disables Audio Packet Error interrupt for INTRQ2 1 = Enables Audio Packet Error interrupt for INTRQ2	
NEW_GAMUT_MDATA_MB2			R/W
0x88	0000000**0**	INTRQ2 interrupt mask for New Gamut Metadata packet interrupt. When set, the New Gamut Metadata packet interrupt triggers the INTRQ2 interrupt and NEW_GAMUT_MDATA_PCKT_ST indicates the interrupt status. 0 = Disables New Gamut metadata InfoFrame interrupt for INTRQ2 1 = Enables New SPD InfoFrame interrupt for INTRQ2	
FIFO_NEAR_OVFL_MB1			R/W
0x89	**0**0000000	INTRQ1 interrupt mask for Audio FIFO Near Overflow interrupt. When set, the Audio FIFO Overflow interrupt triggers the INTRQ1 interrupt and FIFO_NEAR_OVFL_ST indicates the interrupt status. 0 = Disable Audio FIFO Overflow interrupt on INTRQ1 1 = Enable Audio FIFO Overflow interrupt on INTRQ1	
FIFO_UNDERFLO_MB1			R/W
0x89	0**0**000000	INTRQ1 interrupt mask for Audio FIFO Overflow interrupt. When set, the Audio FIFO Overflow interrupt triggers the INTRQ1 interrupt and FIFO_OVERFLO_ST indicates the interrupt status. 0 = Disable Audio FIFO Overflow interrupt on INTRQ1 1 = Enable Audio FIFO Overflow interrupt on INTRQ1	
FIFO_OVERFLO_MB1			R/W
0x89	00**0**00000	INTRQ1 interrupt mask for Audio FIFO Overflow interrupt. When set, the Audio FIFO Overflow interrupt triggers the INTRQ1 interrupt and FIFO_OVERFLO_ST indicates the interrupt status. 0 = Disable Audio FIFO Overflow interrupt on INTRQ1 1 = Enable Audio FIFO Overflow interrupt on INTRQ1	
CTS_PASS_THRSH_MB1			R/W
0x89	0000**0**0000	INTRQ1 interrupt mask for ACR CTS Value Exceed Threshold interrupt. When set, the ACR CTS Value Exceed Threshold interrupt triggers the INTRQ1 interrupt and CTS_PASS_THRSH_ST indicates the interrupt status. 0 = Disable ACR CTS Value Exceeded Threshold interrupt on INTRQ1 1 = Enable ACR CTS Value Exceeded Threshold interrupt on INTRQ1	
CHANGE_N_MB1			R/W
0x89	0000**0**000	INTRQ1 interrupt mask for ACR N Value changed interrupt. When set, the ACR N Value changed interrupt triggers the INTRQ1 interrupt and CHANGE_N_ST indicates the interrupt status. 0 = Disables ACR N Value Changed interrupt for INTRQ1 1 = Enables ACR N Value Changed interrupt for INTRQ1	
PACKET_ERROR_MB1			R/W
0x89	00000**0**00	INTRQ1 interrupt mask for Packet Error interrupt. When set, the Audio Packet Error interrupt triggers the INTRQ1 interrupt and PACKET_ERROR_ST indicates the interrupt status. 0 = Disables Packet Error interrupt for INTRQ1 1 = Enables Packet Error interrupt for INTRQ1	

Reg	Bits	Description
AUDIO_PCKT_ERR_MB1		R/W
0x89	0000000**0**	INTRQ1 interrupt mask for Audio Packet Error interrupt. When set, the Audio Packet Error interrupt triggers the INTRQ1 interrupt and AUDIO_PCKT_ERR_ST indicates the interrupt status. 0 = Disables Audio Packet Error interrupt for INTRQ1 1 = Enables Audio Packet Error interrupt for INTRQ1
NEW_GAMUT_MDATA_MB1		R/W
0x89	0000000**0**	INTRQ1 interrupt mask for New Gamut Metadata packet interrupt. When set, the New Gamut Metadata packet interrupt triggers the INTRQ1 interrupt and NEW_GAMUT_MDATA_PCKT_ST indicates the interrupt status. 0 = Disables New Gamut METADATA InfoFrame interrupt for INTRQ1 1 = Enables New SPD InfoFrame interrupt for INTRQ1
VCLK_CHNG_RAW		R
0x8A	0**0**000000	Status of Video Clock Changed Interrupt signal. When set to 1, it indicates that irregular or missing pulses are detected in the TMDS clock. Once set, this bit remains high until it is cleared via VCLK_CHNG_CLR. 0 = No irregular or missing pulse detected in TMDS clock 1 = Irregular or missing pulses detected in TMDS clock triggered this interrupt
AUDIO_MODE_CHNG_RAW		R
0x8A	00**0**00000	Status of Audio Mode Change Interrupt signal. When set to 1, it indicates that the type of audio packet received has changed. The following are considered: Audio modes, No Audio Packets, or Audio Sample Packet. Once set, this bit remains high until it is cleared via AUDIO_MODE_CHNG_CLR. 0 = Audio mode has not changed. 1 = Audio mode has changed.
PARITY_ERROR_RAW		R
0x8A	000**0**0000	Status of Parity Error Interrupt signal. When set to 1, it indicates an audio sample packet has been received with parity error. Once set, this bit remains high until it is cleared via PARITY_ERROR_CLR. 0 = No parity error detected in audio packets 1 = Parity error has been detected in an audio packet
NEW_SAMP_RT_RAW		R
0x8A	0000**0**000	Status of new sampling rate interrupt signal. When set to 1, it indicates that audio sampling frequency field in channel status data has changed. Once set, this bit remains high until it is cleared via NEW_SAMP_RT_CLR. 0 = Sampling rate bits of the channel status data on audio channel 0 have not changed 1 = Sampling rate bits of the channel status data on audio channel 0 have changed
AUDIO_FLT_LINE_RAW		R
0x8A	00000**0**00	Status of Audio Flat Line interrupt signal. When set to 1, it indicates audio sample packet has been received with the Flat line bit set to 1. Once set, this bit remains high until it is cleared via AUDIO_FLT_LINE_CLR. 0 = Audio sample packet with flat line bit set has not been received 1 = Audio sample packet with flat line bit set has been received
NEW_TMDS_FRQ_RAW		R
0x8A	000000**0**0	Status of New TMDS Frequency interrupt signal. When set to 1, it indicates the TMDS Frequency has changed by more than the tolerance set in FREQTOLERANCE[3:0]. Once set, this bit remains high until it is cleared via NEW_TMDS_FREQ_CLR. 0 = TMDS frequency has not changed by more than tolerance set in FREQTOLERANCE[3:0] in the HDMI Map 1 = TMDS frequency has changed by more than tolerance set in FREQTOLERANCE[3:0] in the HDMI Map
FIFO_NEAR_UFLO_RAW		R
0x8A	0000000**0**	Status of Audio FIFO Near Underflow interrupt signal. When set to 1, it indicates the Audio FIFO is near underflow as the number of FIFO registers containing stereo data is less or equal to value set in AUDIO_FIFO_ALMOST_EMPTY_THRESHOLD. Once set, this bit remains high until it is cleared via FIFO_NEAR_UFLO_CLR. 0 = Audio FIFO has not reached low threshold defined in AUDIO_FIFO_ALMOST_EMPTY_THRESHOLD[6:0] 1 = Audio FIFO has reached low threshold defined in AUDIO_FIFO_ALMOST_EMPTY_THRESHOLD[6:0]
VCLK_CHNG_ST		R
0x8B	0**0**000000	Latched status of Video Clock Change Interrupt. Once set, this bit remains high until the interrupt has been cleared via VCLK_CHNG_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No irregular or missing pulse detected in TMDS clock 1 = Irregular or missing pulses detected in TMDS clock
AUDIO_MODE_CHNG_ST		R
0x8B	00**0**00000	Latched status of Audio Mode Change Interrupt. Once set, this bit remains high until the interrupt has been cleared via AUDIO_MODE_CHNG_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio mode has not changed 1 = Audio mode has changed. The following are considered Audio modes, No Audio, PCM, DSD, HBR, or DST.

Reg	Bits	Description	
PARITY_ERROR_ST			R
0x8B	0000**0**0000	Latched status of Parity Error Interrupt. Once set, this bit remains high until the interrupt has been cleared via PARITY_ERROR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No parity error detected in audio packets 1 = Parity error detected in an audio packet	
NEW_SAMP_RT_ST			R
0x8B	0000**0**000	Latched status of New Sampling Rate Interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_SAMP_RT_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Sampling rate bits of the channel status data on Audio Channel 0 have not changed 1 = Sampling rate bits of the channel status data on Audio Channel 0 have changed.	
AUDIO_FLT_LINE_ST			R
0x8B	00000**0**00	Latched status of Audio Flat Line Interrupt. Once set, this bit remains high until the interrupt has been cleared via AUDIO_FLT_LINE_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio sample packet with flat line bit set has not been received 1 = Audio sample packet with flat line bit set has been received	
NEW_TMDS_FRQ_ST			R
0x8B	000000**0**0	Latched status of New TMDS Frequency Interrupt. Once set, this bit remains high until the interrupt has been cleared via NEW_TMDS_FREQ_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = TMDS frequency has not changed by more than tolerance 1 = TMDS frequency has changed by more than tolerance	
FIFO_NEAR_UFLO_ST			R
0x8B	0000000**0**	Latched status for the Audio FIFO Near Underflow interrupt. Once set, this bit remains high until the interrupt has been cleared via FIFO_UFLO_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = Audio FIFO has not reached low threshold 1 = Audio FIFO has reached low threshold	
VCLK_CHNG_CLR			SC
0x8C	0**0**000000	Clear bit for the Video Clock Change Interrupt. 0 = Does not clear VCLK_CHNG_ST 1 = Clears VCLK_CHNG_ST	
AUDIO_MODE_CHNG_CLR			SC
0x8C	00**0**00000	Clear bit for the Audio Mode Change Interrupt. 0 = Does not clear AUDIO_MODE_CHNG_ST 1 = Clears AUDIO_MODE_CHNG_ST	
PARITY_ERROR_CLR			SC
0x8C	0000**0**0000	Clear bit for the Parity Error Interrupt. 0 = Does not clear 1 = Clears PARRITY_ERROR_ST	
NEW_SAMP_RT_CLR			SC
0x8C	0000**0**000	Clear bit for the New Sample Rate Interrupt. 0 = Does not clear NEW_SAMP_RT_ST 1 = Clears NEW_SAMP_RT_ST	
AUDIO_FLT_LINE_CLR			SC
0x8C	00000**0**00	Clear bit for the Audio Flat line Interrupt. 0 = Does not clear 1 = Clears AUDIO_FLT_LINE_ST	
NEW_TMDS_FRQ_CLR			SC
0x8C	000000**0**0	Clear bit for the New TMDS Frequency Interrupt. 0 = Does not clear NEW_TMDS_FRQ_ST 1 = Clears NEW_TMDS_FRQ_ST	
FIFO_NEAR_UFLO_CLR			SC
0x8C	0000000**0**	Clear bit for the Audio FIFO Near Underflow interrupt. 0 = Does not clear 1 = Clears FIFO_NEAR_UFLO_ST	

Reg	Bits	Description
VCLK_CHNG_MB2		R/W
0x8D	0<u>0</u>000000	INTRQ2 interrupt mask for Video Clock Changed interrupt. When set, the Video Clock Changed interrupt triggers the INTRQ2 interrupt and VCLK_CHNG_ST indicates the interrupt status. 0 = Disable Video Clock Changed interrupt on INTRQ2 1 = Enable Video Clock Changed interrupt on INTRQ2
AUDIO_MODE_CHNG_MB2		R/W
0x8D	00<u>0</u>00000	INTRQ2 interrupt mask for Audio Mode Change interrupt. When set, the Audio Mode Change interrupt triggers the INTRQ2 interrupt and AUDIO_MODE_CHNG_ST indicates the interrupt status. 0 = Disable Audio Mode Changed interrupt on INTRQ2 1 = Enable Audio Mode Changed interrupt on INTRQ2
PARITY_ERROR_MB2		R/W
0x8D	000<u>0</u>0000	INTRQ2 interrupt mask for Parity Error interrupt. When set, the Parity Error interrupt triggers the INTRQ2 interrupt and PARITY_ERROR_ST indicates the interrupt status. 0 = Disable Parity Error interrupt on INTRQ2 1 = Enable Parity Error interrupt on INTRQ2
NEW_SAMP_RT_MB2		R/W
0x8D	0000<u>0</u>000	INTRQ2 interrupt mask for New Sample Rate interrupt. When set, the New Sample interrupt triggers the INTRQ2 interrupt and NEW_SAMP_RT_ST indicates the interrupt status. 0 = Disable New Sample Rate interrupt on INTRQ2 1 = Enable New Sample Rate interrupt on INTRQ2
AUDIO_FLT_LINE_MB2		R/W
0x8D	00000<u>0</u>00	INTRQ2 interrupt mask for Audio Flat line interrupt. When set, the Audio Flat line interrupt triggers the INTRQ2 interrupt and AUDIO_FLT_LINE_ST indicates the interrupt status. 0 = Disable Audio Flat Line interrupt on INTRQ2 1 = Enable Audio Flat Line interrupt on INTRQ2
NEW_TMDS_FRQ_MB2		R/W
0x8D	000000<u>0</u>0	INTRQ2 interrupt mask for New TMDS Frequency interrupt. When set, the New TMDS Frequency interrupt triggers the INTRQ2 interrupt and NEW_TMDS_ST indicates the interrupt status. 0 = Disable New TMDS Frequency interrupt on INTRQ2 1 = Enable New TMDS Frequency interrupt on INTRQ2
FIFO_NEAR_UFLO_MB2		R/W
0x8D	0000000<u>0</u>	INTRQ2 interrupt mask for Audio FIFO Near Underflow interrupt. When set, the Audio FIFO Near Underflow interrupt triggers the INTRQ2 interrupt and FIFO_NEAR_UFLO_ST indicates the interrupt status. 0 = Disable Audio FIFO Near Underflow interrupt on INTRQ2 1 = Enable Audio FIFO Near Underflow interrupt on INTRQ2
VCLK_CHNG_MB1		R/W
0x8E	0<u>0</u>000000	INTRQ1 interrupt mask for Video Clock Changed interrupt. When set, the Video Clock Changed interrupt triggers the INTRQ1 interrupt and VCLK_CHNG_ST indicates the interrupt status. 0 = Disable Video Clock Change interrupt on INTRQ1 1 = Enable Video Clock Change interrupt on INTRQ1
AUDIO_MODE_CHNG_MB1		R/W
0x8E	00<u>0</u>00000	INTRQ1 interrupt mask for Audio Mode Changed interrupt. When set, the Audio Mode Changed interrupt triggers the INTRQ1 interrupt and AUDIO_MODE_CHNG_ST indicates the interrupt status. 0 = Disable Audio Mode Change interrupt on INTRQ1 1 = Enable Audio Mode Change interrupt on INTRQ1
PARITY_ERROR_MB1		R/W
0x8E	000<u>0</u>0000	INTRQ1 interrupt mask for Parity Error interrupt. When set, the Parity Error interrupt triggers the INTRQ1 interrupt and PARITY_ERROR_ST indicates the interrupt status. 0 = Disable Parity Error interrupt on INTRQ1 1 = Enable Parity Error interrupt on INTRQ1
NEW_SAMP_RT_MB1		R/W
0x8E	0000<u>0</u>000	INTRQ1 interrupt mask for New Sample Rate interrupt. When set, the New Sample Rate interrupt triggers the INTRQ1 interrupt and NEW_SAMP_RT_ST indicates the interrupt status. 0 = Disable New Sample Rate interrupt on INTRQ1 1 = Enable New Sample Rate interrupt on INTRQ1

Reg	Bits	Description	
AUDIO_FLT_LINE_MB1			R/W
0x8E	00000<u>0</u>00	INTRQ1 interrupt mask for Audio Flat Line interrupt. When set, the Audio Flat Line interrupt triggers the INTRQ1 interrupt and AUDIO_FLT_LINE_ST indicates the interrupt status. 0 = Disable Audio Flat Line interrupt on INTRQ1 1 = Enable Audio Flat Line interrupt on INTRQ1	
NEW_TMDS_FRQ_MB1			R/W
0x8E	000000<u>0</u>0	INTRQ1 interrupt mask for New TMDS Frequency interrupt. When set, the New TMDS Frequency interrupt triggers the INTRQ1 interrupt and NEW_TMDS_FREQ_ST indicates the interrupt status. 0 = Disable New TMDS Frequency interrupt on INTRQ1 1 = Enable New TMDS Frequency interrupt on INTRQ1	
FIFO_NEAR_UFLO_MB1			R/W
0x8E	0000000<u>0</u>	INTRQ1 interrupt mask for Audio FIFO Near Underflow interrupt. When set, the Audio FIFO Near Underflow interrupt triggers the INTRQ1 interrupt and FIFO_UFLO_ST indicates the interrupt status. 0 = Disable Audio FIFO Overflow interrupt on INTRQ1 1 = Enable Audio FIFO Overflow interrupt on INTRQ1	
MS_INF_CKS_ERR_RAW			R
0x8F	<u>0</u>0000000	Status of MPEG Source InfoFrame Checksum Error interrupt signal. When set to 1, it indicates that a checksum error has been detected for an MPEG Source InfoFrame. Once set, this bit remains high until it is cleared via MS_INF_CKS_ERR_CLR. 0 = No MPEG source InfoFrame checksum error has occurred 1 = An MPEG source InfoFrame checksum error has occurred	
SPD_INF_CKS_ERR_RAW			R
0x8F	0<u>0</u>000000	Status of SPD InfoFrame Checksum Error interrupt signal. When set to 1, it indicates that a checksum error has been detected for an SPD InfoFrame. Once set, this bit remains high until it is cleared via ASPD_INF_CKS_ERR_CLR. 0 = No SPD InfoFrame checksum error has occurred 1 = An SPD InfoFrame checksum error has occurred	
AUD_INF_CKS_ERR_RAW			R
0x8F	00<u>0</u>00000	Status of Audio InfoFrame Checksum Error interrupt signal. When set to 1, it indicates that a checksum error has been detected for an Audio InfoFrame. Once set, this bit remains high until it is cleared via AUDIO_INF_CKS_ERR_CLR. 0 = No Audio InfoFrame checksum error has occurred 1 = An Audio InfoFrame checksum error has occurred	
AVI_INF_CKS_ERR_RAW			R
0x8F	000<u>0</u>0000	Status of AVI InfoFrame Checksum Error interrupt signal. When set to 1, it indicates that a checksum error has been detected for an AVI InfoFrame. Once set, this bit remains high until it is cleared via AVI_INF_CKS_ERR_CLR. 0 = No AVI InfoFrame checksum error has occurred 1 = An AVI InfoFrame checksum error has occurred	
AKSV_UPDATE_A_RAW			R
0x8F	0000<u>0</u>000	Status of Port A AKSV Update Interrupt signal. When set to 1, it indicates that transmitter has written its AKSV into HDCP registers for Port A. Once set, this bit remains high until it is cleared via AKSV_UPDATE_A_CLR. 0 = No AKSV updates on port A 1 = Detected a write access to the AKSV register on port A	
RI_EXPIRED_A_RAW			R
0x8F	00000<u>0</u>00	Status of Port A Ri expired Interrupt signal. When set to 1, it indicates that HDCP cipher Ri value for Port A expired. Once set, this bit remains high until it is cleared via RI_EXPIRED_A_CLR. 0 = No Ri expired on port A 1 = Ri expired on port A	
VS_INF_CKS_ERR_RAW			R
0x8F	0000000<u>0</u>	Status of Vendor Specific InfoFrame Checksum Error interrupt signal. When set to 1, it indicates that a checksum error has been detected for a vendor specific InfoFrame. Once set, this bit remains high until it is cleared via VS_INF_CKS_ERR_CLR. 0 = No vendor specific InfoFrame checksum error has occurred 1 = A vendor specific InfoFrame checksum error has occurred	
MS_INF_CKS_ERR_ST			R
0x90	<u>0</u>0000000	Latched status of MPEG Source InfoFrame Checksum Error interrupt. Once set, this bit remains high until the interrupt has been cleared via MS_INF_CKS_ERR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No change in MPEG source InfoFrame checksum error 1 = An MPEG source InfoFrame checksum error has triggered this interrupt	

Reg	Bits	Description	
SPD_INF_CKS_ERR_ST			R
0x90	0<u>0</u>000000	Latched status of SPD InfoFrame Checksum Error interrupt. Once set, this bit remains high until the interrupt has been cleared via SPD_INF_CKS_ERR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No change in SPD InfoFrame checksum error 1 = An SPD InfoFrame checksum error has triggered this interrupt	
AUD_INF_CKS_ERR_ST			R
0x90	000<u>0</u>0000	Latched status of Audio InfoFrame Checksum Error interrupt. Once set, this bit remains high until the interrupt has been cleared via AUDIO_INF_CKS_ERR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No change in Audio InfoFrame checksum error 1 = An Audio InfoFrame checksum error has triggered this interrupt	
AVI_INF_CKS_ERR_ST			R
0x90	0000<u>0</u>000	Latched status of AVI InfoFrame Checksum Error interrupt. Once set, this bit remains high until the interrupt has been cleared via AVI_INF_CKS_ERR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No change in AVI InfoFrame checksum error 1 = An AVI InfoFrame checksum error has triggered this interrupt	
AKSV_UPDATE_A_ST			R
0x90	0000<u>0</u>000	Latched status of Port A AKSV Update Interrupt. Once set, this bit remains high until the interrupt has been cleared via AKSV_UPDATE_A_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No AKSV updates on Port A 1 = Detected a write access to the AKSV register on Port A	
RI_EXPIRED_A_ST			R
0x90	00000<u>0</u>00	Latched status of port A Ri expired Interrupt. Once set, this bit remains high until the interrupt has been cleared via RI_EXPIRED_A_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No Ri expired on Port A 1 = Ri expired on Port A	
VS_INF_CKS_ERR_ST			R
0x90	0000000<u>0</u>	Latched status of MPEG Source InfoFrame Checksum Error interrupt. Once set, this bit remains high until the interrupt has been cleared via MS_INF_CKS_ERR_CLR. This bit is only valid if enabled via the corresponding INTRQ1 or INTRQ2 interrupt mask bit. 0 = No change in vendor specific InfoFrame checksum error 1 = A vendor specific InfoFrame checksum error has triggered this interrupt	
MS_INF_CKS_ERR_CLR			SC
0x91	<u>0</u>0000000	Clear bit for the MPEG Source InfoFrame Checksum Error Interrupt. 0 = Does not clear MS_INF_CKS_ERR_ST 1 = Clears MS_INF_CKS_ERR_ST	
SPD_INF_CKS_ERR_CLR			SC
0x91	0<u>0</u>000000	Clear bit for the SPD InfoFrame Checksum Error Interrupt. 0 = Does not clear 1 = Clears SPD_INF_CKS_ERR_ST	
AUD_INF_CKS_ERR_CLR			SC
0x91	000<u>0</u>0000	Clear bit for the Audio InfoFrame Checksum Error Interrupt. 0 = Does not clear AUD_INF_CKS_ERR_ST 1 = Clears AUD_INF_CKS_ERR_ST	
AVI_INF_CKS_ERR_CLR			SC
0x91	0000<u>0</u>000	Clear bit for the AVI InfoFrame Checksum Error Interrupt. 0 = Does not clear AVI_INF_CKS_ERR_ST 1 = Clears AVI_INF_CKS_ERR_ST	
AKSV_UPDATE_A_CLR			SC
0x91	0000<u>0</u>000	Clear bit for the Port A AKSV Update Interrupt. 0 = Does not clear 1 = Clears AKSV_UPDATE_A_ST	
RI_EXPIRED_A_CLR			SC
0x91	00000<u>0</u>00	Clear bit for the Port A Ri expired Interrupt. 0 = Does not clear RI_EXPIRED_D_ST 1 = Clears RI_EXPIRED_D_ST	

Reg	Bits	Description
VS_INF_CKS_ERR_CLR		SC
0x91	0000000**0**	Clear bit for the Vendor Specific InfoFrame Checksum Error Interrupt. 0 = Does not clear 1 = Clears VS_INF_CKS_ERR_ST
MS_INF_CKS_ERR_MB2		R/W
0x92	**0**0000000	INTRQ2 interrupt mask for MPEG Source InfoFrame Checksum Error interrupt. When set, the MPEG Source InfoFrame Checksum Error interrupt triggers the INTRQ2 interrupt and MS_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable MPEG Source InfoFrame Checksum Error interrupt on INTRQ2 1 = Enable MPEG Source InfoFrame Checksum Error interrupt on INTRQ2
SPD_INF_CKS_ERR_MB2		R/W
0x92	0**0**000000	INTRQ2 interrupt mask for SPD InfoFrame Checksum Error interrupt. When set, the SPD InfoFrame Checksum Error interrupt triggers the INTRQ2 interrupt and SPD_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable SPD InfoFrame Checksum Error interrupt on INTRQ2 1 = Enable SPD InfoFrame Checksum Error interrupt on INTRQ2
AUD_INF_CKS_ERR_MB2		R/W
0x92	00**0**00000	INTRQ2 interrupt mask for Audio InfoFrame Checksum Error interrupt. When set, the Audio InfoFrame Checksum Error interrupt triggers the INTRQ2 interrupt and AUDIO_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable Audio InfoFrame Checksum Error interrupt on INTRQ2 1 = Enable Audio InfoFrame Checksum Error interrupt on INTRQ2
AVI_INF_CKS_ERR_MB2		R/W
0x92	000**0**0000	INTRQ2 interrupt mask for AVI InfoFrame Checksum Error interrupt. When set, the AVI InfoFrame Checksum Error interrupt triggers the INTRQ2 interrupt and AVI_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable AVI InfoFrame Checksum Error interrupt on INTRQ2 1 = Enable AVI InfoFrame Checksum Error interrupt on INTRQ2
AKSV_UPDATE_A_MB2		R/W
0x92	0000**0**000	INTRQ2 interrupt mask for Port A AKSV Update interrupt. When set, the Port A AKSV Update interrupt triggers the INTRQ2 interrupt and AKSV_UPDATE_A_ST indicates the interrupt status. 0 = Disable Port A AKSV Update interrupt on INTRQ2 1 = Enable Port A AKSV Update interrupt on INTRQ2
RI_EXPIRED_A_MB2		R/W
0x92	00000**0**00	INTRQ2 interrupt mask for Port A Ri expired interrupt. When set, the port A Ri expired interrupt triggers the INTRQ2 interrupt and RI_EXPIRED_A_ST indicates the interrupt status. 0 = Disable port A Ri expired interrupt on INTRQ2 1 = Enable port A Ri expired interrupt on INT 2
VS_INF_CKS_ERR_MB2		R/W
0x92	0000000**0**	INTRQ2 interrupt mask for Vendor Specific InfoFrame Checksum Error interrupt. When set, the Vendor Specific InfoFrame Checksum Error interrupt triggers the INTRQ2 interrupt and VS_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable Vendor Specific InfoFrame Checksum Error interrupt on INTRQ2 1 = Enable Vendor Specific InfoFrame Checksum Error interrupt on INTRQ2
MS_INF_CKS_ERR_MB1		R/W
0x93	**0**0000000	INTRQ1 interrupt mask for MPEG Source InfoFrame Checksum Error interrupt. When set, the MPEG Source InfoFrame Checksum Error interrupt triggers the INTRQ1 interrupt and MS_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable SPD InfoFrame Checksum Error interrupt on INTRQ1 1 = Enable SPD InfoFrame Checksum Error interrupt on INTRQ1
SPD_INF_CKS_ERR_MB1		R/W
0x93	0**0**000000	INTRQ1 interrupt mask for SPD InfoFrame Checksum Error interrupt. When set, the SPD InfoFrame Checksum Error interrupt triggers the INTRQ1 interrupt and SPD_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable SPD InfoFrame Checksum Error interrupt on INTRQ1 1 = Enable SPD InfoFrame Checksum Error interrupt on INTRQ1
AUD_INF_CKS_ERR_MB1		R/W
0x93	00**0**00000	INTRQ1 interrupt mask for Audio InfoFrame Checksum Error interrupt. When set, the Audio InfoFrame Checksum Error interrupt triggers the INTRQ1 interrupt and AUDIO_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable Audio InfoFrame Checksum Error interrupt on INTRQ1 1 = Enable Audio InfoFrame Checksum Error interrupt on INTRQ1

Reg	Bits	Description	
AVI_INF_CKS_ERR_MB1			R/W
0x93	0000**0**0000	INTRQ1 interrupt mask for AVI InfoFrame Checksum Error interrupt. When set, the AVI InfoFrame Checksum Error interrupt triggers the INTRQ1 interrupt and AVI_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable AVI InfoFrame Checksum Error interrupt on INTRQ1 1 = Enable AVI InfoFrame Checksum Error interrupt on INTRQ1	
AKSV_UPDATE_A_MB1			R/W
0x93	0000**0**000	INTRQ1 interrupt mask for Port A AKSV Update interrupt. When set, the Port A AKSV Update interrupt triggers the INTRQ1 interrupt and AKSV_UPDATE_A_ST indicates the interrupt status. 0 = Disable Port A AKSV Update interrupt on INTRQ1 1 = Enable Port A AKSV Update interrupt on INTRQ1	
RI_EXPIRED_A_MB1			R/W
0x93	00000**0**00	INTRQ1 interrupt mask for Port A Ri expired interrupt. When set, the port A AKSV Update interrupt triggers the INTRQ1 interrupt and RI_EXPIRED_A_ST indicates the interrupt status. 0 = Disable Port A Ri expired interrupt on INTRQ1 1 = Enable Port A Ri expired interrupt on INTRQ1	
VS_INF_CKS_ERR_MB1			R/W
0x93	0000000**0**	INTRQ1 interrupt mask for Vendor Specific InfoFrame Checksum Error interrupt. When set, the Vendor Specific InfoFrame Checksum Error interrupt triggers the INTRQ1 interrupt and VS_INF_CKS_ERR_ST indicates the interrupt status. 0 = Disable Vendor Specific Checksum Error interrupt on INTRQ1 1 = Enable Vendor Specific Checksum Error interrupt on INTRQ1	
RD_INFO[15:0]			R
0xDF 0xE0	00000000 00000000	This readback indicates the chip revision. 0x2140 = ES1 0x2141 = ES2 0x2142 = ES3 0x2143 = ES3.1	
READ_AUTO_INC_EN			R/W
0xF2	0000000**1**	This bit is used to auto-increment I²C addresses in the device for consecutive reads. 0 = No auto-increment of I²C address for consecutive reads 1 = Auto-increment of I²C address for consecutive reads	
DPLL_SLAVE_ADDR[6:0]			R/W
0xF3	0000000**0**	This control is used to program the I²C slave address for the DPLL Map.	
CP_SLAVE_ADDR[7:1]			R/W
0xF4	0000000**0**	This control is used to program the I²C slave address for the CP Map.	
HDMI_SLAVE_ADDR[7:1]			R/W
0xF5	0000000**0**	This control is used to program the I²C slave address for the HDMI Map.	
EDID_SLAVE_ADDR[7:1]			R/W
0xF6	0000000**0**	This control is used to program the I²C slave address for the EDID Map.	
REPEATER_SLAVE_ADDR[6:0]			R/W
0xF7	0000000**0**	This control is used to program the I²C slave address for the Repeater Map.	
INFOFRAME_SLAVE_ADDR[7:1]			R/W
0xF8	0000000**0**	This control is used to program the I²C slave address for the InfoFrame Map.	
CBUS_SLAVE_ADDR[6:0]			R/W
0xF9	0000000**0**	This control is used to program the I²C slave address for the CBUS Map.	
CEC_SLAVE_ADDR[7:1]			R/W
0xFA	0000000**0**	This control is used to program the I²C slave address for the CEC Map.	
SD_CORE_SLAVE_ADDR[7:1]			R/W
0xFB	0000000**0**	This control is used to program the I²C slave address for the SD core Map.	

Reg	Bits	Description	
CSI1_TX_SLAVE_ADDR[7:1]			R/W
0xFC	0000000	This control is used to program the I²C slave address for the CSI 1 lane Tx Map.	
CSI4_TX_SLAVE_ADDR[7:1]			R/W
0xFD	0000000	This control is used to program the I²C slave address for the CSI 4 lane Tx Map.	
MAIN_RESET			SC
0xFF	0	This bit is used to initiate a global reset for the device. All I²C maps and all clock domains are reset. 0 = Normal Operation 1 = Apply Main I²C reset	

2.2 HDMI RX MAP

Reg	Bits	Description	
HDCP_A0			R/W
0x00	00000000	This bit is used to set the second LSB of the HDCP port I²C address. 0 = I²C address for HDCP port is 0x74. Used for Single-Link Mode or first receiver in Dual-Link Mode 1 = I²C address for HDCP port is 0x76. Used only for a second receiver Dual-link Mode.	
HDCP_ONLY_MODE			R/W
0x00	00000000	This control is used to configure a HDCP only mode for simultaneous analog and HDMI/MHL modes. Refer to the ADC_HDMI_SIMULTANEOUS_MODE bit. By selecting HDCP only mode HDMI/MHL activity is reduced and it can be used as a power saving feature in simultaneous analog and HDMI/MHL operation. 0 = Normal Operation 1 = HDCP Only Mode	
TERM_AUTO			R/W
0x01	00000000	This bit allows the user to select automatic or manual control of clock and data termination. If automatic mode termination is enabled, then the termination on the port is enabled. 0 = Disable Termination automatic control 1 = Enable Termination automatic control	
I2SOUTMODE[1:0]			R/W
0x03	00011000	This control is used to configure the I²S output interface. 00 = I²S Mode 01 = Right Justified 10 = Left Justified 11 = Raw SPDIF (IEC60958) Mode	
I2SBITWIDTH[4:0]			R/W
0x03	00011000	This control is used to adjust the bit width for right justified mode on the I²S interface. 00000 = 0 bit 00001 = 1 bit 00010 = 2 bits 11000 = 24 bits 11110 = 30 bits 11111 = 31 bits	
AV_MUTE			R
0x04	00000000	This bit is a readback of AVMUTE status received in the last General Control packet received. 0 = AVMUTE not set 1 = AVMUTE set	
HDCP_KEYS_READ			R
0x04	00000000	This bit is a readback to indicate a successful read of the HDCP keys and/or KSV from the internal HDCP Key OTP ROM. A logic high is returned when the read is successful. 0 = HDCP keys and/or KSV not yet read 1 = HDCP keys and/or KSV HDCP keys read	
HDCP_KEY_ERROR			R
0x04	00000000	This bit is a readback to indicate if a checksum error occurred while reading the HDCP and/or KSV from the HDCP Key ROM. Returns high when the HDCP Key master encounters an error while reading the HDCP Key OTP ROM. 0 = No error occurred while reading HDCP keys 1 = HDCP keys read error	
HDCP_RI_EXPIRED			R
0x04	00000000	This bit is a readback to indicate an Ri has not been calculated. This readback is high when a calculated Ri has not been read by the source Tx, on the active port. It remains high until next AKSV update.	
TMDS_PLL_LOCKED			R
0x04	00000000	This bit is a readback to indicate if the TMDS PLL is locked to the TMDS clock input of the HDMI/MHL port. 0 = The TMDS PLL is not locked 1 = The TMDS PLL is locked to the TMDS clock input of the HDMI/MHL port.	
AUDIO_PLL_LOCKED			R
0x04	00000000	This bit is a readback to indicate the Audio DPLL lock status. 0 = The audio DPLL is not locked 1 = The audio DPLL is locked	

Reg	Bits	Description	
HDMI_MODE			R
0x05	0000000	This bit is a readback to indicate whether the stream processed by the HDMI/MHL core is a DVI or an HDMI/MHL stream. 0 = DVI Mode Detected 1 = HDMI/MHL Mode Detected	
HDMI_CONTENT_ENCRYPTED			R
0x05	0000000	This bit is a readback to indicate if the input stream processed by the HDMI/MHL core is HDCP encrypted or not. 0 = The input stream processed by the HDMI/MHL core is not HDCP encrypted 1 = The input stream processed by the HDMI/MHL core is HDCP encrypted	
DVI_HSYNC_POLARITY			R
0x05	0000000	This bit is a readback to indicate the polarity of the HS encoded in the HDMI/MHL input stream 0 = The HS is active low 1 = The HS is active high	
DVI_VSYNC_POLARITY			R
0x05	0000000	This bit is a readback to indicate the polarity of the VS encoded in the HDMI/MHL input stream 0 = The VS is active low 1 = The VS is active high	
HDMI_PIXEL_REPETITION[3:0]			R
0x05	00000000	This signal is a readback to provide the current HDMI/MHL pixel repetition value decoded from the AVI InfoFrame received. The HDMI/MHL receiver automatically discards repeated pixel data and divides the pixel clock frequency appropriately as per the pixel repetition value. 0000 = 1× 0001 = 2× 0010 = 3× 0011 = 4× 0100 = 5× 0101 = 6× 0110 = 7× 0111 = 8× 1000 = 9× 1001 = 10× 1010 to 1111 = Reserved	
VERT_FILTER_LOCKED			R
0x07	0000000	This bit is a readback to indicate the vertical filter lock status. Indicates whether or not the vertical filter is locked and whether VS parameter measurements are valid for readback. 0 = Vertical filter has not locked 1 = Vertical filter has locked	
AUDIO_CHANNEL_MODE			R
0x07	0000000	This bit is a readback to indicate whether stereo or multichannel audio packets are being received. Note that stereo packets may carry compressed multichannel audio. 0 = Stereo Audio (may be compressed multichannel) 1 = Multichannel uncompressed audio detected (three to eight channels).	
DE_REGEN_FILTER_LOCKED			R
0x07	0000000	This bit is a readback to indicate the DE regeneration filter lock status. Indicates that the DE regeneration section has locked to the received DE and HS parameter measurements are valid for readback. 0 = DE regeneration not locked 1 = DE regeneration locked to incoming DE	
LINE_WIDTH[12:0]			R
0x07 0x08	00000000 00000000	This control is a readback indicating the line width. Line width is a horizontal synchronization measurement. It gives the number of active pixels in a line. This measurement is only valid when the DE regeneration filter is locked. xxxxxxxxxxx = Number of active pixels per line.	
FIELD0_HEIGHT[12:0]			R
0x09 0x0A	00000000 00000000	This control is a readback indicating the Field 0 height. Field 0 Height is a vertical filter measurement. This readback gives the number of active lines in Field 0. This measurement is valid only when the vertical filter has locked. xxxxxxxxxxxx = The number of active lines in Field 0	

Reg	Bits	Description	
DEEP_COLOR_MODE[1:0]			R
0x0B	00000000	This control is a readback indicating the deep color mode information extracted from the general control packet. 00 = 8 bits per channel 01 = 10 bits per channel 10 = Reserved 11 = Reserved	
HDMI_INTERLACED			R
0x0B	00000000	This bit is a readback indicating the HDMI/MHL input Interlace status. HDMI/MHL input Interlace status is a vertical filter measurement. 0 = Progressive Input 1 = Interlaced Input	
FIELD1_HEIGHT[12:0]			R
0x0B 0x0C	00000000 00000000	This control is a readback indicating the Field 1 height. Field 1 height is a vertical filter measurement. This readback gives the number of active lines in field. This measurement is valid only when the vertical filter has locked. Field 1 measurements are only valid when HDMI_INTERLACED is set to 1. xxxxxxxxxxxxx = The number of active lines in Field 1	
FREQTOLERANCE[3:0]			R/W
0x0D	00000100	This control is used to set the tolerance in MHz for the new TMDS frequency detection. This tolerance is used for the audio mute mask, MT_MSK_VCLK_CHNG, and the HDMI/MHL status bit, NEW_TMDS_FRQ_RAW. 0100 = Default tolerance in MHz for new TMDS frequency detection xxxx = Tolerance in MHz for new TMDS frequency detection	
MAN_AUDIO_DL_BYPASS			R/W
0x0F	00011111	This bit is used to enable the audio delay bypass manual control. The audio delay line is automatically active for stereo samples and bypassed for multichannel samples. By setting MAN_AUDIO_DL_BYPASS to 1, the audio delay bypass configuration can be set by the user with the AUDIO_DELAY_LINE_BYPASS control. 0 = Audio delay line is automatically bypassed if multichannel audio is received. The audio delay line is automatically enabled if stereo audio is received. 1 = Overrides automatic bypass of audio delay line. Audio delay line is applied depending on the AUDIO_DELAY_LINE_BYPASS control.	
AUDIO_DELAY_LINE_BYPASS			R/W
0x0F	00011111	This bit is used to control the manual bypass for the audio delay line. Only valid if MAN_AUDIO_DL_BYPASS is set to 1. 0 = Enables the audio delay line. 1 = Bypasses the audio delay line.	
AUDIO_MUTE_SPEED[4:0]			R/W
0x0F	00011111	This signal is used to set the number of samples between each volume change of 1.5 dB when muting and unmuting. xxxxx = Number of samples between each volume change of 1.5dB	
CTS_CHANGE_THRESHOLD[5:0]			R/W
0x10	00100101	This signal is used to set the tolerance for change in the CTS value. This tolerance is used for the audio mute mask, MT_MSK_NEW_CTS and the HDMI/MHL status bit CTS_PASS_THRSH_RAW and the HDMI/MHL interrupt status bit, CTS_PASS_THRSH_ST. This register controls the amounts of LSBs that the CTS can change before an audio mute, status change, or interrupt is triggered. 100101 = Default tolerance of CTS value for CTS_PASS_THRSH_RAW and MT_MSK_NEW_CTS xxxxxx = Tolerance of CTS value for CTS_PASS_THRSH_RAW and MT_MSK_NEW_CTS	
AUDIO_FIFO_ALMOST_FULL_THRESHOLD[6:0]			R/W
0x11	01111101	This signal is used to set the threshold used for FIFO_NEAR_OVFL_RAW. The FIFO_NEAR_OVFL_ST interrupt is triggered if the audio FIFO reaches this level.	
AUDIO_FIFO_ALMOST_EMPTY_THRESHOLD[6:0]			R/W
0x12	00000010	This signal is used to set the threshold used for FIFO_NEAR_UFLO_RAW. The FIFO_NEAR_UFLO_ST interrupt is triggered if the audio FIFO goes below this level.	
AC_MSK_VCLK_CHNG			R/W
0x13	01111111	This bit is used to enable the Audio Coast Mask for TMDS clock change. When set, the audio DPLL coasts if the TMDS clock has any irregular/missing pulses. 1 = Audio DPLL coasts if TMDS clock contains irregular/missing pulses. 0 = Audio DPLL does not coast if TMDS clock contains irregular/missing pulses.	

Reg	Bits	Description
AC_MSK_VPLL_UNLOCK		R/W
0x13	011[1]1111	This bit is used to control the Audio Coast Mask for TMDS PLL Unlock. When set, the audio DPLL coasts if the TMDS PLL unlocks. 1 = Audio DPLL coasts if TMDS DPLL unlocks. 0 = Audio DPLL does not coast if TMDS DPLL unlocks.
AC_MSK_NEW_CTS		R/W
0x13	0111[1]111	This bit is used to control the Audio Coast Mask for a new ACR CTS value. When set, the audio DPLL coasts if CTS changes by more than threshold defined in CTS_CHANGE_THRESHOLD[5:0]. 1 = Audio DPLL coasts if CTS changes by more than the threshold set in CTS_CHANGE_THRESHOLD[5:0]. 0 = Audio DPLL does not coast if CTS changes by more than the threshold set in CTS_CHANGE_THRESHOLD[5:0].
AC_MSK_NEW_N		R/W
0x13	01111[1]11	This bit is used to control the Audio Coast Mask for a new ACR N value. When set, the audio DPLL coasts if N value changes. 1 = Audio DPLL coasts if a change in the N value occurs. 0 = Audio DPLL does not coast if a change in the N value occurs.
AC_MSK_VCLK_DET		R/W
0x13	0111111[1]	This bit is used to control the Audio Coast Mask for a TMDS clock detection. It sets the audio DPLL to coast if no TMDS clock is detected on the active port. 1 = Audio DPLL coasts if a TMDS clock is not detected on the active port. 0 = Audio DPLL does not coast if a TMDS clock is not detected on the active port.
MT_MSK_COMPRS_AUD		R/W
0x14	001[1]1111	This bit is used to control the Audio Mute Mask for compressed audio. It sets the audio mutes if the audio received is in a compressed format. Note that NEW_MUTE_COMPR (HDMI RX Map, Address 0x1A[7]) must also be set to 0 for this mute condition to be effective. 1 = Audio mute occurs if audio is received in compressed format.
MT_MSK_AUD_MODE_CHNG		R/W
0x14	001[1]1111	This bit is used to control the Audio Mute Mask for audio mode change. It sets audio mutes if audio changes between any of the following PCM, DSD, HBR or DST formats. 1 = Audio mute occurs if audio changes between any of the following PCM, DSD, HBR, or DST formats.
MT_MSK_PARITY_ERR		R/W
0x14	00111[1]1	This bit is used to control the Audio Mute Mask for a parity error. It sets the audio mutes if an audio sample packet is received with an incorrect parity bit. 1 = Audio mute occurs if an audio sample packet is received with an incorrect parity bit.
MT_MSK_VCLK_CHNG		R/W
0x14	0011111[1]	This bit is used to control the Audio Mute Mask for TMDS Clock Change. It sets the audio mutes if the TMDS clock has irregular/missing pulses. 1 = Audio mute occurs if the TMDS clock has irregular/missing pulses.
MT_MSK_APLL_UNLOCK		R/W
0x15	[1]1111111	This bit is used to control the Audio Mute Mask for Audio PLL Unlock. It sets the audio mutes if the Audio PLL unlocks. 1 = Audio mute occurs if the Audio PLL unlocks.
MT_MSK_VPLL_UNLOCK		R/W
0x15	1[1]111111	This bit is used to control the Audio Mute Mask for TMDS PLL Unlock. When set audio mutes if the TMDS PLL unlocks. 1 = Audio mute occurs if the TMDS PLL unlocks.
MT_MSK_ACR_NOT_DET		R/W
0x15	11[1]11111	This bit is used to control the Audio Mute Mask for ACR packet. When set, the audio mutes if an ACR packet has not been received within one VS. 1 = Audio mute occurs if an ACR packet has not been received within one VS.
MT_MSK_FLATLINE_DET		R/W
0x15	1111[1]111	This bit is used to control the Audio Mute Mask for Flatline bit. When set, the audio mutes if an audio packet is received with the flatline bit set. 1 = Audio mute occurs if an audio packet is received with the flatline bit set.
MT_MSK_FIFO_UNDERLFOW		R/W
0x15	111111[1]	This bit is used to control the Audio Mute Mask for FIFO Underflow. 1 = Audio mute occurs if the FIFO underflows.

Reg	Bits	Description	
MT_MSK_FIFO_OVERFLOW			R/W
0x15	1111111<u>1</u>	This bit is used to control the Audio Mute Mask for FIFO Overflow. 1 = Audio mute occurs if the FIFO overflows.	
MT_MSK_AVMUTE			R/W
0x16	<u>1</u>1111111	This bit is used to control the Audio Mute Mask for AVMUTE. When set, the audio mutes if a general Control packet is received with the SET_AVMUTE bit set. 1 = Audio mute occurs if AVMUTE is set by a general control packet	
MT_MSK_NOT_HDMIMODE			R/W
0x16	1<u>1</u>111111	This bit is used to control the Audio Mute Mask for a non HDMI/MHL input stream. When set, the audio mutes if the HDMI_MODE bit goes low. 1 = Audio mute occurs if HDMI mode bit goes low	
MT_MSK_NEW_CTS			R/W
0x16	11<u>1</u>11111	This bit is used to control the Audio Mute Mask for a change of ACR CTS. When set, the audio mutes if the CTS changes by more than the specified threshold. CTS_CHANGE_THRESHOLD register sets this threshold. 1 = Audio mute occurs if CTS changes	
MT_MSK_NEW_N			R/W
0x16	111<u>1</u>1111	This bit is used to control the Audio Mute Mask for a New ACR N. If set the audio mutes if there is a change in the N value. 1 = Audio mute occurs if N changes	
MT_MSK_CHMODE_CHNG			R/W
0x16	1111<u>1</u>111	This bit is used to control the Audio Mute Mask for an audio channel mode change. When set, the audio mutes if the channel mode changes between stereo and multichannel. 1 = Audio mute occurs if channel mode changes	
MT_MSK_APCKT_ECC_ERR			R/W
0x16	11111<u>1</u>11	This bit is used to control the Audio Mute Mask for Audio Packet ECC Error. When set, the audio mutes if an uncorrectable error is detected in the audio packet by the ECC block. 1 = Audio mute occurs if an uncorrectable error is detected in the audio packet by the ECC block	
MT_MSK_VCLK_DET			R/W
0x16	1111111<u>1</u>	This bit is used to control the Audio Mute Mask for TMDS Clock. When set, the audio mutes if a TMDS clock is not detected. 1 = Audio mute occurs if TMDS is not detected	
AUDIO_SAMPLE_PCKT_DET			R
0x18	0000000<u>0</u>	This bit indicates if Audio Sample Packets have been detected. It resets to zero on the 11th HS leading edge following an Audio packet if a subsequent audio sample packet has not been received. 0 = No linear pulse coded modulated (LPCM) or IEC 61937 compressed audio sample packet received within the last 10 HS signals. 1 = LPCM or IEC 61937 compressed audio sample packet received within the last 10 HS signals.	
IGNORE_PARITY_ERR			R/W
0x1A	1<u>0</u>000000	This control is used to select the processing of audio samples even when they have a parity error. 0 = Discard audio sample packet that have an invalid parity bit. 1 = Process audio sample packets that have an invalid parity bit.	
MUTE_AUDIO			R/W
0x1A	100<u>0</u>0000	This control is used to force an internal mute independently of the mute mask conditions. 0 = Audio in normal operation 1 = Force audio mute	
WAIT_UNMUTE[2:0]			R/W
0x1A	1000<u>000</u>0	This control is used to delay the audio unmute. Once all mute conditions are inactive, WAIT_UNMUTE[2:0] can specify a further delay time before unmuting. NOT_AUTO_UNMUTE must be set to 0 for this control to be effective. 000 = Disables/cancels delayed unmute. Audio unmutes directly after all mute conditions become inactive 001 = Unmutes 250 ms after all mute conditions become inactive 010 = Unmutes 500 ms after all mute conditions become inactive 011 = Unmutes 750 ms after all mute conditions become inactive 100 = Unmutes 1 sec after all mute conditions become inactive	
NOT_AUTO_UNMUTE			R/W
0x1A	1000000<u>0</u>	This control is used to disable the auto unmute feature. When set to 1 audio can be unmuted manually if all mute conditions are inactive by setting NOT_AUTO_UNMUTE to 0 and then back to 1. 0 = Audio unmutes following a delay set by WAIT_UNMUTE after all mute conditions have become inactive. 1 = Prevents audio from unmuting automatically	

Reg	Bits	Description	
DCFIFO_RESET_ON_LOCK			R/W
0x1B	00011000	This bit is used to enable the reset/recentering of video FIFO on video PLL unlock. 0 = Do not reset on video PLL lock 1 = Reset FIFO on video PLL lock	
DCFIFO_KILL_NOT_LOCKED			R/W
0x1B	00011000	This bit control is used to control whether or not the output of the Video FIFO is set to zero when the video PLL is unlocked. 0 = FIFO data is output regardless of video PLL lock status 1 = FIFO output is zeroed if video PLL is unlocked	
DCFIFO_KILL_DIS			R/W
0x1B	00011000	This bit is used to control whether or not the Video FIFO output is zeroed if there is more than one resynchronization of the pointers within two FIFO cycles. This behavior can be disabled with this bit. 0 = FIFO output set to zero if more than one resynchronization is necessary during two FIFO cycles 1 = FIFO output never set to zero regardless of how many resynchronizations occur	
DCFIFO_LOCKED			R
0x1C	00000000	This bit is a readback to indicate if the Video FIFO is locked. 0 = Video FIFO is not locked. Video FIFO had to resynchronize between previous two VS signals 1 = Video FIFO is locked. Video FIFO did not have to resynchronize between previous two VS signals	
DCFIFO_LEVEL[2:0]			R
0x1C	00000000	This signal is a readback to indicate the distance between the read and write pointers. Overflow and underflow read as Level 0. The ideal centered functionality reads as 0b100. 000 = FIFO has underflowed or overflowed 001 = FIFO is about to overflow 010 = FIFO has some margin 011 = FIFO has some margin 100 = FIFO perfectly balanced 101 = FIFO has some margin 110 = FIFO has some margin 111 = FIFO is about to underflow	
PDN_PKT_PROCESSOR			R/W
0x1D	00000000	This bit is used to enable a power saving feature that disables the clocking of the TMDS rate section of the packet processor. Note that the audio clocking can be stopped separately, from the digital PLL (clock generator). 0 = Packet processor is active 1 = Packet processor is stopped, (that is, powered down)	
UP_CONVERSION_MODE			R/W
0x1D	00000000	This bit is used to select linear or interpolated 4:2:2 to 4:4:4 conversion. A 4:2:2 incoming stream is always upconverted to a 4:4:4 stream before being sent to the CP. 0 = Cr and Cb samples are repeated in their respective channel. 1 = Interpolate Cr and Cb values.	
TOTAL_LINE_WIDTH[13:0]			R
0x1E 0x1F	00000000 00000000	This readback is used to indicate the total line width. Total line width is a horizontal synchronization measurement. This gives the total number of pixels per line. This measurement is valid only when the DE regeneration filter has locked. xxxxxxxxxxxx = Total number of pixels per line.	
HSYNC_FRONT_PORCH[12:0]			R
0x20 0x21	00000000 00000000	This readback is used to indicate the HS front porch width. HS front porch width is a horizontal synchronization measurement. The unit of this measurement is unique pixels. This measurement is valid only when the DE regeneration filter has locked. xxxxxxxxxxx = Total number of pixels in the front porch.	
HSYNC_PULSE_WIDTH[12:0]			R
0x22 0x23	00000000 00000000	This readback is used to indicate the HS pulse width. HS pulse width is a horizontal synchronization measurement. The unit of this measurement is unique pixels. This measurement is valid only when the DE regeneration filter has locked. xxxxxxxxxxx = Total number of pixels in the HS pulse width.	
HSYNC_BACK_PORCH[12:0]			R
0x24 0x25	00000000 00000000	This readback is used to indicate the HS Back Porch width. HS Back Porch width is a horizontal synchronization measurement. The unit of this measurement is unique pixels. This measurement is valid only when the DE regeneration filter has locked. xxxxxxxxxxx = Total number of pixels in the back porch.	

Reg	Bits	Description
FIELD0_TOTAL_HEIGHT[13:0]		R
0x26 0x27	00000000 00000000	This readback is used to indicate the Field 0 total height. Field 0 total height is a vertical synchronization measurement. This readback gives the total number of half lines in Field 0. This measurement is valid only when the vertical filter has locked. xxxxxxxxxxxxxx = The total number of half lines in Field 0. (Divide readback by 2 to obtain number of lines)
FIELD1_TOTAL_HEIGHT[13:0]		R
0x28 0x29	00000000 00000000	This readback is used to indicate the Field 1 total height. Field 1 total height is a vertical synchronization measurement. This readback gives the total number of half lines in Field 1. This measurement is valid only when the vertical filter has locked. Field 1 measurements are valid when HDMI_INTERLACED is set to 1. xxxxxxxxxxxxxx = The total number of half lines in Field 1. (Divide readback by 2 to get number of lines)
FIELD0_VS_FRONT_PORCH[13:0]		R
0x2A 0x2B	00000000 00000000	This readback is used to indicate the Field 0 VS front porch width. Field 0 VS front porch width is a vertical synchronization measurement. The unit of this measurement is half lines. This measurement is valid only when the vertical filter has locked. xxxxxxxxxxxxxx = The number of half lines in the VS Front Porch of Field 0. (Divide readback by 2 to get number of lines)
FIELD1_VS_FRONT_PORCH[13:0]		R
0x2C 0x2D	00000000 00000000	This readback is used to indicate the Field 1 VS front porch width. Field 1 VS front porch width is a vertical synchronization measurement. The unit of this measurement is half lines. This measurement is valid only when the vertical filter has locked. Field 1 measurements are valid when HDMI_INTERLACED is set to 1. xxxxxxxxxxxxxx = The number of half lines in the VS Front Porch of Field 1. (Divide readback by 2 to get number of lines)
FIELD0_VS_PULSE_WIDTH[13:0]		R
0x2E 0x2F	00000000 00000000	This readback is used to indicate the Field 0 VS width. Field 0 VS width is a vertical synchronization measurement. The unit for this measurement is half lines. This measurement is valid only when the vertical filter has locked. xxxxxxxxxxxxxx = The number of half lines in the VS Pulse of Field 0. (Divide readback by 2 to get number of lines)
FIELD1_VS_PULSE_WIDTH[13:0]		R
0x30 0x31	00000000 00000000	This readback is used to indicate the Field 1 VS width. Field 1 VS width is a vertical synchronization measurement. The unit for this measurement is half lines. This measurement is valid only when the vertical filter has locked. Field 1 measurements are valid when HDMI_INTERLACED is set to 1. xxxxxxxxxxxxxx = The number of half lines in the VS Pulse of Field 1. (Divide readback by 2 to get number of lines)
FIELD0_VS_BACK_PORCH[13:0]		R
0x32 0x33	00000000 00000000	This readback is used to indicate the Field 0 VS back porch width. Field 0 VS back porch width is a vertical synchronization measurement. The unit for this measurement is half lines. xxxxxxxxxxxxxx = The number of half lines in the VS Back Porch of Field 0. (Divide readback by 2 to get number of lines)
FIELD1_VS_BACK_PORCH[13:0]		R
0x34 0x35	00000000 00000000	This readback is used to indicate the Field 1 VS back porch width. Field 1 VS back porch width is a vertical synchronization measurement. The unit for this measurement is half lines. This measurement is valid only when the vertical filter has locked. Field 1 measurements are valid when HDMI_INTERLACED is set to 1. xxxxxxxxxxxxxx = The number of half lines in the VS Back Porch of Field 1. (Divide readback by 2 to get number of lines)
CS_DATA[39:0]		R
0x36 0x37 0x38 0x39 0x3A	00000000 00000000 00000000 00000000 00000000	This signal is a readback for the Channel Status data bits collected from audio channel 0.
OVERRIDE_DEEP_COLOR_MODE		R/W
0x40	00000000	This bit is used to override the Deep Color mode. 0 = The HDMI section unpacks the video data according to the Deep Color information extracted from the General Control packets. (Normal operation) 1 = Override the Deep Color mode extracted from the General Control Packet. The HDMI section unpacks the video data according to the Deep Color mode set in DEEP_COLOR_MODE_USER[1:0].

Reg	Bits	Description	
DEEP_COLOR_MODE_USER[1:0]			R/W
0x40	00<u>00</u>0000	This control is used to manually set the Deep Color mode. The value set in this register is effective when OVERRIDE_DEEP_COLOR_MODE is set to 1. 00 = 8-bits per channel 01 = 10-bits per channel 10 = Reserved 11 = Reserved	
DEREP_N_OVERRIDE			R/W
0x41	010<u>0</u>0000	This bit is used to allow the user to override the pixel repetition factor. DEREP_N is then used instead of HDMI_PIXEL_REPETITION[3:0] to discard video pixel data from the incoming HDMI/MHL stream. 0 = Automatic detection and processing of pixel repeated modes using the AVI InfoFrame information. 1 = Enables manual setting of the pixel repetition factor as per DEREP_N[3:0].	
DEREP_N[3:0]			R/W
0x41	0100<u>0000</u>	This signal is used to set the derepetition value if DEREP_N_OVERRIDE is set to 1. xxxx = DEREP_N + 1 indicates the pixel and clock discard factor	
QZERO_ITC_DIS			R/W
0x47	00000<u>0</u>00	This bit is used to select manual control of the RGB colorimetry when the AVI InfoFrame Field Q[1:0] = 00. To be used in conjunction with QZERO_RGB_FULL. 0 = AVI InfoFrame ITC bit determines RGB full or limited range when Q[1:0] = 00 1 = Manual RGB range as per QZERO_RGB_FULL	
QZERO_RGB_FULL			R/W
0x47	000000<u>0</u>0	This signal is used to manually select the HDMI/MHL colorimetry when the AVI InfoFrame Field Q[1:0] = 00. It is valid only when QZERO_ITC_DIS is set to 1. 0 = RGB limited range when Q[1:0] = 00 1 = RGB full when Q[1:0] = 00	
ALWAYS_STORE_INF			R/W
0x47	0000000<u>0</u>	This bit is used to force InfoFrames with checksum errors to be stored. 0 = Stores data from received InfoFrames only if their checksum is correct 1 = Always store the data from received InfoFrame regardless of their checksum	
DIS_CABLE_DET_RST			R/W
0x48	0<u>0</u>000000	This bit is used to disable the reset effects of cable detection. Set it to 1 if the 5 V pins are unused and left unconnected. 0 = Resets the HDMI section if the 5 V input pins are inactive 1 = Do not use the 5 V input pins as reset signal for the HDMI section	
GAMUT_IRQ_NEXT_FIELD			R/W
0x50	000<u>0</u>0000	This bit is used to set the NEW_GAMUT_MDATA_RAW interrupt to detect when the new contents are applicable to next field or to indicate that the Gamut packet is new. This is done using header information of the gamut packet. 0 = Interrupt flag indicates that Gamut packet is new 1 = Interrupt flag indicates that Gamut packet is to be applied next field	
CS_COPYRIGHT_MANUAL			R/W
0x50	000000<u>0</u>0	This bit is used to select automatic or manual setting of the copyright value of the channel status bit that is passed to the S/PDIF output. Manual control is set with the CS_COPYRIGHT_VALUE bit. 0 = Automatic channel status (CS) copyright control 1 = Manual CS copyright control. Manual value is set by CS_COPYRIGHT_VALUE	
CS_COPYRIGHT_VALUE			R/W
0x50	0000000<u>0</u>	This bit is used to set the value of the CS copyright bit in the S/PDIF output. This bit is only available when CS_COPYRIGHT_MANUAL is set to 1. 0 = Copyright value of channel status bit is 0. Valid only if CS_COPYRIGHT_MANUAL is set to 1 1 = Copyright value of channel status bit is 1. Valid only if CS_COPYRIGHT_MANUAL is set to 1	
TMDSFREQ[8:0]			R
0x51 0x52	<u>00000000</u> <u>0</u>0000000	This register is used to provide a full precision integer TMDS frequency measurement. xxxxxxxxx = 9-bit TMDS frequency measurement in MHz	
TMDSFREQ_FRAC[6:0]			R
0x52	<u>0</u>0000000	This readback is used to indicate the fractional bits of measured frequency of PLL recovered TMDS clock. The unit is 1/128 MHz. xxxxxxx = 7-bit TMDS fractional frequency measurement in 1/128 MHz	

Reg	Bits	Description	
HDMI_COLORSPACE[3:0]			R
0x53	0000<u>0000</u>	This signal is used to provide a readback of the HDMI/MHL input color space decoded from the AVI InfoFrame. 0000 = RGB_LIMITED 0001 = RGB_FULL 0010 = YUV_601 0011 = YUV_709 0100 = XVYCC_601 0101 = XVYCC_709 0110 = YUV_601_FULL 0111 = YUV_709_FULL 1000 = sYCC 601 1001 = Adobe YCC 601 1010 = Adobe RGB	
FILT_5V_DET_DIS			R/W
0x56	<u>0</u>1011000	This bit is used to disable the digital glitch filter on the HDMI 5 V detect signals. The filtered signals are used as interrupt flags and to reset the HDMI section. The filter works from an internal ring oscillator clock and is therefore available in power-down mode. The clock frequency of the ring oscillator is 42 MHz ± 10%. Note: If the 5 V pins are not used and left unconnected, disconnect the 5 V detect circuitry from the HDMI reset signal by setting DIS_CABLE_DET_RST to 1. This avoids holding the HDMI section in reset. 0 = Enabled 1 = Disabled	
FILT_5V_DET_TIMER[6:0]			R/W
0x56	0<u>1011000</u>	This bit is used to set the timer for the digital glitch filter on the HDMI 5 V detect inputs. The unit of this parameter is 2 clock cycles of the ring oscillator (~ 47 ns). The input must be constantly high for the duration of the timer, otherwise the filter output remains low. The output of the filter returns low as soon as any change in the 5 V power signal is detected. 1011000 = Approximately 4.2 µs xxxxxxx = Time duration of 5 V deglitch filter. The unit of this parameter is 2 clock cycles of the ring oscillator (~47 ns)	
HDCP_REPT_EDID_RESET			SC
0x5A	0000<u>0</u>000	This bit is used to reset the E-EDID/Repeater controller. When asserted it resets the E-EDID/Repeater controller. This is a self clearing bit. 0 = Normal operation 1 = Resets the E-EDID/Repeater controller	
DCFIFO_RECENTER			SC
0x5A	0000<u>0</u>000	This bit is used as a reset to recenter the Video FIFO. This is a self clearing bit. 0 = Video FIFO normal operation 1 = Video FIFO to recenter	
FORCE_N_UPDATE			SC
0x5A	0000000<u>0</u>	This control is used to force an N and CTS value update to the audio DPLL. The audio DPLL regenerates the audio clock. This is a self clearing bit. 0 = No effect 1 = Forces an update on the N and CTS values for audio clock regeneration	
CTS[19:0]			R
0x5B 0x5C 0x5D	<u>00000000</u> <u>00000000</u> <u>0000</u>0000	This control is used to provide a readback for the CTS value received in the HDMI/MHL data stream. xxxxxxxxxxxxxxxxxxxx = CTS value readback from HDMI/MHL stream	
N[19:0]			R
0x5D 0x5E 0x5F	0000<u>0000</u> <u>00000000</u> <u>00000000</u>	This control is used to provide a readback for the N value received in the HDMI/MHL data stream. xxxxxxxxxxxxxxxxxxxx = N value readback from HDMI/MHL stream	
HPA_DELAY_SEL[3:0]			R/W
0x6C	<u>1010</u>0011	This control is used to set a delay between 5 V detection and hot plug assertion on the HPA output pin in HDMI input mode, in increments of 100ms per bit. 0000 = No Delay 0001 = 100 ms Delay 0010 = 200 ms Delay 1010 = 1 sec Delay 1111 = 1.5 sec Delay	

Reg	Bits	Description
HPA_OVR_TERM		R/W
0x6C	10100011	This bit is used to set termination control to be overridden by the HPA setting in HDMI input mode. When this bit is set, termination on a specific port is set according to the HPA status of that port. 0 = Automatic or manual I²C control of port termination 1 = Termination controls disabled and overridden by HPA controls
HPA_AUTO_INT_EDID[1:0]		R/W
0x6C	10100011	This control is used to select the type of automatic control on the HPA output pin in HDMI input mode. This bit has no effect when HPA_MANUAL is set to 1 00 = The HPA of an HDMI port is asserted high immediately after the internal EDID has been activated for that port. The HPA of a specific HDMI port is deasserted low immediately after the internal E-EDID is de-activated for that port. 01 = The HPA of an HDMI port is asserted high following a programmable delay after the device detects an HDMI cable plug on that port. The HPA of an HDMI port is immediately de-asserted after the device detects a cable disconnect on that HDMI port. 10 = The HPA of an HDMI port is asserted high after two conditions have been met. The conditions are detailed as follows: The internal EDID is active for that Port. The delayed version of the cable detect signal CABLE_DET_X_RAW for that port is high. The HPA of an HDMI port is immediately deasserted after any of the following two conditions have been met: The internal EDID is deactivated for that port The cable detect signal CABLE_DET_X_RAW for that port is low. 11 = The HPA of an HDMI port is asserted high after three conditions have been met. The conditions are detailed as follows: The internal EDID is active for that port. The delayed version of the cable detect signal CABLE_DET_X_RAW for that port is high. The user has set the manual HPA control for that port to 1 via the HPA_MAN_VALUE_X controls. The HPA of an HDMI port is immediately deasserted after any of the following three conditions have been met: The internal EDID is deactivated for that port. The cable detect signal CABLE_DET_X_RAW for that port is low. The user sets the manual HPD control for that port to 0 via the HPA_MAN_VALUE_X controls.
HPA_MANUAL		R/W
0x6C	10100011	This bit is used to enable manual control for the Hot Plug Assert output pin in HDMI input mode. By setting this bit any automatic control of this pin is disabled. Manual control is determined by HPA_MAN_VALUE_PORT_A 0 = HPA takes its value based on HPA_AUTO_INT_EDID 1 = HPA takes its value from HPA_MAN_VALUE_PORT_A
I2S_TDM_MODE_ENABLE		R/W
0x6D	00000000	This bit is used to enable I²S time division multiplexing (TDM) output mode, where all four stereo pairs are output on the I2S_SDATA pin. This mode can only be used in multi-channel modes. In 8-channel mode, the maximum sample rate supported is 48 kHz. 0 = Disable TDM mode. Only stereo audio can be output on the I2S_SDATA output pin 1 = Enable TDM mode. Up to 8 audio channels (4 stereo pairs) is time multiplexed on the I2S_SDATA output pin
DDC_PWRDN[7:0]		R/W
0x73	00000000	This control is used to control the power-down for the DDC pads. 0 = Power up DDC pads. 1 = Power down DDC pads.
HDMI_TERMA_DISABLE		R/W
0x83	11111111	This bit is used to disable the termination on Port A in HDMI input mode. It can be used when TERM_AUTO is set to 0. 0 = Enable Termination Port A 1 = Disable Termination Port A
EQ_DYN_EN_A		R/W
0x89	00000000	This bit is used to set the HDMI/MHL equalizer mode for Port A. 0 = Disables equalizer dynamic mode. The equalizer is configured in static mode. This configuration is not recommended. 1 = Enables equalizer dynamic mode. This configuration is recommended.
EQ_DYN_FREQ2[3:0]		R/W
0x8A	10100011	This control is used to set the upper limit, Limit 2, for the HDMI/MHL equalizer dynamic control frequency range. The frequency must be specified in MHz divided by 16. 0000 = Reserved. Do not use. 1010 = Default dynamic equalizer frequency Limit 2. The default value corresponds to 160 MHz. xxxx = Frequency for Limit 2.

Reg	Bits	Description	
EQ_DYN_FREQ1[3:0]			R/W
0x8A	10100011	This control is used to set the lower limit, Limit 1, for the HDMI/MHL equalizer dynamic control frequency range. The frequency must be specified in MHz, divided by 16. 0000 = Reserved. Do not use. 0011 = Default dynamic equalizer frequency Limit 1. The default value corresponds to 48 MHz. xxxx = Frequency for Limit 1.	
HPA_MAN_VALUE_PORT_A			R/W
0xF8	0000000 0	Manual value for HPA Port A. 0 = HPA deasserted 1 = HPA asserted	
HPA_TRISTATE_PORT_A			R/W
0xF9	0000000 0	Manual value for HPA tristate Port A. 0 = HPA pin active 1 = Tristate HPA pin	

2.3 HDMI RX REPEATER MAP

Reg	Bits	Description	
BKSV[39:0]			R
0x00 0x01 0x02 0x03 0x04	00000000 00000000 00000000 00000000 00000000	This readback is used to indicate the HDMI/MHL receiver Key Selection Vector (BKSV). It can be read back once the device has successfully accessed the HDCP ROM. The following registers contain the BKSV read from the EEPROM. 0x00[7:0] = BKSV[7:0], 0x01[7:0] = BKSV[15:8], 0x02[7:0] = BKSV[23:16], 0x03[7:0] = BKSV[31:24] and 0x04[7:0] = BKSV[39:32]	
RI[15:0]			R
0x08 0x09	00000000 00000000	This readback is used to indicate the Ri generated by the HDCP core.	
PJ[7:0]			R
0x0A	00000000	This readback is used to indicate the Pj generated by HDCP core.	
AKSV[39:0]			R/W
0x10 0x11 0x12 0x13 0x14	00000000 00000000 00000000 00000000 00000000	This readback is used to indicate the AKSV of the transmitter attached to the active HDMI/MHL port. It can be read back after an AKSV update. The following registers contain the AKSV written by the Tx. 0x10[7:0] = AKSV[7:0], 0x11[7:0] = AKSV[15:8], 0x12[7:0] = AKSV[23:16], 0x13[7:0] = AKSV[31:24] and 0x14[7:0] = AKSV[39:32].	
AINFO[7:0]			R/W
0x15	00000000	This control is used to read back the AINFO written by Tx.	
AINFO_RB[7:0]			R
0x16	00000000	This readback is used to indicate the AINFO value. The value in the previous address is cleared after an AKSV update.	
AN[63:0]			R/W
0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This control contains the AN written by Tx.	
SHA_A[31:0]			R/W
0x20 0x21 0x22 0x23	00000000 00000000 00000000 00000000	This control contains the SHA Hash Part A generated by on-chip calculation.	
SHA_B[31:0]			R/W
0x24 0x25 0x26 0x27	00000000 00000000 00000000 00000000	This control contains the SHA Hash Part B generated by on-chip calculation.	
SHA_C[31:0]			R/W
0x28 0x29 0x2A 0x2B	00000000 00000000 00000000 00000000	This control contains the SHA Hash Part C generated by on-chip calculation.	
SHA_D[31:0]			R/W
0x2C 0x2D 0x2E 0x2F	00000000 00000000 00000000 00000000	This control contains the SHA Hash Part D generated by on-chip calculation.	

Reg	Bits	Description	
SHA_E[31:0]			R/W
0x30 0x31 0x32 0x33	00000000 00000000 00000000 00000000	This control contains the SHA Hash Part E generated by on-chip calculation.	
BCAPS[7:0]			R/W
0x40	10000011	This control contains the BCAPS presented to the Tx attached to the active HDMI/MHL port. 10000011 = Default BCAPS register value presented to the Tx xxxxxxxx = BCAPS register value presented to the Tx	
BSTATUS[15:0]			R/W
0x41 0x42	00000000 00000000	These registers contain the BSTATUS information presented to the Tx attached to the active HDMI/MHL port. Bits[11:0] must be set by the system software acting as a repeater. 0x41[7:0] = BSTATUS[7:0], 0x42[7:0] = BSTATUS[15:8] xxxxxxxxxxxxxxxx = BSTATUS register presented to Tx 0000000000000000 = Reset value; BSTATUS register is reset only after power up	
KSV_LIST_READY_PORT_A			R/W
0x69	0000000 0	The system sets this bit in order to indicate that the KSV list has been read from the Tx IC(s) and written into the Repeater Map. The system must also set Bits[11:0] of BSTATUS before setting this bit. This bit only applies to Port A	
PRIMARY_EDID_SIZE[3:0]			R/W
0x70	0000 0000	This control is used to specify the size for the primary EDID image, in 128-byte blocks. 0000 = EDID image not present 0001 = 128 bytes 0010 = 256 bytes 0011 = 384 bytes 0100 = 512 bytes Others = Reserved	
MAN_EDID_A_ENABLE			R/W
0x74	0000000 0	This bit is used to control the manual enable for I²C access to internal EDID ram from DDC port A, when the EDID has been loaded and CKSUM_CALC are finished. 0 = Manual enable not active for E-EDID on Port A 1 = Manual enable active for E-EDID on Port A	
EDID_A_ENABLED			R
0x76	0000000 0	This readback is used to indicate the enable state for EDID access on Port A, after a combination of manual and automatic functions. 0 = Disabled 1 = Enabled	
CLEAR_KSV_LIST			SC
0x77	0000 0000	This bit is used to clear the KSV list from memory.	
CKSUM_CALC			SC
0x77	00000 000	This bit is used to force a full re-calculation of all checksums for the internal E-EDID for all ports. 0 = No effect 1 = Calculate checksums of all internal EDID contents for all ports	
KSV_LIST_READY_CLR_A			SC
0x78	0000000 0	This bit is used to clear the BCAPS KSV list ready bit in Port A. 0 = No effect 1 = Clears BCAPS KSV list ready bit (Self clearing bit)	
EDID_SEGMENT_POINTER[2:0]			R/W
0x7A	000 0000	This control is used to select which 256-byte block is accessed via the main I²C, inside the space selected by EDID_SRAM_SPACE_SELECT. 000 = First 256 bytes of EDID data area 001 = Second 256 bytes of EDID data area	
DISABLE_AUTO_EDID			R/W
0x7A	0000 0000	This bit is used to disable all automatic enables for the internal E-EDID. 0 = Automatic enable of internal E-EDID on HDMI/MHL port when the device comes out of power-down mode 1 = Disable automatic enable of internal E-EDID on HDMI/MHL port when the device comes out of power-down mode	

Reg	Bits	Description	
EDID_SRAM_SPACE_SELECT[1:0]			R/W
0x7A	000000<u>00</u>	This control is used to select which SRAM memory bank is to be accessed via the main I²C from primary or secondary HDMI/MHL EDID. 00 = access primary HDMI/MHL EDID image 01 = access secondary HDMI/MHL EDID image 10 = Reserved 11 = Reserved	
DUAL_EDID_ENABLE_PORT_A			R/W
0x7D	0000000<u>0</u>	This bit selects which EDID data area is accessed (primary or secondary). This also affects SPA and checksum access. 0 = Primary EDID data area 1 = Secondary EDID data area	
SECONDARY_EDID_SIZE[3:0]			R/W
0x7E	<u>0000</u>0000	This control is used to specify the size for the secondary EDID image, in 128-byte blocks. 0000 = EDID image not present 0001 = 128 bytes 0010 = 256 bytes 0011 = 384 bytes 0100 = 512 bytes Others = Reserved	
KSV_BYTE_0[7:0]			R/W
0x80	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_1[7:0]			R/W
0x81	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_2[7:0]			R/W
0x82	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_3[7:0]			R/W
0x83	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_4[7:0]			R/W
0x84	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_5[7:0]			R/W
0x85	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_6[7:0]			R/W
0x86	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_7[7:0]			R/W
0x87	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_8[7:0]			R/W
0x88	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_9[7:0]			R/W
0x89	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_10[7:0]			R/W
0x8A	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_11[7:0]			R/W
0x8B	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_12[7:0]			R/W
0x8C	<u>00000000</u>	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

Reg	Bits	Description	
KSV_BYTE_13[7:0]			R/W
0x8D	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_14[7:0]			R/W
0x8E	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_15[7:0]			R/W
0x8F	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_16[7:0]			R/W
0x90	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_17[7:0]			R/W
0x91	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_18[7:0]			R/W
0x92	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_19[7:0]			R/W
0x93	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_20[7:0]			R/W
0x94	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_21[7:0]			R/W
0x95	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_22[7:0]			R/W
0x96	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_23[7:0]			R/W
0x97	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_24[7:0]			R/W
0x98	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_25[7:0]			R/W
0x99	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_26[7:0]			R/W
0x9A	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_27[7:0]			R/W
0x9B	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_28[7:0]			R/W
0x9C	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_29[7:0]			R/W
0x9D	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_30[7:0]			R/W
0x9E	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_31[7:0]			R/W
0x9F	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

Reg	Bits	Description	
KSV_BYTE_32[7:0]			R/W
0xA0	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_33[7:0]			R/W
0xA1	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_34[7:0]			R/W
0xA2	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_35[7:0]			R/W
0xA3	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_36[7:0]			R/W
0xA4	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_37[7:0]			R/W
0xA5	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_38[7:0]			R/W
0xA6	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_39[7:0]			R/W
0xA7	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_40[7:0]			R/W
0xA8	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_41[7:0]			R/W
0xA9	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_42[7:0]			R/W
0xAA	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_43[7:0]			R/W
0xAB	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_44[7:0]			R/W
0xAC	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_45[7:0]			R/W
0xAD	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_46[7:0]			R/W
0xAE	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_47[7:0]			R/W
0xAF	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_48[7:0]			R/W
0xB0	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_49[7:0]			R/W
0xB1	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_50[7:0]			R/W
0xB2	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

Reg	Bits	Description	
KSV_BYTE_51[7:0]			R/W
0xB3	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_52[7:0]			R/W
0xB4	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_53[7:0]			R/W
0xB5	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_54[7:0]			R/W
0xB6	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_55[7:0]			R/W
0xB7	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_56[7:0]			R/W
0xB8	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_57[7:0]			R/W
0xB9	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_58[7:0]			R/W
0xBA	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_59[7:0]			R/W
0xBB	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_60[7:0]			R/W
0xBC	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_61[7:0]			R/W
0xBD	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_62[7:0]			R/W
0xBE	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_63[7:0]			R/W
0xBF	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_64[7:0]			R/W
0xC0	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_65[7:0]			R/W
0xC1	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_66[7:0]			R/W
0xC2	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_67[7:0]			R/W
0xC3	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_68[7:0]			R/W
0xC4	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_69[7:0]			R/W
0xC5	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

Reg	Bits	Description	
KSV_BYTE_70[7:0]			R/W
0xC6	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_71[7:0]			R/W
0xC7	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_72[7:0]			R/W
0xC8	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_73[7:0]			R/W
0xC9	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_74[7:0]			R/W
0xCA	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_75[7:0]			R/W
0xCB	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_76[7:0]			R/W
0xCC	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_77[7:0]			R/W
0xCD	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_78[7:0]			R/W
0xCE	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_79[7:0]			R/W
0xCF	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_80[7:0]			R/W
0xD0	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_81[7:0]			R/W
0xD1	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_82[7:0]			R/W
0xD2	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_83[7:0]			R/W
0xD3	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_84[7:0]			R/W
0xD4	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_85[7:0]			R/W
0xD5	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_86[7:0]			R/W
0xD6	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_87[7:0]			R/W
0xD7	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_88[7:0]			R/W
0xD8	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

Reg	Bits	Description	
KSV_BYTE_89[7:0]			R/W
0xD9	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_90[7:0]			R/W
0xDA	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_91[7:0]			R/W
0xDB	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_92[7:0]			R/W
0xDC	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_93[7:0]			R/W
0xDD	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_94[7:0]			R/W
0xDE	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_95[7:0]			R/W
0xDF	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_96[7:0]			R/W
0xE0	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_97[7:0]			R/W
0xE1	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_98[7:0]			R/W
0xE2	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_99[7:0]			R/W
0xE3	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_100[7:0]			R/W
0xE4	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_101[7:0]			R/W
0xE5	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_102[7:0]			R/W
0xE6	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_103[7:0]			R/W
0xE7	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_104[7:0]			R/W
0xE8	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_105[7:0]			R/W
0xE9	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_106[7:0]			R/W
0xEA	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_107[7:0]			R/W
0xEB	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

Reg	Bits	Description	
KSV_BYTE_108[7:0]			R/W
0xEC	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_109[7:0]			R/W
0xED	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_110[7:0]			R/W
0xEE	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_111[7:0]			R/W
0xEF	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_112[7:0]			R/W
0xF0	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_113[7:0]			R/W
0xF1	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_114[7:0]			R/W
0xF2	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_115[7:0]			R/W
0xF3	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_116[7:0]			R/W
0xF4	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_117[7:0]			R/W
0xF5	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_118[7:0]			R/W
0xF6	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_119[7:0]			R/W
0xF7	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_120[7:0]			R/W
0xF8	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_121[7:0]			R/W
0xF9	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_122[7:0]			R/W
0xFA	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_123[7:0]			R/W
0xFB	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_124[7:0]			R/W
0xFC	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_125[7:0]			R/W
0xFD	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	
KSV_BYTE_126[7:0]			R/W
0xFE	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

Reg	Bits	Description	
KSV_BYTE_127[7:0]			R/W
0xFF	00000000	This readback displays a byte in the KSV list used for the HDCP repeater protocol.	

2.4 HDMI RX INFOFRAME MAP

Reg	Bits	Description	
AVI_INF_PB[223:0]			R
0x00	00000000	This readback is used to indicate the AVI InfoFrame Data.	
0x01	00000000		
0x02	00000000		
0x03	00000000		
0x04	00000000		
0x05	00000000		
0x06	00000000		
0x07	00000000		
0x08	00000000		
0x09	00000000		
0x0A	00000000		
0x0B	00000000		
0x0C	00000000		
0x0D	00000000		
0x0E	00000000		
0x0F	00000000		
0x10	00000000		
0x11	00000000		
0x12	00000000		
0x13	00000000		
0x14	00000000		
0x15	00000000		
0x16	00000000		
0x17	00000000		
0x18	00000000		
0x19	00000000		
0x1A	00000000		
0x1B	00000000		
AUD_INF_PB[111:0]			R
0x1C	00000000	This readback is used to indicate the Audio InfoFrame Data.	
0x1D	00000000		
0x1E	00000000		
0x1F	00000000		
0x20	00000000		
0x21	00000000		
0x22	00000000		
0x23	00000000		
0x24	00000000		
0x25	00000000		
0x26	00000000		
0x27	00000000		
0x28	00000000		
0x29	00000000		

Reg	Bits	Description	
SPD_INF_PB[223:0]			R
0x2A	00000000	This readback is used to indicate the Source Product Descriptor InfoFrame Data.	
0x2B	00000000		
0x2C	00000000		
0x2D	00000000		
0x2E	00000000		
0x2F	00000000		
0x30	00000000		
0x31	00000000		
0x32	00000000		
0x33	00000000		
0x34	00000000		
0x35	00000000		
0x36	00000000		
0x37	00000000		
0x38	00000000		
0x39	00000000		
0x3A	00000000		
0x3B	00000000		
0x3C	00000000		
0x3D	00000000		
0x3E	00000000		
0x3F	00000000		
0x40	00000000		
0x41	00000000		
0x42	00000000		
0x43	00000000		
0x44	00000000		
0x45	00000000		
MS_INF_PB[111:0]			R
0x46	00000000	This readback is used to indicate the Moving Picture Expert Group (MPEG) Source InfoFrame Data.	
0x47	00000000		
0x48	00000000		
0x49	00000000		
0x4A	00000000		
0x4B	00000000		
0x4C	00000000		
0x4D	00000000		
0x4E	00000000		
0x4F	00000000		
0x50	00000000		
0x51	00000000		
0x52	00000000		
0x53	00000000		

Reg	Bits	Description	
VS_INF_PB[223:0]			R
0x54	00000000	This readback is used to indicate the Vendor Specific InfoFrame Data.	
0x55	00000000		
0x56	00000000		
0x57	00000000		
0x58	00000000		
0x59	00000000		
0x5A	00000000		
0x5B	00000000		
0x5C	00000000		
0x5D	00000000		
0x5E	00000000		
0x5F	00000000		
0x60	00000000		
0x61	00000000		
0x62	00000000		
0x63	00000000		
0x64	00000000		
0x65	00000000		
0x66	00000000		
0x67	00000000		
0x68	00000000		
0x69	00000000		
0x6A	00000000		
0x6B	00000000		
0x6C	00000000		
0x6D	00000000		
0x6E	00000000		
0x6F	00000000		
ACP_PB[223:0]			R
0x70	00000000	This readback is used to indicate the ACP InfoFrame Data.	
0x71	00000000		
0x72	00000000		
0x73	00000000		
0x74	00000000		
0x75	00000000		
0x76	00000000		
0x77	00000000		
0x78	00000000		
0x79	00000000		
0x7A	00000000		
0x7B	00000000		
0x7C	00000000		
0x7D	00000000		
0x7E	00000000		
0x7F	00000000		
0x80	00000000		
0x81	00000000		
0x82	00000000		
0x83	00000000		
0x84	00000000		
0x85	00000000		
0x86	00000000		
0x87	00000000		
0x88	00000000		
0x89	00000000		
0x8A	00000000		
0x8B	00000000		

Reg	Bits	Description	
ISRC1_PB[223:0]			R
0x8C	00000000	This readback is used to indicate the ISRC 1 InfoFrame Data.	
0x8D	00000000		
0x8E	00000000		
0x8F	00000000		
0x90	00000000		
0x91	00000000		
0x92	00000000		
0x93	00000000		
0x94	00000000		
0x95	00000000		
0x96	00000000		
0x97	00000000		
0x98	00000000		
0x99	00000000		
0x9A	00000000		
0x9B	00000000		
0x9C	00000000		
0x9D	00000000		
0x9E	00000000		
0x9F	00000000		
0xA0	00000000		
0xA1	00000000		
0xA2	00000000		
0xA3	00000000		
0xA4	00000000		
0xA5	00000000		
0xA6	00000000		
0xA7	00000000		
ISRC2_PB[223:0]			R
0xA8	00000000	This readback is used to indicate the ISRC 2 InfoFrame Data.	
0xA9	00000000		
0xAA	00000000		
0xAB	00000000		
0xAC	00000000		
0xAD	00000000		
0xAE	00000000		
0xAF	00000000		
0xB0	00000000		
0xB1	00000000		
0xB2	00000000		
0xB3	00000000		
0xB4	00000000		
0xB5	00000000		
0xB6	00000000		
0xB7	00000000		
0xB8	00000000		
0xB9	00000000		
0xBA	00000000		
0xBB	00000000		
0xBC	00000000		
0xBD	00000000		
0xBE	00000000		
0xBF	00000000		
0xC0	00000000		
0xC1	00000000		
0xC2	00000000		
0xC3	00000000		

Reg	Bits	Description	
GBD[223:0]			R
0xC4	00000000	This readback is used to indicate the Gamut InfoFrame Data.	
0xC5	00000000		
0xC6	00000000		
0xC7	00000000		
0xC8	00000000		
0xC9	00000000		
0xCA	00000000		
0xCB	00000000		
0xCC	00000000		
0xCD	00000000		
0xCE	00000000		
0xCF	00000000		
0xD0	00000000		
0xD1	00000000		
0xD2	00000000		
0xD3	00000000		
0xD4	00000000		
0xD5	00000000		
0xD6	00000000		
0xD7	00000000		
0xD8	00000000		
0xD9	00000000		
0xDA	00000000		
0xDB	00000000		
0xDC	00000000		
0xDD	00000000		
0xDE	00000000		
0xDF	00000000		
AVI_PACKET_ID[7:0]			R/W
0xE0	10000010	This control is used to set the AVI InfoFrame ID. xxxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x00 to Address 0x1B	
AVI_INF_VERS[7:0]			R
0xE1	00000000	This readback is used to indicate the AVI InfoFrame Version.	
AVI_INF_LEN[7:0]			R
0xE2	00000000	This readback is used to indicate the AVI InfoFrame Length.	
AUD_PACKET_ID[7:0]			R/W
0xE3	10000100	This control is used to set the Audio InfoFrame ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x1C to Address 0x29 1xxxxxxx = Packet type value of InfoFrame stored in InfoFrame Map, Address 0x1C to Address 0x29	
AUD_INF_VERS[7:0]			R
0xE4	00000000	This readback is used to indicate the Audio InfoFrame Version.	
AUD_INF_LEN[7:0]			R
0xE5	00000000	This readback is used to indicate the Audio InfoFrame Length.	
SPD_PACKET_ID[7:0]			R/W
0xE6	10000011	This control is used to set the Source Product Descriptor InfoFrame ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x2A to Address 0x45 1xxxxxxx = Packet type value of InfoFrame stored in InfoFrame Map, Address 0x2A to Address 0x45	
SPD_INF_VERS[7:0]			R
0xE7	00000000	This readback is used to indicate the Source Product Descriptor InfoFrame Version.	
SPD_INF_LEN[7:0]			R
0xE8	00000000	This readback is used to indicate the Source Product Descriptor InfoFrame Length.	
MS_PACKET_ID[7:0]			R/W
0xE9	10000101	This control is used to set the MPEG Source InfoFrame ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x46 to Address 0x53 1xxxxxxx = Packet type value of InfoFrame stored in InfoFrame Map, Address 0x46 to Address 0x53	

Reg	Bits	Description	
MS_INF_VERS[7:0]			R
0xEA	00000000	This readback is used to indicate the MPEG Source InfoFrame Version.	
MS_INF_LEN[7:0]			R
0xEB	00000000	This readback is used to indicate the MPEG Source InfoFrame Length.	
VS_PACKET_ID[7:0]			R/W
0xEC	10000001	This control is used to set the Vendor Specific InfoFrame ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x54 to Address 0x6F 1xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x54 to Address 0x6F	
VS_INF_VERS[7:0]			R
0xED	00000000	This readback is used to indicate the Vendor Specific InfoFrame Version.	
VS_INF_LEN[7:0]			R
0xEE	00000000	This readback is used to indicate the Vendor Specific InfoFrame Length.	
ACP_PACKET_ID[7:0]			R/W
0xEF	00000100	This control is used to set the ACP Packet ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x70 to Address 0x8B 1xxxxxxx = Packet type value of InfoFrame stored in InfoFrame Map, Address 0x70 to Address 0x8B	
ACP_TYPE[7:0]			R
0xF0	00000000	This readback is used to indicate the ACP Type.	
ACP_HEADER2[7:0]			R
0xF1	00000000	This readback is used to indicate the ACP Header 2.	
ISRC1_PACKET_ID[7:0]			R/W
0xF2	00000101	This control is used to set the ISRC1 Packet ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0x8C to Address 0xA7 1xxxxxxx = Packet type value of InfoFrame stored in InfoFrame Map, Address 0x8C to Address 0xA7	
ISRC1_HEADER1[7:0]			R
0xF3	00000000	This readback is used to indicate the ISRC1 Header 1.	
ISRC1_HEADER2[7:0]			R
0xF4	00000000	This readback is used to indicate the ISRC1 Header 2.	
ISRC2_PACKET_ID[7:0]			R/W
0xF5	00000110	This control is used to set the ISRC2 Packet ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0xA8 to Address 0xC3 1xxxxxxx = Packet type value of InfoFrame stored in InfoFrame Map, Address 0xA8 to Address 0xC3	
ISRC2_HEADER1[7:0]			R
0xF6	00000000	This readback is used to indicate the ISRC2 Header 1.	
ISRC2_HEADER2[7:0]			R
0xF7	00000000	This readback is used to indicate the ISRC2 Header 2.	
GAMUT_PACKET_ID[7:0]			R/W
0xF8	00001010	This control is used to set the Gamut Metadata Packet ID. 0xxxxxxx = Packet type value of packet stored in InfoFrame Map, Address 0xC4 to Address 0xDF 1xxxxxxx = Packet type value of InfoFrame stored in InfoFrame Map, Address 0xC4 to Address 0xDF	
GAMUT_HEADER1[7:0]			R
0xF9	00000000	This readback is used to indicate the Gamut Metadata Header 1.	
GAMUT_HEADER2[7:0]			R
0xFA	00000000	This readback is used to indicate the Gamut Metadata Header 2.	

Reg	Bits	Description	
PKT_CNT_ID[7:0]			R/W
0xFD	10000001	Select which type of packet is going to be counted during each frame. Any packet header ID is supported to detect the count, but only collectable packets can be stored. Default is 0x81 for Vendor Specific InfoFrame.	
EN_PKT_CNT_SEL			R/W
0xFE	0000**0**000	This control is used to enable the feature whereby one can choose which of the multiple received packets per frame of TYPE PKT_CNT_ID is to be collected. This feature is only relevant for packet types that can be stored in the InfoFrame map. 0 = Disabled. Collect every packet as it is received 1 = Enable selectively collecting one of multiple packets per frame of the same type	
PKT_CNT_SEL[3:0]			R/W
0xFE	0000**0000**	This control is used to select which one of the multiple received packets per frame of TYPE PKT_CNT_ID is to be collected. Must be enabled with EN_PKT_CNT_SEL. It is recommended that PKT_CNT_ID be manually changed after receiving the corresponding new packet detect flag. 0000 = Select the 1st packet after VS rising edge … 1110 = Select the 15th packet after VS rising edge 1111 = Undefined	
RB_PKT_CNT[3:0]			R
0xFF	0000**0000**	This readback is used to indicate the count of number of packets of type PKT_CNT_ID per frame, as detected between VS rising edges. The count readback is supported for any PKT_CNT_ID, even those that cannot be stored in the InfoFrame map (for example, general control packets, and even custom ID packets). The count saturates to the 15 maximum, even if more packets are received. A count of zero means that no packet were detected.	

2.5 CBUS MAP

Reg	Bits	Description	
REQ_TX_CLEAR_BUFFERS			SC
0x00	0xxxxxxx	This control is a self clearing bit used to clear the Requester Tx buffer to start a new transaction 0 = Do not clear Requester Tx buffer 1 = Clear Requester Tx buffer	
REQ_RX_CLEAR_BUFFERS			SC
0x00	x0xxxxxx	This control is a self clearing bit used to clear the Requester Rx buffer to start a new transaction 0 = Do not clear Requester Rx buffer 1 = Clear Requester Rx buffer	
REQ_TX_BUFFER_SIZE[4:0]			SC
0x00	xx00000x	This self-clearing control can be used to set the number of command/data bytes to be sent in the next MSC Requester transaction 0x00 = No byte to be sent 0x01 = 1 byte to be sent 0x1F = 31 bytes to be sent	
REQ_TX_EN			SC
0x00	xxxxxxx0	This control is a self-clearing bit used to start a Requester Tx transmission. Assert it once the command/data has been filled 0 = Not ready to start Requester Tx transmission 1 = Start Requester Tx transmission	
RESP_TX_CLEAR_BUFFERS			SC
0x01	0xxxxxxx	This control is a self-clearing bit used to clear the Responder Tx buffer to start a new transaction 0 = Do not clear Responder Tx buffers 1 = Clear Responder Tx buffers	
RESP_RX_CLEAR_BUFFERS			SC
0x01	x0xxxxxx	This control is a self-clearing bit used to clear the Responder Rx buffer to start a new transaction 0 = Do not clear Responder Rx buffers 1 = Clear Responder Rx buffers	
RESP_TX_BUFFER_SIZE[4:0]			SC
0x01	xx00000x	This self-clearing control can be used to set the number of command/data bytes to be sent in next MSC Responder transaction 0x00 = No byte to be sent 0x01 = 1 byte to be sent 0x1F = 31 bytes to be sent	
RESP_TX_EN			SC
0x01	xxxxxxx0	This control is a self-clearing bit used to start a Responder Tx transmission. Assert it once the command/data has been filled 0 = Not ready to start Responder Tx transmission 1 = Start Responder Tx transmission	
EDID_READY			R
0x02	xxxx0xxx	This bit is a readback to indicate that the internal EDID has already been programmed 0 = EDID has not been programmed yet 1 = EDID has been programmed	
MSC_STATE[2:0]			R
0x02	xxxxx000	This control is a readback indicating the current status of the state machine controlling the MSC layer 000 = IDLE 001 = Responder Rx 010 = Responder Tx 011 = Requester Tx 100 = Requester Rx	
ENABLE_TIMEOUT_COUNTERS			R/W
0x03	0xxx1xxx	This control is used to enable the hardware timeout counters for CBUS packets transactions. When these counters are enabled, the meaning of the MSC_RESP_TX_PACKET_SENT interrupt changes. MSC_RESP_TX_PACKET_SENT is set to 1 when a timeout has occurred. 0 = Hardware timeout counters disabled 1 = Hardware timeout counters enabled	

Reg	Bits	Description	
CLEAR_BUFFERS_ON_ABORT			R/W
0x05	01100000	This control can be used to configure all the buffers to be automatically cleared after an MSC ABORT command has been received 0 = Responder/Requester buffers are not cleared after an MSC ABORT is received 1 = Responder/Requester buffers are automatically cleared after an MSC ABORT is received	
RB_MHL_MODE			R
0x06	0000000 0	Internal state of the MHL mode of operation 0 = HDMI/DVI mode 1 = MHL mode	
SPI_MODE[1:0]			R/W
0x07	00000011	This control can be used to set the SPI mode 00 = SPI Mode 0 (Data sampled on rising edge of SCLK) 01 = SPI Mode 1 (Data sampled on falling edge of SCLK) 10 = SPI Mode 2 (Data sampled on falling edge of SCLK) 11 = SPI Mode 3 (Data sampled on rising edge of SCLK)	
MHL_INTERRUPT_EN			R/W
0x0F	00001011	This control can be used to enable INTRQ3 to be used as the MHL CBUS interrupt 0 = INTRQ3 disabled 1 = INTRQ3 enabled	
INTRQ3_OP_SEL[1:0]			R/W
0x0F	00001011	This signal selects the interrupt signal configuration for INTRQ3. 00 = Open drain 01 = Drives low when active 10 = Drives high when active 11 = Disabled	
INTRQ3_DUR_SEL[1:0]			R/W
0x0F	00001011	This signal selects the interrupt signal duration for the interrupt signal on INTRQ3. 00 = 4 XTAL periods 01 = 16 XTAL periods 10 = 64 XTAL periods 11 = Active until cleared	
MSC_GOT_ABORT_ST			R
0x10	00000000	Latched status of MSC_GOT_ABORT interrupt. This bit is set to one after an ABORT MSC command is received. Once set, this bit remains high until the interrupt has been cleared via MSC_GOT_ABORT_CLR. This bit is only valid if enabled via the MSC_GOT_ABORT_MB interrupt mask bit. 0 = No MSC_GOT_ABORT interrupt generated 1 = MSC_GOT_ABORT interrupt has been generated	
MSC_GOT_NACK_ST			R
0x10	00000000	Latched status of MSC_GOT_NACK interrupt. This bit is set to one after a NACK MSC command is received. Once set, this bit remains high until the interrupt has been cleared via MSC_GOT_NACK_CLR. This bit is only valid if enabled via the MSC_GOT_NACK_MB interrupt mask bit. 0 = No MSC_GOT_NACK interrupt generated 1 = MSC_GOT_NACK interrupt has been generated	
MSC_STATE_REQ_RX_ST			R
0x10	00000000	Latched status of MSC_STATE_REQ_RX interrupt. This bit is set to one after the MSC Layer State transitioned into Requester Rx. Once set, this bit remains high until the interrupt has been cleared via MSC_STATE_REQ_RX_CLR. This bit is only valid if enabled via the MSC_STATE_REQ_RX_MB interrupt mask bit. 0 = No MSC_STATE_REQ_RX interrupt generated 1 = MSC_STATE_REQ_RX interrupt has been generated	
MSC_STATE_REQ_TX_ST			R
0x10	00000000	Latched status of MSC_STATE_REQ_TX interrupt. This bit is set to one after the MSC Layer State transitioned into Requester Tx. Once set, this bit remains high until the interrupt has been cleared via MSC_STATE_REQ_TX_CLR. This bit is only valid if enabled via the MSC_STATE_REQ_TX_MB interrupt mask bit. 0 = No MSC_STATE_REQ_TX interrupt generated 1 = MSC_STATE_REQ_TX interrupt has been generated	

Reg	Bits	Description	
MSC_STATE_RESP_TX_ST			R
0x10	0000**0**000	Latched status of MSC_STATE_RESP_TX interrupt. This bit is set to one after the MSC Layer State transitions into Responder Tx. Once set, this bit remains high until the interrupt has been cleared via MSC_STATE_RESP_TX_CLR. This bit is only valid if enabled via the MSC_STATE_RESP_TX_MB interrupt mask bit. 0 = No MSC_STATE_RESP_TX interrupt generated 1 = MSC_STATE_RESP_TX interrupt has been generated	
MSC_RESP_RX_SIZE_MATCH_ST			R
0x10	00000**0**00	Latched status of MSC_RESP_RX_SIZE_MATCH interrupt. This bit is set to one after the Responder Rx FIFO reaches the expected size set in RESP_EXP_RX_BUFF_SIZE. Once set, this bit remains high until the interrupt has been cleared via MSC_RESP_RX_SIZE_MATCH_CLR. This bit is only valid if enabled via the MSC_RESP_RX_SIZE_MATCH_MB interrupt mask bit. 0 = No MSC_RESP_RX_SIZE_MATCH interrupt generated 1 = MSC_RESP_RX_SIZE_MATCH interrupt has been generated	
MSC_STATE_RESP_RX_ST			R
0x10	000000**0**0	Latched status of MSC_STATE_RESP_RX interrupt. This bit is set to one after the MSC Layer State transitions into Responder Rx. Once set, this bit remains high until the interrupt has been cleared via MSC_STATE_RESP_RX_CLR. This bit is only valid if enabled via the MSC_STATE_RESP_RX_MB interrupt mask bit. 0 = No MSC_STATE_RESP_RX interrupt generated 1 = MSC_STATE_RESP_RX interrupt has been generated	
MSC_STATE_IDLE_ST			R
0x10	0000000**0**	Latched status of MSC_STATE_IDLE interrupt. This bit is set to one after the MSC Layer State transitions into an idle state, as a result of a NACK or a finished Requester or Responder transaction. Once set, this bit remains high until the interrupt has been cleared via MSC_STATE_IDLE_CLR. This bit is only valid if enabled via the MSC_STATE_IDLE_MB interrupt mask bit. 0 = No MSC_STATE_IDLE interrupt generated 1 = MSC_STATE_IDLE interrupt has been generated	
MSC_RESP_RX_PACKET_RECEIVED_ST			R
0x11	**0**0000000	Latched status of MSC_RESP_RX_PACKET_RECEIVED interrupt. This bit is set to one after the Responder Rx successfully receives a packet. Once set, this bit remains high until the interrupt has been cleared via MSC_RESP_RX_PACKET_RECEIVED_CLR. This bit is only valid if enabled via the MSC_RESP_RX_PACKET_RECEIVED_MB interrupt mask bit. 0 = No MSC_RESP_RX_PACKET_RECEIVED interrupt generated 1 = MSC_RESP_RX_PACKET_RECEIVED interrupt has been generated	
MSC_RESP_TX_PACKET_SENT_ST			R
0x11	0**0**000000	Latched status of MSC_RESP_TX_PACKET_SENT interrupt. When ENABLE_TIMEOUT_COUNTERS is set to 0, this bit is set to one after the Responder Tx successfully sends a packet. When ENABLE_TIMEOUT_COUNTERS is set to 1, this bit is set to one after a timeout has occurred. Once set, this bit remains high until the interrupt has been cleared via MSC_RESP_TX_PACKET_SENT_CLR. This bit is only valid if enabled via the MSC_RESP_TX_PACKET_SENT_MB interrupt mask bit. 0 = No MSC_RESP_TX_PACKET_SENT interrupt generated 1 = MSC_RESP_TX_PACKET_SENT interrupt has been generated	
MSC_REQ_RX_PACKET_RECEIVED_ST			R
0x11	00**0**00000	Latched status of MSC_REQ_RX_PACKET_RECEIVED interrupt. This bit is set to one after the Requester Rx successfully receives a packet. Once set, this bit remains high until the interrupt has been cleared via MSC_REQ_RX_PACKET_RECEIVED_CLR. This bit is only valid if enabled via the MSC_REQ_RX_PACKET_RECEIVED_MB interrupt mask bit. 0 = No MSC_REQ_RX_PACKET_RECEIVED interrupt generated 1 = MSC_REQ_RX_PACKET_RECEIVED interrupt has been generated	
MSC_REQ_TX_PACKET_SENT_ST			R
0x11	000**0**0000	Latched status of MSC_REQ_TX_PACKET_SENT interrupt. This bit is set to one after the Requester Tx successfully sends a packet. Once set, this bit remains high until the interrupt has been cleared via MSC_REQ_TX_PACKET_SENT_CLR. This bit is only valid if enabled via the MSC_REQ_TX_PACKET_SENT_MB interrupt mask bit. 0 = No MSC_REQ_TX_PACKET_SENT interrupt generated 1 = MSC_REQ_TX_PACKET_SENT interrupt has been generated	
NRETRY_NACKS_FOLLOW_ST			R
0x11	0000**0**000	Latched status of NRETRY_NACKS_FOLLOW interrupt. This bit is set to one when the Link layer has not been able to receive a packet after NRETRY. Once set, this bit remains high until the interrupt has been cleared via NRETRY_NACKS_FOLLOW_CLR. This bit is only valid if enabled via the NRETRY_NACKS_FOLLOW_MB interrupt mask bit. 0 = No NRETRY_NACKS_FOLLOW interrupt generated 1 = NRETRY_NACKS_FOLLOW interrupt has been generated	

Reg	Bits	Description	
NRETRY_NACKS_INIT_ST			R
0x11	000000**0**0	Latched status of NRETRY_NACKS_INIT interrupt. This bit is set to one when the link layer has not been able to send a packet after NRETRY. Once set, this bit remains high until the interrupt has been cleared via NRETRY_NACKS_INIT_CLR. This bit is only valid if enabled via the NRETRY_NACKS_INIT_MB interrupt mask bit. 0 = No NRETRY_NACKS_INIT interrupt generated 1 = NRETRY_NACKS_INIT interrupt has been generated	
MHL_ELECTRICAL_DISCOVERY_ST			R
0x12	0000000**0**	Latched status of MHL_ELECTRICAL_DISCOVERY interrupt. This bit is set to one after the full electrical discovery sequence completed successfully, or if the electrical discovery failed and stopped. Once set, this bit remains high until the interrupt has been cleared via MHL_ELECTRICAL_DISCOVERY_CLR. This bit is only valid if enabled via the MHL_ELECTRICAL_DISCOVERY_MB interrupt mask bit. 0 = No MHL_ELECTRICAL_DISCOVERY interrupt generated 1 = MHL_ELECTRICAL_DISCOVERY interrupt has been generated	
MHL_MODE_DETECTED_ST			R
0x12	0000000**0**	Latched status of MHL_MODE_DETECTED interrupt. This bit is set to one after a high value was detected on CD_SENSE, or after the MHL mode was manually forced. Once set, this bit remains high until the interrupt has been cleared via MHL_MODE_DETECTED_CLR. This bit is only valid if enabled via the MHL_MODE_DETECTED_MB interrupt mask bit. 0 = No MHL_MODE_DETECTED interrupt generated 1 = MHL_MODE_DETECTED interrupt has been generated	
MSC_GOT_ABORT_CLR			SC
0x13	**0**0000000	Clear bit for MSC_GOT_ABORT interrupt signal. 0 = Do not clear MSC_GOT_ABORT_ST 1 = Clear MSC_GOT_ABORT_ST	
MSC_GOT_NACK_CLR			SC
0x13	0**0**000000	Clear bit for MSC_GOT_NACK interrupt signal. 0 = Do not clear MSC_GOT_NACK_ST 1 = Clear MSC_GOT_NACK_ST	
MSC_STATE_REQ_RX_CLR			SC
0x13	00**0**00000	Clear bit for MSC_STATE_REQ_RX interrupt signal. 0 = Do not clear MSC_STATE_REQ_RX_ST 1 = Clear MSC_STATE_REQ_RX_ST	
MSC_STATE_REQ_TX_CLR			SC
0x13	000**0**0000	Clear bit for MSC_STATE_REQ_TX interrupt signal. 0 = Do not clear MSC_STATE_REQ_TX_ST 1 = Clear MSC_STATE_REQ_TX_ST	
MSC_STATE_RESP_TX_CLR			SC
0x13	0000**0**000	Clear bit for MSC_STATE_RESP_TX interrupt signal. 0 = Do not clear MSC_STATE_RESP_TX 1 = Clear MSC_STATE_RESP_TX	
MSC_RESP_RX_SIZE_MATCH_CLR			SC
0x13	00000**0**00	Clear bit for MSC_RESP_RX_SIZE_MATCH interrupt signal. 0 = Do not clear MSC_RESP_RX_SIZE_MATCH_ST 1 = Clear MSC_RESP_RX_SIZE_MATCH_ST	
MSC_STATE_RESP_RX_CLR			SC
0x13	000000**0**0	Clear bit for MSC_STATE_RESP_RX interrupt signal. 0 = Do not clear MSC_STATE_RESP_RX 1 = Clear MSC_STATE_RESP_RX	
MSC_STATE_IDLE_CLR			SC
0x13	0000000**0**	Clear bit for MSC_STATE_IDLE interrupt signal. 0 = Do not clear MSC_STATE_IDLE_ST 1 = Clear MSC_STATE_IDLE_ST	
MSC_RESP_RX_PACKET_RECEIVED_CLR			SC
0x14	**0**0000000	Clear bit for MSC_RESP_RX_PACKET_RECEIVED interrupt signal. 0 = Do not clear MSC_RESP_RX_PACKET_RECEIVED_ST 1 = Clear MSC_RESP_RX_PACKET_RECEIVED_ST	

Reg	Bits	Description	
MSC_RESP_TX_PACKET_SENT_CLR			SC
0x14	0<u>0</u>000000	Clear bit for MSC_RESP_TX_PACKET_SENT interrupt signal. 0 = Do not clear MSC_RESP_TX_PACKET_SENT_ST 1 = Clear MSC_RESP_TX_PACKET_SENT_ST	
MSC_REQ_RX_PACKET_RECEIVED_CLR			SC
0x14	00<u>0</u>00000	Clear bit for MSC_REQ_RX_PACKET_RECEIVED interrupt signal. 0 = Do not clear MSC_REQ_RX_PACKET_RECEIVED_ST 1 = Clear MSC_REQ_RX_PACKET_RECEIVED_ST	
MSC_REQ_TX_PACKET_SENT_CLR			SC
0x14	000<u>0</u>0000	Clear bit for MSC_REQ_TX_PACKET_SENT interrupt signal. 0 = Do not clear MSC_REQ_TX_PACKET_SENT_ST 1 = Clear MSC_REQ_TX_PACKET_SENT_ST	
NRETRY_NACKS_FOLLOW_CLR			SC
0x14	0000<u>0</u>000	Clear bit for NRETRY_NACKS_FOLLOW interrupt signal. 0 = Do not clear NRETRY_NACKS_FOLLOW_ST 1 = Clear NRETRY_NACKS_FOLLOW_ST	
NRETRY_NACKS_INIT_CLR			SC
0x14	00000<u>0</u>00	Clear bit for NRETRY_NACKS_INIT_CLR interrupt signal. 0 = Do not clear NRETRY_NACKS_INIT_CLR_ST 1 = Clear NRETRY_NACKS_INIT_CLR_ST	
MHL_ELECTRICAL_DISCOVERY_CLR			SC
0x15	0000000<u>0</u>	Clear bit for MHL_ELECTRICAL_DISCOVERY interrupt signal. 0 = Do not clear MHL_ELECTRICAL_DISCOVERY_ST 1 = Clear MHL_ELECTRICAL_DISCOVERY_ST	
MSC_MODE_DETECTED_CLR			SC
0x15	0000000<u>0</u>	Clear bit for MHL_MODE_DETECTED interrupt signal. 0 = Do not clear MHL_MODE_DETECTED_ST 1 = Clear MHL_MODE_DETECTED_ST	
MSC_GOT_ABORT_MB			R/W
0x16	<u>0</u>0000000	INTRQ3 interrupt mask for MSC_GOT_ABORT interrupt. When set, the MSC_GOT_ABORT interrupt triggers the INTRQ3 interrupt and MSC_GOT_ABORT_ST indicates the interrupt status. 0 = Disable MSC_GOT_ABORT interrupt for INTRQ3 1 = Enable MSC_GOT_ABORT interrupt for INTRQ3	
MSC_GOT_NACK_MB			R/W
0x16	0<u>0</u>000000	INTRQ3 interrupt mask for MSC_GOT_NACK interrupt. When set, the MSC_GOT_NACK interrupt triggers the INTRQ3 interrupt and MSC_GOT_NACK_ST indicates the interrupt status. 0 = Disable MSC_GOT_NACK interrupt for INTRQ3 1 = Enable MSC_GOT_NACK interrupt for INTRQ3	
MSC_STATE_REQ_RX_MB			R/W
0x16	00<u>0</u>00000	INTRQ3 interrupt mask for MSC_STATE_REQ_RX interrupt. When set, the MSC_STATE_REQ_RX interrupt triggers the INTRQ3 interrupt and MSC_STATE_REQ_RX_ST indicates the interrupt status. 0 = Disable MSC_STATE_REQ_RX interrupt for INTRQ3 1 = Enable MSC_STATE_REQ_RX interrupt for INTRQ3	
MSC_STATE_REQ_TX_MB			R/W
0x16	000<u>0</u>0000	INTRQ3 interrupt mask for MSC_STATE_REQ_TX interrupt. When set, the MSC_STATE_REQ_TX interrupt triggers the INTRQ3 interrupt and MSC_STATE_REQ_TX_ST indicates the interrupt status. 0 = Disable MSC_STATE_REQ_TX interrupt for INTRQ3 1 = Enable MSC_STATE_REQ_TX interrupt for INTRQ3	
MSC_STATE_RESP_TX_MB			R/W
0x16	0000<u>0</u>000	INTRQ3 interrupt mask for MSC_STATE_RESP_TX interrupt. When set, the MSC_STATE_RESP_TX interrupt triggers the INTRQ3 interrupt and MSC_STATE_RESP_TX_ST indicates the interrupt status. 0 = Disable MSC_STATE_RESP_TX interrupt for INTRQ3 1 = Enable MSC_STATE_RESP_TX interrupt for INTRQ3	

Reg	Bits	Description
MSC_RESP_RX_SIZE_MATCH_MB		R/W
0x16	00000000	INTRQ3 interrupt mask for MSC_RESP_RX_SIZE_MATCH interrupt. When set, the MSC_RESP_RX_SIZE_MATCH interrupt triggers the INTRQ3 interrupt and MSC_RESP_RX_SIZE_MATCH_ST indicates the interrupt status. 0 = Disable MSC_RESP_RX_SIZE_MATCH interrupt for INTRQ3 1 = Enable MSC_RESP_RX_SIZE_MATCH interrupt for INTRQ3
MSC_STATE_RESP_RX_MB		R/W
0x16	00000000	INTRQ3 interrupt mask for MSC_STATE_RESP_RX interrupt. When set, the MSC_STATE_RESP_RX interrupt triggers the INTRQ3 interrupt and MSC_STATE_RESP_RX_ST indicates the interrupt status. 0 = Disable MSC_STATE_RESP_RX interrupt for INTRQ3 1 = Enable MSC_STATE_RESP_RX interrupt for INTRQ3
MSC_STATE_IDLE_MB		R/W
0x16	00000000	INTRQ3 interrupt mask for MSC_STATE_IDLE interrupt. When set, the MSC_STATE_IDLE interrupt triggers the INTRQ3 interrupt and MSC_STATE_IDLE_ST indicates the interrupt status. 0 = Disable MSC_STATE_IDLE interrupt for INTRQ3 1 = Enable MSC_STATE_IDLE interrupt for INTRQ3
MSC_RESP_RX_PACKET_RECEIVED_MB		R/W
0x17	00000000	INTRQ3 interrupt mask for MSC_RESP_RX_PACKET_RECEIVED interrupt. When set, the MSC_RESP_RX_PACKET_RECEIVED interrupt triggers the INTRQ3 interrupt and MSC_RESP_RX_PACKET_RECEIVED_ST indicates the interrupt status. 0 = Disable MSC_RESP_RX_PACKET_RECEIVED interrupt for INTRQ3 1 = Enable MSC_RESP_RX_PACKET_RECEIVED interrupt for INTRQ3
MSC_RESP_TX_PACKET_SENT_MB		R/W
0x17	00000000	INTRQ3 interrupt mask for MSC_RESP_TX_PACKET_SENT interrupt. When set, the MSC_RESP_TX_PACKET_SENT interrupt triggers the INTRQ3 interrupt and MSC_RESP_TX_PACKET_SENT_ST indicates the interrupt status. 0 = Disable MSC_RESP_TX_PACKET_SENT interrupt for INTRQ3 1 = Enable MSC_RESP_TX_PACKET_SENT interrupt for INTRQ3
MSC_REQ_RX_PACKET_RECEIVED_MB		R/W
0x17	00000000	INTRQ3 interrupt mask for MSC_REQ_RX_PACKET_RECEIVED interrupt. When set, the MSC_REQ_RX_PACKET_RECEIVED interrupt triggers the INTRQ3 interrupt and MSC_REQ_RX_PACKET_RECEIVED_ST indicates the interrupt status. 0 = Disable MSC_REQ_RX_PACKET_RECEIVED interrupt for INTRQ3 1 = Enable MSC_REQ_RX_PACKET_RECEIVED interrupt for INTRQ3
MSC_REQ_TX_PACKET_SENT_MB		R/W
0x17	00000000	INTRQ3 interrupt mask for MSC_REQ_TX_PACKET_SENT interrupt. When set, the MSC_REQ_TX_PACKET_SENT interrupt triggers the INTRQ3 interrupt and MSC_REQ_TX_PACKET_SENT_ST indicates the interrupt status. 0 = Disable MSC_REQ_TX_PACKET_SENT interrupt for INTRQ3 1 = Enable MSC_REQ_TX_PACKET_SENT interrupt for INTRQ3
NRETRY_NACKS_FOLLOW_MB		R/W
0x17	00000000	INTRQ3 interrupt mask for NRETRY_NACKS_FOLLOW interrupt. When set, the NRETRY_NACKS_FOLLOW interrupt triggers the INTRQ3 interrupt and NRETRY_NACKS_FOLLOW_ST indicates the interrupt status. 0 = Disable NRETRY_NACKS_FOLLOW interrupt for INTRQ3 1 = Enable NRETRY_NACKS_FOLLOW interrupt for INTRQ3
NRETRY_NACKS_INIT_MB		R/W
0x17	00000000	INTRQ3 interrupt mask for NRETRY_NACKS_INIT interrupt. When set, the NRETRY_NACKS_INIT interrupt triggers the INTRQ3 interrupt and NRETRY_NACKS_INIT_ST indicates the interrupt status. 0 = Disable NRETRY_NACKS_INIT interrupt for INTRQ3 1 = Enable NRETRY_NACKS_INIT interrupt for INTRQ3
MHL_ELECTRICAL_DISCOVERY_MB		R/W
0x18	00000000	INTRQ3 interrupt mask for MHL_ELECTRICAL_DISCOVERY interrupt. When set, the MHL_ELECTRICAL_DISCOVERY interrupt triggers the INTRQ3 interrupt and MHL_ELECTRICAL_DISCOVERY_ST indicates the interrupt status. 0 = Disable MHL_ELECTRICAL_DISCOVERY interrupt for INTRQ3 1 = Enable MHL_ELECTRICAL_DISCOVERY interrupt for INTRQ3
MHL_MODE_DETECTED_MB		R/W
0x18	00000000	INTRQ3 interrupt mask for MHL_MODE_DETECTED interrupt. When set, the MHL_MODE_DETECTED interrupt triggers the INTRQ3 interrupt and MHL_MODE_DETECTED_ST indicates the interrupt status. 0 = Disable MHL_MODE_DETECTED interrupt for INTRQ3 1 = Enable MHL_MODE_DETECTED interrupt for INTRQ3

Reg	Bits	Description	
DDC_ERROR_CODE[7:0]			R
0x1B	00000000	This signal is a readback indicating the DDC error code to be read by the MCU in order to send it to MHL Transmitter if requested. 0x02 = Unexpected/unsupported command time out error code 0x04 = DDC time out error code	
LINK_MODE_MUTED			R/W
0x1C	0000**0**000	This control is used to indicate that the content stream of the device is muted. 0 = MHL content is unmuted 1 = MHL content is muted	
LINK_MODE_PATH_EN			R/W
0x1C	0000**0**000	This control is used to indicate that the TMDS path of the device is not in use. 0 = MHL Rx is not ready to receive data yet 1 = MHL Rx is ready to receive data	
LINK_MODE_CLK_MODE[2:0]			R/W
0x1C	00000**000**	This control is used to indicate the clock mode on the link. 000 = Reserved 001 = Reserved 010 = Reserved 011 = Normal (24-bit) clock mode Other = Reserved	
STANDBY_IMPL_SEL[1:0]			R/W
0x25	000000**00**	This control can be used to select between State 4A, State 4B, or State 4C in standby mode. 000 = Implement State 4A 001 = Implement State 4B 010 = Implement State 4C	
SEND_ABORT			SC
0x26	000000**0**0	This control can be used to send an MSC ABORT command. 0 = Do not send MSC ABORT 1 = Send MSC ABORT	
RESTART_DISCOVERY			SC
0x26	0000000**0**	This self-clearing bit can be used to force the restart of a discovery sequence. 0 = Do no restart discovery sequence 1 = Restart discovery sequence	
SINK_STANDBY			R/W
0x27	**0**0000010	This control can be used to put the device in Standby state. 0 = Device is in Active mode 1 = Command device to go into Standby mode	
FLOAT_CBUS			R/W
0x27	0**0**000010	This control can be used to float CBUS. 0 = Do not float CBUS 1 = Float CBUS to prompt MHL Transmitter to restart discovery	
MANUAL_VBUS_VALUE			R/W
0x27	0000**0**010	This control can be used to manually set the value for VBUS. It is only active when MANUAL_VBUS_ENABLE is set to 1. 0 = VBUS Low 1 = VBUS High	
MANUAL_VBUS_ENABLE			R/W
0x27	000000**0**10	This control can be used to manually override VBUS. When this control is set to 1, the VBUS value can be set by MANUAL_VBUS_VALUE. 0 = VBUS is asserted automatically depending on the discovery sequence state 1 = VBUS takes value manually programmed by the MANUAL_VBUS_VALUE control	
DISABLE_WAKE_PULSES_SINK1			R/W
0x27	000000**1**10	This control can be used to disable the wake pulses detection in SINK1 state. 0 = Wake pulses are not required before discovery pulses 1 = Discovery sequence is only accepted after correct wake up pulses detected	

Reg	Bits	Description	
DISABLE_WAKE_PULSES_STBY			R/W
0x27	0000001<u>0</u>	This control can be used to disable the wake pulses detection in Standby mode. 0 = Wake up pulses get the device out of standby mode 1 = Wake up pulses do not get the device out of standby mode. Required to implement state SINK4B	
RB_SEQUENCE_STATE[2:0]			R
0x28	000<u>00000</u>	This control is a readback of the discovery sequencing state 0x0 = SINK0 0x1 = SINK1 0x2 = SINK2 0x3 = SINK3 0x4 = SINK4 0x5 = SINK5	
RB_SINK_STANDBY_STATUTS			R
0x29	0000000<u>0</u>	This control is a readback of the standby status. It indicates whether the device is in active mode or in standby mode 0 = Device is in active mode 1 = Device is in standby mode	
REQ_RX_DETECT_BUFFER_SIZE[4:0]			R
0x30	000<u>00000</u>	This signal is a readback indicating the number of command/data bytes received in the Requester Rx buffer 0x00 = No bytes received 0x01 = 1 byte received … 0x1F = 31 bytes received	
REQ_TX_LATCHED_BUFFER_SIZE[4:0]			R
0x31	000<u>00000</u>	This signal is a readback indicating the latched number of bytes to be sent from the Requester Tx buffer 0x00 = No bytes to be sent 0x01 = 1 byte to be sent … 0x1F = 31 bytes to be sent	
RESP_RX_DETECT_BUFFER_SIZE[4:0]			R
0x32	000<u>00000</u>	This signal is a readback indicating the number of command/data bytes received in the Responder Rx buffer 0x00 = No bytes received 0x01 = 1 byte received … 0x1F = 31 bytes received	
RESP_TX_LATCHED_BUFFER_SIZE[4:0]			R
0x33	000<u>00000</u>	This signal is a readback indicating the latched number of bytes to be sent from the Responder Tx buffer 0x00 = No bytes to be sent 0x01 = 1 byte to be sent … 0x1F = 31 bytes to be sent	
REQ_EXPECTED_RX_BUFF_SIZE[4:0]			SC
0x34	000<u>11111</u>	This self-clearing control can be used to set the number of bytes expected to be received in the Requester Rx buffer 0x00 = No bytes expected 0x01 = 1 byte expected … 0x1F = 31 bytes expected	
RESP_EXPECTED_RX_BUFF_SIZE[4:0]			SC
0x35	000<u>11111</u>	This self-clearing control can be used to set the number of bytes expected to be received in the Responder Rx buffer 0x00 = No bytes expected 0x01 = 1 byte expected … 0x1F = 31 bytes expected	
REQ_EXPECTED_RX_BUFF_SIZE_LATCHED[4:0]			R
0x36	000<u>11111</u>	This signal is a readback indicating the latched number of bytes expected to be received in the Requester Rx buffer 0x00 = No bytes expected 0x01 = 1 byte expected … 0x1F = 31 bytes expected	

Reg	Bits	Description
RESP_EXPECTED_RX_BUFF_SIZE_LATCHED[4:0]		R
0x37	00011111	This signal is a readback indicating the latched number of bytes expected to be received in the Responder Rx buffer 0x00 = No bytes expected 0x01 = 1 byte expected … 0x1F = 31 bytes expected
REQ_TX_CMD_DATAB_FLAGS[31:0]		R/W
0x40 0x41 0x42 0x43	00000000 00000000 00000000 00000000	Bit ID for each byte to be sent from the Requester Tx buffer. Each bit identifies one of the bytes in the buffer. For example, Bit 0 identifies Byte 0, Bit 1 identifies Byte 1, and so on. A bit set to 0 indicates that the corresponding byte contains data. A bit set to 1 indicates that the corresponding byte contains a command. 0x00000000 = All bytes in buffer are data bytes 0x00000001 = Byte 0 is a command byte, all other bytes in buffer are data bytes … 0xFFFFFFFF = All bytes in buffer are command bytes
REQ_RX_CMD_DATAB_FLAGS[31:0]		R
0x50 0x51 0x52 0x53	00000000 00000000 00000000 00000000	Bit ID for each byte being received in the Requester Rx buffer. Each bit identifies one of the bytes in the buffer. For example, Bit 0 identifies Byte 0, Bit 1 identifies Byte 1, and so on. A bit set to 0 indicates that the corresponding byte contains data. A bit set to 1 indicates that the corresponding byte contains a command. 0x00000000 = All bytes in buffer are data bytes 0x00000001 = Byte 0 is a command byte, all other bytes in buffer are data bytes … 0xFFFFFFFF = All bytes in buffer are command bytes
RESP_TX_CMD_DATAB_FLAGS[31:0]		R/W
0x60 0x61 0x62 0x63	00000000 00000000 00000000 00000000	Bit ID for each bytes to be sent from the Responder Tx buffer. Each bit identifies one of the bytes in the buffer. For example, Bit 0 identifies Byte 0, Bit 1 identifies Byte 1, and so on. A bit set to 0 indicates that the corresponding byte contains data. A bit set to 1 indicates that the corresponding byte contains a command. '0x00000000 = All bytes in buffer are data bytes 0x00000001 = Byte 0 is a command byte, all other bytes in buffer are data bytes … 0xFFFFFFFF = All bytes in buffer are command bytes
RESP_RX_CMD_DATAB_FLAGS[31:0]		R
0x70 0x71 0x72 0x73	00000000 00000000 00000000 00000000	Bit ID for each bytes being received in the Responder Rx buffer. Each bit identifies one of the bytes in the buffer. For example, Bit 0 identifies Byte 0, Bit 1 identifies Byte 1, and so on. A bit set to 0 indicates that the corresponding byte contains data. A bit set to 1 indicates that the corresponding byte contains a command. '0x00000000 = All bytes in buffer are data bytes 0x00000001 = Byte 0 is a command byte, all other bytes in buffer are data bytes … 0xFFFFFFFF = All bytes in buffer are command bytes
REQ_TX_BYTE_0[7:0]		R/W
0x80	00000000	Byte 0 of Requester Tx buffer.
REQ_TX_BYTE_1[7:0]		R/W
0x81	00000000	Byte 1 of Requester Tx buffer.
REQ_TX_BYTE_2[7:0]		R/W
0x82	00000000	Byte 2 of Requester Tx buffer.
REQ_TX_BYTE_3[7:0]		R/W
0x83	00000000	Byte 3 of Requester Tx buffer.
REQ_TX_BYTE_4[7:0]		R/W
0x84	00000000	Byte 4 of Requester Tx buffer.
REQ_TX_BYTE_5[7:0]		R/W
0x85	00000000	Byte 5 of Requester Tx buffer.
REQ_TX_BYTE_6[7:0]		R/W
0x86	00000000	Byte 6 of Requester Tx buffer.

Reg	Bits	Description	
REQ_TX_BYTE_7[7:0]			R/W
0x87	00000000	Byte 7 of Requester Tx buffer.	
REQ_TX_BYTE_8[7:0]			R/W
0x88	00000000	Byte 8 of Requester Tx buffer.	
REQ_TX_BYTE_9[7:0]			R/W
0x89	00000000	Byte 9 of Requester Tx buffer.	
REQ_TX_BYTE_10[7:0]			R/W
0x8A	00000000	Byte 10 of Requester Tx buffer.	
REQ_TX_BYTE_11[7:0]			R/W
0x8B	00000000	Byte 11 of Requester Tx buffer.	
REQ_TX_BYTE_12[7:0]			R/W
0x8C	00000000	Byte 12 of Requester Tx buffer.	
REQ_TX_BYTE_13[7:0]			R/W
0x8D	00000000	Byte 13 of Requester Tx buffer.	
REQ_TX_BYTE_14[7:0]			R/W
0x8E	00000000	Byte 14 of Requester Tx buffer.	
REQ_TX_BYTE_15[7:0]			R/W
0x8F	00000000	Byte 15 of Requester Tx buffer.	
REQ_TX_BYTE_16[7:0]			R/W
0x90	00000000	Byte 16 of Requester Tx buffer.	
REQ_TX_BYTE_17[7:0]			R/W
0x91	00000000	Byte 17 of Requester Tx buffer.	
REQ_TX_BYTE_18[7:0]			R/W
0x92	00000000	Byte 18 of Requester Tx buffer.	
REQ_TX_BYTE_19[7:0]			R/W
0x93	00000000	Byte 19 of Requester Tx buffer.	
REQ_TX_BYTE_20[7:0]			R/W
0x94	00000000	Byte 20 of Requester Tx buffer.	
REQ_TX_BYTE_21[7:0]			R/W
0x95	00000000	Byte 21 of Requester Tx buffer.	
REQ_TX_BYTE_22[7:0]			R/W
0x96	00000000	Byte 22 of Requester Tx buffer.	
REQ_TX_BYTE_23[7:0]			R/W
0x97	00000000	Byte 23 of Requester Tx buffer.	
REQ_TX_BYTE_24[7:0]			R/W
0x98	00000000	Byte 24 of Requester Tx buffer.	
REQ_TX_BYTE_25[7:0]			R/W
0x99	00000000	Byte 25 of Requester Tx buffer.	

Reg	Bits	Description	
REQ_TX_BYTE_26[7:0]			R/W
0x9A	00000000	Byte 26 of Requester Tx buffer.	
REQ_TX_BYTE_27[7:0]			R/W
0x9B	00000000	Byte 27 of Requester Tx buffer.	
REQ_TX_BYTE_28[7:0]			R/W
0x9C	00000000	Byte 28 of Requester Tx buffer.	
REQ_TX_BYTE_29[7:0]			R/W
0x9D	00000000	Byte 29 of Requester Tx buffer.	
REQ_TX_BYTE_30[7:0]			R/W
0x9E	00000000	Byte 30 of Requester Tx buffer.	
REQ_TX_BYTE_31[7:0]			R/W
0x9F	00000000	Byte 31 of Requester Tx buffer.	
REQ_RX_BYTE_0[7:0]			R
0xA0	00000000	Byte 0 of Requester Rx buffer.	
REQ_RX_BYTE_1[7:0]			R
0xA1	00000000	Byte 1 of Requester Rx buffer.	
REQ_RX_BYTE_2[7:0]			R
0xA2	00000000	Byte 2 of Requester Rx buffer.	
REQ_RX_BYTE_3[7:0]			R
0xA3	00000000	Byte 3 of Requester Rx buffer.	
REQ_RX_BYTE_4[7:0]			R
0xA4	00000000	Byte 4 of Requester Rx buffer.	
REQ_RX_BYTE_5[7:0]			R
0xA5	00000000	Byte 5 of Requester Rx buffer.	
REQ_RX_BYTE_6[7:0]			R
0xA6	00000000	Byte 6 of Requester Rx buffer.	
REQ_RX_BYTE_7[7:0]			R
0xA7	00000000	Byte 7 of Requester Rx buffer.	
REQ_RX_BYTE_8[7:0]			R
0xA8	00000000	Byte 8 of Requester Rx buffer.	
REQ_RX_BYTE_9[7:0]			R
0xA9	00000000	Byte 9 of Requester Rx buffer.	
REQ_RX_BYTE_10[7:0]			R
0xAA	00000000	Byte 10 of Requester Rx buffer.	
REQ_RX_BYTE_11[7:0]			R
0xAB	00000000	Byte 11 of Requester Rx buffer.	
REQ_RX_BYTE_12[7:0]			R
0xAC	00000000	Byte 12 of Requester Rx buffer.	

Reg	Bits	Description	
REQ_RX_BYTE_13[7:0]			R
0xAD	00000000	Byte 13 of Requester Rx buffer.	
REQ_RX_BYTE_14[7:0]			R
0xAE	00000000	Byte 14 of Requester Rx buffer.	
REQ_RX_BYTE_15[7:0]			R
0xAF	00000000	Byte 15 of Requester Rx buffer.	
REQ_RX_BYTE_16[7:0]			R
0xB0	00000000	Byte 16 of Requester Rx buffer.	
REQ_RX_BYTE_17[7:0]			R
0xB1	00000000	Byte 17 of Requester Rx buffer.	
REQ_RX_BYTE_18[7:0]			R
0xB2	00000000	Byte 18 of Requester Rx buffer.	
REQ_RX_BYTE_19[7:0]			R
0xB3	00000000	Byte 19 of Requester Rx buffer.	
REQ_RX_BYTE_20[7:0]			R
0xB4	00000000	Byte 20 of Requester Rx buffer.	
REQ_RX_BYTE_21[7:0]			R
0xB5	00000000	Byte 21 of Requester Rx buffer.	
REQ_RX_BYTE_22[7:0]			R
0xB6	00000000	Byte 22 of Requester Rx buffer.	
REQ_RX_BYTE_23[7:0]			R
0xB7	00000000	Byte 23 of Requester Rx buffer.	
REQ_RX_BYTE_24[7:0]			R
0xB8	00000000	Byte 24 of Requester Rx buffer.	
REQ_RX_BYTE_25[7:0]			R
0xB9	00000000	Byte 25 of Requester Rx buffer.	
REQ_RX_BYTE_26[7:0]			R
0xBA	00000000	Byte 26 of Requester Rx buffer.	
REQ_RX_BYTE_27[7:0]			R
0xBB	00000000	Byte 27 of Requester Rx buffer.	
REQ_RX_BYTE_28[7:0]			R
0xBC	00000000	Byte 28 of Requester Rx buffer.	
REQ_RX_BYTE_29[7:0]			R
0xBD	00000000	Byte 29 of Requester Rx buffer.	
REQ_RX_BYTE_30[7:0]			R
0xBE	00000000	Byte 30 of Requester Rx buffer.	
REQ_RX_BYTE_31[7:0]			R
0xBF	00000000	Byte 31 of Requester Rx buffer.	

Reg	Bits	Description	
RESP_TX_BYTE_0[7:0]			R/W
0xC0	00000000	Byte 0 of Responder Tx buffer.	
RESP_TX_BYTE_1[7:0]			R/W
0xC1	00000000	Byte 1 of Responder Tx buffer.	
RESP_TX_BYTE_2[7:0]			R/W
0xC2	00000000	Byte 2 of Responder Tx buffer.	
RESP_TX_BYTE_3[7:0]			R/W
0xC3	00000000	Byte 3 of Responder Tx buffer.	
RESP_TX_BYTE_4[7:0]			R/W
0xC4	00000000	Byte 4 of Responder Tx buffer.	
RESP_TX_BYTE_5[7:0]			R/W
0xC5	00000000	Byte 5 of Responder Tx buffer.	
RESP_TX_BYTE_6[7:0]			R/W
0xC6	00000000	Byte 6 of Responder Tx buffer.	
RESP_TX_BYTE_7[7:0]			R/W
0xC7	00000000	Byte 7 of Responder Tx buffer.	
RESP_TX_BYTE_8[7:0]			R/W
0xC8	00000000	Byte 8 of Responder Tx buffer.	
RESP_TX_BYTE_9[7:0]			R/W
0xC9	00000000	Byte 9 of Responder Tx buffer.	
RESP_TX_BYTE_10[7:0]			R/W
0xCA	00000000	Byte 10 of Responder Tx buffer.	
RESP_TX_BYTE_11[7:0]			R/W
0xCB	00000000	Byte 11 of Responder Tx buffer.	
RESP_TX_BYTE_12[7:0]			R/W
0xCC	00000000	Byte 12 of Responder Tx buffer.	
RESP_TX_BYTE_13[7:0]			R/W
0xCD	00000000	Byte 13 of Responder Tx buffer.	
RESP_TX_BYTE_14[7:0]			R/W
0xCE	00000000	Byte 14 of Responder Tx buffer.	
RESP_TX_BYTE_15[7:0]			R/W
0xCF	00000000	Byte 15 of Responder Tx buffer.	
RESP_TX_BYTE_16[7:0]			R/W
0xD0	00000000	Byte 16 of Responder Tx buffer.	
RESP_TX_BYTE_17[7:0]			R/W
0xD1	00000000	Byte 17 of Responder Tx buffer.	
RESP_TX_BYTE_18[7:0]			R/W
0xD2	00000000	Byte 18 of Responder Tx buffer.	

Reg	Bits	Description	
RESP_TX_BYTE_19[7:0]			R/W
0xD3	00000000	Byte 19 of Responder Tx buffer.	
RESP_TX_BYTE_20[7:0]			R/W
0xD4	00000000	Byte 20 of Responder Tx buffer.	
RESP_TX_BYTE_21[7:0]			R/W
0xD5	00000000	Byte 21 of Responder Tx buffer.	
RESP_TX_BYTE_22[7:0]			R/W
0xD6	00000000	Byte 22 of Responder Tx buffer.	
RESP_TX_BYTE_23[7:0]			R/W
0xD7	00000000	Byte 23 of Responder Tx buffer.	
RESP_TX_BYTE_24[7:0]			R/W
0xD8	00000000	Byte 24 of Responder Tx buffer.	
RESP_TX_BYTE_25[7:0]			R/W
0xD9	00000000	Byte 25 of Responder Tx buffer.	
RESP_TX_BYTE_26[7:0]			R/W
0xDA	00000000	Byte 26 of Responder Tx buffer.	
RESP_TX_BYTE_27[7:0]			R/W
0xDB	00000000	Byte 27 of Responder Tx buffer.	
RESP_TX_BYTE_28[7:0]			R/W
0xDC	00000000	Byte 28 of Responder Tx buffer.	
RESP_TX_BYTE_29[7:0]			R/W
0xDD	00000000	Byte 29 of Responder Tx buffer.	
RESP_TX_BYTE_30[7:0]			R/W
0xDE	00000000	Byte 30 of Responder Tx buffer.	
RESP_TX_BYTE_31[7:0]			R/W
0xDF	00000000	Byte 31 of Responder Tx buffer.	
RESP_RX_BYTE_0[7:0]			R
0xE0	00000000	Byte 0 of Responder Rx buffer.	
RESP_RX_BYTE_1[7:0]			R
0xE1	00000000	Byte 1 of Responder Rx buffer.	
RESP_RX_BYTE_2[7:0]			R
0xE2	00000000	Byte 2 of Responder Rx buffer.	
RESP_RX_BYTE_3[7:0]			R
0xE3	00000000	Byte 3 of Responder Rx buffer.	
RESP_RX_BYTE_4[7:0]			R
0xE4	00000000	Byte 4 of Responder Rx buffer.	
RESP_RX_BYTE_5[7:0]			R
0xE5	00000000	Byte 5 of Responder Rx buffer.	

Reg	Bits	Description	
RESP_RX_BYTE_6[7:0]			R
0xE6	00000000	Byte 6 of Responder Rx buffer.	
RESP_RX_BYTE_7[7:0]			R
0xE7	00000000	Byte 7 of Responder Rx buffer.	
RESP_RX_BYTE_8[7:0]			R
0xE8	00000000	Byte 8 of Responder Rx buffer.	
RESP_RX_BYTE_9[7:0]			R
0xE9	00000000	Byte 9 of Responder Rx buffer.	
RESP_RX_BYTE_10[7:0]			R
0xEA	00000000	Byte 10 of Responder Rx buffer.	
RESP_RX_BYTE_11[7:0]			R
0xEB	00000000	Byte 11 of Responder Rx buffer.	
RESP_RX_BYTE_12[7:0]			R
0xEC	00000000	Byte 12 of Responder Rx buffer.	
RESP_RX_BYTE_13[7:0]			R
0xED	00000000	Byte 13 of Responder Rx buffer.	
RESP_RX_BYTE_14[7:0]			R
0xEE	00000000	Byte 14 of Responder Rx buffer.	
RESP_RX_BYTE_15[7:0]			R
0xEF	00000000	Byte 15 of Responder Rx buffer.	
RESP_RX_BYTE_16[7:0]			R
0xF0	00000000	Byte 16 of Responder Rx buffer.	
RESP_RX_BYTE_17[7:0]			R
0xF1	00000000	Byte 17 of Responder Rx buffer.	
RESP_RX_BYTE_18[7:0]			R
0xF2	00000000	Byte 18 of Responder Rx buffer.	
RESP_RX_BYTE_19[7:0]			R
0xF3	00000000	Byte 19 of Responder Rx buffer.	
RESP_RX_BYTE_20[7:0]			R
0xF4	00000000	Byte 20 of Responder Rx buffer.	
RESP_RX_BYTE_21[7:0]			R
0xF5	00000000	Byte 21 of Responder Rx buffer.	
RESP_RX_BYTE_22[7:0]			R
0xF6	00000000	Byte 22 of Responder Rx buffer.	
RESP_RX_BYTE_23[7:0]			R
0xF7	00000000	Byte 23 of Responder Rx buffer.	
RESP_RX_BYTE_24[7:0]			R
0xF8	00000000	Byte 24 of Responder Rx buffer.	

Reg	Bits	Description	
RESP_RX_BYTE_25[7:0]			R
0xF9	00000000	Byte 25 of Responder Rx buffer.	
RESP_RX_BYTE_26[7:0]			R
0xFA	00000000	Byte 26 of Responder Rx buffer.	
RESP_RX_BYTE_27[7:0]			R
0xFB	00000000	Byte 27 of Responder Rx buffer.	
RESP_RX_BYTE_28[7:0]			R
0xFC	00000000	Byte 28 of Responder Rx buffer.	
RESP_RX_BYTE_29[7:0]			R
0xFD	00000000	Byte 29 of Responder Rx buffer.	
RESP_RX_BYTE_30[7:0]			R
0xFE	00000000	Byte 30 of Responder Rx buffer.	
RESP_RX_BYTE_31[7:0]			R
0xFF	00000000	Byte 31 of Responder Rx buffer.	

2.6 SDP MAIN MAP

Reg	Bits	Description	
INSEL[4:0]			R/W
0x00	00001110	This control is used to select an input channel as well as the input format. 00000 = CVBS on AIN1 00001 = CVBS on AIN2 00010 = CVBS on AIN3 00011 = CVBS on AIN4 00100 = CVBS on AIN5 00101 = CVBS on AIN6 00110 = CVBS on AIN7 00111 = CVBS on AIN8 01000 = Y input on AIN1, C input on AIN2 01001 = Y input on AIN3, C input on AIN4 01010 = Y input on AIN5, C input on AIN6 01011 = Y input on AIN7, C input on AIN8 01100 = Y input on AIN1, Pb input on AIN2, Pr input on AIN3 01101 = Y input on AIN4, Pb input on AIN5, Pr input on AIN6 01110 = Differential positive on AIN1, differential negative on AIN2 01111 = Differential positive on AIN3, differential negative on AIN4 10000 = Differential positive on AIN5, differential negative on AIN6 10001 = Differential positive on AIN7, differential negative on AIN8	
ENHSPLL			R/W
0x01	11001000	This control is used to enable the HS processor. 0 = Disable HS PLL 1 = Enable HS PLL	
BETACAM			R/W
0x01	11001000	This control is used to set the target value for the AGC operation. 0 = YUV selected as input format. Selecting PAL with pedestal selects MII. Selecting PAL without pedestal selects SMPTE. Selecting NTSC with pedestal selects MII. Selecting NTSC without pedestal selects SMPTE. 1 = YUV selected as input format. Selecting PAL with pedestal selects BETACAM. Selecting PAL without pedestal selects BETACAM variant. Selecting NTSC with pedestal selects BETACAM. Selecting NTSC without pedestal selects BETACAM variant.	
ENVSPROC			R/W
0x01	11001000	This control is used to enable the VS processor. 0 = Disable VS processor 1 = Enable VS processor	
SQPE			R/W
0x01	11001000	The SQPE bit enables the square pixel operation. Square pixel mode is used to compensate for the different aspect ratios of standard television sets (4:3) vs. computer screens (4:4). The different aspect ratios mean that their respective pixels are rectangular (TV) or square (monitor). 0 = Select standard mode 1 = Enable square pixel modes	
VID_SEL[3:0]			R/W
0x02	00000100	This control is used to force the digital core into a specific video standard. This is not necessary under normal circumstances. 0000 = Autodetect PAL B/G/H/I/D, NTSC J (no pedestal), SECAM 0001 = Autodetect PAL B/G/H/I/D, NTSC M (with pedestal), SECAM 0010 = Autodetect PAL N (pedestal), NTSC J (no pedestal), SECAM 0011 = Autodetect PAL N (pedestal), NTSC M (with pedestal), SECAM 0100 = NTSC J 0101 = NTSC M 0110 = PAL 60 0111 = NTSC 4.43 1000 = PAL B/G/H/I/D 1001 = PAL N = PAL B/G/H/I/D (with pedestal) 1010 = PAL M (without pedestal) 1011 = PAL M 1100 = PAL combination N 1101 = PAL combination N (with pedestal) 1110 = SECAM 1111 = SECAM (with pedestal)	

Reg	Bits	Description
VBI_EN		R/W
0x03	01001100	This control is used to pass VBI data, such as CGMS and closed captioning (CCAP), through the luma channel of the SDP decoder. 0 = All video lines filtered or scaled 1 = Only active video region filtered or scaled
BT656_4		R/W
0x04	00110101	This control is used to change the operation of the V bit in accordance with ITU-R BT.656-4. 0 = ITU-R BT.656-3 specification: V bit goes low at EAV of Line 10 and Line 273 1 = ITU-R BT.656-4 specification: V bit goes low at EAV of Line 20 and Line 283
BL_C_VBI		R/W
0x04	00110101	This control is used to set the output of the chroma channels during the VBI region. 0 = Decode and output color during VBI 1 = Blank Cr and Cb value during VBI (no color, 0x80)
RANGE		R/W
0x04	00110101	This control is used to limit the range of values output by the SD core to the recommended value range. It is, however, ensured in any case that the reserved values of 255d (0xFF) and 00d (0x00) are not presented on the output pins unless they are part of an AV code header. 0 = ITU-R BT.656 output range (16 ≤ Y ≤ 235, 16 ≤ C/P ≤ 240) 1 = Extended output range (1 ≤ Y ≤ 254, 1 ≤ C/P ≤ 254)
AD_SEC525_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of 525-line SECAM. 0 = Disable autodetection of 525-line system with SECAM style, frequency modulated color component 1 = Enable detection
AD_SECAM_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of SECAM. 0 = Disable autodetection of SECAM 1 = Enable autodetection
AD_N443_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of NTSC 4.43. 0 = Disable autodetection of NTSC style systems with 4.43 MHz color subcarrier 1 = Enable autodetection
AD_P60_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of PAL 60. 0 = Disable autodetection of PAL systems with 60 Hz field rate 1 = Enable detection
AD_PALN_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of PAL N. 0 = Disable detection of PAL N 1 = Enable detection
AD_PALM_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of PAL M. 0 = Disable autodetection of PAL M 1 = Enable detection
AD_NTSC_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of NTSC. 0 = Disable detection of standard NTSC 1 = Enable detection
AD_PAL_EN		R/W
0x07	01111111	This control allows the user to disable the autodetection of PAL. 0 = Disable detection of standard PAL 1 = Enable detection
CONTRAST[7:0]		R/W
0x08	10000000	This control is used to adjust the contrast of a picture for the SDP block only. 10000000 = Adjust contrast (affects gain on luma channel)

Reg	Bits	Description	
BRIGHTNESS[7:0]			R/W
0x0A	00000000	This control is used to adjust the brightness of a video signal through the SDP core. 00000000 = Adjust brightness (affects offset of luma channel)	
HUE[7:0]			R/W
0x0B	00000000	This control is used to adjust the color hue of a picture. 00000000 = Adjust hue (affects phase of chroma signal)	
DEF_Y[5:0]			R/W
0x0C	001101xx	This control is used to specify a default luma value to be output if the SD core loses lock on the incoming video signal or if there is no input signal. 001101 = Default value of Y	
DEF_VAL_AUTO_EN			R/W
0x0C	xxxxxx1x	This control is used to enable the automatic use of the default values for Y, Cr, and Cb if the SD core cannot lock to the video signal. 0 = Disable free-run mode 1 = Enable automatic free-run mode (blue screen)	
DEF_VAL_EN			R/W
0x0C	xxxxxxx0	This control is used to force the use of the default values for Y, Cr, and Cb. 0 = Free-run mode dependent on DEF_VAL_AUTO_EN 1 = Force free-run mode on and output blue screen	
DEF_C[7:0]			R/W
0x0D	01111100	This control is used with the DEF_Y[5:0] value to define the four MSBs of the Cr and Cb values. 01111100 (blue) = Default values for Cr and Cb	
SUB_USR_EN[1:0]			R/W
0x0E	00xxxxxx	This control is used to select which map is currently being accessed: the SDP Main Map, SDP Map 1, or SDP Map 2. This control takes precedence over R_ONLY_MAPS_SEL. 00 = Access SDP Main Map 01 = Access SDP Map 1 10 = Access SDP Map 2 11 = Reserved	
R_ONLY_MAPS_SEL[2:0]			R/W
0x0E	xxxxx000	This control is used to select which part of the sub map is selected; the read/write area of the sub map; or the read only area of the sub map. This control is a lower priority than SUB_USR_EN; SUB_USR_EN must be set to 00 for this control to be effective. 000 = No Read-Only Map Selected 001 = SDP Read-Only Main Map 010 = SDP Read-Only Map 1 011 = SDP Read-Only Map 2 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	
RES			R/W
0x0F	0xxxxxxx	This control is used to reset the SD core registers. 0 = Normal operation 1 = Start reset sequence	
PWRDN			R/W
0x0F	xx1xxxxx	This control is used to switch the SD core into power-down mode. The I²C interface of the SD core is unaffected by this bit and is operational. 0 = SD Core powered up 1 = SD Core powered down	
TRAQ			SC
0x10	x0xxxxxx	This control forces the SD core to reacquire the video signal. It shortens the time it takes the SD core to acquire a new signal since it does not have to find out that it lost lock before starting a reacquisition. 0 = Normal operation 1 = Start reacquire of video signal immediately	

Reg	Bits	Description	
CCLEN			R/W
0x14	00010000	This control is used to switch off the current sources in the analog front end. 0 = Switch off current sources 1 = Enable current sources	
FREE_RUN_PAT_SEL[2:0]			R/W
0x14	00010000	This control is used to set the free run output pattern. 000 = Single color set by DEF_Y/DEF_C controls 001 = 100% color bars 010 = Grey ramp 011 = Cb ramp 100 = Cr ramp 101 = Boundary box	
DCT[1:0]			R/W
0x15	00000000	This control is used to determine the time constant (TC) of the digital fine clamp circuitry. 00 = Slow (TC: 1 sec) 01 = Medium (TC: 0.5 sec) 10 = Fast (TC: 0.1 sec) 11 = Determined by SD core dependent on video parameters	
CSFM[2:0]			R/W
0x17	00000001	This control selects the C shaping filter mode and allows selection from a range of low-pass chrominance filters; if either auto mode is selected, the decoder selects the optimum C filter depending on the CVBS video source quality (good vs. bad); non auto settings force a C filter for all standards and quality of CVBS video. 000 = Auto selection 1.5 MHz bandwidth 001 = Auto selection 2.17 MHz bandwidth 010 = SH1 011 = SH2 100 = SH3 101 = SH4 110 = SH5 111 = Wideband mode	

Reg	Bits	Description
YSFM[4:0]		R/W
0x17	00000001	This control selects the Y shaping filter mode in CVBS-only mode; allows the user to select a wide range of low-pass and notch filters; if either auto mode is selected, the decoder selects the optimum Y filter depending on the CVBS video source quality (good vs. poor). 00000 = Automatic selection including a wide notch response (PAL/NTSC/SECAM) 00001 = Automatic selection including a narrow notch response (PAL/NTSC/SECAM) 00010 = SVHS 1 00011 = SVHS 2 00100 = SVHS 3 00101 = SVHS 4 00110 = SVHS 5 00111 = SVHS 6 01000 = SVHS 7 01001 = SVHS 8 01010 = SVHS 9 01011 = SVHS 10 01100 = SVHS 11 01101 = SVHS 12 01110 = SVHS 13 01111 = SVHS 14 10000 = SVHS 15 10001 = SVHS 16 10010 = SVHS 17 10011 = SVHS 18 (CCIR 601) 10100 = PAL NN 1 10101 = PAL NN 2 10110 = PAL NN 3 10111 = PAL WN 1 11000 = PAL WN 2 11001 = NTSC NN 1 11010 = NTSC NN 2 11011 = NTSC NN 3 11100 = NTSC WN 1 11101 = NTSC WN 2 11110 = NTSC WN 3 11111 = Reserved
WYSFMOVR		R/W
0x18	10010011	This control is used to enable the use of the WYSFM[4:0] settings for good quality video signals. 0 = Automatic selection of shaping filter for good quality video signals 1 = Enable manual override via WYSFM[4:0]
WYSFM[4:0]		R/W
0x18	10010011	This control selects the wideband Y shaping filter mode and allows the user to select which Y shaping filter is used for the Y component of Y/C, YPrPb, B/W input signals; it is also used when a good quality input CVBS signal is detected; for all other inputs, the Y shaping filter chosen is controlled by YSFM[4:0]. 00000 = Reserved 00001 = Reserved 00010 = SVHS 1 00011 = SVHS 2 00100 = SVHS 3 00101 = SVHS 4 00110 = SVHS 5 00111 = SVHS 6 01000 = SVHS 7 01001 = SVHS 8 01010 = SVHS 9 01011 = SVHS 10 01100 = SVHS 11 01101 = SVHS 12 01110 = SVHS 13 01111 = SVHS 14 10000 = SVHS 15 10001 = SVHS 16 10010 = SVHS 17 10011 = SVHS 18 (CCIR 601) 10100 = Reserved Others = Reserved

Reg	Bits	Description
NSFSEL[1:0]		R/W
0x19	1111<u>00</u>01	This control is used to select how much of the overall signal bandwidth is fed to the combs. A narrow split filter selection gives better performance on diagonal lines, but leaves more dot crawl in the final output image. The opposite is true for selecting a wide split filter. 00 = Narrow 01 = Medium 10 = Medium 11 = Wide
PSFSEL[1:0]		R/W
0x19	111100<u>01</u>	This control is used to select how much of the overall signal bandwidth is fed to the combs in PAL mode only. A wide split filter selection eliminates dot crawl, but shows imperfections on diagonal lines. The opposite is true for selecting a narrow split filter. 00 = Narrow 01 = Medium 10 = Wide 11 = Widest
SWPC		R/W
0x27	<u>0</u>1011000	This control is used to swap Cr and Cb samples of the SDP block only. 0 = No swapping 1 = Swap Cr and Cb values
AUTO_PDC_EN		R/W
0x27	0<u>1</u>011000	This control is used to automatically set LTA[1:0] and CTA[2:0] to have the same chroma and luma data delays for all modes. If set to 1, LTA[1:0] and CTA[2:0] are not used by the system. If set to 0, automatic mode is disabled and the values programmed into LTA[1:0] and CTA[2:0] take effect. 0 = Use LTA[1:0] and CTA[2:0] values for delaying luma and chroma samples 1 = SD core automatically determines LTA and CTA values to have luma and chroma aligned at output
CTA[2:0]		R/W
0x27	0<u>101</u>1000	This control is used to specify a timing difference between chroma and luma samples. This may be used to compensate for external filter group delay differences in the chroma vs. luma path and to allow a different number of pipeline delays while processing the video downstream. 000 = Reserved 001 = Chroma + 2 chroma pixel (early) 010 = Chroma + 1 chroma pixel (early) 011 = No delay 100 = Chroma − 1 chroma pixel (late) 101 = Chroma − 2 chroma pixel (late) 110 = Chroma − 3 chroma pixel (late) 111 = Reserved
LTA[1:0]		R/W
0x27	01011<u>00</u>	This control is used to specify a timing difference between chroma and luma samples. This may be used to compensate for external filter group delay differences in the luma vs. chroma path and to allow a different number of pipeline delays while processing the video downstream. 00 = No delay 01 = Luma 1 clock (37 ns) delayed 10 = Luma 2 clock (74 ns) early 11 = Luma 1 clock (37 ns) early
CKE		R/W
0x2B	1<u>1</u>000001	This control is used to enable the optional color kill function. 0 = Disable color kill 1 = Enable color kill
PW_UPD		R/W
0x2B	1100000<u>1</u>	The peak white and average video algorithms determine the gain based on measurements taken from the active video. This control is used to determine the rate of gain change. 0 = Update gain once per video line 1 = Update gain once per field

Reg	Bits	Description	
LAGC[2:0]			R/W
0x2C	0010 0010	This control is used to select the mode of operation for the gain control in the luma path. 000 = Manual fixed gain (use LMG[11:0]) 001 = AGC (blank level to sync tip), peak white algorithm off 010 = AGC (blank level to sync tip), peak white algorithm on 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Freeze gain	
CAGC[1:0]			R/W
0x2C	0010 0010	This control is used to select the basic mode of operation for the automatic gain control in the chroma path. 00 = Manual fixed gain (use CMG[11:0]) 01 = Use luma gain for chroma 10 = Automatic gain (based on color burst) 11 = Freeze chroma gain	
CAGT[1:0]			R/W
0x2D	1100 0100	This control is used to influence the tracking speed of the chroma automatic gain control. Note that this only has an effect if the CAGC[1:0] register is set to 10 (automatic gain). 00 = Slow (TC: 2 sec) 01 = Medium (TC: 1 sec) 10 = Fast (TC: 0.2 sec) 11 = Adaptive	
CMG[11:0]			R/W
0x2D 0x2E	1100 0100 0000 0000	This control is used to set the required manual chroma gain.	
LAGT[1:0]			R/W
0x2F	1100 0000	This control is used to influence the tracking speed of the luminance automatic gain control. Note that this only has an effect if the LAGC[2:0] register is set to 001, 010, 011 or 100 (automatic gain control modes). Note: If peak white AGC is enabled and active (see FOLLOW_PW (SDP R/O main map, Register 0x10[3])), the actual gain update speed is dictated by the peak white AGC loop and, as a result, the LAGT settings have no effect. As soon as the device leaves peak white AGC, LAGT becomes relevant again. The update speed for the peak white algorithm can be customized by the use of internal parameters. 00 = Slow (TC: 2 sec) 01 = Medium (TC: 1 sec) 10 = Fast (TC: 0.2 sec) 11 = Adaptive	
LMG[11:0]			R/W
0x2F 0x30	1100 0000 0000 0000	This control is used to set the required manual luma gain.	
NEWAVMODE			R/W
0x31	0000 0010	This control is used to enable the manual position of the V and Field bits. The default is ITU-R BT.656 compliant. 0 = Do not enable manual position 1 = Enable manual position	
CTAPSN[1:0]			R/W
0x38	1000 0000	This control is used to select the number of taps in the NTSC 2D comb. 00 = Do not use 01 = NTSC chroma comb adapts 3 lines (3 taps) to 2 lines (2 taps) 10 = NTSC chroma comb adapts 5 lines (5 taps) to 3 lines (3 taps) 11 = NTSC chroma comb adapts 5 lines (5 taps) to 4 lines (4 taps)	

Reg	Bits	Description	
CCMN[2:0]			R/W
0x38	10000000	This control is used to adjust the chroma 2D comb filter. 0xx = Adaptive comb mode: adaptive 3-line chroma comb for CTAPSN = 01, adaptive 4-line chroma comb for CTAPSN = 10, adaptive 5-line chroma comb for CTAPSN = 11 100 = Disable chroma comb 101 = Fixed chroma comb (top lines of line memory): fixed 2-line chroma comb for CTAPSN = 01, fixed 3-line chroma comb for CTAPSN = 10, fixed 4-line chroma comb for CTAPSN = 11 110 = Fixed chroma comb (all lines of line memory): fixed 3-line chroma comb for CTAPSN = 01, fixed 4-line chroma comb for CTAPSN = 10, fixed 5-line chroma comb for CTAPSN = 11 111 = Fixed chroma comb (bottom lines of line memory): fixed 2-line chroma comb for CTAPSN = 01, fixed 3-line chroma comb for CTAPSN = 10, fixed 4-line chroma comb for CTAPSN = 11	
YCMN[2:0]			R/W
0x38	10000000	This control is used to adjust the luma 2D comb filter. 0xx = Adaptive comb mode: adaptive 3-line (3 taps) luma comb 100 = Disable luma comb: use low pass/notch filter 101 = Fixed luma comb (top lines of line memory): fixed 2-line (2 taps) luma comb 110 = Fixed luma comb (all lines of line memory): fixed 3-line (3 taps) luma comb 111 = Fixed luma comb (bottom lines of line memory): fixed 2-line (2 taps) luma comb	
CTAPSP[1:0]			R/W
0x39	11000000	This control is used to adjust the PAL 2D comb filter. 00 = Do not use 01 = PAL chroma comb adapts 5 lines (3 taps) to 3 lines (2 taps) cancel cross luma only 10 = PAL chroma comb adapts 5 lines (5 taps) to 3 lines (3 taps) cancels cross luma and hue error less well 11 = PAL chroma comb adapts 5 lines (5 taps) to 4 lines (4 taps) cancels cross luma and hue error well	
CCMP[2:0]			R/W
0x39	11000000	This control is used to adjust the chroma 2D comb filter. 0xx = Adaptive comb mode: adaptive 3 line chroma comb for CTAPSP = 01, adaptive 4-line chroma comb for CTAPSP = 10, adaptive 5-line chroma comb for CTAPSP = 11 100 = Disable chroma comb 101 = Fixed chroma comb (top lines of line memory): fixed 2-line chroma comb for CTAPSP = 01, fixed 3-line chroma comb for CTAPSP = 10, fixed 4-line chroma comb for CTAPSP = 11 110 = Fixed chroma comb (all lines of line memory): fixed 3-line chroma comb for CTAPSP = 01, fixed 4-line chroma comb for CTAPSP = 10, fixed 5-line chroma comb for CTAPSP = 11 111 = Fixed chroma comb P165(bottom lines of line memory): fixed 2-line chroma comb for CTAPSP = 01, fixed 3-line chroma comb for CTAPSP = 10, fixed 4-line chroma comb for CTAPSP = 11	
YCMP[2:0]			R/W
0x39	11000000	This control is used to adjust the luma 2D comb filter. 0xx = Adaptive comb mode: adaptive 5 lines (3 taps) luma comb 100 = Disable luma comb: use low-pass/notch filter 101 = Fixed luma comb (top lines of line memory): fixed 3 lines (2 taps) luma comb 110 = Fixed luma comb (all lines of line memory): fixed 5 lines (3 taps) luma comb 111 = Fixed luma comb (bottom lines of line memory): fixed 3 lines (2 taps) luma comb	
PDN_ADC0			R/W
0x3A	00000000	This control is used to power down the ADC0 mux circuitry. 0 = ADC0 and associated channels in normal operation 1 = Power down ADC0 and associated channels operation	
PDN_ADC1			R/W
0x3A	00000000	This control is used to power down the ADC1 mux circuitry. 0 = ADC1 and associated channel in normal operation 1 = Power down ADC1 and associated channel operation	
PDN_ADC2			R/W
0x3A	00000000	This control is used to power down the ADC2 mux circuitry. 0 = ADC2 and associated channel in normal operation 1 = Power down ADC2 and associated channel operation	
ADC_PDN_OVERRIDE			R/W
0x3A	00000000	This control is used to set the manual configuration of the power control bits for the ADC. 0 = Automatic 1 = Manual override	

Reg	Bits	Description	
CKILLTHR[2:0]			R/W
0x3D	0010 0010	This control allows the color kill threshold to be configured. 000 = NTSC, PAL color kill at <0.5%, SECAM no color kill 001 = NTSC, PAL color kill at <1.5%, SECAM color kill at <5% 010 = NTSC, PAL color kill at <2.5%, SECAM color kill at <7% 011 = NTSC, PAL color kill at <4%, SECAM color kill at <8% 100 = NTSC, PAL color kill at <8.5%, SECAM color kill at <9.5% 101 = NTSC, PAL color kill at <16%, SECAM color kill at <15% 110 = NTSC, PAL color kill at <32%, SECAM color kill at <32% 111 = Reserved	
GDECEL[15:0]			R/W
0x48 0x49	00000000 00000000	This control is used to select the even video lines of the Gemstar™ data to be decoded. The 16 bits of this control are interpreted as a collection of 16 individual line decode enable signals. Each bit refers to a line of video in an even field. Setting the bit enables the decoder block trying to find Gemstar or CCAP compatible data on that particular line. Setting the bit to 0 prevents the decoder from trying to retrieve data. To retrieve CCAP data services on NTSC (Line 284), GDECEL[11] must be set. To retrieve CCAP data services on PAL (Line 335), GDECEL[14] must be set. 00000000 = Do not attempt to decode Gemstar compatible data or CCAP on any line (even field)	
GDECOL[15:0]			R/W
0x4A 0x4B	00000000 00000000	This control is used to select the even video lines of the GemStar data to be decoded. The 16 bits of this control are interpreted as a collection of 16 individual line decode enable signals. Each bit refers to a line of video in an odd field. Setting the bit enables the decoder block trying to find GemStar or CCAP compatible data on that particular line. Setting the bit to 0 prevents the decoder from trying to retrieve data. To retrieve CCAP data services on NTSC (Line 284), GDECEL[11] must be set. To retrieve CCAP data services on PAL (Line 335), GDECEL[14] must be set. 00000000 = Do not attempt to decode Gemstar compatible data or CCAP on any line (odd field)	
GDE_SEL_OLD_ADF			R/W
0x4C	0000 0000	This control is used to enable an ancillary data system. 0 = Enables new ancillary data system 1 = Enables ancillary data system compatible with the SD core	
GDECAD			R/W
0x4C	0000000 0	This control is used to insert data into the horizontal blanking period. 0 = Split data into half bytes and insert 1 = Output data straight in 8-bit format	
DNR_EN			R/W
0x4D	111 1 01111	This control is used to enable the DNR block or bypass it. 0 = Bypass DNR (disable) 1 = Enable digital noise reduction on luma data	
CTI_AB[1:0]			R/W
0x4D	1110 11 11	This control is used to set the behavior of the alpha-blend circuitry which mixes the sharpened chroma signal with the original one. It controls the visual impact of the CTI on the output data. 00 = Sharpest mixing between sharpened and original chroma signal 01 = Sharp mixing 10 = Smooth mixing 11 = Smoothest alpha blend function	
CTI_AB_EN			R/W
0x4D	11101 1 11	This control is used to enable an alpha-blend mixing function within the CTI block. If set to 1, the alpha-blender mixes the transient improved chroma with the original signal. The sharpness of the alpha-blending can be configured using the CTI_AB[1:0] bits. 0 = Disable CTI alpha-blend mixing function 1 = Enable CTI alpha-blend mixing function	
CTI_EN			R/W
0x4D	1110111 1	This control is used to enable the CTI function. 0 = Disable CTI 1 = Enable CTI block	
CTI_C_TH[7:0]			R/W
0x4E	00001000	This control is used to specify how big the amplitude step has to be in a chroma transition to be steepened by the CTI block. 00001000 = Threshold for chroma edges prior to CTI	

Reg	Bits	Description
DNR_TH[7:0]		R/W
0x50	00001000	This control is used to determine the maximum edge that is still interpreted as noise and, therefore, blanked from the luma data. 00001000 = Threshold for maximum luma edges to be interpreted as noise
FSCLE		R/W
0x51	00100100	This control is used to choose whether the status of the color subcarrier loop is taken into account when the overall lock status is determined and presented via Bits[1:0] in Status Register 1. 0 = Overall lock status only dependent on HS lock 1 = Overall lock status dependent on HS lock and f_{SC} lock
SRLS		R/W
0x51	00100100	This control is used to choose between two sources for the determination of the lock status (as per Bits[1:0] in the Status 1 register). The first source, the TIME_WIN signal, is based on a line-to-line evaluation of the horizontal synchronization pulse of the incoming video. It reacts quickly. The second source, the FREE_RUN signal, evaluates the properties of the incoming video over several fields and takes vertical synchronization information into account. 0 = Select FREE_RUN signal 1 = Select TIME_WIN signal
COL[2:0]		R/W
0x51	00100100	This control is used to determine the number of consecutive lines for which the out of lock condition has to be true before the system switches into the unlocked state. 000 = 1 line of video 001 = 2 lines of video 010 = 5 lines of video 011 = 10 lines of video 100 = 100 lines of video 101 = 500 lines of video 110 = 1000 lines of video 111 = 10000 lines of video
CIL[2:0]		R/W
0x51	00100100	This control is used to determine the number of consecutive lines for which the into lock condition has to be true before the system switches into the locked state. 000 = 1 line of video 001 = 2 lines of video 010 = 5 lines of video 011 = 10 lines of video 100 = 100 lines of video 101 = 500 lines of video 110 = 1000 lines of video 111 = 10000 lines of video
ADC0N_SW[3:0]		R/W
0x60	00010000	This control is used to select the ADC0 negative analog input. 0000 = No connect 0001 = No connect 0010 = AIN2 0011 = No connect 0100 = AIN4 0101 = No connect 0110 = AIN6 0111 = No connect 1000 = AIN8
ADC1_SW[3:0]		R/W
0xC3	00000000	This control is used to select the ADC1 analog input. 0000 = No connect 0001 = No connect 0010 = AIN2 0011 = No connect 0100 = AIN4 0101 = AIN5 0110 = AIN6 0111 = No connect 1000 = AIN 8

Reg	Bits	Description	
ADC0_SW[3:0]			R/W
0xC3	0000<u>0000</u>	This control is used to select the ADC0 analog input. 0000 = No connection 0001 = AIN1 0010 = AIN2 0011 = AIN3 0100 = AIN4 0101 = AIN5 0110 = AIN6 0111 = AIN7 1000 = AIN8	
ADC_SW_MAN			R/W
0xC4	<u>0</u>0000000	This control is used to enable manual setting of the ADC input signal muxing. 0 = Disable 1 = Enable	
ADC2_SW[3:0]			R/W
0xC4	0000<u>0000</u>	This control is used to select the ADC2 analog input. 0000 = No connect 0001 = No connect 0010 = AIN2 0011 = AIN3 0100 = No connect 0101 = No connect 0110 = AIN 6 0111 = No connect 1000 = No connect	
LB_TH[4:0]			R/W
0xDC	101<u>01100</u>	This control is used to set the threshold value that determines if a line is black. 01101 to 10000 = Increase threshold 00000 to 01011 = Decrease threshold	
LB_SL[3:0]			R/W
0xDD	<u>0100</u>1100	This control is used to program the start line of the activity window for Letterbox detection (start of field). 0100 = Letterbox detection aligned with the start of active video (23/286 NTSC)	
LB_EL[3:0]			R/W
0xDD	0100<u>1100</u>	This control is used to program the end line of the activity window for Letterbox detection (end of field). 1100 = Letterbox detection ends with the last line of active video on a field (262/525 NTSC)	
SD_OFF_U[7:0]			R/W
0xE1	<u>10000000</u>	This control is used to provide an offset for the U channel only. 00000000 = −312 mV offset applied to the U channel 10000000 = 0 mV offset applied to the U channel 11111111 = +312 mV offset applied to the U channel	
SD_OFF_V[7:0]			R/W
0xE2	<u>10000000</u>	This control is used to provide an offset for the V channel only. 00000000 = −312 mV offset applied to the V channel 10000000 = 0 mV offset applied to the V channel 11111111 = +312 mV offset applied to the V channel	
SD_SAT_U[7:0]			R/W
0xE3	<u>10000000</u>	This control is used to adjust the gain of the U channel only. 00000000 = Gain on U channel set to −42 dB 10000000 = Gain on U channel set to 0 dB 11111111 = Gain on U channel set to +6 dB	
SD_SAT_V[7:0]			R/W
0xE4	<u>10000000</u>	This control is used to adjust the gain of the V channel only. 00000000 = Gain on V channel set to −42 dB 10000000 = Gain on V channel set to 0 dB 11111111 = Gain on V channel set to +6 dB	

Reg	Bits	Description	
NVBEGDELO			R/W
0xE5	00100101	This control is used to delay the start of the odd field by a line relative to NVBEG. 0 = No delay 1 = Delay positive edge of VS on odd field	
NVBEGDELE			R/W
0xE5	00100101	This control is used to delay the start of the even field by a line relative to NVBEG. 0 = No delay 1 = Delay positive edge of VS on even field	
NVBEGSIGN			R/W
0xE5	00100101	This control is used to delay the start of the VS for NTSC. Setting this control to 0 in manual programming mode delays the start of VS. Setting this control to 1 (default) advances the start of VS; this is not recommended for use in manual mode however. 0 = Set to low when manual programming 1 = Not suitable for user programming	
NVBEG[4:0]			R/W
0xE5	00100101	This control is used to set the NTSC VS start position. 00101 = NTSC (ITU-R BT.656)	
NVENDDELO			R/W
0xE6	00000100	This control is used to delay the end of the odd field by a line relative to NVEND. 0 = No delay 1 = Delay VS from going low on odd field	
NVENDDELE			R/W
0xE6	00000100	This control is used to delay the end of the even field by a line relative to NVEND. 0 = No delay 1 = Delay VS from going low on even field	
NVENDSIGN			R/W
0xE6	00000100	This control is used to delay the end of the VS for NTSC. Setting this control to 0 in manual programming mode delays the end of VS. Setting this control to 1 (default) advances the end of VS; this is not recommended for use in manual mode however. 0 = Delays end of VS; set for user manual programming. 1 = Advances end of VS (not recommended for user programming)	
NVEND[4:0]			R/W
0xE6	00000100	This control is used to set the NTSC VS end position. It affects the V bit in the AV codes. 00100 = NTSC (ITU-R BT.656)	
NFTOGDELO			R/W
0xE7	01100011	This control is used to delay the NTSC Field transition on the odd field by a line relative to NFTOG. 0 = No delay 1 = Delay field toggle/transition on odd field	
NFTOGDELE			R/W
0xE7	01100011	This control is used to delay the NTSC Field transition on the even field by a line relative to NFTOG. 0 = No delay 1 = Delay field toggle/transition on even field	
NFTOGSIGN			R/W
0xE7	01100011	This control is used to delay the Field toggle/transition for NTSC. Setting this control to 0 in manual programming mode delays the field toggle/transition. Setting this control to 1 (default) advances the field toggle/transition; this is not recommended for use in manual mode however. 0 = Delays field transition. Set for user manual programming 1 = Advances field transition (not recommended for user programming)	
NFTOG[4:0]			R/W
0xE7	01100011	This control is used to position the NTSC Field bit. It affects the F bit in the AV cdoes. 00011 = NTSC (ITU-R BT.656)	
PVBEGDELO			R/W
0xE8	01100101	This control is used to delay the VS transition on the odd field by a line relative to PVBEG. 0 = No delay 1 = Delay VS going high on odd field	

Reg	Bits	Description	
PVBEGDELE			R/W
0xE8	01100101	This control is used to delay the VS transition on the even field by a line relative to PVBEG. 0 = No delay 1 = Delay VS going high on even field	
PVBEGSIGN			R/W
0xE8	01100101	This control is used to delay the start of the VS for PAL. Setting this control to 0 in manual programming mode delays the start of VS. Setting this control to 1 (default) advances the start of VS; this is not recommended for use in manual mode however. 0 = Delays the beginning of VS. Set for user manual programming. 1 = Advances the beginning of VS (not recommended for user programming)	
PVBEG[4:0]			R/W
0xE8	01100101	This control is used to set the PAL VS start position. 00101 = PAL (ITU-R BT.656)	
PVENDDELO			R/W
0xE9	00010100	This control is used to delay the VS low transition on an odd field by a line relative to PVEND. 0 = No delay 1 = Delay VS going low on odd field	
PVENDDELE			R/W
0xE9	00010100	This control is used to delay the VS low transition on an even field by a line relative to PVEND. 0 = No delay 1 = Delay VS going low on an even field	
PVENDSIGN			R/W
0xE9	00010100	This control is used to delay the end of the VS for PAL. Setting this control to 0 in manual programming mode delays the end of VS. Setting this control to 1 (default) advances the end of VS; this is not recommended for use in manual mode however. 0 = Delay end of VS (set for user manual programming) 1 = Advance end of VS (not recommended for user programming)	
PVEND[4:0]			R/W
0xE9	00010100	This control is used to set the PAL VS end position. 10100 = PAL (ITU-R BT.656)	
PFTOGDELO			R/W
0xEA	01100011	This control is used to delay the Field transition on the odd Field by a line relative to PFTOG. 0 = No delay 1 = Delay Field transition on odd field	
PFTOGDELE			R/W
0xEA	01100011	This control is used to delay the Field transition on the even Field by a line relative to PFTOG. 0 = No delay 1 = Delay Field transition on even field	
PFTOGSIGN			R/W
0xEA	01100011	This control is used to delay the Field toggle/transition for PAL. Setting this control to 0 in manual programming mode delays the field toggle/transition. Setting this control to 1 (default) advances the field toggle/transition; this is not recommended for use in manual mode however. 0 = Delay field transition. Set for user manual programming. 1 = Advances field transition (not recommended for user programming)	
PFTOG[4:0]			R/W
0xEA	01100011	This control is used to set the PAL Field transition position. For all NTSC/PAL Field timing controls, the Field bit in the code is modified. 00011 = PAL (ITU-R BT.656)	
NVBIOLCM[1:0]			R/W
0xEB	01010101	This control is used to set the position of the first active (comb filtered) line after VBI on an odd field in NTSC. 00 = VBI ends one line earlier (Line 20) 01 = ITU-R BT.470 compliant (Line 21) 10 = VBI ends one line later (Line 22) 11 = VBI ends two lines later (Line 23)	

Reg	Bits	Description	
NVBIELCM[1:0]			R/W
0xEB	0101**01**0101	This control is used to set the position of the first active (comb filtered) line after VBI on an even field in NTSC. 00 = VBI ends one line earlier (Line 282) 01 = ITU-R BT.470 compliant (Line 283) 10 = VBI ends one line later (Line 284) 11 = VBI ends two lines later (Line 285)	
PVBIOLCM[1:0]			R/W
0xEB	0101**01**01	This control is used to set the position of the first active (comb filtered) line after VBI on an odd field in PAL. 00 = VBI ends one line earlier (Line 22) 01 = ITU-R BT.470 compliant (Line 23) 10 = VBI ends one line later (Line 24) 11 = VBI ends two lines later (Line 25)	
PVBIELCM[1:0]			R/W
0xEB	010101**01**	This control is used to set the position of the first active (comb filtered) line after VBI on an even field in PAL. 00 = VBI ends one line earlier (Line 335) 01 = ITU-R BT.470 compliant (Line 336) 10 = VBI ends one line later (Line 337) 11 = VBI ends two lines later (Line 338)	
NVBIOCCM[1:0]			R/W
0xEC	**01**010101	This control is used to set the position of the first line that outputs color after VBI on an odd field in NTSC. 00 = Color output start Line 20 01 = ITU-R BT.470 compliant color output start Line 21 10 = Color output start Line 22 11 = Color output start Line 23	
NVBIECCM[1:0]			R/W
0xEC	0101**01**0101	This control is used to set the position of the first line that outputs color after VBI on an even field in NTSC. 00 = Color output start Line 282 01 = ITU-R BT.470-compliant color output start Line 283 10 = VBI ends one line later (Line 284) 11 = Color output start Line 285	
PVBIOCCM[1:0]			R/W
0xEC	0101**01**01	This control is used to set the position of the first line that outputs color after VBI on an odd field in PAL. 00 = Color output start Line 22 01 = ITU-R BT.470-compliant color output start Line 23 10 = Color output start Line 24 11 = Color output start Line 25	
PVBIECCM[1:0]			R/W
0xEC	010101**01**	This control is used to set the position of the first line that outputs color after VBI on an even field in PAL. 00 = Color output start Line 335 01 = ITU-R BT.470 compliant color output start Line 336 10 = Color output start Line 337 11 = Color output start Line 338	
AA_FILT_MAN_OVR			R/W
0xF3	0000**0**0000	This control is used to override the antialiasing filter. 0 = Disable override 1 = Enable override	
AA_FILT_EN[3:0]			R/W
0xF3	0000**0000**	This control is used to enable the antialiasing filter. 0000 = Disable all antialiasing filters 0001 = Enable antialiasing Filter 1. Disable all other antialiasing filters 0010 = Enable antialiasing Filter 2. Disable all other antialiasing filters 0100 = Enable antialiasing Filter 3. Disable all other antialiasing filters 1000 = Enable antialiasing Filter 4. Disable all other antialiasing filters	

Reg	Bits	Description	
IFFILTSEL[2:0]			R/W
0xF8	00000000	This control is used to select an IF filter for PAL and NTSC. 000 = Bypass mode (0 dB) 001 = −3 dB (2 MHz, NTSC filters) 010 = −6 dB (2 MHz, NTSC filters) 011 = −10 dB (2 MHz, NTSC filters) 100 = Reserved 101 = −2 dB (3 MHz, PAL filters) 110 = −5 dB (3 MHz, PAL filters) 111 = −7 dB (3 MHz, PAL filters)	
VS_COAST_MODE[1:0]			R/W
0xF9	00000011	This control is used to set the output coast frequency for the SDP. The default setting allows each block (VS processor, horizontal PLL, and free run block) to decide independently what frequency to coast at (PAL vs. NTSC timing). If this control is set to 01, the SDP is only allowed to coast using the PAL timing (50 Hz). If it is set to 10, the SDP is only allowed to coast using the NTSC timing. If it is set to 11, the coast frequency for all blocks in the SDP is set by the VS processor, and the coast frequency used is the last vertically locked standard. 00 = Auto coast mode: coast independently controlled by each subblock 01 = 50 Hz coast mode 10 = 60 Hz coast mode 11 = Auto coast mode: controlled by VS processor	
EXTEND_VS_MIN_FREQ			R/W
0xF9	00000011	This control is used to extend the minimum vertical locking range of the SDP core. When set to 1, this control extends the minimum VS frequency allowed by the VS processor to 39.51 Hz (791 lines/frame). 0 = Limit minimum VS frequency to 42.75 Hz (731 lines/frame) 1 = Limit minimum VS frequency to 39.51 Hz (791 lines/frame)	
EXTEND_VS_MAX_FREQ			R/W
0xF9	00000011	This control is used to extend the minimum vertical locking range of the SDP core. When set to 1, this control extends the maximum VS frequency allowed by the VS processor to 70.09 Hz (449 lines/frame). 0 = Limit maximum VS frequency to 66.25 Hz (475 lines/frame) 1 = Limit maximum VS frequency to 70.09 Hz (449 lines/frame)	
PEAKING_GAIN[7:0]			R/W
0xFB	01000000	This control is used to increases or decrease the gain for high frequency portions of the video signal.	
DNR_TH_2[7:0]			R/W
0xFC	00000100	This control is used to specify the maximum luma edge that is interpreted as noise and is therefore blanked. 00000100 = Threshold for maximum luma edges to be interpreted as noise	

2.7 SDP R/O MAIN MAP

Reg	Bits	Description	
SUB_USR_EN_RB[1:0]			R
0x0E	00000000	This indicates which map is currently being accessed: the SDP Main Map, SDP Map 1, SDP Map 2. This readback takes precedence over R_ONLY_MAPS_SEL_RB. 00 = Access SDP Main Map 01 = Access SDP Map 1 10 = Access SDP Map 2 11 = Reserved	
R_ONLY_MAPS_SEL_RB[2:0]			R
0x0E	00000000	This readback indicates which part of the sub map is selected; the read/write area of the sub map; or the read only area of the sub map. This readback has a lower priority than SUB_USR_EN_RB; SUB_USR_EN_RB must be set to 00 for this readback to be effective. 000 = No Read-Only Map Selected 001 = SDP Read-Only Main Map 010 = SDP Read-Only Map 1 011 = SDP Read-Only Map 2 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	
COL_KILL			R
0x10	00000000	This readback indicates if color kill is active. 0 = Color kill not active 1 = Color kill active	
AD_RESULT[2:0]			R
0x10	00000000	This readback displays the connected input from the SDP autodetection block. 000 = NTSM-MJ 001 = NTSC-443 010 = PAL-M 011 = PAL-60 100 = PAL-BGHID 101 = SECAM 110 = PAL-Combination N 111 = SECAM 525	
FOLLOW_PW			R
0x10	00000000	This readback displays information about the internal status of the decoder. 0 = Not active 1 = Peak white AGC mode active	
FSC_LOCK			R
0x10	00000000	This readback displays information about the internal status of the decoder. 0 = Not locked 1 = f_{SC} lock (now)	
LOST_LOCK			R
0x10	00000000	This readback displays information about the internal status of the decoder. 0 = No change in lock status 1 = Lost lock (since last read)	
IN_LOCK			R
0x10	00000000	This readback displays information about the internal status of the decoder. 0 = Not locked 1 = In lock (now)	
FSC NSTD			R
0x12	00000000	This readback indicates if the f_{SC} frequency is nonstandard. 0 = Standard f_{SC} detected 1 = Nonstandard f_{SC} detected	
LL NSTD			R
0x12	00000000	This readback indicates if the line length is nonstandard. 0 = Standard line length detected 1 = Nonstandard line length detected	

Reg	Bits	Description
MV AGC DET		R
0x12	0000**0**000	This readback indicates if Macrovision AGC pulses are detected. 0 = Macrovision AGC pulses not detected 1 = Macrovision AGC pulses detected
MV PS DET		R
0x12	00000**0**00	This readback indicates if Macrovision pseudo sync pulses are detected. 0 = Macrovision pseudo sync pulses not detected 1 = Macrovision pseudo sync pulses detected
MVCS T3		R
0x12	000000**0**0	This readback displays information about Macrovision color striping protection. 0 = Type 2 1 = Type 3
MVCS DET		R
0x12	0000000**0**	This readback indicates if Macrovision color striping is detected. 0 = Macrovision color striping not detected 1 = Macrovision color striping detected
PAL SW LOCK		R
0x13	**0**0000000	This readback indicates if a reliable swinging burst sequence is detected. 0 = Reliable swinging burst sequence not detected 1 = Reliable swinging burst sequence detected
INTERLACE		R
0x13	0**0**000000	This readback indicates if a Field sequence was found. 0 = Interlaced video not detected 1 = Interlaced video detected
STD FLD LEN		R
0x13	00**0**00000	This readback indicates if the Field length is correct for the currently selected video standard. 0 = Field length not correct 1 = Field length correct
FREE_RUN_ACT		R
0x13	000**0**0000	This readback indicates if free run mode is active. 0 = Free run mode not active 1 = Free run mode active
SD_OP_50HZ_RB		R
0x13	00000**0**00	This general status readback displays information about the SD field rate detected. 0 = SD 60 Hz detected 1 = SD 50 Hz detected
INST_HLOCK		R
0x13	0000000**0**	This readback indicates the horizontal lock status of the decoder. 0 = Currently not locked to horizontal signal 1 = Horizontal lock achieved
CGMSD		R
0x90	0000**0**000	This readback indicates that the data in the CGMS1, CGMS2, and CGMS3 registers are valid. The CGMSD bit goes high if the rising edge of the start bit is detected within a time window. The user is requested to check the validity of data by reading the CRC check bits which are contained in the CGMS3[7:0] register. No internal CRC check is performed. 0 = No CGMS transmission detected 1 = CGMS sequence decoded
EDTVD		R
0x90	00000**0**00	This readback indicates that the data in the EDTV1, EDTV2, and EDTV3 registers are valid. The EDTVD bit goes high if the rising edge of the start bit is detected within a time window and if the polarity of the parity bit matches the data transmitted. 0 = No EDTV sequence detected 1 = EDTV sequence detected

Reg	Bits	Description
CCAPD_RB		R
0x90	0000000**0**	This general status readback indicates that the data in the CCAP1 and CCAP2 registers are valid. The CCAPD bit goes high if the rising edge of the start bit is detected within a time window and if the polarity of the parity bit matches the data transmitted. 0 = No CCAP signals detected 1 = CCAP sequence detected
WSSD		R
0x90	0000000**0**	This readback indicates that the data in the WSS1 and WSS2 registers are valid. The WSSD bit goes high if the rising edge of the start bit is detected within a time window and if the polarity of the parity bit matches the data transmitted. 0 = No WSS detected 1 = WSS detected
WSS1[7:0]		R
0x91	00000000	This readback displays the WSS data. 00000000 = Default
WSS2[7:0]		R
0x92	00000000	This readback displays the WSS data. 00000000 = Default
EDTV1[7:0]		R
0x93	00000000	This readback displays the EDTV data. 00000000 = Default
EDTV2[7:0]		R
0x94	00000000	This readback displays the EDTV data. 00000000 = Default
EDTV3[7:0]		R
0x95	00000000	This readback displays the EDTV data. 00000000 = Default
CGMS1[7:0]		R
0x96	00000000	This readback displays the CGMS data. 00000000 = Default
CGMS2[7:0]		R
0x97	00000000	This readback displays the CGMS data. 00000000 = Default
CGMS3[7:0]		R
0x98	00000000	This readback displays the CGMS data. 00000000 = Default
CCAP1[7:0]		R
0x99	00000000	This readback displays the CCAP data. 00000000 = Default
CCAP2[7:0]		R
0x9A	00000000	This readback displays the CCAP data. 00000000 = Default
LB_LCT[7:0]		R
0x9B	00000000	This readback displays the number of black lines detected at the top of active video. 00000000 = Default
LB_LCM[7:0]		R
0x9C	00000000	This readback displays the number of black lines detected in the bottom half of active video if subtitles are detected. 00000000 = Default
LB_LCB[7:0]		R
0x9D	00000000	This readback displays the number of black lines detected at the bottom of active video. 00000000 = Default
ST_NOISE_VLD		R
0xDE	0000**0**000	This readback indicates if the sync tip noise measurement value, ST_NOISE[10:0], is valid. 0 = ST_NOISE[10:0] measurement not valid 1 = ST_NOISE[10:0] measurement valid

Reg	Bits	Description
ST_NOISE[10:0]		R
0xDE 0xDF	00000000 00000000	This readback provides information on the average sync tip noise on the preceding four fields of video. One bit of ST_NOISE[10:0] is equal to one ADC code. That is, one bit of ST_NOISE[10:0] = 1.6 V/4096 = 390.625 µV. This readback value is only valid if ST_NOISE_VLD is set to 1. 000 = Average noise present on video sync tip

2.8 SDP MAP 1

Reg	Bits	Description	
MV_INTRQ_SEL[1:0]			R/W
0x40	0001**0**000	This control is used to configure the Macrovision interrupt. 00 = Reserved 01 = Pseudo sync only 10 = Color stripe only 11 = Pseudo sync or color stripe	
MPU_STIM_INTRQ			R/W
0x40	0001**0**000	This control is used to configure the manual interrupt mode. 0 = Disable manual interrupt mode 1 = Enable manual interrupt mode	
MV_PS_CS_CLR			SC
0x43	0**0**000000	This control is used to clear the MV_PS_CS_Q interrupt. 0 = Do not clear MV_PS_CS 1 = Clear MV_PS_CS	
SD_FR_CHNG_CLR			SC
0x43	000**0**0000	This control is used to clear the SD_FR_CHNG interrupt. 0 = Do not clear SD_FR_CHNG 1 = Clear SD_FR_CHNG	
SD_UNLOCK_CLR			SC
0x43	0000000**0**	This control is used to clear the SD_UNLOCK interrupt. 0 = Do not clear SD_UNLOCK 1 = Clear SD_UNLOCK	
SD_LOCK_CLR			SC
0x43	0000000**0**	This control is used to clear the SD_LOCK interrupt. 0 = Do not clear SD_LOCK 1 = Clear SD_LOCK	
MV_PS_CS_MSKB			R/W
0x44	0**0**000000	This control is used to mask the MV_PS_CS interrupt. 0 = Mask MV_PS_CS 1 = Unmask MV_PS_CS	
SD_FR_CHNG_MSKB			R/W
0x44	000**0**0000	This control is used to mask the SD_FR_CHNG interrupt. 0 = Mask SD_FR_CHNG 1 = Unmask SD_FR_CHNG	
SD_UNLOCK_MSKB			R/W
0x44	0000000**0**	This control is used to mask the SD_UNLOCK interrupt. 0 = Mask SD_UNLOCK 1 = Unmask SD_UNLOCK	
SD_LOCK_MSKB			R/W
0x44	0000000**0**	This control is used to mask the SD_LOCK interrupt. 0 = Mask SD_LOCK 1 = Unmask SD_LOCK	
MPU_STIM_INTRQ_CLR			SC
0x47	**0**0000000	This control is used to clear the MPU_STIM_INTRQ interrupt. 0 = Do not clear MPU_STIM_INTRQ 1 = Clear MPU_STIM_INTRQ	
CHX_MIN_MAX_INTRQ_CLR			SC
0x47	0**0**000000	This control is used to clear the CHX_MIN_MAX_INTRQ interrupt. 0 = Do not clear CHX_MIN_MAX_INTRQ 1 = Clear CHX_MIN_MAX_INTRQ	
SD_FIELD_CHNGD_CLR			SC
0x47	000**0**0000	This control is used to clear the SD_FIELD_CHNGD interrupt. 0 = Do not clear SD_FIELD_CHNGD 1 = Clear SD_FIELD_CHNGD	

Reg	Bits	Description	
CCAPD_CLR			SC
0x47	0000000**0**	This control is used to clear the CCAPD interrupt. 0 = Do not clear CCAPD (VBI System 2) 1 = Clear CCAPD (VBI System 2)	
MPU_STIM_INTRQ_MSKB			R/W
0x48	**0**0000000	This control is used to mask the MPU_STIM_INTRQ interrupt. 0 = Mask MPU_STIM_INTRQ 1 = Unmask MPU_STIM_INTRQ	
CHX_MIN_MAX_INTRQ_MSKB			R/W
0x48	00**0**00000	This control is used to mask the CHX_MIN_MAX_INTRQ interrupt. 0 = Mask CHX_MIN_MAX_INTRQ 1 = Unmask CHX_MIN_MAX_INTRQ	
SD_FIELD_CHNGD_MSKB			R/W
0x48	000**0**0000	This control is used to mask the SD_FIELD_CHNGD interrupt. 0 = Mask SD_FIELD_CHNGD 1 = Unmask SD_FIELD_CHNGD	
CCAPD_MSKB			R/W
0x48	0000000**0**	This control is used to mask the CCAPD interrupt. 0 = Mask CCAPD 1 = Unmask CCAPD	
PAL_SW_LK_CHNG_CLR			SC
0x4B	00**0**00000	This control is used to clear the PAL_SW_LK_CHNG interrupt. 0 = Do not clear PAL_SW_LK_CHNG 1 = Clear PAL_SW_LK_CHNG	
SCM_LOCK_CHNG_CLR			SC
0x4B	000**0**0000	This control is used to clear the SCM_LOCK_CHNG interrupt. 0 = Do not clear SCM_LOCK_CHNG 1 = Clear SCM_LOCK_CHNG	
SD_AD_CHNG_CLR			SC
0x4B	0000**0**000	This control is used to clear the SD_AD_CHNG interrupt. 0 = Do not clear SD_AD_CHNG 1 = Clear SD_AD_CHNG	
SD_H_LOCK_CHNG_CLR			SC
0x4B	00000**0**00	This control is used to clear the SD_H_LOCK_CHNG interrupt. 0 = Do not clear SD_H_LOCK_CHNG 1 = Clear SD_H_LOCK_CHNG	
SD_V_LOCK_CHNG_CLR			SC
0x4B	000000**0**0	This control is used to clear the SD_V_LOCK_CHNG interrupt. 0 = Do not clear SD_V_LOCK_CHNG 1 = Clear SD_V_LOCK_CHNG	
SD_OP_CHNG_CLR			SC
0x4B	0000000**0**	This control is used to clear the SD_OP_CHNG interrupt. 0 = Do not clear SD_OP_CHNG 1 = Clear SD_OP_CHNG	
PAL_SW_LK_CHNG_MSKB			R/W
0x4C	00**0**00000	This control is used to mask the PAL_SW_LK_CHNG interrupt. 0 = Mask PAL_SW_LK_CHNG 1 = Unmask PAL_SW_LK_CHNG	
SCM_LOCK_CHNG_MSKB			R/W
0x4C	000**0**0000	This control is used to mask the SCM_LOCK_CHNG interrupt. 0 = Mask SCM_LOCK_CHNG 1 = Unmask SCM_LOCK_CHNG	
SD_AD_CHNG_MSKB			R/W
0x4C	0000**0**000	This control is used to mask the SD_AD_CHNG interrupt. 0 = Mask SD_AD_CHNG 1 = Unmask SD_AD_CHNG	

Reg	Bits	Description	
SD_H_LOCK_CHNG_MSKB			R/W
0x4C	00000000	This control is used to mask the SD_H_LOCK_CHNG interrupt. 0 = Mask SD_H_LOCK_CHNG 1 = Unmask SD_H_LOCK_CHNG	
SD_V_LOCK_CHNG_MSKB			R/W
0x4C	00000000	This control is used to mask the SD_V_LOCK_CHNG interrupt. 0 = Mask SD_V_LOCK_CHNG 1 = Unmask SD_V_LOCK_CHNG	
SD_OP_CHNG_MSKB			R/W
0x4C	00000000	This control is used to mask the SD_OP_CHNG interrupt. 0 = Mask SD_OP_CHNG 1 = Unmask SD_OP_CHNG	
VDP_VITC_CLR			SC
0x4F	00000000	This control is used to clear the VDP_VITC interrupt. 0 = Self clearing 1 = Clear VDP_VITC	
VDP_PDC_VPS_UTC_CHNG_CLR			SC
0x4F	00000000	This control is used to clear the VDP_GS_VPS_PDC_UTC_CHNG interrupt. 0 = Self clearing 1 = Clear VDP_GS_VPS_PDC_UTC_CHNG	
VDP_CGMS_WSS_CHNGD_CLR			SC
0x4F	00000000	This control is used to clear the VDP_CGMS_WSS_CHNGD interrupt. 0 = Do not clear VDP_CGMS_WSS_CHNGD 1 = Clear VDP_CGMS_WSS_CHNGD	
VDP_CCAPD_CLR			SC
0x4F	00000000	This control is used to clear the VDP_CCAPD interrupt. 0 = Do not clear VDP_CCAPD 1 = Clear VDP_CCAPD	
VDP_VITC_MSKB			R/W
0x50	00000000	This control is used to mask the VDP_VITC interrupt. 0 = Do not mask VDP_VITC 1 = Mask VDP_VITC	
VDP_PDC_VPS_UTC_CHNG_MSKB			R/W
0x50	00000000	This control is used to mask the VDP_GS_VPS_PDC_UTC_CHNG interrupt. 0 = Do not mask VDP_GS_VPS_PDC_UTC_CHNG 1 = Mask VDP_GS_VPS_PDC_UTC_CHNG	
VDP_CGMS_WSS_CHNGD_MSKB			R/W
0x50	00000000	This control is used to mask the VDP_CGMS_WSS_CHNGD interrupt. 0 = Unmask VDP_CGMS_WSS_CHNGD 1 = Mask VDP_CGMS_WSS_CHNGD	
VDP_CCAPD_MSKB			R/W
0x50	00000000	This control is used to mask the VDP_CCAPD interrupt. 0 = Unmask VDP_CCAPD 1 = Mask VDP_CCAPD	
AUTO_DETECT_GS_TYPE			R/W
0x61	00010000	This control is used for the autodetection of the Gemstar type. 0 = Disable autodetection of Gemstar type 1 = Enable autodetection of Gemstar type	
ADF_ENABLE			R/W
0x62	00010101	This control is used to insert VBI decoded data into the ITU-R BT.656 datastream as ancillary data. 0 = VBI decoded data not inserted into ancillary ITU-R BT.656 stream 1 = VBI decoded data inserted into ancillary ITU-R BT.656 stream	

Reg	Bits	Description
ADF_MODE[1:0]		R/W
0x62	00010101	This control is used to set the ADF mode for ancillary data. 00 = Nibble mode 01 = Byte mode, no code restrictions 10 = Byte mode with 0x00 and 0xFF prevented 11 = Reserved
ADF_DID[4:0]		R/W
0x62	00010101	This control is used to specify the value of the data identification (DID) sent in the ancillary stream with VBI decoded data. 10101 = User specified DID sent in the ancillary data stream with VDP decoded data
DUPLICATE_ADF		R/W
0x63	00101010	This control is used to duplicate the ancillary data across the Y and C data streams. 0 = Ancillary data packet spread across Y and C data streams 1 = Ancillary data packet duplicated on Y and C data streams
ADF_SDID[5:0]		R/W
0x63	00101010	This control is used to specify the secondary data identification (SDID) sent in the ancillary data stream with VDP decoded data. 101010 = User specified SDID sent in the ancillary data stream with VDP decoded data
MAN_LINE_PGM		R/W
0x64	00000000	This control allows the user to manually program the VBI standards for each of the lines in the VBI region. 0 = Decode default standards on lines. 1 = Manually program VBI standard to be decoded on each line. All VBI_DATA_Px_Ny bits can be set as desired.
VBI_DATA_P318[3:0]		R/W
0x64	00000000	This control is used to set the VBI standard to be decoded from Line 318 (PAL). It is not available for NTSC. MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 0101 = Reserved 0110 = Reserved 0111 = CCAP
VBI_DATA_P6_N23[3:0]		R/W
0x65	00000000	This control is used to set the VBI standard to be decoded from Line 6 (PAL), Line 23 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P319_N286[3:0]		R/W
0x65	00000000	This control is used to set the VBI standard to be decoded from Line 319 (PAL), Line 286 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)

Reg	Bits	Description	
VBI_DATA_P7_N24[3:0]			R/W
0x66	0000 0000	This control is used to set the VBI standard to be decoded from Line 7 (PAL), Line 24 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P320_N287[3:0]			R/W
0x66	0000 0000	This control is used to set the VBI standard to be decoded from Line 320 (PAL), Line 287 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P8_N25[3:0]			R/W
0x67	0000 0000	This control is used to set the VBI standard to be decoded from Line 8 (PAL), Line 25 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P321_N288[3:0]			R/W
0x67	0000 0000	This control is used to set the VBI standard to be decoded from Line 321 (PAL), Line 288 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P9[3:0]			R/W
0x68	0000 0000	This control is used to set the VBI standard to be decoded from Line 9 (PAL). It is not available for NTSC. MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 0101 = Reserved 0110 = Reserved 0111 = CCAP	

Reg	Bits	Description
VBI_DATA_P322[3:0]		R/W
0x68	0000<u>0000</u>	This control is used to set the VBI standard to be decoded from Line 322 (PAL). It is not available for NTSC. MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 0101 = Reserved 0110 = Reserved 0111 = CCAP
VBI_DATA_P10[3:0]		R/W
0x69	<u>0000</u>0000	This control is used to set the VBI standard to be decoded from Line 10 (PAL). It is not available for NTSC. MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 0101 = Reserved 0110 = Reserved 0111 = CCAP
VBI_DATA_P323[3:0]		R/W
0x69	0000<u>0000</u>	This control is used to set the VBI standard to be decoded from Line 323 (PAL). It is not available for NTSC. MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 0101 = Reserved 0110 = Reserved 0111 = CCAP
VBI_DATA_P11[3:0]		R/W
0x6A	<u>0000</u>0000	This control is used to set the VBI standard to be decoded from Line 11 (PAL). It is not available for NTSC. MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 0101 = Reserved 0110 = Reserved 0111 = CCAP
VBI_DATA_P324_N272[3:0]		R/W
0x6A	0000<u>0000</u>	This control is used to set the VBI standard to be decoded from Line 324 (PAL), Line 272 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)

Reg	Bits	Description
VBI_DATA_P12_N10[3:0]		R/W
0x6B	0000 0000	This control is used to set the VBI standard to be decoded from Line 12 (PAL), Line 10 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P325_N273[3:0]		R/W
0x6B	0000 0000	This control is used to set the VBI standard to be decoded from Line 325 (PAL), Line 273 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P13_N11[3:0]		R/W
0x6C	0000 0000	This control is used to set the VBI standard to be decoded from Line 13 (PAL), Line 11 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P326_N274[3:0]		R/W
0x6C	0000 0000	This control is used to set the VBI standard to be decoded from Line 326 (PAL), Line 274 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P14_N12[3:0]		R/W
0x6D	0000 0000	This control is used to set the VBI standard to be decoded from Line 14 (PAL), Line 12 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)

Reg	Bits	Description	
VBI_DATA_P327_N275[3:0]			R/W
0x6D	00000000	This control is used to set the VBI standard to be decoded from Line 327 (PAL), Line 275 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P15_N13[3:0]			R/W
0x6E	00000000	This control is used to set the VBI standard to be decoded from Line 15 (PAL), Line 13 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P328_N276[3:0]			R/W
0x6E	00000000	This control is used to set the VBI standard to be decoded from Line 328 (PAL), Line 276 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P16_N14[3:0]			R/W
0x6F	00000000	This control is used to set the VBI standard to be decoded from Line 16 (PAL), Line 14 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VBI_DATA_P329_N277[3:0]			R/W
0x6F	00000000	This control is used to set the VBI standard to be decoded from Line 329 (PAL), Line 277 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	

Reg	Bits	Description
VBI_DATA_P17_N15[3:0]		R/W
0x70	0000 0000	This control is used to set the VBI standard to be decoded from Line 17 (PAL), Line 15 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P330_N278[3:0]		R/W
0x70	0000 0000	This control is used to set the VBI standard to be decoded from Line 330 (PAL), Line 278 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P18_N16[3:0]		R/W
0x71	0000 0000	This control is used to set the VBI standard to be decoded from Line 18 (PAL), Line 16 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P331_N279[3:0]		R/W
0x71	0000 0000	This control is used to set the VBI standard to be decoded from Line 331 (PAL), Line 279 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P19_N17[3:0]		R/W
0x72	0000 0000	This control is used to set the VBI standard to be decoded from Line 19 (PAL), Line 17 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)

Reg	Bits	Description
VBI_DATA_P332_N280[3:0]		R/W
0x72	00000000	This control is used to set the VBI standard to be decoded from Line 332 (PAL), Line 280 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P20_N18[3:0]		R/W
0x73	00000000	This control is used to set the VBI standard to be decoded from Line 20 (PAL), Line 18 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P333_N281[3:0]		R/W
0x73	00000000	This control is used to set the VBI standard to be decoded from Line 333 (PAL), Line 281 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P21_N19[3:0]		R/W
0x74	00000000	This control is used to set the VBI standard to be decoded from Line 21 (PAL), Line 19 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P334_N282[3:0]		R/W
0x74	00000000	This control is used to set the VBI standard to be decoded from Line 334 (PAL), Line 282 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)

Reg	Bits	Description
VBI_DATA_P22_N20[3:0]		R/W
0x75	0000 0000	This control is used to set the VBI standard to be decoded from Line 22 (PAL), Line 20 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P335_N283[3:0]		R/W
0x75	0000 0000	This control is used to set the VBI standard to be decoded from Line 335 (PAL), Line 283 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P23_N21[3:0]		R/W
0x76	0000 0000	This control is used to set the VBI standard to be decoded from Line 23 (PAL), Line 21 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P336_N284[3:0]		R/W
0x76	0000 0000	This control is used to set the VBI standard to be decoded from Line 336 (PAL), Line 284 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)
VBI_DATA_P24_N22[3:0]		R/W
0x77	0000 0000	This control is used to set the VBI standard to be decoded from Line 24 (PAL), Line 22 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)

Reg	Bits	Description	
VBI_DATA_P337_N285[3:0]			R/W
0x77	0000<u>0000</u>	This control is used to set the VBI standard to be decoded from Line 337 (PAL), Line 285 (NTSC). MAN_LINE_PGM must be set to 1 for these bits to be effective. 0000 = Disable VDP 0001 = Reserved 0010 = VPS-ETSI EN 300 231 V 1.3.1 (PAL)/Reserved (NTSC) 0011 = VITC 0100 = WSS ITU-R BT.1119-1/ETSI.EN.300294 (PAL)/CGMS EIA-J CPR-1204/IEC 61880 (NTSC) 0101 = Reserved 0110 = Reserved 0111 = CCAP (PAL)/CCAP EIA-608 (NTSC)	
VITC_CLR			SC
0x78	0<u>0</u>000000	This control is used to clear the VITC readback registers. 0 = Do not clear VITC readback registers 1 = Clear VITC readback registers	
GS_PDC_VPS_UTC_CLR			SC
0x78	0000<u>0</u>0000	This control is used to clear the VPS, PDC or UTC data readback registers. 0 = Do not clear GS/PDC/VPS/UTC readback registers 1 = Clear GS/PDC/VPS/UTC readback registers	
CGMS_WSS_CLR			SC
0x78	000000<u>0</u>00	This control is used to clear the CGMS/WSS readback registers. 0 = Do not clear CGMS/WSS readback registers 1 = Clear CGMS/WSS readback registers	
CC_CLR			SC
0x78	0000000<u>0</u>	This control is used to clear the CCAP readback registers. 0 = Do not clear CCAP readback registers 1 = Clear CCAP readback registers	
WSS_CGMS_CB_CHANGE			R/W
0x9C	001<u>1</u>0000	This control can be used to enable content-based updating for WSS and CGMS. For standards such as WSS and CGMS, the information content in the signal transmitted remains the same over numerous lines; therefore, the user may want to be notified only when there is a change in the information content or loss of the information content. 0 = Disable content-based updating 1 = Enable content-based updating	

2.9 SDP R/O MAP 1

Reg	Bits	Description	
SUB_USR_EN_RB[1:0]			R
0x0E	00000000	This indicates which map is currently being accessed: the SDP Main Map, SDP Map 1, or SDP Map 2. This readback takes precedence over R_ONLY_MAPS_SEL_RB. 00 = Access SDP Main Map 01 = Access SDP Map 1 10 = Access SDP Map 2 11 = Reserved	
R_ONLY_MAPS_SEL_RB[2:0]			R
0x0E	00000000	This readback indicates which part of the sub map is selected; the read/write area of the sub map or the read only area of the sub map. This readback has a lower priority than SUB_USR_EN_RB; SUB_USR_EN_RB must be set to 00 for this readback to be effective. 000 = No Read-Only Map Selected 001 = SDP Read-Only Main Map 010 = SDP Read-Only Map 1 011 = SDP Read-Only Map 2 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	
MV_PS_CS_Q			R
0x42	00000000	This readback indicates that Macrovision pseudo syncs or color striping was detected. 0 = Pseudo sync/color striping not detected 1 = Pseudo sync/color striping detected	
SD_FR_CHNG_Q			R
0x42	00000000	This readback indicates that the free run status has changed. 0 = No change 1 = Change in free run status	
SD_UNLOCK_Q			R
0x42	00000000	This readback indicates that the SD core has changed from a locked to an unlocked state. 0 = No change 1 = SD input caused decoder to change from locked to unlocked state	
SD_LOCK_Q			R
0x42	00000000	This readback indicates that the SD core has changed from an unlocked to a locked state. 0 = No change 1 = SD input caused decoder to change from unlocked to locked state	
MPU_STIM_INTRQ			R
0x45	00000000	This readback indicates the manual interrupt set mode. 0 = Manual interrupt mode disabled 1 = Manual interrupt mode enabled	
CHX_MIN_MAX_INTRQ			R
0x45	00000000	This readback indicates if the input to ADC is within the correct range, as set by the controls in SDP Map 2. 0 = Input to ADC within correct range 1 = Input to ADC outside range	
EVEN_FIELD			R
0x45	00000000	This readback indicates if the current field is identified as odd or even. 0 = Odd field 1 = Even field	
CCAPD			R
0x45	00000000	This raw status readback indicates if CCAP signals or CCAP sequence is detected. If it is set to 1, it indicates that the data in the CCAP1 and CCAP2 registers is valid. The CCAPD bit goes high if the rising edge of the start bit is detected within a time window and if the polarity of the parity bit matches the data transmitted. 0 = No CCAP signals detected, confidence in decoded data low 1 = CCAP sequence detected, confidence high	
MPU_STIM_INTRQ_Q			R
0x46	00000000	This readback indicates if a manual interrupt was set. 0 = Manual interrupt not set 1 = Manual interrupt set	

Reg	Bits	Description	
CHX_MIN_MAX_INTRQ_Q			R
0x46	00**0**00000	This readback indicates if the input to the ADC is within the correct range. The range is set by SDP Map 2. 0 = Input to ADC within correct range 1 = Input to ADC outside range	
SD_FIELD_CHNGD_Q			R
0x46	000**0**0000	This readback indicates a changed Field polarity. 0 = SD signal did not change Field from odd to even or vice versa 1 = SD signal changed Field from odd to even or vice versa	
CCAPD_Q			R
0x46	0000000**0**	This readback indicates if CCAP data was detected. 0 = CCAP data not detected in input video signal 1 = CCAP data detected in video input signal	
SCM_LOCK			R
0x49	000**0**0000	This readback indicates if a SECAM lock was established. 0 = SECAM lock not established 1 = SECAM lock established	
SD_H_LOCK			R
0x49	00000**0**00	This readback indicates if an SD horizontal sync lock was established. 0 = SD horizontal sync lock not established 1 = SD horizontal sync lock established	
SD_V_LOCK			R
0x49	0000000**0**	This readback indicates if an SD vertical sync lock was established. 0 = SD vertical sync lock not established 1 = SD vertical sync lock established	
SD_OP_50HZ			R
0x49	0000000**0**	This raw status readback indicates if the frame rate at output was SD 50 Hz or 60 Hz. 0 = SD 60 Hz signal output 1 = SD 50 Hz signal output	
PAL_SW_LK_CHNG_Q			R
0x4A	00**0**00000	This readback indicates if there was a change in the PAL swinging burst lock status. 0 = No change in PAL swinging burst lock status 1 = Change in PAL swinging burst lock status	
SCM_LOCK_CHNG_Q			R
0x4A	000**0**0000	This readback indicates if there was a change in the SECAM lock status. 0 = No change in SECAM lock status 1 = Change in SECAM lock status	
SD_AD_CHNG_Q			R
0x4A	0000**0**000	This readback indicates if there was a change in the autodetection status. 0 = No change in AD_RESULT status 1 = Change in AD_RESULT status	
SD_H_LOCK_CHNG_Q			R
0x4A	00000**0**00	This readback indicates if there was a change in the HS lock status. 0 = No change in HS lock status 1 = Change in HS lock status	
SD_V_LOCK_CHNG_Q			R
0x4A	0000000**0**	This readback indicates if there was a change in the VS lock status. 0 = No change in VS lock status 1 = Change in VS lock status	
SD_OP_CHNG_Q			R
0x4A	0000000**0**	This readback indicates if there was a change from SD 60 Hz to 50 Hz frame rate (or vice versa) at the output. 0 = No change in SD signal standard detected at output 1 = Change in SD signal standard detected at output	
VDP_VITC_Q			R
0x4E	0**0**000000	This readback indicates if VITC data was detected. 0 = VITC data not detected 1 = VITC data detected	

Reg	Bits	Description	
VDP_PDC_VPS_UTC_CHNG_Q			R
0x4E	0000**0**0000	This readback indicates if a change in PDC, VPS, or UTC status was detected. 0 = No change in PDC, VPS, or UTC status 1 = Change in PDC, VPS, or UTC status	
VDP_CGMS_WSS_CHNGD_Q			R
0x4E	000000**0**00	This readback indicates if a change in CGMS or WSS status was detected. 0 = No change in CGMS or WSS status 1 = Change in CGMS or WSS status	
VDP_CCAPD_Q			R
0x4E	0000000**0**	This readback displays the CCAP detection status. 0 = CCAP not detected 1 = CCAP detected	
Y CHANNEL MIN VIOLATION			R
0x51	00**0**00000	This readback indicates if the Y value has reached the programmed minimum value. 0 = Y value above programmed minimum value 1 = Y value below programmed minimum value	
Y CHANNEL MAX VIOLATION			R
0x51	0000**0**000	This readback indicates if the Y value has reached the programmed maximum value. 0 = Y value below programmed maximum value 1 = Y value above programmed maximum value	
CB CHANNEL MIN VIOLATION			R
0x51	0000**0**000	This readback indicates if the Cb value has reached the programmed minimum value. 0 = Cb value above programmed minimum value 1 = Cb value below programmed minimum value	
CB CHANNEL MAX VIOLATION			R
0x51	000000**0**0	This readback indicates if the Cb value has reached the programmed maximum value. 0 = Cb value below programmed maximum value 1 = Cb value above programmed maximum value	
CR CHANNEL MIN VIOLATION			R
0x51	0000000**0**	This readback indicates if the Cr value has reached the programmed minimum value. 0 = Cr value above programmed minimum value 1 = Cr value below programmed minimum value	
CR CHANNEL MAX VIOLATION			R
0x51	0000000**0**	This readback indicates if the Cr value has reached the programmed maximum value. 0 = Cr value below programmed maximum value 1 = Cr value above programmed maximum value	
DIAG_TRI2_L1			R
0x53	0000**0**000	This readback indicates if the programmed voltage level was exceeded on DIAG2. 0 = Programmed voltage level not exceeded on DIAG2 1 = Programmed voltage level exceeded on DIAG2	
DIAG_TRI2_L0			R
0x53	000000**0**0	This readback indicates the diagnostic lower level trip point interrupt status. 0 = Lower level trip point active 1 = Lower level trip point inactive	
DIAG_TRI1_L1			R
0x53	000000**0**0	This readback indicates if the programmed voltage level was exceeded on DIAG1. 0 = Programmed voltage level not exceeded on DIAG1 1 = Programmed voltage level exceeded on DIAG1	
DIAG_TRI1_L0			R
0x53	0000000**0**	This readback indicates the diagnostic lower level trip point interrupt status. 0 = Lower level trip point active 1 = Lower level trip point inactive	
CGMS_WSS_AVL			R
0x78	000000**0**0	This readback displays the WSS or CGMS Type A data detection status. 0 = CGMS/WSS not detected 1 = CGMS/WSS detected	

Reg	Bits	Description	
CC_EVEN_FIELD			R
0x78	0000000<u>0</u>	This readback displays the CCAP data in the even field status. 0 = CCAP decoded from odd field 1 = CCAP decoded from even field	
CC_AVL			R
0x78	0000000<u>0</u>	This readback displays the CCAP data detection status. 0 = CCAP not detected 1 = CCAP detected	
VDP_CCAP_DATA_0[7:0]			R
0x79	<u>00000000</u>	This readback displays byte 1 of the decoded CCAP data. 00000000 = Decoded CCAP Byte 1	
VDP_CCAP_DATA_1[7:0]			R
0x7A	<u>00000000</u>	This readback displays byte 2 of the decoded CCAP data. 00000000 = Decoded CCAP Byte 2	
VDP_CGMS_WSS_DATA_CRC[5:0]			R
0x7D 0x7E	0000<u>0000</u> <u>00</u>000000	This readback indicates if the decoded CGMS (SD) data passed the CRC check. 0 = CGMS failed CRC 1 = CGMS passed CRC	
VDP_CGMS_WSS_DATA[13:0]			R
0x7E 0x7F	00<u>000000</u> <u>00000000</u>	This readback displays decoded data for CGMS Type A and WSS. 000000 = Decoded CGMS/WSS data	
VPS_PDC_UTC_BYTE_0[7:0]			R
0x84	<u>00000000</u>	This readback displays the decoded VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded Gemstar/VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_1[7:0]			R
0x85	<u>00000000</u>	This readback displays the decoded VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded Gemstar/VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_2[7:0]			R
0x86	<u>00000000</u>	This readback displays the decoded VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded Gemstar/VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_3[7:0]			R
0x87	<u>00000000</u>	This readback displays the decoded VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded Gemstar/VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_4[7:0]			R
0x88	<u>00000000</u>	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_5[7:0]			R
0x89	<u>00000000</u>	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_6[7:0]			R
0x8A	<u>00000000</u>	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_7[7:0]			R
0x8B	<u>00000000</u>	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_8[7:0]			R
0x8C	<u>00000000</u>	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_9[7:0]			R
0x8D	<u>00000000</u>	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	

Reg	Bits	Description	
VPS_PDC_UTC_BYTE_10[7:0]			R
0x8E	00000000	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_11[7:0]			R
0x8F	00000000	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_PDC_UTC_BYTE_12[7:0]			R
0x90	00000000	This readback displays the VPS, PDC, UTC, or CGMS Type B data. 00000000 = Decoded VPS/PDC/UTC data	
VPS_BIPHASE_ERROR_COUNT[7:0]			R
0x91	00000000	This readback indicates the number of errors encountered while decoding the biphase VPS standard. Each bit is represented by a positive or negative transition. The error count is increased if these transitions are not detected by the VDP during the bit time.	
VITC_DATA_1[7:0]			R
0x92	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 1	
VITC_DATA_2[7:0]			R
0x93	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 2	
VITC_DATA_3[7:0]			R
0x94	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 3	
VITC_DATA_4[7:0]			R
0x95	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 4	
VITC_DATA_5[7:0]			R
0x96	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 5	
VITC_DATA_6[7:0]			R
0x97	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 6	
VITC_DATA_7[7:0]			R
0x98	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 7	
VITC_DATA_8[7:0]			R
0x99	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 8	
VITC_DATA_9[7:0]			R
0x9A	00000000	This readback displays decoded VITC data. 00000000 = Decoded VITC Data Byte 9	
VITC_CRC[7:0]			R
0x9B	00000000	This readback indicates the calculated CRC value for decoded VITC data. 00000000 = Decoded VITC CRC	

2.10 SDP MAP 2

Reg	Bits	Description	
ACE_ENABLE			R/W
0x80	00000000	This control is used to enable ACE. 0 = Bypass ACE 1 = Enable ACE	
ACE_LUMA_GAIN[4:0]			R/W
0x83	00001101	This control is used to set the autocontrast level for the luma channel when ACE_ENABLE is set to 1. 00000 = Set to minimum 01101 = Default 11111 = Set to maximum	
ACE_CHROMA_MAX[3:0]			R/W
0x84	10001000	This control is used to set the maximum autosaturation for the color channels when ACE_ENABLE is set to 1. 0000 = Set to minimum 1000 = Default 1111 = Set to maximum	
ACE_CHROMA_GAIN[3:0]			R/W
0x84	10001000	This control is used to set the autosaturation level for the color channels when ACE_ENABLE is set to 1. 0000 = Set to minimum 1000 = Default 1111 = Set to maximum	
ACE_RESPONSE_SPEED[3:0]			R/W
0x85	11111000	This control is used to set the response speed for autocontrast to take effect when ACE_ENABLE is set to 1. The updated image is alpha blended over the old image using this response speed. 0001 = Set to minimum 1111 = Set to maximum	
ACE_GAMMA_GAIN[3:0]			R/W
0x85	11111000	This control is used to provide further contrast enhancement to the luma and chroma gain controls and is particularly effective in the darker areas of an image. 1000 = Default	
SDP_BR_DITHER_MODE			R/W
0x92	00000110	This control is used to activate the dither function. The dither function converts the digital output of the SD core from 8-bit pixel data down to 6-bit pixel data. This makes it easier to communicate with some LCD panels. 0 = 8-bit to 6-bit down dither disabled 1 = 8-bit to 6-bit down dither enabled	
MIN_THRESH_Y[7:0]			R/W
0xD9	00000000	This control is used to select the minimum threshold for the incoming luma video signal. 00000000 = Default	
MAX_THRESH_Y[7:0]			R/W
0xDA	11111111	This control is used to select the maximum threshold for the incoming luma video signal. 11111111 = Default	
MIN_THRESH_C[7:0]			R/W
0xDB	00000000	This control is used to select the minimum threshold for the incoming chroma video signal. 00000000 = Default	
MAX_THRESH_C[7:0]			R/W
0xDC	11111111	This control is used to select the maximum threshold for the incoming chroma video signal. 11111111 = Default	
MIN_SAMPLES_ALLOWED_Y[3:0]			R/W
0xDD	11001100	This control is used to select the number of minimum luma samples allowed in a given window before an interrupt is triggered. 1100 = Default	
MAX_SAMPLES_ALLOWED_Y[3:0]			R/W
0xDD	11001100	This control is used to select the number of minimum luma samples allowed in a given window before an interrupt is triggered. 1100 = Default	

Reg	Bits	Description	
MIN_SAMPLES_ALLOWED_C[3:0]			R/W
0xDE	1100 1100	This control is used to select the number of minimum chroma samples allowed in a given window before an interrupt is triggered. 1100 = Default	
MAX_SAMPLES_ALLOWED_C[3:0]			R/W
0xDE	1100 1100	This control is used to select the number of maximum chroma samples allowed in a given window before an interrupt is triggered. 1100 = Default	
FL_ENABLE			R/W
0xE0	0000000 0	This control is used to enable fast locking of the SD core. 0 = Disable fast locking 1 = Enable fast locking	
LINE_START[8:0]			R/W
0xE1 0xE5	00010001 0010001 1	This control is used to select the start line for field averaging.	
LINE_END[8:0]			R/W
0xE2 0xE5	10001000 0010001 1	This control is used to select the end line for field averaging.	
SAMPLE_START[9:0]			R/W
0xE3 0xE5	00011011 0010 0011	This control is used to select the start sample for line averaging.	
SAMPLE_END[9:0]			R/W
0xE4 0xE5	11010111 00 100011	This control is used to select the end sample for line averaging.	
Y_AVG_TIME_CONST[2:0]			R/W
0xE6	000 100 00	This control is used to select the filter cutoff for filtering the Y averaged data. 1xx = Least filtered 000 = Next least filtered 011 = Heavily filtered	
Y_AVG_FILT_EN			R/W
0xE6	00010 0 0	This control is used to enable low-pass filtering of the Y averaged signal. 0 = Disable low-pass filtering 1 = Enable low-pass filtering	
CAPTURE_VALUE			SC
0xE7	0000000 0	This control is used as a trigger to store the readback value. 0 = Not used 1 = Store readback value	

2.11 SDP R/O MAP 2

Reg	Bits	Description	
SUB_USR_EN_RB[1:0]			R
0x0E	00000000	This indicates which map is currently being accessed: the SDP Main Map, SDP Map 1, or SDP Map 2. This readback takes precedence over R_ONLY_MAPS_SEL_RB. 00 = Access SDP Main Map 01 = Access SDP Map 1 10 = Access SDP Map 2 11 = Reserved	
R_ONLY_MAPS_SEL_RB[2:0]			R
0x0E	00000000	This readback indicates which part of the sub map is selected; the read/write area of the sub map; or the read only area of the sub map. This readback has a lower priority than SUB_USR_EN_RB; SUB_USR_EN_RB must be set to 00 for this readback to be effective. 000 = No Read-Only Map Selected 001 = SDP Read-Only Main Map 010 = SDP Read-Only Map 1 011 = SDP Read-Only Map 2 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	
RB_Y_AVERAGE_DATA[9:0]			R
0xE7 0xE8	00000000 00000000	This readback displays the averaged video data.	

2.12 DPLL MAP

Reg	Bits	Description	
MCLK_FS_N[2:0]			R/W
0xB5	00000001	Selects the multiple of 128 f_S used for MCLK output. 000 = 128 f_S 001 = 256 f_S 010 = 384 f_S 011 = 512 f_S 100 = 640 f_S 101 = 768 f_S 110 = Not Valid 111 = Not Valid	

2.13 CP MAP

Reg	Bits	Description	
RB_CSC_SCALE[1:0]			R
0x0B	00000000	This readback displays the CSC scale applied to CSC coefficients.	
RB_A4[12:0]			R
0x0B 0x0C	00000000 00000000	This readback displays the CSC coefficient, A4, modified by the video adjustment block.	
RB_A3[12:0]			R
0x0D 0x0E	00000000 00000000	This readback displays the CSC coefficient, A3, modified by the video adjustment block.	
RB_A2[12:0]			R
0x0E 0x0F 0x10	00000000 00000000 00000000	This readback displays the CSC coefficient, A2, modified by the video adjustment block.	
RB_A1[12:0]			R
0x10 0x11	00000000 00000000	This readback displays the CSC coefficient, A1, modified by the video adjustment block.	
RB_B4[12:0]			R
0x12 0x13	00000000 00000000	This readback displays the CSC coefficient, B4, modified by the video adjustment block.	
RB_B3[12:0]			R
0x14 0x15	00000000 00000000	This readback displays the CSC coefficient, B3, modified by the video adjustment block.	
RB_B2[12:0]			R
0x15 0x16 0x17	00000000 00000000 00000000	This readback displays the CSC coefficient, B2, modified by the video adjustment block.	
RB_B1[12:0]			R
0x17 0x18	00000000 00000000	This readback displays the CSC coefficient, B1, modified by the video adjustment block.	
RB_C4[12:0]			R
0x19 0x1A	00000000 00000000	This readback displays the CSC coefficient, C4, modified by the video adjustment block.	
RB_C3[12:0]			R
0x1B 0x1C	00000000 00000000	This readback displays the CSC coefficient, C3, modified by the video adjustment block.	
RB_C2[12:0]			R
0x1C 0x1D 0x1E	00000000 00000000 00000000	This readback displays the CSC coefficient, C2, modified by the video adjustment block.	
RB_C1[12:0]			R
0x1E 0x1F	00000000 00000000	This readback displays the CSC coefficient, C1, modified by the video adjustment block.	
DE_V_START_R[3:0]			R/W
0x30	00000000	This control is used to vary the position of the start of the extra VBI region between the left and right fields during the odd field in the field alternative packing in 3D TV video format. It stores a signed value represented in a twos complement format. The unit of DE_V_START_R is one line. 1000 to 1111 = −8 lines to −1 line 0000 = Default value (0 lines) 0001 to 0111 = 1 line to 7 lines	

Reg	Bits	Description
DE_V_END_R[3:0]		R/W
0x30	00000000	This control is used to vary the position of the end of the extra VBI region between the left and right fields during the odd field in the field alternative packing in 3D TV video format. It stores a signed value represented in a twos complement format. The unit of DE_V_END_R is one line. 1000 to 1111 = −8 lines to −1 line 0000 = Default value (0 lines) 0001 to 0111 = 1 line to 7 lines
DE_V_START_EVEN_R[3:0]		R/W
0x31	00000000	This control is used to vary the position of the start of the extra VBI region between L and R fields during the even field in the field alternative packing in 3D TV video format. It stores a signed value represented in a twos complement format. The unit of DE_V_START_EVEN_R[3:0] is one line. 1000 to 1111 = −8 lines to −1 line 0000 = Default value (0 lines) 0001 to 0111 = 1 line to 7 lines
DE_V_END_EVEN_R[3:0]		R/W
0x31	00000000	This control is used to vary the position of the end of the extra VBI region between L and R fields during the even field in the field alternative packing in 3D TV video format. It stores a signed value represented in a twos complement format. The unit of DE_V_END_EVEN_R[3:0] is one line. 1000 to 1111 = −8 lines to −1 line 0000 = Default value (0 lines) 0001 to 0111 = 1 line to 7 lines
CP_INT_PAT_GEN_EN		R/W
0x37	00000000	This control can be used to enable the internal pattern generator in the CP block. 0 = CP internal pattern generator disabled 1 = CP internal pattern generator enabled
CP_BORDER_PIX_EN		R/W
0x37	00000000	This control enables a border of 1 or 2 pixels to be added to the output of the CP pattern generator. It is only applicable if the CP pattern generator is enabled. 0 = Pixel border disabled 1 = Pixel border enabled
CP_BORDER_WIDTH		R/W
0x37	00000000	This control can be used to set the width of the pixel border added to the output of the CP pattern generator. 0 = Border width is 1 pixel 1 = Border width is 2 pixels
CP_PAT_GEN_SEL[2:0]		R/W
0x37	00000000	This control can be used to select the pattern output by the CP pattern generator. 000 = Plain blue 001 = Color bars 010 = Ramp grey 011 = Ramp blue 100 = Ramp red 101 = Checker board
CP_CONTRAST[7:0]		R/W
0x3A	10000000	This control is used to set the contrast. It is an unsigned value represented in a 1.7 binary format. The MSB represents the integer part of the contrast value, which is either 0 or 1. The seven LSBs represent the fractional part of the contrast value. The fractional part has the range 0 to 0.99. This control is functional if VID_ADJ_EN is set to 1. 00000000 = Contrast set to minimum 10000000 = Default 11111111 = Contrast set to maximum
CP_SATURATION[7:0]		R/W
0x3B	10000000	This control is used to set the saturation. It is an unsigned value represented in a 1.7 binary format. The MSB represents the integer part of the saturation value, which is either 0 or 1. The seven LSBs represent the fractional part of the saturation value. The fractional part has a 0 to 0.99 range. This control is functional if VID_ADJ_EN is set to 1. 00000000 = Saturation set to minimum 10000000 = Default 11111111 = Saturation set to maximum

Reg	Bits	Description	
CP_BRIGHTNESS[7:0]			R/W
0x3C	00000000	This control is used to set the brightness. It is a signed value. The effective brightness value applied to the luma is obtained by multiplying the programmed value CP_BRIGHTNESS with a gain of 4. The brightness applied to the luma has a range of −512 to +508. This control is functional if VID_ADJ_EN is set to 1. 00000000 = Offset applied to luma is 0. 01111111 = Offset applied to luma is 508d. This value corresponds to brightest setting. 10000000 = Offset applied to luma is −512d. This value corresponds to darkest setting.	
CP_HUE[7:0]			R/W
0x3D	00000000	This control is used to set the hue. This control represents an unsigned value which provides hue adjustment. The effective hue applied to the chroma is [(CP_HUE[7:0] × 180)/256 − 90]. The range of the effective hue applied to the chroma is 0° to 360°. This control is functional if VID_ADJ_EN is set to 1. 00000000 = Hue of 0° applied to chroma 01000000 = Hue of 90° applied to chroma 10000000 = Hue of 180° applied to chroma 11000000 = Hue of 270° applied to chroma 11111111 = Hue of 90° applied to chroma	
VID_ADJ_EN			R/W
0x3E	0 0000000	This control is used to enable video adjustment. It is used to select whether or not the color controls feature is enabled. The color controls feature is configured via the CP_CONTRAST[7:0], CP_SATURATION[7:0], CP_BRIGHTNESS[7:0], and CP_HUE[7:0] parameters. The CP CSC must also be enabled for the color controls to be effective. 0 = Disable color controls 1 = Enable color controls	
CP_UV_ALIGN_SEL[1:0]			R/W
0x3E	0000 0000	This control is used to adjust the timing of the chroma stream. It adjusts the timing of the Cr and Cb interleaved stream output by the CP core for 4:2:2 output modes. 00 = Chroma stream is synchronous with start of active video 01 = Chroma stream is synchronous with leading edge of HS 10 = Chroma stream is synchronous with leading edge of DE 11 = Chroma stream is synchronized with start of active video	
CP_UV_DVAL_INV			R/W
0x3E	0000 0000	This control is used to swap the order of the Cr and Cb in the chroma stream for 4:2:2 output modes. 0 = Do not swap order of Cr and Cb samples 1 = Swap order of Cr and Cb samples	
ALT_SAT_UV_MAN			R/W
0x3E	0000000 0	This control is used to define the U and V saturation range. 0 = Range of saturator on Cr and Cb channels determined by CP_OP_656_RANGE and CP_ALT_DATA_SAT 1 = Range of saturator on Cr and Cb channels determined by ALT_SAV_UV if CP_OP_656_RANGE or CP_ALT_DATA_SAT set to 0	
ALT_SAT_UV			R/W
0x3E	0000000 0	This control is used to define the Cr and Cb saturation range. Refer to the description of ALT_SAT_UV_MAN for additional details. 0 = Range of saturators on Cr and Cb channels is 15 to 235 1 = Range of saturators on Cr and Cb channels is 16 to 240	
CSC_SCALE[1:0]			R/W
0x52	01 000000	This control is used to set the CSC coefficient scalar. 00 = CSC scalar set to 1 01 = CSC scalar set to 2 10 = Reserved 11 = Reserved	
A4[12:0]			R/W
0x52 0x53	010 00000 00000000	This control is used to set the CSC coefficient, A4. It contains a 13-bit A4 coefficient for the A channel. 0x0000 = Default value	
A3[12:0]			R/W
0x54 0x55	0 0000000 000000 00	This control is used to set the CSC coefficient, A3. It contains a 13-bit A3 coefficient for the A channel. 0x0000 = Default value	

Reg	Bits	Description	
A2[12:0]			R/W
0x55 0x56 0x57	00000000 00000000 00001000	This control is used to set the CSC coefficient, A2. It contains s 13-bit A2 coefficient for the A channel. 0x0000 = Default value	
A1[12:0]			R/W
0x57 0x58	00001000 00000000	This control is used to set the CSC coefficient, A1. It contains a 13-bit A1 coefficient for the A channel. 0x0800 = Default value	
B4[12:0]			R/W
0x59 0x5A	00000000 00000000	This control is used to set the CSC coefficient, B4. It contains a 13-bit B4 coefficient for the B channel. 0x0000 = Default value	
B3[12:0]			R/W
0x5B 0x5C	00000000 00000001	This control is used to set the CSC coefficient, B3. It contains a 13-bit B3 coefficient for the B channel. 0x0000 = Default value	
B2[12:0]			R/W
0x5C 0x5D 0x5E	00000001 00000000 00000000	This control is used to set the CSC coefficient, B2. It contains a 13-bit B2 coefficient for the B channel. 0x0800 = Default value	
B1[12:0]			R/W
0x5E 0x5F	00000000 00000000	This control is used to set the CSC coefficient, B1. It contains a 13-bit B1 coefficient for the B channel. 0x0000 = Default value	
C4[12:0]			R/W
0x60 0x61	00000000 00000000	This control is used to set the CSC coefficient, C4. It contains a 13-bit C4 coefficient for the C channel. 0x0000 = Default value	
C3[12:0]			R/W
0x62 0x63	00100000 00000000	This control is used to set the CSC coefficient, C3. It contains a 13-bit C3 coefficient for the C channel. 0x0800 = Default value	
C2[12:0]			R/W
0x63 0x64 0x65	00000000 00000000 00000000	This control is used to set the CSC coefficient, C2. It contains a 13-bit C2 coefficient for the C channel. 0x0000 = Default value	
C1[12:0]			R/W
0x65 0x66	00000000 00000000	This control is used to set the CSC coefficient, C1. It contains a 13-bit C1 coefficient for the C channel. 0x0000 = Default value	
CSC_COEFF_SEL[3:0]			R/W
0x68	11110000	This control is used to select the mode in which the CP CSC operates. 0000 = CP CSC configuration in manual mode 1111 = CP CSC configured in automatic mode xxxx = All other values	
CP_CHROMA_LOW_EN			R/W
0x68	11110000	This control is used to set filter response for the 444 to 422 chroma decimation filter. 0 = High bandwidth, sharp transition filter for Channel B/Channel C 1 = Soft filter with minimized ringing for Channel B/Channel C	
MAN_CP_DECIM_EN			R/W
0x69	00000100	This control is used to manually enable the CP chroma decimation filter. By default, the CP filter is automatically enabled in the case where the output format is required to be 422 based on OP_FORMAT_SEL. 0 = CP decimation automatically enabled if required 1 = Manual override to force CP decimation to be enabled	

Reg	Bits	Description
MAN_CP_CSC_EN		R/W
0x69	00000100	This control is used to manually enable the CP CSC. By default, the CP CSC is automatically enabled when either a color space conversion or video adjustment (hue, saturation, contrast, or brightness) is determined to be required due to other I^2C settings. If MAN_CP_CSC_EN is set to 1, the CP CSC is forced into the enabled state. 0 = CP CSC automatically enabled if required 1 = Manual override to force CP CSC to be enabled
EIA_861_COMPLIANCE		R/W
0x69	00000100	This control is used to implement compliance to the CEA 861 standard for 525p inputs. It affects the start of the VBI for the 525p standard only. 0 = VBI region starts on Line 1 1 = VBI region starts on Line 523 (compliant with 861 specification)
CP_PREC[1:0]		R/W
0x77	11111111	This control is used to set the precision of the data output by the CP core for A, B, and C channels. 00 = Rounds and truncates data in A, B and C channels to 10-bit precision 01 = Rounds and truncates data in A, B and C channels to 12-bit precision 10 = Rounds and truncates data in A, B and C channels to 8-bit precision 11 = Reserved
AV_INV_F		R/W
0x7B	00000101	This control is used to invert the F bit in the AV codes. 0 = Insert F bit with default polarity 1 = Invert F bit before inserting it into AV code
AV_INV_V		R/W
0x7B	00000101	This control is used to invert the V bit in AV codes. 0 = Do not invert V bit polarity before inserting it into AV code 1 = Invert V bit polarity before inserting it into AV code
AV_POS_SEL		R/W
0x7B	00000101	This control is used to select the position of AV codes. 0 = SAV code at HS falling edge and EAV code at HS rising edge 1 = Use predetermined (default) positions for AV codes
CP_INV_HS		R/W
0x7C	11000000	A control to set the polarity of the HS/composite sync output by the CP core. 0 = The CP outputs a HS/composite sync with positive polarity 1 = The CP outputs a HS/composite sync with negative polarity
CP_INV_VS		R/W
0x7C	11000000	A control to set the polarity of the VS output by the CP core. 0 = The CP outputs a VS with positive polarity 1 = The CP outputs a VS with negative polarity
START_HS[9:0]		R/W
0x7C 0x7E	11000000 00000000	This control is used to shift the position of the leading edge of the HS output by the CP core. It stores a signed value in a twos complement format. This control is the number of pixel clocks by which the leading edge of the HS is shifted (for example, 0x3FF corresponds to a shift of one pixel clock away from the active video, and 0x005 corresponds to a shift of five pixel clocks toward the active video). 0x000 = Default value 0x000 to 0x1FF = Leading edge of HS shifted toward active video 0x200 to 0x3FF = Leading edge of HS shifted away from active video
END_HS[9:0]		R/W
0x7C 0x7D	11000000 00000000	This control is used to shift the position of the trailing edge of the HS output by the CP core. It stores a signed value in a twos complement format. This control is the number of pixel clocks by which the trailing edge of the HS is shifted (for example, 0x3FF corresponds to a shift of one pixel clock away from the active video, and 0x005 corresponds to a shift of five pixel clocks towards the active video). 0x000 = Default value 0x000 to 0x1FF = Trailing edge of HS shifted toward active video 0x200 to 0x3FF = Trailing edge of HS shifted away from active video

Reg	Bits	Description
START_VS[3:0]		R/W
0x7F	0000 0000	This control is used to shift the position of the leading edge of the VS output by the CP core. It stores a signed value in a twos complement format. This control is the number of lines by which the leading edge of the VS is shifted (for example, 0x0F corresponds to a shift by one line toward the active video, and 0x01 corresponds to a shift of one line away from the active video). 0x0 = Default value 0x0 to 0x7 = Leading edge of VS shifted toward active video 0x8 to 0xF = Leading edge of VS shifted away from active video
END_VS[3:0]		R/W
0x7F	0000 0000	This control is used to shift the position of the trailing edge of the VS output by the CP core. It stores a signed value in a twos complement format. This control is the number of lines by which the trailing edge of the VS is shifted (for example, 0x0F corresponds to a shift of one line toward the active video, and 0x01 corresponds to a shift of one line away from the active video). 0x0 = Default value 0x0 to 0x7 = Trailing edge of VS shifted toward active video 0x8 to 0xF = Trailing edge of VS shifted away from active video
DE_V_START_EVEN[5:0]		R/W
0x87	00 000000	This control is used to vary the start position of the VBI region in an even field. It stores a signed value represented in a twos complement format. The unit of adjustment is one line. 100000…111111 = −32 lines … −1 line 000000 = Default value (0 lines) 000000…011111 = 1 line … 31 lines
DE_V_END_EVEN[5:0]		R/W
0x88	00 000000	This control is used to vary the position of the end of the VBI region in an even field. It stores a signed value represented in a twos complement format. The unit of adjustment is one line. 100000…111111 = −32 lines … −1 line 000000 = Default value (0 lines) 000000…011111 = 1 line … 31 lines
START_VS_EVEN[3:0]		R/W
0x89	0000 0000	This control is used to shift the position of the leading edge of the VS output by the CP core. It stores a signed value in a twos complement format. START_VS_EVEN[3:0] is the number of lines by which the leading edge of the VS is shifted (for example, 0x0F corresponds to a shift of one line toward the active video, and 0x01 corresponds to a shift of one line away from the active video). 0x0 to 0x7 = Leading edge of even VS shifted towards active video 0x8 to 0xF = Leading edge of even VS shifted away from active video
END_VS_EVEN[3:0]		R/W
0x89	0000 0000	This control is used to shift the position of the trailing edge of the VS output by the CP core. It stores a signed value in a twos complement format. END_VS_EVEN[3:0] is the number of lines by which the trailing edge of the VS is shifted (for example, 0x0F corresponds to a shift of one line toward the active video, and 0x01 corresponds to a shift of one line away from the active video). 0x0 to 0x7 = Trailing edge of even VS shifted toward active video 0x8 to 0xF = Trailing edge of even VS shifted away from active video
DE_H_START[9:0]		R/W
0x8B 0x8D	0100 0000 00000000	This control is used to vary the leading edge position of the DE signal output by the CP core. It stores a signed value in a twos complement format. The unit of DE_H_START[9:0] is one pixel clock. 0x200 = −512 pixels of shift 0x3FF = −1 pixel of shift 0x000 = Default value (no shift) 0x001 = +1 pixel of shift 0x1FF = +511 pixels
DE_H_END[9:0]		R/W
0x8B 0x8C	0100 0000 00000000	This control is used to vary the trailing edge position of the DE signal output by the CP core. It stores a signed value in a twos complement format. The unit of DE_H_END[9:0] is one pixel clock. 0x200 = −512 pixels of shift 0x3FF = −1 pixel of shift 0x000 = Default value (no shift) 0x001 = +1 pixel of shift 0x1FF = +511 pixels

Reg	Bits	Description	
INTERLACED			R/W
0x91	01000000	This control is used to set the interlaced/progressive mode of the incoming video processed in CP mode. 0 = CP core expects video mode is progressive 1 = CP core expects video mode is interlaced	
INTERLACED_3D			R/W
0x91	01000000	This control is used to set the interlaced/progressive mode of the incoming 3D video processed in CP mode. 0 = CP core expects 3D video mode is progressive 1 = CP core expects 3D video mode is interlaced	
DE_V_START[5:0]			R/W
0x98	00000000	This control is used to vary the start position of the VBI region. It stores a signed value represented in a twos complement format. The unit of DE_V_START[5:0] is one line. 100000 = −32 lines of shift 1111 11 = −1 line of shift 000000 = no shift (default) 000001 = +1 line of shift 011111 = +31 lines of shift	
DE_V_END[5:0]			R/W
0x99	00000000	This control is used to vary the position of the end of the VBI region. It stores a signed value represented in a twos complement format. The unit of DE_V_END[5:0] is one line. 100000 = −32 lines of shift 1111 11 = −1 line of shift 000000 = no shift (default) 000001 = +1 line of shift 011111 = +31 lines of shift	
HDMI_FRUN_MODE			R/W
0xBA	00000001	This control is used to configure the free run feature in HDMI/MHL mode. 0 = HDMI/MHL free run Mode 0. Device free runs when TMDS clock is not detected on selected HDMI/MHL port. 1 = HDMI/MHL free run Mode 1. CP core free runs when TMDS clock is not detected on selected HDMI/MHL port or if video resolution of HDMI/MHL stream processed by device does not match video resolution programmed in CP_VID_STD[5:0] and CP_V_FREQ[2:0].	
HDMI_FRUN_EN			R/W
0xBA	00000001	This control is used to enable free run in HDMI/MHL mode. 0 = Disable free run feature in HDMI/MHL mode 1 = Enable free run feature in HDMI/MHL mode	
HCOUNT_ALIGN_ADJ[4:0]			R/W
0xBE 0xBF	00000100 00010010	This control is used to manually adjust for internally generated HCOUNT offset. The control allows an adjustment of 15 pixels to the left or to the right. The MSB sets the direction (left or right) and the 4 LSBs set the number of pixels to move. This is an unsigned control. 00000 = no adjustment (default value)	
CP_DEF_COL_MAN_VAL			R/W
0xBF	00010010	This control is used to enable the manual selection of the color used when the CP core free runs. 0 = Use default color blue 1 = Output default colors as given in DEF_COL_CHA[7:0], DEF_COL_CHB[7:0], DEF_COL_CHC[7:0].	
CP_DEF_COL_AUTO			R/W
0xBF	00010010	This control is used to enable the insertion of the default color when the CP free runs. 0 = Disable automatic insertion of default color 1 = Output default colors when CP free runs	
CP_FORCE_FREERUN			R/W
0xBF	00010010	This control is used to force the CP to free run. 0 = Do not force CP core free run 1 = Force CP core to free run	
DEF_COL_CHA[7:0]			R/W
0xC0	00000000	This control is used to set the default color for Channel A. It is used if CP_DEF_COL_MAN_VAL is set at 1. 0x00 = Default value	
DEF_COL_CHB[7:0]			R/W
0xC1	00000000	This control is used to set the default color for Channel B. It is used if CP_DEF_COL_MAN_VAL is set at 1. 0x00 = Default value	

Reg	Bits	Description	
DEF_COL_CHC[7:0]			R/W
0xC2	00000000	This control is used to set the default color for Channel C. It is used if CP_DEF_COL_MAN_VAL is set at 1. 0x00 = Default value	
SWAP_SPLIT_AV			R/W
0xC9	00101100	This control is used to swap the luma and chroma AV codes in DDR modes. 0 = Swap luma and chroma AV codes in DDR mode 1 = Do not swap luma and chroma AV codes in DDR mode	
DIS_AUTO_PARAM_BUFF			R/W
0xC9	00101100	This control is used to disable the buffering of the timing parameters used for free run in HDMI/MHL mode. 0 = Enable buffering of measured parameters in HDMI/MHL mode. Free run standard corresponds to the last measured parameters. 1 = Disable buffering of measured parameters in HDMI/MHL mode. Free run standard determined by CP_VID_STD[5:0] and CP_V_FREQ[2:0].	
HDMI_CP_AUTOPARM_LOCKED			R
0xE0	00000000	This readback reports the lock status of the parameter buffering block in HDMI/MHL mode. 0 = Parameter buffering block is not locked to the synchronization signals from the HDMI/MHL core 1 = Parameter buffering block is locked to the synchronization signals from the HDMI/MHL core	
CH1_FL_FR_THRESHOLD[2:0]			R/W
0xF3	11010100	This readback indicates the threshold for the difference between the input video field length and the internally stored standard to enter and exit free run. 000 = Minimum difference to switch into free run is 36 lines. Maximum difference to switch out of free run is 31 lines. 001 = Minimum difference to switch into free run is 18 lines. Maximum difference to switch out of free run is 15 lines. 010 = Minimum difference to switch into free run is 10 lines. Maximum difference to switch out of free run is 7 lines. 011 = Minimum difference to switch into free run is 4 lines. Maximum difference to switch out of free run is 3 lines. 100 = Minimum difference to switch into free run is 51 lines. Maximum difference to switch out of free run is 46 lines. 101 = Minimum difference to switch into free run is 69 lines. Maximum difference to switch out of free run is 63 lines. 110 = Minimum difference to switch into free run is 134 lines. Maximum difference to switch out of free run is 127 lines. 111 = Minimum difference to switch into free run is 263 lines. Maximum difference to switch out of free run is 255 lines.	
CH1_F_RUN_THR[2:0]			R/W
0xF3	11010100	This control is used to select the free run threshold for Sync Channel 1. It determines the horizontal conditions under which free run mode is entered or left. The length of the incoming video line is measured based on the crystal clock and is compared to an internally stored parameter. The magnitude of the difference decides whether or not Sync Channel 1 enters free run mode. 000 = Minimum difference to switch into free run is 2. Maximum difference to switch out of free run is 1. 001 = Minimum difference to switch into free run is 256. Maximum difference to switch out of free run is 200. 010 = Minimum difference to switch into free run is 128. Maximum difference to switch out of free run is 112. 011 = Minimum difference to switch into free run is 64. Maximum difference to switch out of free run is 48. 100 = Minimum difference to switch into free run is 32. Maximum difference to switch out of free run is 24. 101 = Minimum difference to switch into free run is 16. Maximum difference to switch out of free run is 12. 110 = Minimum difference to switch into free run is 8. Maximum difference to switch out of free run is 6. 111 = Minimum difference to switch into free run is 4. Maximum difference to switch out of free run is 3.	
CSC_COEFF_SEL_RB[3:0]			R
0xF4	00000000	This readback displays the CP CSC conversion when configured in automatic mode. 0000 = CSC bypassed 0001 = YPbPr 601 to RGB 0011 = YPbPr 709 to RGB 0101 = RGB to YPbPr 601 0111 = RGB to YPbPr 709 1001 = YPbPr 709 to YPbPr 601 1010 = YPbPr 601 to YPbPr 709 1111 = CSC in manual mode xxxx = All other values	

Reg	Bits	Description	
CP_FREE_RUN			R
0xFF	0000<u>0</u>0000	This readback indicates the component processor free run status. 0 = CP not free running 1 = CP free running	

2.14 CEC MAP

Reg	Bits	Description	
CEC_TX_FRAME_HEADER[7:0]			R/W
0x00	00000000	This register is used to set the Tx header block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA0[7:0]			R/W
0x01	00000000	This register is used to set Data Byte 1 of the data block for the CEC message to be transmitted. The message includes opcodes and operands, total 15 bytes.	
CEC_TX_FRAME_DATA1[7:0]			R/W
0x02	00000000	This register is used to set Data Byte 2 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA2[7:0]			R/W
0x03	00000000	This register is used to set Data Byte 3 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA3[7:0]			R/W
0x04	00000000	This register is used to set Data Byte 4 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA4[7:0]			R/W
0x05	00000000	This register is used to set Data Byte 5 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA5[7:0]			R/W
0x06	00000000	This register is used to set Data Byte 6 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA6[7:0]			R/W
0x07	00000000	This register is used to set Data Byte 7 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA7[7:0]			R/W
0x08	00000000	This register is used to set Data Byte 8 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA8[7:0]			R/W
0x09	00000000	This register is used to set Data Byte 9 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA9[7:0]			R/W
0x0A	00000000	This register is used to set Data Byte 10 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA10[7:0]			R/W
0x0B	00000000	This register is used to set Data Byte 11 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA11[7:0]			R/W
0x0C	00000000	This register is used to set Data Byte 12 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA12[7:0]			R/W
0x0D	00000000	This register is used to set Data Byte 13 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA13[7:0]			R/W
0x0E	00000000	This register is used to set Data Byte 14 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_DATA14[7:0]			R/W
0x0F	00000000	This register is used to set Data Byte 15 of the data block for the CEC message to be transmitted.	
CEC_TX_FRAME_LENGTH[4:0]			R/W
0x10	00000000	This signal is used to specify the message size of the CEC message to be transmitted. This is the number of byte in the outgoing message, including the header. Allowable range: 0x00 to 0x10. xxxxx = Total number of bytes (including header byte) to be sent	

Reg	Bits	Description	
CEC_TX_ENABLE			R/W
0x11	0000000<u>0</u>	This bit enables the Tx section. When set to 1, it initiates the start of transmission of the message in the outgoing message buffer. When the message transmission is completed, this bit is automatically reset to 0. If it is manually set to 0 during a message transmission, it may terminate the transmission depending on what stage of the transmission process has been reached. If the message transmission is still in the signal free time stage, the message transmission is terminated. If data transmission has begun, then the transmission continues until the message is fully sent, or until an error condition occurs. 0 = Transmission mode disabled 1 = Transmission mode enabled and message transmission started	
CEC_TX_RETRY[2:0]			R/W
0x12	0<u>001</u>0011	The signal is used to specify the number of times the CEC Tx tries to retransmit the message if an error condition is encountered. Per the CEC specification, do not set this value greater than 5. 001 = Try to retransmit the message 1 time if an error occurs xxx = Try to retransmit the message xxx times if an error occurs	
CEC_RETRY_SFT[3:0]			R/W
0x12	0001<u>0011</u>	This signal is used to specify the signal free time of periods for retransmission retry. Set this parameter to a value equal to or greater than 3 and strictly less than 5.	
CEC_TX_SFT[3:0]			R/W
0x13	<u>0101</u>0111	This signal is used to specify the signal free time if the device is a new initiator. Set this parameter to a value equal to or greater than 5 and strictly less than 7.	
CEC_TX_SFT[3:0]			R/W
0x13	0101<u>0111</u>	This signal is used to specify the signal free time if the device transmits a next frame immediately after its previous frame. Set this parameter to a value equal to or greater than 7 and strictly less than 10.	
CEC_TX_LOWDRIVE_COUNTER[3:0]			R
0x14	<u>0000</u>0000	This signal is used to specify the number of times that the LOWDRIVE error condition was encountered while trying to send the current message. This register is reset to 0b0000 when CEC_TX_ENABLE is set to 1. 0000 = No error condition xxxx = The number of times the LOWDRIVE error condition was encountered	
CEC_TX_NACK_COUNTER[3:0]			R
0x14	0000<u>0000</u>	The signal is used to specify the number of times that the NACK error condition was encountered while trying to send the current message. This register is reset to 0b0000 when CEC_TX_ENABLE is set to 1. 0000 = No error condition xxxx = The number of times the NACK error condition was encountered	
CEC_BUF0_RX_FRAME_HEADER[7:0]			R
0x15	<u>00000000</u>	This register is used to readback byte 0 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA0[7:0]			R
0x16	<u>00000000</u>	This register is used to read back Byte 1 of the received CEC message in Frame Buffer 0. The message includes opcodes and operands, total = 15 bytes.	
CEC_BUF0_RX_FRAME_DATA1[7:0]			R
0x17	<u>00000000</u>	This register is used to read back Byte 2 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA2[7:0]			R
0x18	<u>00000000</u>	This register is used to read back Byte 3 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA3[7:0]			R
0x19	<u>00000000</u>	This register is used to read back Byte 4 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA4[7:0]			R
0x1A	<u>00000000</u>	This register is used to read back Byte 5 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA5[7:0]			R
0x1B	<u>00000000</u>	This register is used to read back Byte 6 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA6[7:0]			R
0x1C	<u>00000000</u>	This register is used to read back Byte 7 of the received CEC message in Frame Buffer 0.	

Reg	Bits	Description	
CEC_BUF0_RX_FRAME_DATA7[7:0]			R
0x1D	00000000	This register is used to read back Byte 8 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA8[7:0]			R
0x1E	00000000	This register is used to read back Byte 9 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA9[7:0]			R
0x1F	00000000	This register is used to read back Byte 10 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA10[7:0]			R
0x20	00000000	This register is used to read back Byte 11 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA11[7:0]			R
0x21	00000000	This register is used to read back Byte 12 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA12[7:0]			R
0x22	00000000	This register is used to read back Byte 13 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA13[7:0]			R
0x23	00000000	This register is used to read back Byte 14 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_DATA14[7:0]			R
0x24	00000000	This register is used to read back Byte 15 of the received CEC message in Frame Buffer 0.	
CEC_BUF0_RX_FRAME_LENGTH[4:0]			R
0x25	00000000	This signal is used to read back the message size of the CEC message received in frame buffer 0.	
CEC_LOGICAL_ADDRESS_MASK[2:0]			R/W
0x27	00010000	This signal is used to specify the logical address mask of the CEC logical devices, support up to 3 logical devices. When the bit is one, the related logical device is enabled, and the messages whose destination address is matched with the logical address. 001 = Use CEC_LOGICAL_ADDRESS0 for CEC controller 010 = Use CEC_LOGICAL_ADDRESS1 for CEC controller 100 = Use CEC_LOGICAL_ADDRESS2 for CEC controller	
CEC_ERROR_REPORT_MODE			R/W
0x27	00010000	This bit is used to specify the error reporting. 0 = Only report short bit period errors 1 = Report both short and long bit period errors	
CEC_ERROR_DET_MODE			R/W
0x27	00010000	This bit is used to specify the error detection. 0 = If any short bit period error, except for start bit, is detected, the CEC controller immediately drives the CEC line low for 3.6 ms 1 = If a short bit period is detected in the data block where the destination is the CEC section or a target CEC device, the CEC controller immediately drives the CEC line low for 3.6 ms	
CEC_FORCE_NACK			R/W
0x27	00010000	This bit is used to force NACK control setting. This bit forces the CEC controller to not acknowledge any received messages. 0 = ACK the relevant messages 1 = NACK all messages	
CEC_FORCE_IGNORE			R/W
0x27	00010000	This bit is used to force Ignore Control. This bit forces the CEC controller to ignore any directly addressed messages. Keep normal operation for the broadcast message. 0 = ACK the direct-addressed messages 1 = NACK all direct-addressed messages	
CEC_LOGICAL_ADDRESS1[3:0]			R/W
0x28	11111111	This signal is used to specify Logical Address 1. This address must be enabled by setting CEC_LOGICAL_ADDRESS_MASK[1] to 1. 1111 = Logical address set to this value by default xxxx = User specified logical address	

Reg	Bits	Description	
CEC_LOGICAL_ADDRESS0[3:0]			R/W
0x28	1111<u>1111</u>	This signal is used to specify Logical Address 0. This address must be enabled by setting CEC_LOGICAL_ADDRESS_MASK[0] to 1. 1111 – Logical address set to this value by default xxxx = User specified logical address	
CEC_LOGICAL_ADDRESS2[3:0]			R/W
0x29	0000<u>1111</u>	This signal is used to specify Logical Address 2. This address must be enabled by setting CEC_LOGICAL_ADDRESS_MASK[2] to 1. 1111 = Logical address set to this value by default xxxx = User specified logical address	
CEC_POWER_UP			R/W
0x2A	0011111<u>0</u>	This signal is used to set the power mode of the CEC controller. 0 = Power down the CEC module 1 = Power up the CEC module	
CEC_GLITCH_FILTER_CTRL[5:0]			R/W
0x2B	00<u>000111</u>	This signal is used to control the glitch filter. The CEC input signal is sampled by the input clock (XTAL clock). CEC_GLITCH_FILTER_CTRL specifies the minimum pulse width requirement in input clock cycles. Pulses of widths less than the minimum specified width are considered glitches and is removed. 000000 = Disable the glitch filter 000001 = Filter out pulses with width less than 1 clock cycle 000010 = Filter out pulses with width less than 2 clock cycles … 000111 = Filter out pulses with width less than 7 clock cycles … 111111 = Filter out pulses with width less than 63 clock cycles	
CEC_CLR_RX_RDY2			SC
0x2C	0000<u>0</u>000	This bit is used to control the clear signal for RX_RDY2. 0 = Retain the value of the interrupt RX_RDY2 1 = Clear out the interrupt RX_RDY2	
CEC_CLR_RX_RDY1			SC
0x2C	00000<u>0</u>00	This bit is used to control the clear signal for RX_RDY1. 0 = Retain the value of the interrupt RX_RDY1 1 = Clear out the interrupt RX_RDY1	
CEC_CLR_RX_RDY0			SC
0x2C	000000<u>0</u>0	This bit is used to control the clear signal for RX_RDY0. 0 = Retain the value of the interrupt RX_RDY0 1 = Clear out the interrupt RX_RDY0	
CEC_SOFT_RESET			SC
0x2C	0000000<u>0</u>	This bit is used to reset the CEC controller. 0 = Do not reset the CEC controller 1 = Reset the CEC controller	
CEC_BUF2_TIMESTAMP[1:0]			R
0x53	00<u>00</u>0000	This signal is used to read back the time stamp for the frame stored in Receiver Frame Buffer 2. This can be used to determine which frame is read next from the receiver frame buffers. 00 = Invalid timestamp, no frame is available in this frame buffer 01 = Of the frames currently buffered, this frame was the first to be received 10 = Of the frames currently buffered, this frame was the second to be received 11 = Of the frames currently buffered, this frame was the third to be received	
CEC_BUF1_TIMESTAMP[1:0]			R
0x53	0000<u>00</u>00	This signal is used to read back the time stamp for the frame stored in Receiver Frame Buffer 1. This can be used to determine which frame is read next from the receiver frame buffers. 00 = Invalid timestamp, no frame is available in this frame buffer 01 = Of the frames currently buffered, this frame was the first to be received 10 = Of the frames currently buffered, this frame was the second to be received 11 = Of the frames currently buffered, this frame was the third to be received	

Reg	Bits	Description	
CEC_BUF0_TIMESTAMP[1:0]			R
0x53	000000<u>00</u>	This signal is used to read back the time stamp for the frame stored in Receiver Frame Buffer 0. This can be used to determine which frame is read next from the receiver frame buffers. 00 = Invalid timestamp, no frame is available in this frame buffer 01 = Of the frames currently buffered, this frame was the first to be received 10 = Of the frames currently buffered, this frame was the second to be received 11 = Of the frames currently buffered, this frame was the third to be received	
CEC_BUF1_RX_FRAME_HEADER[7:0]			R
0x54	<u>00000000</u>	This register is used to read back Byte 0 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA0[7:0]			R
0x55	<u>00000000</u>	This register is used to read back Byte 1 of the received CEC message in Frame Buffer 1. The message includes opcodes and operands, total 15 bytes.	
CEC_BUF1_RX_FRAME_DATA1[7:0]			R
0x56	<u>00000000</u>	This register is used to read back Byte 2 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA2[7:0]			R
0x57	<u>00000000</u>	This register is used to read back Byte 3 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA3[7:0]			R
0x58	<u>00000000</u>	This register is used to read back Byte 4 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA4[7:0]			R
0x59	<u>00000000</u>	This register is used to read back Byte 5 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA5[7:0]			R
0x5A	<u>00000000</u>	This register is used to read back Byte 6 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA6[7:0]			R
0x5B	<u>00000000</u>	This register is used to read back Byte 7 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA7[7:0]			R
0x5C	<u>00000000</u>	This register is used to read back Byte 8 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA8[7:0]			R
0x5D	<u>00000000</u>	This register is used to read back Byte 9 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA9[7:0]			R
0x5E	<u>00000000</u>	This register is used to read back Byte 10 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA10[7:0]			R
0x5F	<u>00000000</u>	This register is used to read back Byte 11 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA11[7:0]			R
0x60	<u>00000000</u>	This register is used to read back Byte 12 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA12[7:0]			R
0x61	<u>00000000</u>	This register is used to read back Byte 13 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA13[7:0]			R
0x62	<u>00000000</u>	This register is used to read back Byte 14 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_DATA14[7:0]			R
0x63	<u>00000000</u>	This register is used to read back Byte 15 of the received CEC message in Frame Buffer 1.	
CEC_BUF1_RX_FRAME_LENGTH[4:0]			R
0x64	000<u>00000</u>	This signal is used to read back the message size of the CEC message received in Frame Buffer 1. xxxxx = The total number of bytes (including header byte) that were received into Buffer 1	

Reg	Bits	Description	
CEC_BUF2_RX_FRAME_HEADER[7:0]			R
0x65	00000000	This register is used to read back Byte 0 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA0[7:0]			R
0x66	00000000	This register is used to read back Byte 1 of the received CEC message in Frame Buffer 2. The message includes opcodes and operands, total 15 bytes.	
CEC_BUF2_RX_FRAME_DATA1[7:0]			R
0x67	00000000	This register is used to read back Byte 2 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA2[7:0]			R
0x68	00000000	This register is used to read back Byte 3 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA3[7:0]			R
0x69	00000000	This register is used to read back Byte 4 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA4[7:0]			R
0x6A	00000000	This register is used to read back Byte 5 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA5[7:0]			R
0x6B	00000000	This register is used to read back Byte 6 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA6[7:0]			R
0x6C	00000000	This register is used to read back Byte 7 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA7[7:0]			R
0x6D	00000000	This register is used to read back Byte 8 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA8[7:0]			R
0x6E	00000000	This register is used to read back Byte 9 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA9[7:0]			R
0x6F	00000000	This register is used to read back Byte 10 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA10[7:0]			R
0x70	00000000	This register is used to read back Byte 11 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA11[7:0]			R
0x71	00000000	This register is used to read back Byte 12 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA12[7:0]			R
0x72	00000000	This register is used to read back Byte 13 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA13[7:0]			R
0x73	00000000	This register is used to read back Byte 14 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_DATA14[7:0]			R
0x74	00000000	This register is used to read back Byte 15 of the received CEC message in Frame Buffer 2.	
CEC_BUF2_RX_FRAME_LENGTH[4:0]			R
0x75	00000000	This signal is used to read back the message size of the CEC message received in Frame Buffer 2. xxxxx = The total number of bytes (including header byte) that were received into Buffer 2	
CEC_RX_RDY2			R
0x76	0000000	This bit is used to read back the interrupt for frame presence in Buffer 2. 0 = No CEC frame available in Buffer 2 1 = A CEC frame is available in Buffer 2	

Reg	Bits	Description	
CEC_RX_RDY1			R
0x76	0000000<u>0</u>	This bit is used to read back the interrupt for frame presence in Buffer 1. 0 = No CEC frame available in Buffer 1 1 = A CEC frame is available in Buffer 1	
CEC_RX_RDY0			R
0x76	0000000<u>0</u>	This bit is used to read back the interrupt for frame presence in Buffer 0. 0 = No CEC frame available in Buffer 0 1 = A CEC frame is available in Buffer 0	
CEC_USE_ALL_BUFS			R/W
0x77	0000000<u>0</u>	This bit is used to select whether the new frames is received in all three buffers or only one buffer. 0 = Use only Buffer 0 to store CEC frames (Legacy mode) 1 = Use all three buffers to stores the CEC frames (Nonlegacy mode)	
CEC_WAKE_OPCODE0[7:0]			R/W
0x78	<u>01101101</u>	This register is used to set CEC Wake Opcode 0. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 01101101 = Power on xxxxxxxx = User specified opcode to respond to	
CEC_WAKE_OPCODE1[7:0]			R/W
0x79	<u>10001111</u>	This register is used to set CEC Wake Opcode 1. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 10001111 = Give power status xxxxxxxx = User specified OPCODE to respond to	
CEC_WAKE_OPCODE2[7:0]			R/W
0x7A	<u>10000010</u>	This register is used to set CEC Wake Opcode 2. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 10000010 = Active source xxxxxxxx = User specified OPCODE to respond to	
CEC_WAKE_OPCODE3[7:0]			R/W
0x7B	<u>00000100</u>	This register is used to set CEC Wake Opcode 3. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 00000100 = Image view on xxxxxxxx = User specified opcode to respond to	
CEC_WAKE_OPCODE4[7:0]			R/W
0x7C	<u>00001101</u>	This register is used to set CEC Wake Opcode 4. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 00001101 = Text view on xxxxxxxx = User specified opcode to respond to	
CEC_WAKE_OPCODE5[7:0]			R/W
0x7D	<u>01110000</u>	This register is used to set CEC Wake Opcode 5. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 01110000 = System audio mode request xxxxxxxx = User specified opcode to respond to	
CEC_WAKE_OPCODE6[7:0]			R/W
0x7E	<u>01000010</u>	This register is used to set CEC Wake Opcode 6. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 01000010 = Deck control xxxxxxxx = User specified opcode to respond to	

Reg	Bits	Description	
CEC_WAKE_OPCODE7[7:0]			R/W
0x7F	01000001	This register is used to set CEC Wake Opcode 7. To make use of the wake opcode interrupts when the device is powered down, the CEC interrupts must first be unmasked in the IO map. This value can be set to any CEC opcode that requires a response. On receipt of this opcode the Rx generates an interrupt that can be used to alert the system that a CEC opcode of interest has been received and requires a response. 01000001 = Play xxxxxxxx = User specified opcode to respond to	

2.15 CSI-TXA MAP

Reg	Bits	Description	
CSITX_PWRDN			R/W
0x00	10000001	The register disables or enables CSI Tx. 0 = CSI Tx On 1 = CSI Tx Off	
EN_AUTOCALC_DPHY_PARAMS			R/W
0x00	1000**0**001	This bit can be used to enable the automatic computation of the DPHY timing parameters 0 = Do not use automatically computed DPHY timing parameters 1 = Enable and use automatically computed DPHY timing parameters	
NUM_LANES[2:0]			R/W
0x00	10000**001**	This control can be used to program the number of active lanes of the CSI Tx. 0 = Not valid 1 = 1-Lane configuration 2 = 2-Lane configuration 3 = Not valid 4 = 4-Lane configuration 5 to 7 = Not valid	
VC_REF[1:0]			R/W
0x0D	**00**000000	Programmable Virtual Channel number.	
INTERPRET_FS_AS_LS			R/W
0x1E	**0**0000000	This control can be used to enable the non line start (LS)/line end (LE) packet mode. This mode can be useful in cases where the horizontal blanking is too short to transmit LS and LE packets. 0 = Transmit both LS/LE and frame start (FS)/frame end (FE) packets 1 = Transmit only FS/FE packets; do not transmit LS/LE packets	
FRAMENUMBER_INTERLACED			R/W
0x1F	**0**0000000	This control sets association of frame number in the FS and FE packets with the F bit in the EAV/SAV codes. This control is only valid for interlaced video modes. 0 = Frame number is 2 for fields with F Bit 0, and 1 for fields with F Bit 1 1 = Frame number is 1 for fields with F Bit 0, and 2 for fields with F Bit 1	
LINENUMBER_INCR_INTERLACED			R/W
0x20	**0**0000000	The line numbers in the LS and LE packets for interlaced video must increment by more than 1. This control gives the user the option to use an incremental step of 2, or an incremental step of 3. This control is only for interlaced video. 0 = Increment line numbers by 2 1 = Increment line numbers by 3	
LINENUMBER1_F1_INTERLACED[7:0]			R/W
0x21	**00000011**	Arbitrary nonzero start value of the logical line number for the first line after FS packet in an even field (F bit = 1). This I²C register is only for interlaced video. 3 = Default value	
LINENUMBER1_F2_INTERLACED[7:0]			R/W
0x22	**00000010**	Arbitrary nonzero start value of the logical line number for the first line after FS packet in an odd field (F bit = 0). This I²C register is only for interlaced video. 2 = Default value	
ESC_MODE_EN_CLK			R/W
0x26	000**0**0000	Escape Mode Enable for Clock Lane 1 = Escape Mode Enabled in CSI Tx for Clock Lane 0 = Escape Mode Disabled in CSI Tx for Clock Lane	
ESC_XSHUTDOWN_CLK			R/W
0x26	000**0**0000	External Sleep Command in Escape Mode for Clock Lane 1 = Disable Sleep mode, Clock Lane 0 = Enable Sleep mode, Clock Lane	
ESC_MODE_EN_D3			R/W
0x27	**0**0000000	Escape Mode Enable for Data Lane3 1 = Escape Mode Enabled in CSI Tx for Data Lane3 0 = Escape Mode Disabled in CSI Tx for Data Lane3	

Reg	Bits	Description	
ESC_XSHUTDOWN_D3			R/W
0x27	0<u>0</u>000000	External Sleep Command in Escape Mode for Data Lane3 1 = Disable Sleep mode, Data Lane3 0 = Enable Sleep mode, Data Lane3	
ESC_MODE_EN_D2			R/W
0x27	00<u>0</u>00000	Escape Mode Enable for Data Lane2 1 = Escape Mode Enabled in CSI Tx for Data Lane2 0 = Escape Mode Disabled in CSI Tx for Data Lane2	
ESC_XSHUTDOWN_D2			R/W
0x27	000<u>0</u>0000	External Sleep Command in Escape Mode for Data Lane2 1 = Disable Sleep mode, Data Lane2 0 = Enable Sleep mode, Data Lane2	
ESC_MODE_EN_D1			R/W
0x27	0000<u>0</u>000	Escape Mode Enable for Data Lane1 1 = Escape Mode Enabled in CSI Tx for Data Lane1 0 = Escape Mode Disabled in CSI Tx for Data Lane1	
ESC_XSHUTDOWN_D1			R/W
0x27	00000<u>0</u>00	External Sleep Command in Escape Mode for Data Lane1 1 = Disable Sleep mode, Data Lane1 0 = Enable Sleep mode, Data Lane1	
ESC_MODE_EN_D0			R/W
0x27	000000<u>0</u>0	Escape Mode Enable for Data Lane0 1 = Escape Mode Enabled in CSI Tx for Data Lane0 0 = Escape Mode Disabled in CSI Tx for Data Lane0	
ESC_XSHUTDOWN_D0			R/W
0x27	0000000<u>0</u>	External Sleep Command in Escape Mode for Data Lane0 1 = Disable Sleep mode, Data Lane0 0 = Enable Sleep mode, Data Lane0	
ESC_BYTE[7:0]			R/W
0x2B	<u>00011110</u>	Escape Mode Command to be sent 00011110 = ULPS Entry Command	
F_BIT_INV_POL			R/W
0x70	0001<u>0</u>000	This control can be used to invert the polarity of the F bit. 0 = Do not invert polarity of F bit 1 = Invert polarity of F bit	
EN_MAN_FMT			R/W
0x7E	<u>0</u>0000000	This control can be used to enable the MAN_FMT control, which allows the manual setting of the format. 0 = Format set automatically 1 = Format set manually by the MAN_FMT control	
MAN_FMT[2:0]			R/W
0x7E	0<u>000</u>0000	This control can be used to manually set the format. The EN_MAN_FMT control must be set to 0b1 for this control to be available. 000 = RGB 4:4:4 format (available for 18-bit or 24-bit modes only) 001 = RGB 4:4:4 format (available for 16-bit mode only) 010 = YUV 4:2:2 format (available for 8-bit or 10-bit modes) Other = Reserved	
EN_MAN_BPP			R/W
0x7E	0000<u>0</u>000	This control can be used to enable the MAN_BPP control, which allows the manual setting of the number of bits per pixel. 0 = Number of bits per pixel set automatically 1 = Number of bits per pixel set manually by the MAN_BPP control	

Reg	Bits	Description	
MAN_BPP[2:0]			R/W
0x7E	00000000	This control can be used to manually set the number of bits per pixel. The EN_MAN_BPP control must be set to 0b1 for this control to be available. 000 = 16 bits per pixel 001 = 18 bits per pixel 010 = 20 bits per pixel 011 = 24 bits per pixel Other = Reserved	
LANE1_NUM[2:0]			R/W
0x9C	00010000	Cross-pointing control for Data Lane1 0 = Send out data corresponding to Lane0 1 = Send out data corresponding to Lane1 2 = Send out data corresponding to Lane2 3 = Send out data corresponding to Lane3 4 = Send out data corresponding to clock lane	
LANE0_NUM[2:0]			R/W
0x9C	00010000	Cross-pointing control for Data lane0 0 = Send out data corresponding to Lane0 1 = Send out data corresponding to Lane1 2 = Send out data corresponding to Lane2 3 = Send out data corresponding to Lane3 4 = Send out data corresponding to clock lane	
LANE3_NUM[2:0]			R/W
0x9D	00110010	Cross-pointing control for Data Lane3 0 = Send out data corresponding to Lane0 1 = Send out data corresponding to Lane1 2 = Send out data corresponding to Lane2 3 = Send out data corresponding to Lane3 4 = Send out data corresponding to clock lane	
LANE2_NUM[2:0]			R/W
0x9D	00110010	Cross-pointing control for Data Lane2 0 = Send out data corresponding to Lane0 1 = Send out data corresponding to Lane1 2 = Send out data corresponding to Lane2 3 = Send out data corresponding to Lane3 4 = Send out data corresponding to clock lane	
LANECLK_NUM[2:0]			R/W
0x9E	00000100	Cross-pointing control for clock lane 0 = Send out data corresponding to Lane0 1 = Send out data corresponding to Lane1 2 = Send out data corresponding to Lane2 3 = Send out data corresponding to Lane3 4 = Send out data corresponding to clock lane	
CLKLN_IS_CLOCK_LANE			R/W
0xC1	00111011	Configures this lane to be the clock lane or not. Only one lane can be the clock lane. 0 = This lane is not the clock lane 1 = This lane is the clock lane	
CLKLN_ZERO_CLK_LANE			R/W
0xC1	00111011	This control allows the output of this lane (if configured as a clock lane) to be glitchlessly forced to zero, or it allows it to toggle. 0 = The clock output is allowed to toggle 1 = The clock output is forced to Logic 0	
CLKLN_LANE_IS_MASTER			R/W
0xC1	00111011	This control defines which lanes generate the master synchronization signal for the MIPI Tx. Only the clock lane and the data lane corresponding to Lane 0 can be set as the master lanes. 0 = This data lane is a synchronization slave lane 1 = This data lane is the synchronization master lane	

Reg	Bits	Description	
D0LN_IS_CLOCK_LANE			R/W
0xC4	0000**0**1010	Configures this lane to be the clock lane or not. Only one lane can be the clock lane. 0 = This lane is not the clock lane 1 = This lane is the clock lane	
D0LN_ZERO_CLK_LANE			R/W
0xC4	0000**0**1010	This control allows the output of this lane (if configured as a clock lane) to be glitchlessly forced to zero, or it allows it to toggle. 0 = The clock output is allowed to toggle 1 = The clock output is forced to Logic 0	
D0LN_LANE_IS_MASTER			R/W
0xC4	0000**1**010	This control defines which lanes generate the master synchronization signal for the MIPI Tx. Only the clock lane and the data lane corresponding to Lane 0 can be set as the master lanes. 0 = This data lane is a synchronization slave lane 1 = This data lane is the synchronization master lane	
D1LN_IS_CLOCK_LANE			R/W
0xC7	0000**0**0010	Configures this lane to be the clock lane or not. Only one lane can be the clock lane. 0 = This lane is not the clock lane 1 = This lane is the clock lane	
D1LN_ZERO_CLK_LANE			R/W
0xC7	0000**0**0010	This control allows the output of this lane (if configured as a clock lane) to be glitchlessly forced to zero, or it allows it to toggle. 0 = The clock output is allowed to toggle 1 = The clock output is forced to Logic 0	
D1LN_LANE_IS_MASTER			R/W
0xC7	0000**0**010	This control defines which lanes generate the master synchronization signal for the MIPI Tx. Only the clock lane and the data lane corresponding to Lane 0 can be set as the master lanes. 0 = This data lane is a synchronization slave lane 1 = This data lane is the synchronization master lane	
D2LN_IS_CLOCK_LANE			R/W
0xCA	0000**0**0010	Configures this lane to be the clock lane or not. Only one lane can be the clock lane. 0 = This lane is not the clock lane 1 = This lane is the clock lane	
D2LN_ZERO_CLK_LANE			R/W
0xCA	0000**0**0010	This control allows the output of this lane (if configured as a clock lane) to be glitchlessly forced to zero, or it allows it to toggle. 0 = The clock output is allowed to toggle 1 = The clock output is forced to Logic 0	
D2LN_LANE_IS_MASTER			R/W
0xCA	0000**0**010	This control defines which lanes generate the master synchronization signal for the MIPI Tx. Only the clock lane and the data lane corresponding to Lane 0 can be set as the master lanes. 0 = This data lane is a synchronization slave lane 1 = This data lane is the synchronization master lane	
D3LN_IS_CLOCK_LANE			R/W
0xCD	0000**0**0010	Configures this lane to be the clock lane or not. Only one lane can be the clock lane. 0 = This lane is not the clock lane 1 = This lane is the clock lane	
D3LN_ZERO_CLK_LANE			R/W
0xCD	0000**0**0010	This control allows the output of this lane (if configured as a clock lane) to be glitchlessly forced to zero, or it allows it to toggle. 0 = The clock output is allowed to toggle 1 = The clock output is forced to Logic 0	
D3LN_LANE_IS_MASTER			R/W
0xCD	0000**0**010	This control defines which lanes generate the master synchronization signal for the MIPI Tx. Only the clock lane and the data lane corresponding to Lane 0 can be set as the master lanes. 0 = This data lane is a synchronization slave lane 1 = This data lane is the synchronization master lane	

Reg	Bits	Description	
MIPI_PLL_LOCK_FLAG			R
0xDA	00000000	The register allows checking PLL lock status. Note: MIPI_PLL_CLK_DET must be set first. 0 = PLL unlocked 1 = PLL locked	
MIPI_PLL_CLK_DET			R/W
0xDA	00000000	The register allows enabling lock detection circuitry for the MIPI PLL. 0 = lock detect disable 1 = lock detect enable	
MIPI_PLL_EN			R/W
0xDA	00000000	The register enables/disables MIPI PLL. 0 = Power down MIPI PLL 1 = Power up MIPI PLL	
DPHY_PWDN			R/W
0xF0	00000001	This control can be used to power down the DPHY. 0 = DPHY is not powered down 1 = DPHY is powered down	

2.16 CSI-TXB MAP

Reg	Bits	Description	
CSITX_PWRDN			R/W
0x00	10000001	The register disables or enables CSI TX. 0 = CSI Tx On 1 = CSI Tx Off	
EN_AUTOCALC_DPHY_PARAMS			R/W
0x00	10000001	This bit can be used to enable the automatic computation of the DPHY timing parameters 0 = Do not use automatically computed DPHY timing parameters 1 = Enable and use automatically computed DPHY timing parameters	
VC_REF[1:0]			R/W
0x0D	00000000	Programmable Virtual Channel number.	
INTERPRET_FS_AS_LS			R/W
0x1E	00000000	This control can be used to enable the non LS/LE packet mode. This mode can be useful in cases where the horizontal blanking is too short to transmit LS and LE packets. 0 = Transmit both LS/LE and FS/FE packets 1 = Transmit only FS/FE packets, do not transmit LS/LE packets	
FRAMENUMBER_INTERLACED			R/W
0x1F	00000000	This control sets association of frame number in the FS and FE packets with the F bit in the EAV/SAV codes. This control is only valid for interlaced video modes. 0 = Frame number is 2 for fields with F Bit 0, and 1 for fields with F Bit 1 1 = Frame number is 1 for fields with F Bit 0, and 2 for fields with F Bit 1	
LINENUMBER_INCR_INTERLACED			R/W
0x20	00000000	The line numbers in the Line Start (LS) and Line End (LE) packets for interlaced video must increment by more than 1. This control gives the user the option to use an incremental step of 2, or an incremental step of 3. This control is only for interlaced video. 0 = Increment line numbers by 2 1 = Increment line numbers by 3	
LINENUMBER1_F1_INTERLACED[7:0]			R/W
0x21	00000011	Arbitrary nonzero start value of the logical line number for the first line after FS packet in an even field (F bit = 1). This I^2C register is only for interlaced video. 3 = Default value	
LINENUMBER1_F2_INTERLACED[7:0]			R/W
0x22	00000010	Arbitrary nonzero start value of the logical line number for the first line after FS packet in an odd field (F bit = 0). This I^2C register is only for interlaced video. 2 = Default value	
ESC_MODE_EN_CLK			R/W
0x26	00000000	Escape Mode Enable for Clock Lane 1 = Escape Mode Enabled in CSI Tx for Clock Lane 0 = Escape Mode Disabled in CSI Tx for Clock Lane	
ESC_XSHUTDOWN_CLK			R/W
0x26	00000000	External Sleep Command in Escape Mode for Clock Lane 1 = Disable Sleep mode, Clock Lane 0 = Enable Sleep mode, Clock Lane	
ESC_MODE_EN_D3			R/W
0x27	00000000	Escape Mode Enable for Data Lane3 1 = Escape Mode Enabled in CSI Tx for Data Lane3 0 = Escape Mode Disabled in CSI Tx for Data Lane3	
ESC_XSHUTDOWN_D3			R/W
0x27	00000000	External Sleep Command in Escape Mode for Data Lane3 1 = Disable Sleep mode, Data Lane3 0 = Enable Sleep mode, Data Lane3	
ESC_MODE_EN_D2			R/W
0x27	00000000	Escape Mode Enable for Data Lane2 1 = Escape Mode Enabled in CSI Tx for Data Lane2 0 = Escape Mode Disabled in CSI Tx for Data Lane2	

Reg	Bits	Description	
ESC_XSHUTDOWN_D2			R/W
0x27	0000**0**0000	External Sleep Command in Escape Mode for Data Lane2 1 = Disable Sleep mode, Data Lane2 0 = Enable Sleep mode, Data Lane2	
ESC_MODE_EN_D1			R/W
0x27	0000**0**000	Escape Mode Enable for Data Lane1 1 = Escape Mode Enabled in CSI Tx for Data Lane1 0 = Escape Mode Disabled in CSI Tx for Data Lane1	
ESC_XSHUTDOWN_D1			R/W
0x27	00000**0**00	External Sleep Command in Escape Mode for Data Lane1 1 = Disable Sleep mode, Data Lane1 0 = Enable Sleep mode, Data Lane1	
ESC_MODE_EN_D0			R/W
0x27	000000**0**0	Escape Mode Enable for Data Lane0 1 = Escape Mode Enabled in CSI Tx for Data Lane0 0 = Escape Mode Disabled in CSI Tx for Data Lane0	
ESC_XSHUTDOWN_D0			R/W
0x27	0000000**0**	External Sleep Command in Escape Mode for Data Lane0 1 = Disable Sleep mode, Data Lane0 0 = Enable Sleep mode, Data Lane0	
ESC_BYTE[7:0]			R/W
0x2B	**00011110**	Escape Mode Command to be sent 00011110 = ULPS Entry Command	
F_BIT_INV_POL			R/W
0x70	0001**0**000	This control can be used to invert the polarity of the F bit. 0 = Do not invert polarity of F bit 1 = Invert polarity of F bit	
MIPI_PLL_LOCK_FLAG			R
0xDA	00000**0**00	The register allows checking PLL lock status. Note: MIPI_PLL_CLK_DET must be set first. 0 = PLL unlocked 1 = PLL locked	
MIPI_PLL_CLK_DET			R/W
0xDA	000000**0**0	The register allows enabling lock detection circuitry for the MIPI PLL. 0 = lock detect disable 1 = lock detect enable	
MIPI_PLL_EN			R/W
0xDA	0000000**0**	The register enables/disables MIPI PLL. 0 = Power down MIPI PLL 1 = Power up MIPI PLL	
DPHY_PWDN			R/W
0xF0	0000000**1**	This control can be used to power down the DPHY. 0 = DPHY is not powered down 1 = DPHY is powered down	

3 INDEX

NOTES

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

UG12558-0-12/14(0)