ANALOG DEVICES Engineer To Engineer Note EE-125

Technical Notes on using Analog Devices' DSP components and development tools

Phone: (800) ANALOG-D, FAX: (781) 461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp

ADSP-218x Embedded System Software Management and In-System-Programming (ISP)

Contributed by HS, last change 16-SEP-2000

Introduction:

Modern embedded DSP systems are these days equipped with non-volatile memory devices such as EEPROM or FLASH memories. This allows in an easy way a later reprogramming with a new content either to correct problems of the current firmware or to enhance capabilities of the system.

All members of the ADSP-218x processor family can be booted from a single 8bit wide FLASH memory device. The Engineer- to- Engineer Note EE-39 discusses the connection to and the access of byte- programmable FLASH memory components thoroughly. To understand and transfer the content of this application note to the own system, the user should be familiar with the content of EE-39, describing the library functions used in this EE note in detail.

This Engineer-to-Engineer note describes an easy way of safe in-system programming and update of the user application code by running a software management system on the ADSP-218x. The provided software assumes an external 4Mbit FLASH component connected to the DSP in full memory mode (BDMA) and an RS-232 port, which is emulated by the ADSP-218x programmable flags. Furthermore, a jumper or switch is installed on the data memory line DATA[7], which can be pulled-up or down weakly.

Declarations and Naming:

Before heading on, it is necessary to define common terms: the user application program (UAP) is the software written by the user which will later function on the embedded DSP system as application software running as the main task. The boot image of this application is called UAB, user application boot-image.

The systems management software (SMS) takes care of flashing the user code into the FLASH memory and allowing FLASH maintenance like erasing or programming sections of the FLASH.

Application Idea:

As the systems management software should not reduce the available internal memory of the ADSP-218x and must be protected against the user application code, the easiest approach is to split the two tasks into two different executables. One contains the UAP, the other the SMS. One of their boot images is loaded at a time from the common FLASH memory at system reset.

The start-up selection is handled by a change to the ADI common 218x boot loader and the setting of the jumper on DATA[7]. On low level of the pin at reset, the user program will be booted, on high the system software. If the SMS was selected, it will prompt a welcome screen to an ASCII terminal connected to the RS-232 interface of the system.

Hardware Design:

The ADSP-218x is connected in full memory mode to an AM29F004 4Mbit flash memory through the BDMA port. The DSP is set-up for system booting from the FLASH memory.

A dedicated UART chip is not required. The protocol is implemented in software using PF3 as input and FL0 as output pin. A suitable RS-232 level converter would be an ADM3311 or ADM202. To select between user code or system software a dip switch is added to DATA[7].

System Software Design:

Upon booting it must be selected whether to boot either the user code or the system software by setting the dip switch on DATA[7] accordingly. So after power-up either the user code beginning from 0x0 in the FLASH memory or the service kernel from 0x70000 in the FLASH is loaded.

To keep this function alive after download of the user program to the FLASH memory beginning from 0x0, the original 32 program memory words boot strap loader, created by the 21xx splitter tool, is modified (exchanged) during software download.

Whenever the service software shall be run, a connection to an ASCII terminal at 9600,N,8,1 (9600 baud, no parity, 8 data bits, 1 stop bit) is necessary. A first splash screen comes up after reset, giving access to the user interface. Pressing <??> provides the user further hints on available commands built into the software. All user inputs will be dispatched by the command module for e.g. FLASH maintenance, like erasing <SE> or FLASH memory info <FI>, receiving and programming data to the user space of the FLASH memory (0x0..0x70000), using the implemented FLASH routines.

During user program download via RS232 and programming to the FLASH the system software cares for the exchange of the first 96 byte kernel loader, thus allowing future calls of the system management software.

The system software allows furthermore to specify the desired start sector for the application, to have several applications stored into the FLASH memory. These applications can afterwards be easily accessed from the user program.

First Time Preparations:

Before the system software can be booted at system reset from the FLASH memory, it must be stored to it. Therefore, please start up your ADSP-218x emulator, download the systems management software to your target and start executing it by pressing either the start button or hitting F5. The software will show you via the RS-232 interface of your target board on the connected ASCII terminal the spash screen and waits for commands.

Upon the first dry- run of the system software, it is possible by the hidden command <HS> to upload the system software itself into the FLASH memory. All necessary modifications for software restart will be done to the SMS Motorola S2 image file during download. A change of the system management software is recommended at factory only.

Having downloaded the monitor program itself, please clear sector 0 of the FLASH memory using <SE>, <0> and download the bootcode <DB>.

With the completion of the first download to your target, unplug the emulator, set DATA[7] to system software mode and start up the recently downloaded system monitor software by pressing the reset button. Now you have via RS-232 full control over your system, can erase the user accessible FLASH memory parts, collect sector information and download your Motorola S2 image file (UAB) of your application.

Before applying the in- system- programming sequence to your own application code, please insure that it is well tested. Please verify first proper execution of your UAP source before programming it into the FLASH device, otherwise a successful start-up of your own application can’t be guaranteed.

Having passed this first requirement, the user can continue with the process by generating the boot image of the user application code (UAB). Please use the spl21 splitter tool with following command line call:

SPL21 yourexe yourS2 -loader -2181 -s2

This will generate the required Motorola S2 boot stream for your ADSP-218x processor. For further into the available command line switches, please consult your v documentation.

Programming User Boot Images:

To program or update the UAB, set the embedded system in maintenance mode and press reset. The SMS will prompt you first to erase your destination sectors of the FLASH. With the sectors cleared, pressing <DL> will prompt for the file to download. After a successful download, please set the jumper on DATA[7] back to user mode and press reset – the new UAP should be running.

Caveats:

If Sector 0 of the FLASH is unpopulated by a user program, please insure that the modified boot code is written to the first 96 words of your FLASH eprom using the <db> command - otherwise the system software will be inaccessible after reset!

Please insure before programming that sector 0 of the FLASH holds no data, as a FLASH memory cell can’t be reprogrammed before erasing it. It is safest to run a sector erase <se> for sector 0 first and afterwards download the boot code <db>.

References:

http://www.analog.com/dsp
v Documentation
EE-39: 218x FLASH Application Note
ADmsp403 Evaluation Board Schematics
ADmsp403 Embedded Systems Software