Engineer-to-Engineer Note EE-379

Technical notes on using Analog Devices products and development tools
Visit our Web resources http://www.analog.com/ee-notes and http://www.analog.com/processors or e-mail processor.support@analog.com or processor.tools.support@analog.com for technical support.

ADSP-214xx vs ADSP-SC58x/ADSP-2158x - Peripheral Considerations

Contributed by Sachin-V Kumar

Rev 1 – June 12, 2015

Introduction

The ADSP-SC58x/ADSP-2158x SHARC® processors have a rich set of peripherals to support next-generation embedded applications. The processor infrastructure and peripheral designs are improved significantly to meet growing industry demands. This EE-Note is intended to assist software/hardware designers migrating from the ADSP-214xx family of processors to the ADSP-SC58x/ADSP-2158x processor family (hereafter referred to as ADSP-SC58x).

The objective of this EE-note is to familiarize the readers with key enhancements and design changes among I/O peripherals common to both ADSP-214xx and ADSP-SC58x SHARC processors.

For complete information describing use and programming of the peripherals described herein, please refer to the respective product *Hardware Reference Manuals*[1][2] and *Processor Datasheets*[3][4][5][6]. This document should not be considered as a replacement to this literature.

This EE-note describes important architectural changes to the following blocks/peripherals:

- Clock Generation/Distribution
- Peripheral Interrupt Handling
- Direct Memory Access (DMA)
- Digital Peripheral Interface (DPI) and PORT
- Core Flag Pins
- Digital Audio Interface (DAI)
- Serial Port (SPORT)
- SPDIF-RX
- SPI
- UART
- Watchdog Timer
- Peripheral Timer
- Accelerators

Clock Generation/Distribution

ADSP-SC58x processors have greater control and flexibility over generation and distribution of various on-chip clocks. They feature two Clock Generation Units (CGUs) and one Clock Distribution Unit (CDU). The CGU generates various on-chip clocks from a clock input source. Each CGU can have its own clock input source. The CDU allows routing any of the on-chip clocks generated by the CGU to cores, memories, and peripherals. The important differences in the new architecture are:

- ADSP-214xx processors have hardware pins (CLK_CFGx) which determine the PLL multipliers and dividers upon hardware reset. ADSP-SC58x processors have no such hardware pins; rather, the default on-chip clocks depend on the register reset values of CGU registers.
- The peripheral clock (PCLK) in ADSP-214xx processors is a fixed value (CCLK/2). All the peripherals except for the SDRAM controller use PCLK for their internal operation. On

ADSP-SC58x processors, the CGU generates PLLCLK, which is used to derive various on-chip clocks like CCLK, DCLK, SYSCLK, SCLK0, and SCLK1. Unique dividers are available for each of these clocks. Most of the peripherals source their clocks from CGU0's SCLK0 or SCLK1. Some of the high-speed peripherals (e.g., SPDIF, Gigabit Ethernet, PCIe, CAN, DDR, Link Port, and SDIO) have additional configurable options available in the CDU, which provides the ability to choose an on-chip clock as the internal clock for these peripheral. This provide greater flexibility in generating necessary clocks for every peripheral.

- On the ADSP-214xx processors, peripheral clocks to individual peripheral modules can be shut off using the PMCTL1 register. With the ADSP-SC58x processors, this feature is moved to the Dynamic Power Management (DPM) module, controlled via the DPM_PER_DIS register. The CGU also supports gating of core clocks.
- The CGU on the ADSP-SC58x processors allows routing a divided version of any of the on-chip clocks to a processor pin (SYS_CLKOUT), providing a test point to capture clock waveforms during debugging.

Peripheral Interrupt Handling

An ADSP-214xx processor has its DAI System Interrupt Controller (SIC) and DPI SIC connected to a Core Interrupt Controller (CIC). On ADSP-SC58x processors, peripheral interrupts are routed to the cores using the System Event Controller (SEC) on the SHARC cores and the Generic Interrupt Controller (GIC) for the ARM core. All the non-DAI peripheral interrupts and SPORT interrupts are routed through the SEC to the CIC and then to the SHARC cores. DAI peripherals (except the SPORT) still use the DAI SIC, so DAI interrupts are routed through the DAI SIC to the SEC, then to the SHARC core via the CIC.

The SEC manages the enabling, prioritization, and routing of events to the SHARC cores. The SEC also routes fault sources to the integrated Fault Management Unit (FMU). When two or more interrupts occur at a time, the highest priority peripheral interrupt will be forwarded to the Core Interrupt Controllers (CIC). From the CIC perspective, all the peripheral interrupts occur at the same core priority level. As a result, the CIC must and does support self-nesting of SEC interrupts, which means that nesting of interrupts is the default behavior for all peripheral sources handled by the SEC.

The programmable interrupt priority feature of CIC events is not available in ADSP-SC58x processors.

Some peripherals interrupts (like FIR and IIR) are edge-sensitive. It is necessary to program the SEC for the same for proper operation.

Direct Memory Access (DMA)

On ADSP-SC58x processors, all DMA channels are assigned to external peripherals, with most having a dedicated DMA channel (or pair of channels). Some DMA channels perform memory-to-memory transfers (MDMA), while the rest perform transfers between memory and a peripheral (PDMA). The new DMA controller supports several operating modes like stop mode, autobuffer mode, and descriptor mode. Stop mode configures DMA for a single-shot transfer. In autobuffer mode, DMA performs repeated transfers to/from the same transmit/receive buffer. Descriptor mode is similar to the Chain Pointer Mode (TCB) on ADSP-214xx processors, but descriptor mode in the ADSP-SC58x processor is much more flexible, as it provides configurable options to set the size for each descriptors (chain pointers/TCB). This will be useful in reducing unnecessary bandwidth overhead while fetching descriptors.

ADSP-SC58x processors feature a System Protection Unit (SPU), which restricts peripheral/DMA access to various system memories. It is necessary to configure the SPU for DMAs to provide read/write access to DMAs before enabling DMA.

If a DMA channel needs to access either of the SHARC core's L1 memory for a data transfer or a descriptor fetch, then the DMA engine must be provided with the system address (L1 memory multiprocessor equivalent) rather than the L1 memory private address.

The DMA engines on ADSP-214xx processors are within the peripheral modules. Hence, configuration and associated programming models are a little different among the various modules. On ADSP-SC58x processors, most of the peripheral DMA channels are grouped under the DMAC controller (though a few peripherals have their own DMA channels), which brings uniformity to the design and programming model.

Below is the pseudo-code for enabling and disabling a peripheral DMA channel on the ADSP-SC58x processors:

```
/* pseudo-code for enabling DMA in
ADSP-SC58x processors */

Configure_DMA_Descriptors();

Enable_SPU_for_DMA ();

Configure_DMA ();

Configure_peripheral ();

Enable_DMA ();

Enable_peripheral ();


/* pseudo-code for disabling ADSP-SC58x
processors */

Disable_DMA ();

Disable_peripheral ();

Disable_SPU_for_DMA (); /* optional */
```

The EMDMA channels on the ADSP-SC58x processors are similar to the EPDMA channels of the ADSP-214xx processors. EMDMA channels support all the features supported by the EPDMA channels.

Digital Peripheral Interface (DPI) and PORT

The Digital Peripheral Interface (DPI) on the ADSP-214xx processors has been replaced by the General-Purpose Port module (PORT) on the ADSP-SC58x processors. All the DPI peripheral signals like TIMER, SPI, TWI, and PWM can be routed to any of the DPI pins. This flexibility is not provided by the PORT module. In the PORT module, a pin multiplexing table dictates the pin functionality. Every PORT pin can be individually configured either as a GPIO pin or as a peripheral pin. When configured for peripheral functionality, each pin is then further controlled by a pin multiplexing scheme such that numerous peripherals can share the same pin. Each pin can be configured as any individual peripheral's pin, depending on the multiplexing scheme. Refer to the processor datasheet for the pin multiplexing table.

As peripheral signals are shared among PORT pins using a defined multiplexing scheme, it is important to consult the pin multiplexing table to understand the various combinations of GPIO and peripherals that can be supported by a given processor.

Core Flag Pins

On ADSP-214xx processors, the FLAG0-3 pins are dedicated pins, and the FLAG15-4 pins are available via the DPI. On ADSP-SC58x processors, the FLAG15-4 pins do not exist. Each SHARC core has FLAG0-3 pins, which are available via the PORT module.

Digital Audio Interface (DAI)

ADSP-SC58x processors feature two DAIs, each with eight half-SPORTs (four full SPORTs), two Precision Clock Generators (PCGs), four ASRCs, one SPDIF TX/RX pair. Most of the peripherals and pin signals in each DAI cannot be connected to peripherals/pins allocated to the other DAI. The DAIs do allow for sharing of a few signals between them in order to support synchronization of peripherals across DAIs. These signals are referred to as cross-mode connections:

- PCG's clock/FS outputs
- two DAI pin buffers (ideal for sharing Clock and Frame Sync between DAIs)
- SRC3 and SRC7 data lines

On ADSP-214xx processors, there are only four ASRCs; therefore, in TDM mode, only eight channels are supported through daisy chaining. On ADSP-SC58x processors, there are eight ASRCs, and clock, frame sync, and SRC data line sharing allows daisy chaining of eight ASRCs, which allows for support up to 16 TDM channels.

Serial Port (SPORT)

The ADSP-SC58x processor SPORT naming convention differs slightly from that of the ADSP-214xx processors. On ADSP-SC58x processors each SPORT top module is assumed to be comprised of two half-SPORTs called half-SPORT A and half-SPORT B. Each half-SPORT is equivalent to a full SPORT on ADSP-214xx processors.

Each half-SPORT can be independently configured as a transmitter or a receiver and has two data lines (primary and secondary). DMAs are outside the SPORT module, and each half-SPORT has a dedicated DMA channel. The key differences between SPORT designs are:

- The minimum PCLK:SPORT_CLK ratio for the ADSP-214xx is 4:1, whereas it is 1:1 in the ADSP-SC58x SPORT design. This gives better granularity for clock generation.
- The ADSP-SC58x SPORT added support for Right-Justified (RJ) mode (not supported on ADSP-214xx processors).
- The ADSP-214xx SPORT clock and frame sync signals must be in the same direction (both signals are either outputs or inputs) in Stereo modes. The ADSP-SC58x SPORT design removes this limitation.
- The ADSP-214xx SPORT supports only MSB first mode in stereo modes. The ADSP-SC58x design removes this limitation.
- The ADSP-214xx SPORT supports companding only on the primary data channel, and compression can only be performed by even-numbered SPORTs while expansion can only be performed by odd-numbered SPORTs. The ADSP-SC58x design supports companding on both the primary and secondary channels of each half-SPORT, and compression/expansion capability is decided by the SPORT's direction (transmit or receive, respectively).
- The ADSP-SC58x SPORT design supports word lengths of 4-32 in DSP serial mode and 5-32 in other modes, whereas the ADSP-214xx SPORT supports 8-32 bit word lengths in stereo mode and 3-32 bits in other modes.
- The ADSP-SC58x SPORT's multichannel mode features a Window Offset that allows choosing 128 continuous channels among a much larger 1024-channel window.
- The ADSP-SC58x SPORT design has a SPMUX feature which allows internal sharing of clock and frame sync signals between two half-SPORTs of a given full SPORT pairing.
- The ADSP-SC58x SPORT design also has a gated clock feature. In this mode, the clock will only be generated when data is valid. This feature is useful when interfacing to ADCs having this bit clock requirement.

SPDIF-RX

The ADSP-SC58x processor has a newly designed SPDIF receiver, and jitter frequency response is improved over the entire frequency range of operation. The key differences are:

- The ADSP-SC58x SPDIF receiver supports sampling rates from 24-96 kHz.
- The default state of the ADSP-SC58x SPDIF-RX is disabled, whereas it is enabled by default in ADSP-214xx processors.

- The ADSP-SC58x SPDIF-RX can detect word length from the channel status bits of the biphase stream, as specified by IEC 60958-3.
- The ADSP-SC58x SPDIF receiver can detect compression type of the digital audio stream, as specified by the burst info table included in IEC 61937-2.
- The ADSP-SC58x SPDIF receiver can detect whether the compression type is DTS or AC3.
- With the ADSP-214xx SPDIF, the action to be taken on biphase error is configurable. This feature is not supported in the ADSP-SC58x design.
- The ADSP-SC58x SPDIF does not include hardware support for single-channel double-frequency mode. It has to be implemented in software.
- Support for generating an interrupt on a CRC error, emphasized audio parity error, and no audio error is not available in ADSP-SC58x processors.
- The ADSP-SC58x SPDIF receiver can also detect word length of the SPDIF channel.
- The ADSP-SC58x SPDIF receiver doesn't decode any compressed data streams, but it can detect several standard compressed streams.

SPI

The ADSP-SC58x SPI design is similar to that of the ADSP-BF60x and ADSP-70x Blackfin processors. The key differences are:

- The minimum PCLK:SPI_CLK ratio for the ADSP-214xx SPI is 4:1, whereas it is 1:1 in the ADSP-SC58x SPI design. This gives better granularity for baud generation.
- The master mode lead/lag times in the older SPI design hardware-generated slave-select signals is fixed to 0.5xSPI_CLK. In the ADSP-SC58x design, it is configurable.
- The ADSP-SC58x SPI design supports flexible FIFOs (eight deep for 8-bit transfer size, four deep for 16-bit, and two deep for 32-bit).
- In the previous SPI design, transfers have to be initiated with a read/write from/to the receive/transmit FIFO. The ADSP-SC58x design allows for more flexible SPI transfer initiation options, as the SPI can be configured to initiate transfers either when the transmit FIFO is not empty, the receive FIFO is not full, or when both the transmit and receive FIFOs can accommodate new transfers.
- The ADSP-SC58x SPI design also allows flexible interrupt generation from the SPI module. It is possible to set a watermark for the SPI transmit/receive FIFO. The SPI will generate interrupt only when the transmit FIFO content is less than the watermark and receive FIFO content is more than the watermark, which can greatly reduce SPI interrupt frequency.
- The ADSP-SC58x SPI also features word count registers, where the SPI will stop interrupting the core after the word count has been reached. In DMA mode, the Word Count register ensures that the number of SPI transfers is exactly equal to the value programmed in the DMA channel.
- The ADSP-SC58x SPI design also supports an urgent watermark, which is helpful in generating urgent requests when the transmit FIFO is nearly empty or the receive FIFO is nearly full.
- The ADSP-SC58x SPI design provides support for hardware flow control. Flow control also supports the watermark feature, which provides additional flexibility.
- The ADSP-214xx SPI has only one DMA channel, so full-duplex operation is not possible. The ADSP-SC58x SPI design allocates a transmit DMA and a receive DMA channel, thereby providing full-duplex DMA support.

- The ADSP-SC58x SPI design supports dual-I/O mode, quad-I/O mode, and fast mode.
- The ADSP-SC58x processor's SPI2 module also supports memory-mapped mode and execute-in-place (XIP) mode. In memory-mapped mode, communication to a SPI memory device is automated such that memory contents are accessible by directly reading the processor's address space. Once configured, a SPI memory device will behave similar to an L2 ROM. XIP mode allows the processor to execute code directly from SPI memory.

Please refer to the pin multiplexing table in the data sheet for information regarding SPI modules which support dual- and quad-I/O modes and flow control.

UART

The UART design of the ADSP-SC58x processor is similar to that of the ADSP-BF60x and ADSP-70x Blackfin processors. The key differences between ADSP-SC58x and ADSP-214xx are listed below:

- The ADSP-SC58x UART has hardware flow control which automatically generates the CTS/RTS signals.
- 9-bit transmission from the ADSP-214xx UART is still supported in the ADSP-SC58x design through Multi-Drop Bus mode.
- The ADSP-SC58x UART design has improved bit granularity, as the module has both a bit clock and a sample clock. In normal mode, the bit clock is equal to 1/16th of the sample clock. Setting the bit clock equal to the sample clock increases the bit accuracy.
- The ADSP-SC58x UART also supports IrDA SIR mode, useful for implementing the IrDA SIR physical layer protocol.
- The ADSP-SC58x UART also supports inter-frame gap transmission and LIN break commands.

Watchdog Timer (WDT)

The ADSP-SC58x Watchdog Timer (WDT) design is similar to that of the ADSP-BF60x and ADSP-70x Blackfin processors. Unlike the ADSP-214xx processor watchdog timer, the new design is not capable of resetting the processor on its own. The ADSP-SC58x WDT can only raise an interrupt event. If the interrupt is enabled in the SEC, the core can issue a software reset in the WDT ISR. If the WDT interrupt is mapped to the Fault Management Unit (FMU), the FMU can be configured to reset the processor upon expiration of the watchdog timer.

In a system where multiple sources can reset the processor through the FMU, it is not possible to identify which module performed the reset.

Peripheral Timer

The ADSP-SC58x peripheral timer design is similar to that of the ADSP-BF60x and ADSP-70x Blackfin processors. The ADSP-SC58x design supports all the modes supported by the ADSP-214xx timer, along with some additional functionality:

- In the ADSP-SC58x design, separate set and clear registers are available for the Timer Enable and Timer Stop registers, which makes enabling and disabling of individual timers without affecting operations of other timers much easier.
- In PWM mode, the timer can be configured to take its clock from the internal system clock (SCLK0) or from a clock supplied on one of the external pins (TM_ALT_CLK or TM_CLK).
- In Width-Capture mode, the timer can be configured to capture the width of an input signal on either the TM_TMR or TM_ACI pin.
- The ADSP-SC58x design still supports single pulse generation, but it is through a separate mode (single-pulse PWMOUT mode).

- The ADSP-SC58x design also supports stopping the timer either abruptly or gracefully (waiting until the end of the current period).
- The pin interrupt (PINT) mode of the ADSP-SC58x timer design can generate an interrupt on the rising or falling edge of the signal on the TM_TMR pin or the TM_ACI pin.

Accelerators

- The ADSP-SC58x FIR/IIR accelerator design is the same as on the ADSP-214xx processors. On ADSP-214xx processors, only one accelerator can be enabled at a time. The ADSP-SC58x design removes this limitation, and all accelerators can be in use simultaneously.
- The ADSP-SC58x FFT accelerator design supports features like pipelined DMA operation and compute engines, IFFT, and hardware support for squaring of magnitude.

Conclusion

The ADSP-SC58x peripherals have been improved significantly as compared to the ADSP-214xx predecessors, providing innovative solutions for an ever-growing embedded application world. Understanding these key differences and important enhancements will greatly reduce the effort associated with migrating a design from the ADSP-214xx family of SHARC processors to the ADSP-SC58x family.

References

[1] *ADSP-SC58x SHARC+ Processor Hardware Reference*. Rev 0.2, June 2015. Analog Devices, Inc.

[2] *ADSP-214xx SHARC Processor Hardware Reference*. Rev 0.3, June 2010. Analog Devices, Inc.

[3] *SHARC+ Dual Core DSP with ARM Cortex-A5 Datasheet*. Rev. PrC. May 2015. Analog Devices, Inc.

[4] *ADSP-21483_21486_21487_21488_21489 Processor Data Sheet*. Rev B. March 2013 Analog Devices, Inc.

[5] *ADSP-21467_21489 Processor Data Sheet*. Rev B. March 2013 Analog Devices, Inc.

[6] *ADSP-21477_21478_21479 Processor Data Sheet*. Rev C. July 2013 Analog Devices, Inc.

Document History

Revision	Description
Rev 1 – June 12, 2015 *by Sachin-V Kumar*	Initial Release