Actel
POWER MATTERS

Cortex-M1 Enabled Fusion

Development Kit User's Guide

Version 1.1

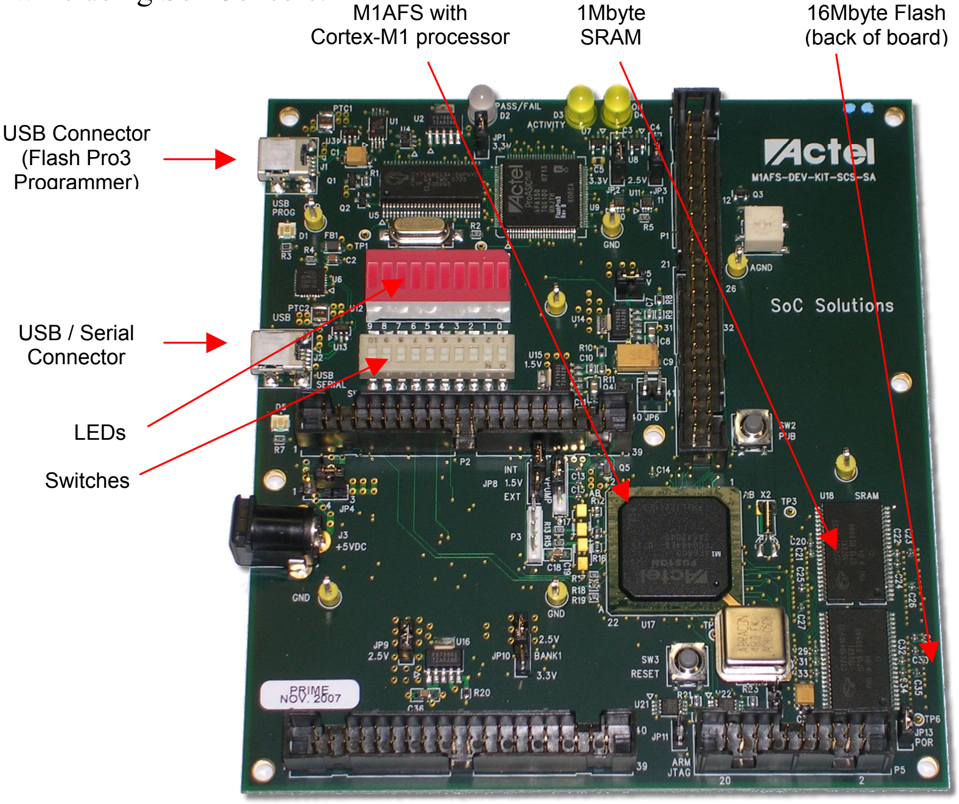
Introduction

Thank you for purchasing the Actel Cortex-M1 Enabled Fusion™ Development Kit.

Key Features

- In-system configurable analog supports a wide variety of applications
- Up to 8 Mbits of user flash memory
- Extensive clocking resources
  - Analog PLLs
  - 1% RC oscillator
  - Crystal oscillator circuit
  - Real-Time Counter (RTC)
- Flash FPGA fabric
  - Single chip
  - Low power
  - Secure
  - Live at Power-Up (LAPU)
  - Firm error immune

The Cortex-M1 Enabled Fusion ^{TM} Development Kit is an advanced microprocessor based FPGA development and evaluation kit. The purpose of the kit is to help the user become familiar with the Fusion FPGA features by providing a useful Sample Design, with a “How To” tutorial for implementing the FPGA hardware design using Libero Project Manager and CoreConsole. The tutorial also shows how to implement Cortex-M1 embedded software using SoftConsole.



M1AFS Development Board

Table of Contents

Document Assumptions

This user’s guide assumes the following:

- You intend to use Actel Libero® Integrated Design Environment (IDE) software.
- You intend to use Actel CoreConsole® and SoftConsole® software.
- You have installed and are familiar with:
    - Actel Libero IDE v8.1 or later software.
    - Actel CoreConsole® v1.4 or later software.
    - Actel SoftConsole® v2.0 or later software.
- You are familiar with PCs and the Windows® operating system.

Contents and System Requirements

This chapter details the contents of the Cortex-M1 Enabled Fusion Development Kit and lists the power supply and software system requirements.

Cortex-M1 Enabled Fusion Development Kit Contents

The Cortex-M1 Enabled Fusion Development Kit includes the following:

- M1AFS Development Board
- Libero IDE v8.1 CD (in DVD case)
- Development Kit Install CD with sample design
- 2 USB A to Mini-B Cables
- +5.0 V external power supply with international adaptors
- 4 Self-Adhesive Rubber Pads
- Quick Start Guide

System Requirements

The Cortex-M1 Enabled Fusion Development Kit requires the following:

- PC or Laptop running Windows XP or Windows 2000
- 2 USB ports (connectors) on the PC or Laptop

Additional Information

For further information, refer to the following appendices:

Appendix A – “Package Connections”

Appendix B – “Board Schematics”

Appendix C – “User Tests”

Appendix D – “Support”

Hardware Components

This chapter describes the hardware components of the Cortex-M1 Enabled Fusion Development Kit.

Ideal Uses for the development kit

Ideal uses for the Cortex-M1 Enabled Fusion Development Kit are the following:

- development and verification of embedded microprocessor based systems or subsystems
- product development platform
- algorithm development

Applications

The Cortex-M1 Enabled Fusion Development Kit is ideal for use in the following applications:

- mixed-signal system on a chip
- smart controllers
- Digital Signal Processing
- CPU or DSP (inside FPGA)
- wireless baseband processors
- communications
- display controllers
- sensor controllers

M1AFS Development Board

The M1AFS Development Board contains these features:

- Actel M1AFS600 Fusion
- 1 MByte SRAM
- 16 MByte Flash
- Analog Test Circuits
- USB–RS232 converter chip
- GPIO connectors
- 3.3V and 2.5V I/O
- On-board FlashPro3 circuitry
- 20-Pin Cortex-M1 JTAG connector
- Socketed Crystal Oscillator
- Pushbutton power-on reset circuit
- 10 test LEDs
- 10 test switches
- Expansion connectors including Analog Connector with signals from 6 analog quads

Detailed Board Description and Usage

The Cortex-M1 Enabled Fusion Development Kit has various advanced features that are covered in later sections of this chapter. The architecture provides access to a one-chip FPGA solution containing a Cortex-M1 32bit RISC processor, and mixed-signal peripheral components.

Note that the Actel FPGA is soldered directly to the board. The development board is available only in a directly soldered configuration. A socketed configuration is not available.

Full schematics are available on the Install CD supplied with the development kit. See Appendix B for M1AFS Development Board schematics contents.

Block Diagram

The following simplified block diagram shows the main features of the M1AFS Development Board. The blocks inside the FPGA block are CoreConsole™ and SmartGen™ components.

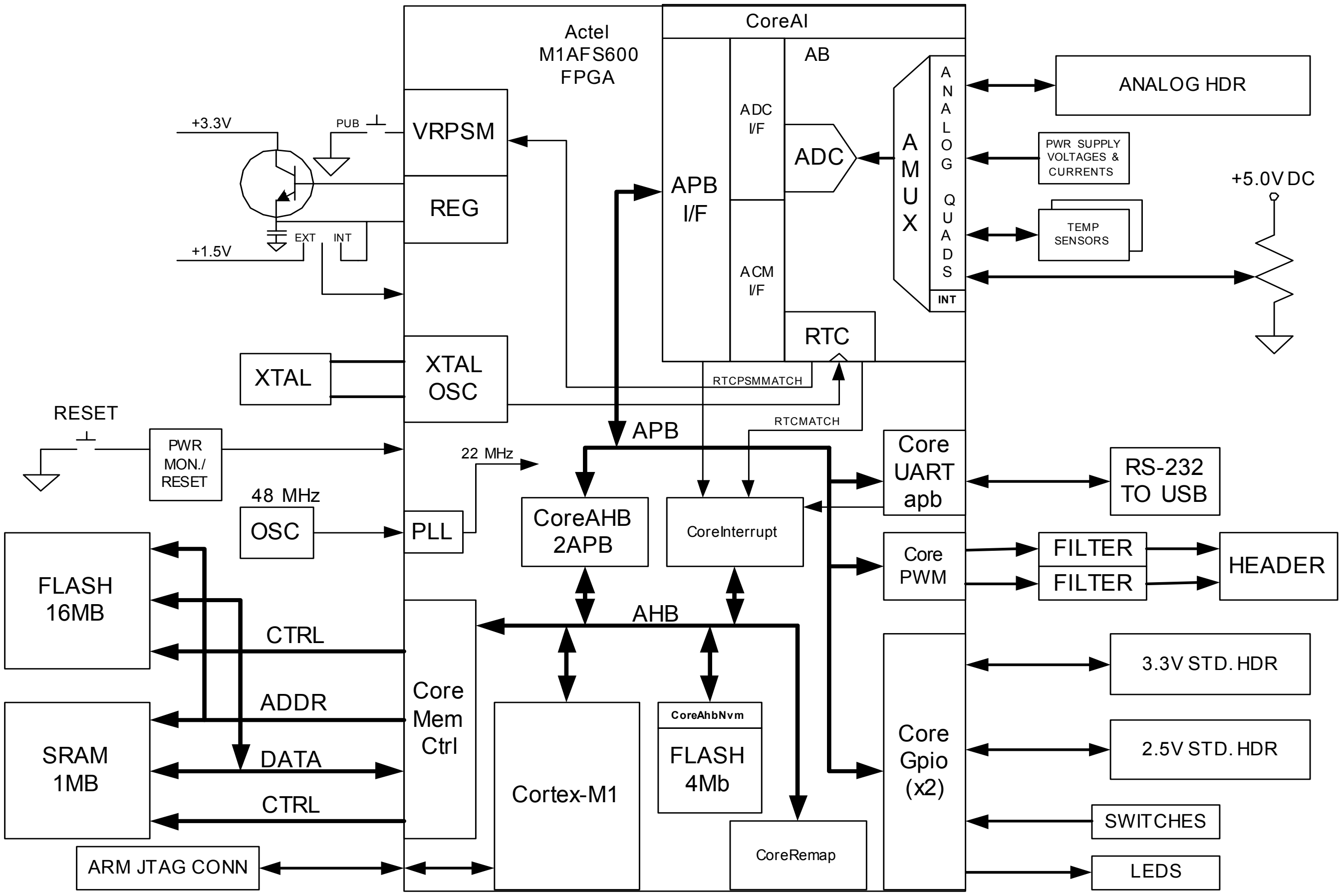


Figure 2.1 - M1AFS Development Board Block Diagram

Power

The development kit may only be powered through the J3 connector with the external +5.0 V 2.1 mm positive-center power supply that is included with the kit. Jumper JP4 should be installed in the 1-4 position to allow this. This jumper is a convenient place to measure the input current. The board was originally designed to allow USB powering options; however, the USB inrush current was too high so these options were removed.

A USB “mini-B to A” cable is provided for the USB (FlashPro3) connector. A second cable is also provided for the USB-to-RS232 interface. Both USB ports can be used simultaneously.

To program the Fusion FPGA, the USB cable should be plugged into the top, or J1 connector. The circuitry along the top of the board emulates the Actel FlashPro3 programming adaptor. The FlashPro software is then used to program the device.

The FlashPro3 logic uses 3 voltage regulators (blocks marked R). A 3.3V regulator provides power to the USB interface. The USB interface can then enable 2.5 V and 3.3 V regulators for the APA150 FPGA which implements the JTAG programming logic for the Fusion FPGA.

Aside from the regulators for the FlashPro3 circuit, there are three regulator components on the board to provide 1.5 V, 2.5 V, and 3.3 V to the Fusion FPGA.

In programming mode, the “ON” LED (D4) illuminates to indicate that J1 has been connected to a USB port. LED D5 indicates that the 3.3V regulator is operating.

The 3.3 V supply is used to provide the VPUMP programming voltage.

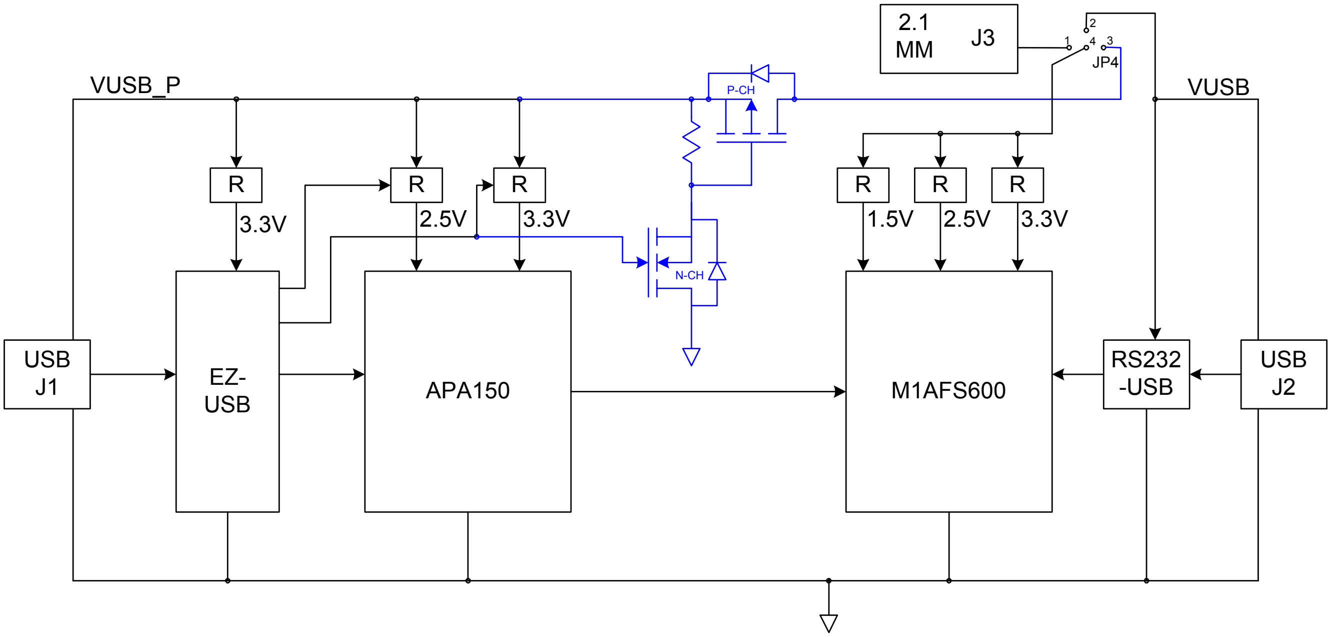


Figure 2.2- M1AFS Development Board Power Configuration

Accessory Card Power Supply Connections

Limited power may be supplied by the Fusion Development Kit to an accessory card. Connectors for accessory cards (headers P1, P2, & P5) are shown on page 2 of the schematics. The 5 V input voltage and the 2.5 V and 3.3 V regulated voltages are provided to the accessory card connectors.

Cortex-M1 Support

A standard 20-pin Cortex-M1 JTAG header (P5) is supplied to enable use of any ARM standard emulator and debug environment. Alternatively USB port J1 and the on-board FlashPro3 circuitry may be used as a debug path to the Cortex-M1's internal debug interface. Note that the FPGA design must instantiate the Cortex-M1 component with the FlashPro3/UJTAG debug option to include the debug logic for software development.

Connected to the FPGA for use by the Cortex-M1 are Flash and SRAM memories. The SRAM configuration is a byte-addressable 256Kx32, or 1MB. The Flash configuration is 4Mx32 or 16 MB (expandable to 32 MB). The Flash is not byte-addressable but each 16-bit word may be individually addressed. All memories are asynchronous. Access times may vary due to component availability so check the datasheets for the memories installed.

The clock for the FPGA logic can come from U20, a Socketed 48 MHz 3.3 V, 50% duty cycle crystal oscillator. This frequency may be modified inside the FPGA.

Two power-on resets are provided to the FPGA. One has a pushbutton feature to provide a warm reset. The non-push-button reset is usually used for the Cortex-M1's JTAG nTRST signal. Other reset schemes may also be used.

USB Serial Interface

USB Connector J2 provides a USB-to-RS-232 interface through U6. In this configuration, the Fusion FPGA should contain a UART core.

Analog Features

The Cortex-M1 Enabled Fusion Development Kit board has several features for exercising the Fusion chip's analog quads.

A potentiometer can provide a voltage between 0V and +5V DC.

Two transistors are configured as diodes to provide temperature sensors. These are placed at different locations on the board so that temperature differences may be measured.

Current and voltage sensing is offered for the 3.3V, 2.5V, and 1.5V supplies.

A transistor is provided to complete the internal 1.5V regulator circuit. A jumper is provided so that internal or external voltage can be selected.

A 32.768 KHz crystal is provided for the RTC circuitry.

A 2 pin header is provided so to either give access to the internal 2.56V voltage reference or to allow it to be replaced with an external voltage reference.

A pushbutton switch to ground is provided on the PUB input.

A 50-pin analog header is provided for access to 6 of the analog quads.

Other Features

The development kit also contains a 10 position DIP switch bank and a 10 LED module for general purpose use. All signals are connected to the FPGA. The LEDs and switches are active high and the switches have 10K pull-down resistors.

Programming or Re-Programming the Sample Design

On the Cortex-M1 Enabled Fusion Development Kit Install CD, you will find a *Sample Design* folder containing a STAPL file for programming the target design. Select the *.STP* file from the CD and use that as the STAPL file in the FlashPro software. Selecting **PROGRAM** will erase, program, and verify the part.

M1AFS Development Board Jumper Descriptions

Jumper	Development Kit Function	Factory Default	Notes
JP1	Provides 3.3V to Prog. USB interface	Installed	Current can be measured at this point.
JP2	Provides 3.3V to FlashPro3 FPGA	Installed	Current can be measured at this point.
JP3	Provides 2.5V to FlashPro3 FPGA	Installed	Current can be measured at this point.
JP4	Selects input power (5V) to the main board	Installed 1-4	1-4 selects J3 2.1mm barrel connector. Other jumper positions have been removed and are not supported or recommended. Current can be measured at this point.
JP5	Provides 3.3V to non-FlashPro3 portion of board	Installed	Current can be measured at this point.
JP6	Provides for measurement or replacement of analog reference voltage.	Not Installed	Do not install jumper here or you will short out reference voltage.
JP7	Connects 3.3V to VPUMP pin on FPGA	Installed 2-3	Current can be measured at this point.
JP8	Selects internal or external 1.5V regulator.	Installed 2-3	1-2 selects internal voltage regulator 2-3 selects external voltage regulator Current can be measured at this point.
JP9	Provides 2.5V to non-FlashPro3 portion of board	Installed	Current can be measured at this point.
JP10	Selects 3.3V or 2.5V for bank 1 of Fusion FPGA	Installed 1-2	1-2 selects 2.5V 2-3 selects 3.3V Current can be measured at this point.
JP11	Connects pushbutton reset to P3	Not installed	This functionality is usually not required and can add noise to the reset.
JP12	Allows for enabling/disabling crystal oscillator U20.	Not Installed	Installing jumper ties enable input to GND. Enable/disable function depends on polarity of enable input of installed oscillator. Note that factory installed oscillator has no enable input.
JP13	Connects power on reset to FPGA	Installed	Can be uninstalled if this input is needed for another purpose.

Table 2.2- M1AFS Development Board Jumper Descriptions

Test Points

All ground test points on the board are fitted with small test loops. They are labeled only as “GND” or “AGND” for digital or analog ground. Signal test points are labeled on the silkscreen as TP1, TP2, etc. The test points have holes that a scope probe can access. Power voltages may be probed at the jumpers that connect them to the circuitry that they power. Voltages will be 5.0 V, 3.3 V, 2.5 V, 1.5 V, or GND. When measuring the voltage at a test point with a DVM (digital voltage multimeter) the ground lead should be connected to a test point labeled GND and the voltage lead should be connected to the voltage to be tested. All voltage labels on the board are relative to a 0 V ground reference or GND.

Physical Characteristics of Board

The printed circuit board assembly, including all components, is completely lead-free RoHS compliant.

The board is fabricated with six copper layers. The layers are arranged as follows from top to bottom:

- Layer 1 – Top Signal Layer
- Layer 2 – Analog & Digital Ground Plane
- Layer 3 – Signal Layer
- Layer 4 – Analog & Digital Ground Plane
- Layer 5 – Signal Layer, Power Plane cutouts
- Layer 6 – Signal Layer, Power Plane cutouts
- Layer 7 – Analog & Digital Ground Plane
- Layer 8 – Signal Layer
- Layer 9 – Power Plane
- Layer 10 – Bottom Signal Layer, Power Plane cutouts

3

Installation and Setup

This chapter outlines how to set up the Cortex-M1 Enabled Fusion Development Kit using the Install CD and the Libero DVD. This chapter also describes the initial M1AFS Development Board configuration and Power sequence.

Installing Libero IDE v8.x

Place the Libero DVD in the DVD Drive on your Personal Computer or Laptop. The DVD should automatically start an auto-run session. At this point, follow the instructions (prompts) on the "Libero IDE" dialog box. Install the complete Actel design environment, including the CoreConsole and SoftConsole tools.

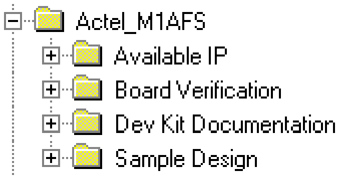
For more Libero IDE v8.x software installation instructions, please refer to the documentation supplied in the Libero IDE DVD case or refer to the *Actel Libero IDE / Designer Installation and Licensing Guide for Software v8.x*.

Note: Libero IDE , CoreConsole and SoftConsole tools will be used in for the Sample Design Tutorial in Chapter 4.

Installing Cortex-M1 Enabled Development Kit using the "Install CD".

Place the Development Kit "Install CD" in the CD Drive on your PC (PC refers to either your Personal Computer or Laptop). The CD should automatically start an auto-run session. Follow the instructions (prompts) on the "Install" dialog box.

The "Install" application will properly place all the documentation and sample project files in the C:\Actel_M1AFS folder (default) or the user selected folder. Figure 3.1 shows the installed directory structure.



```
⊟ Actel_M1AFS
   ⊞ Available IP
   ⊞ Board Verification
   ⊞ Dev Kit Documentation
   ⊞ Sample Design
```

Figure 3.1 – Installed directory structure

The "Install" application will also install the SFE USB to RS232 Controller driver set for the Serial USB connector J2 (SERIAL). The drivers are copied to: C:\Program Files\SparkFunElectronics\USB (default location).

Note: The FlashPro3 USB drivers for the J1 USB (PROG) interface are located in the Libero IDE Installation location.

The Install CD contains the following documention:

- This Cortex-M1 Fusion Enabled Development Kit User Guide.
- Cortex-M1 Fusion Enabled Development Kit Quick Start Guide.
- M1AFS Board Schematics and Layout.
- Available IP marketing briefs from Actel and also 3rd Party IP Vendors.

Initial M1AFS Development Board configuration

Before powering up the M1AFS Development board for the first time, please make sure the switches and jumpers are in the following, factory set, positions:

SW1: All switches (0-9) are in the ON position.

Jumpers: **JP1, JP2, JP3, JP4(1-4), JP5, JP7(2-3), JP8(2-3), JP9, JP10(1-2), JP13** are installed.

All others are not installed.

Powering up the M1AFS Development Board

Apply power to the board by connecting one end of the 5-Volt power supply to the J3 connector on the M1AFS board and the other end to a power outlet.

Next, connect one end of a supplied USB cable to a USB port (connector) on your PC or Laptop. Connect the other end to M1AFS connector J1 (PROG). After a few seconds, you should see the big yellow "ON" LED at the top right of the board illuminate.

The PC will detect that new hardware is installed. The "add new hardware wizard" will take care of the Actel FlashPro3 driver installation. The wizard will ask for a location for the drivers.

Note: See the Actel FlashPro3 documentation for the location of these drivers. These are usually located in the "<FlashPro install location>\Drivers".

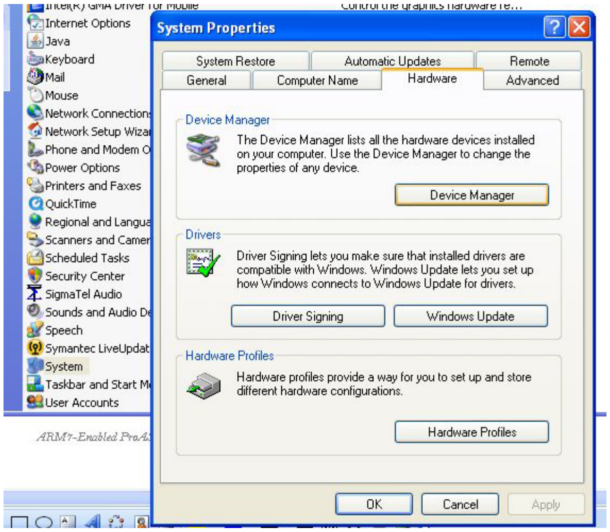
Now connect one end of the second supplied USB cable to a second USB port (connector) on your PC or Laptop. Connect the other end to M1AFS Development Board port J2 (SERIAL). You should see the LED closest to the J2 connector illuminate. This USB port is used to transport a serial RS-232 interface over USB 2.0. The PC will recognize this interface as a COM port.

The PC will detect that new hardware is installed. The "add new hardware wizard" will take care of the USB driver installation. The wizard may ask for a location for the drivers. The drivers are located at "<M1AFS Install Location>\Sample Design\USB_Drivers".

Determining the Serial COM Port

Verify the PC COM port that was enumerated with the RS-232 USB chip at J2.

1. On the PC, click the Start button and open up the "Device Manager" by navigating to "Control Panel -> System". Then click the "Hardware" tab, then the "Device Manager" button as shown in figure 3.2.



The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

Driver Signing lets you make sure that installed drivers are compatible with Windows. Windows Update lets you set up how Windows connects to Windows Update for drivers.

Hardware profiles provide a way for you to set up and store different hardware configurations.

Figure 3.2- Running the Device Manager

2. Expand the "Ports (COM & LPT)" tree in the view.

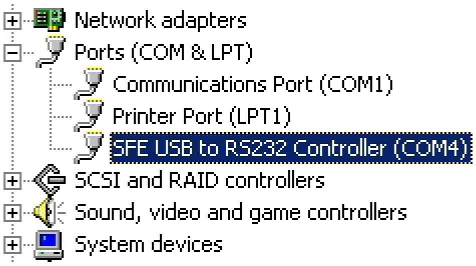

Network adapters
Ports (COM & LPT)
Communications Port (COM1)
Printer Port (LPT1)
SFE USB to RS232 Controller (COM4)
SCSI and RAID controllers
Sound, video and game controllers
System devices

Figure 3.3- Finding the COM Port

3. Figure 3.3 shows the COM port associated with the "**SFE USB to RS232 Controller**". This is the COM port associated with the J2 USB (Serial) Connector.

Note: Concerning the SFE USB to RS232 Controller, if the board is powered down while any PC application is using this resource, then the PC application will need to close the COM port and reopen the COM port after the board has been powered up again. In general, adhere to the following sequence: close COM port, power down board, power up board, open COM port. Otherwise, the communication channel will not function if the COM port is not restarted after a board powerdown/powerup sequence.

Factory Configuration

The M1AFS Development Board is shipped from the manufacturer with the Sample Design loaded in the M1AFS FPGA. Also the Analog Serial Display embedded software image is loaded into External FLASH. *See Chapter 4 – Sample Design Tutorial.*

After powering up the M1AFS Development Board for the first time, you should observe the LEDs at U12 illuminate in a pattern that reflects a voltage measurement across a potentiometer at R6. Change the measured voltage across R6 by rotating the dial on the potentiometer with a screwdriver, and watch the LED pattern change.

Sample Design Tutorial

This sample design is created specifically for the Actel Cortex-M1 Enabled Fusion Development Kit. This tutorial will guide you though the following sections:

Sample System Hardware Overview

Libero Project

- Top Level File
- Additional Source Files
- Constraints Files
- Device Options

CoreConsole Design

- Sample Design Hardware Overview
- Functional Description
- Memory Map

Programming the FPGA on the M1AFS Development Board

SoftConsole Designs

- Importing and compiling the SoftConsole designs
- Debugging the Analog Serial Display sample design
- Using the Memory Loader Utility to program the on board Flash with the Analog Serial Display embedded code image
- Rebooting the board to run the Analog Serial Display software from Flash

Sample System Hardware Overview

The Sample system hardware design was produced using two tools in the standard Actel toolchain: Libero Project Manager and CoreConsole.

The Libero design contains the top-level file, other source files, the CoreConsole subsystem, constraints files, and build scripts. These are used by Libero to compile, synthesize, and place-and-route the sample design. The Libero project used in the sample hardware design is provided for reference in the development kit installation.

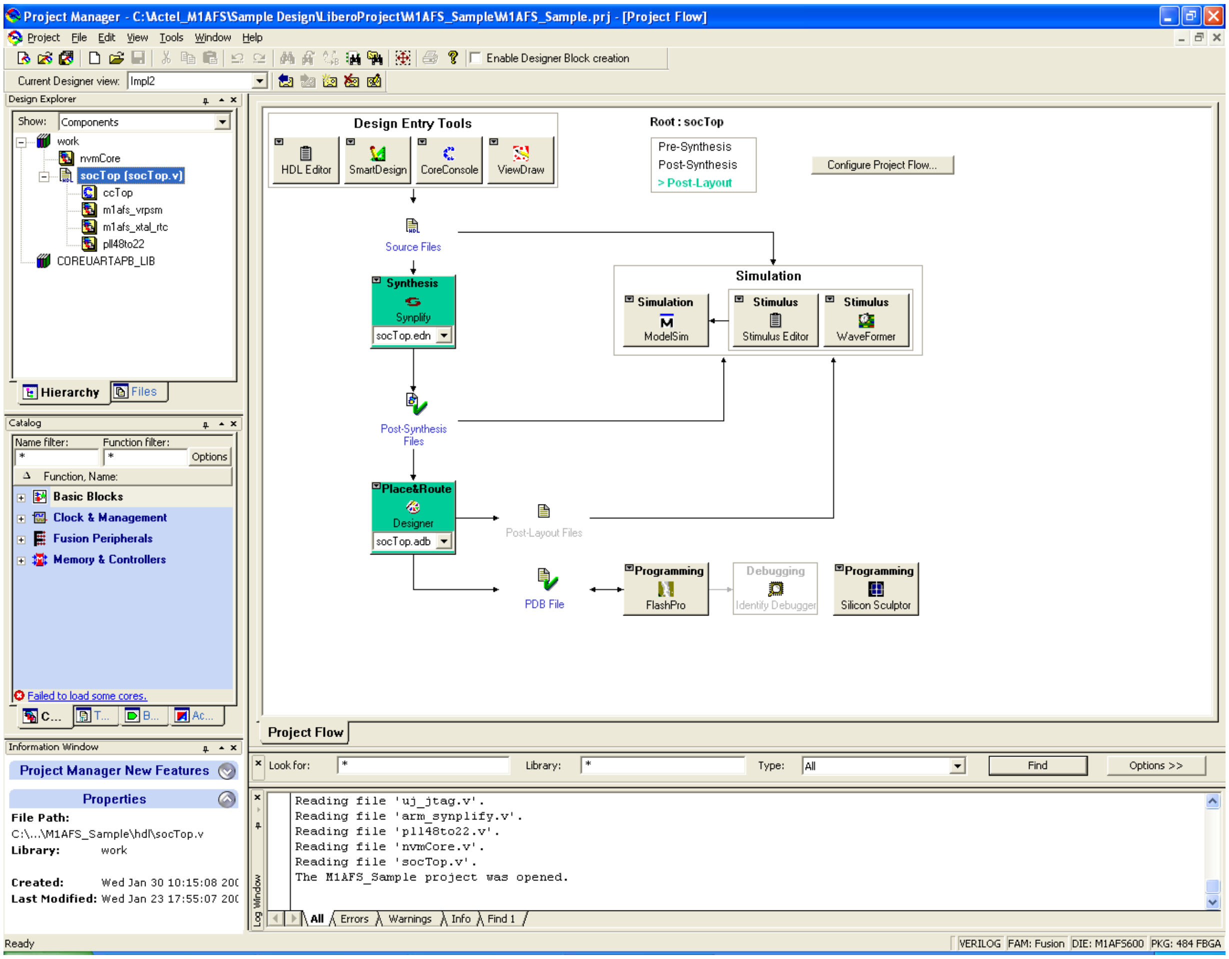
The CoreConsole component contains the processor, memory controllers, UART, Timer, Interrupt Controller, Analog Block and interface, Remap, GPIO, PWM, as well as instantiations of the AHB and APB buses. The CoreConsole component contains the bulk of the functionality and logic of the sample design. The CoreConsole project used in the sample hardware design is provided for reference in the development kit installation.

Note: It is not necessary to synthesize, place-and-route, or generate a programming file for the sample hardware design. Also, it is not necessary to regenerate the CoreConsole subsystem. These files are included in the M1AFS Development Kit installation.

Libero Project

Start Libero Project Manager by selecting Actel Libero IDE 8.x -> Project Manager …from the Start Menu.

Open the Libero project file M1AFS_Sample.prj by selecting the Project -> Open Project… menu item and then selecting M1AFS_Sample.prj. The project is shown in Figure 4.1.



Project Manager - C:\Actel_M1AFS\Sample Design\LiberoProject\M1AFS_Sample\M1AFS_Sample.prj - [Project Flow]

Current Designer view: Impl2

Show: Components

Root : socTop

Properties

File Path: C:\...\M1AFS_Sample\hdl\socTop.v
Library: work

Created: Wed Jan 30 10:15:08 200
Last Modified: Wed Jan 23 17:55:07 200

```
Reading file 'uj_jtag.v'.
Reading file 'arm_synplify.v'.
Reading file 'pll48to22.v'.
Reading file 'nvmCore.v'.
Reading file 'socTop.v'.
The M1AFS_Sample project was opened.
```

Ready | VERILOG | FAM: Fusion | DIE: M1AFS600 | PKG: 484 FBGA

Figure 4.1 - Libero Project

This project file is located in the <install folder>\Sample Design\LiberoProject\M1AFS_Sample\ folder.

Top Level Design

The socTop.v design, supplied in the "Sample Designs\Hardware\Source" folder, instantiates the following modules:

- PLL module
- Clock Buffers
- M1AFS Real-Time Clock Interface module
- M1AFS Voltage Regulator Power Supply Module
- Cortex-M1 Subsystem including Analog interface module (CoreConsole component)

The following code is from the socTop Verilog module. Note that CLKINT buffers are necessary on the reset signals since they are not on global buffer pins.

```
//////////////////////////////////
//     Cortex-M1 SAMPLE DESIGN
//////////////////////////////////

module socTop (
    pbRstN,
    poRstN,

...

  // I/O definitions:

    // System clocks:
    input  sysClk;

    // Resets:
    input  pbRstN;                  // external reset connected to pushbutton
    input  poRstN;                  // external reset not connected to pushbutton
    output flashRstN;               // reset to FLASH

    // RS232 connections:
    output rs232Atx;                // serial transmit line
    input  rs232Arx;                // serial receive line

  ...

// Instantiations:

    // Instantiate PLL:
    pll48to22 pll48to22(
        .POWERDOWN(poRstNi),
        .CLKA(sysClk_48MHz),
     ...

    CLKINT ci0 (.A(extClk), .Y(sysClk_48MHz));
    CLKINT ci1 (.A(pbRstN), .Y(pbRstNi));
    CLKINT ci2 (.A(poRstN), .Y(poRstNi));

    m1afs_vrpsm m1afs_vrpsm (
        .PUB            (PUB),
        .VRPU           (VRPU),
    ...

    m1afs_xtal_rtc m1afs_xtal_rtc (
        .XTL            (XTLCLK),
        .CLKOUT         (RTCCLK),
    ...

    ccTop ccTop_0(
        .DDGDON_DDGDON0({1{1'b0}}),
        .DDGDON_DDGDON1({1{1'b0}}),
    ...
```

Constraints Files

The following is the list of the Libero Constraint files for the Sample Design:

- constraints.sdc

  Timing constraints for synthesis

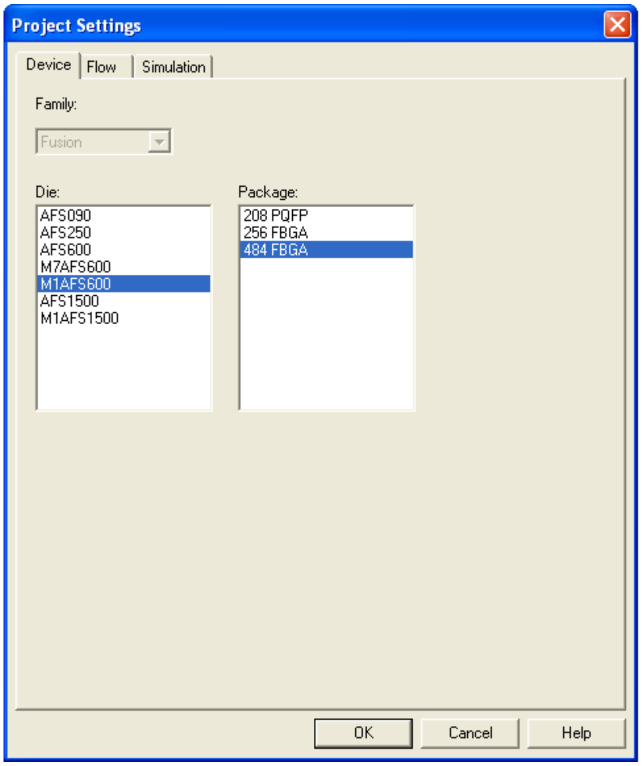
- socTop.pdc

  Physical design constraints for place and route (ie I/O placement)

These files are located in the <install folder>\Sample Design\LiberoProject\M1AFS_Sample\Constraints folder.

Device Options

Select Project -> Settings … from the menu to view the Libero device options as shown in Figure 4.2.



Family: Fusion

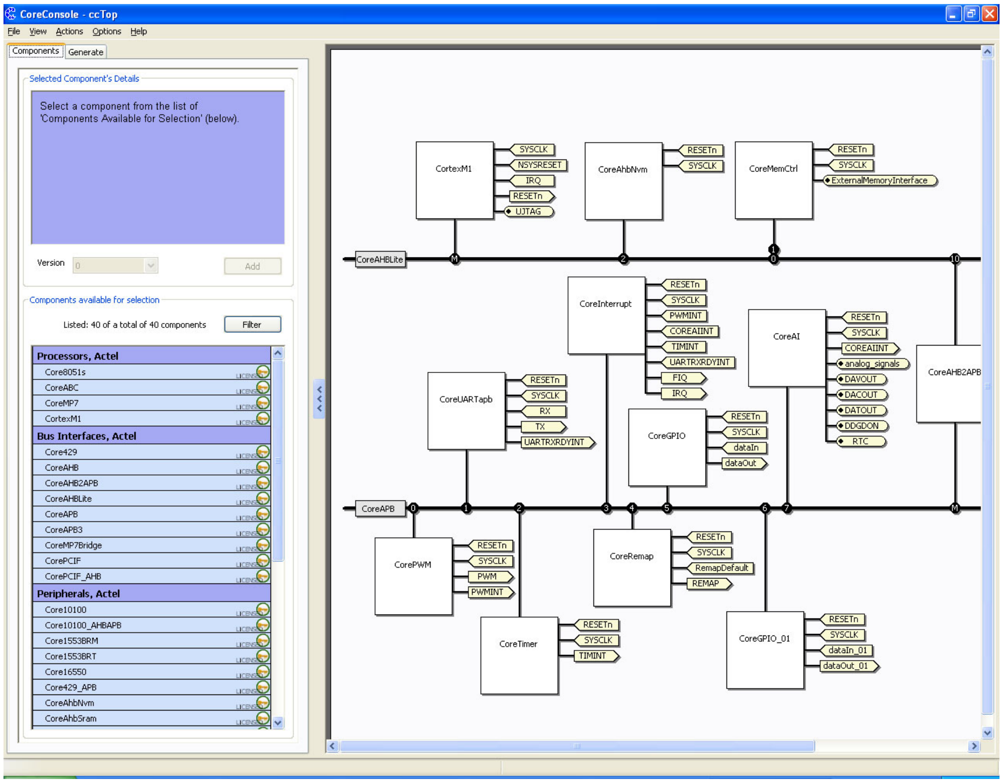
Die: M1AFS600

Package: 484 FBGA

Figure 4.2 - Libero Device Options

CoreConsole Design

To open the CoreConsole portion of the M1AFS_Sample project, expand the top level of the design in Libero Project Manager, right click on ccTop, and select Open Component.   The project is shown in Figure 4.3.



Select a component from the list of 'Components Available for Selection' (below).

Version: 0

Listed: 40 of a total of 40 components

Figure 4.3 - CoreConsole M1AFS_Sample Project

Sample Design Hardware Overview

To see more of the design you can expand the CoreConsole window to full screen and click on the 3 “<” symbols between the panes to give a view as in Figure 4.4.

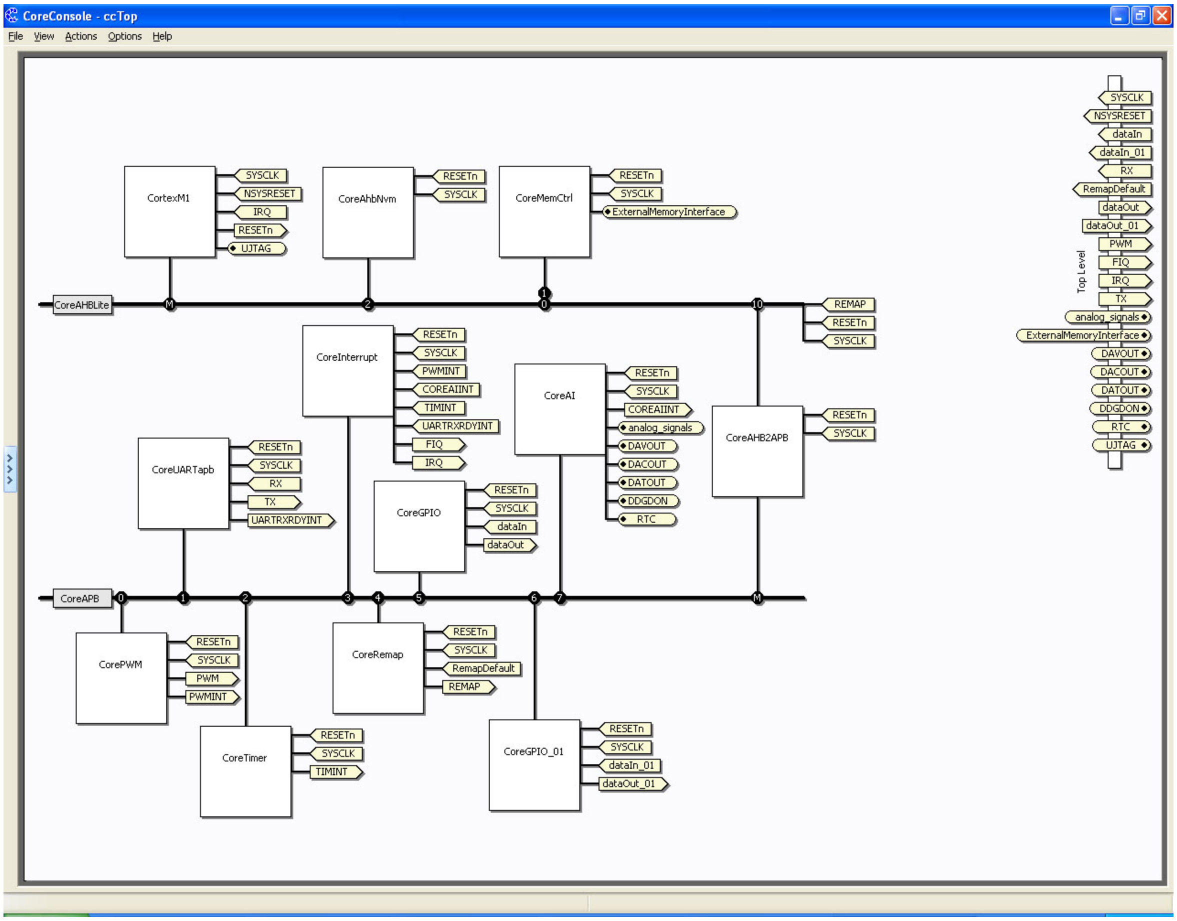


Figure 4.4 - Sample Design Block Diagram

This block diagram shows the CoreConsole components including the CoreAI block that contains the analog components. Note that the Cortex-M1 processor interfaces to the CoreAHBLite bus and that many of the peripherals are on the CoreAPB bus and accessed through the CoreAHB2APB bridges module. To get familiar with the modules you can right click on each and see their configuration parameters. You can then unexpand the window by clicking on the 3 “>” characters and click on the “Generate” tab near the top to see the “Generate” menus. If this were a new design you would click on the “Save & Generate” button and the design would be checked, saved and hardware and software files would be generated for use by Libero and SoftConsole. This has already been done for you.

SmartGen Components

The Sample Hardware design contains these SmartGen components:

- m1afs_vrpsm (Voltage Regulator Power Supply Module)
- m1afs_xtal_rtc (Crystal Oscillator in RTC mode)
- pll48to22 (Phase Locked Loop module to convert 48 MHz input clock to 22 MHz)

To explore these modules and their options right click on them in Libero and select **Open Component**.

Functional Description

The Sample Hardware design is a mixed-signal microcontroller system design. This design contains hardware that is typically used for supporting software, such as Real Time Operating Systems (RTOS) or basic controller software. This design specifically supports a Cortex-M1 processor based system.

The following blocks are contained in the CoreConsole sample design:

- Cortex-M1 (Processor)
- CoreAhbNvm (Internal Flash Memory Controller)
- CoreMemCtrl (External SRAM and Flash Controller)
- CoreAHB2APB (AHB to APB Bridge)
- CorePWM
- CoreUARTapb
- CoreTimer
- CoreInterrupt
- CoreRemap
- CoreGPIO (two General Purpose I/O modules)
- CoreAI (contains Analog Block)
- CoreAHBLite (bus)
- CoreAPB (bus)

The Cortex-M1 processor can be configured to use from 0 to 32 interrupts. In this example, the Cortex-M1 is configured to use one interrupt input (IRQ0). There are four interrupt sources in the design that share the same Cortex-M1 interrupt via the CoreInterrupt (interrupt controller module). The interrupt sources are the "Receive Ready" interrupt from the CoreUARTapb peripheral (UARTRXRDYINT), the "Timer Expired" interrupt from the CoreTimer peripheral (TIMINT), the CoreAI interrupt (COREAIINT), and the PWM interrupt from the CorePWM peripheral (PWMINT).

The Sample subsystem interfaces to both internal Flash memory and external SRAM and Flash memory. The internal Flash is connected to the CoreAHBLite bus through the CoreAhbNvm internal memory controller. The external (off chip) SRAM and Flash are also connected to the CoreAHBLite bus through the CoreMemCtrl module. The CoreRemap module is used to swap AHBLite bus positions 0 & 1 to swap external Flash & SRAM in the memory map.

The slower peripherals, CorePWM, CoreUARTapb, CoreTimer, CoreInterrupt, CoreRemap, CoreGPIO, and CoreAI modules, are connected to the CoreAPB bus. The CoreAPB bus is connected to the CoreAHBLite bus through the CoreAHB2APB bridge.

Memory Map

The memory map is largely determined by what slot on the AHB/APB bus a particular module occupies. An external Remap input is an input to the CoreRemap module that is used to toggle the mapping of SRAM and FLASH at AHB slots 0 and 1.

The following memory locations are determined by the position of switch SW1 as follows:

With SW1 #9 in the ON position:

External FLASH	0x00000000
External SRAM	0x10000000

With SW1 #9 in the OFF position:

External SRAM	0x00000000
External FLASH	0x10000000

The rest of the memory map is fixed:

Internal Flash	0x20000000
CorePWM	0xA0000000
CoreUARTapb	0xA1000000
CoreTimer	0xA2000000
CoreInterrupt	0xA3000000
CoreRemap	0xA4000000
CoreGPIO	0xA5000000
CoreGPIO_01	0xA6000000
CoreAI	0xA7000000

See the CoreConsole project for additional settings and connections. Exit CoreConsole by clicking File -> Exit.

If desired, the project can be rebuilt by clicking on the Synthesis, then Place&Route boxes in Libero; however, this has already been done for you. Clicking on the Synthesis block opens the Synplify synthesis application. Clicking on the Place&Route box opens the Designer application. If you have opened either of these applications then close them and get back to the Libero Project Manager before proceeding to the next section.

Programming the FPGA on the M1AFS Development Board

This section describes how to load the FPGA with the sample hardware design. It is necessary to reprogram the FPGA before running the sample software design described later in the SoftConsole Designs section.

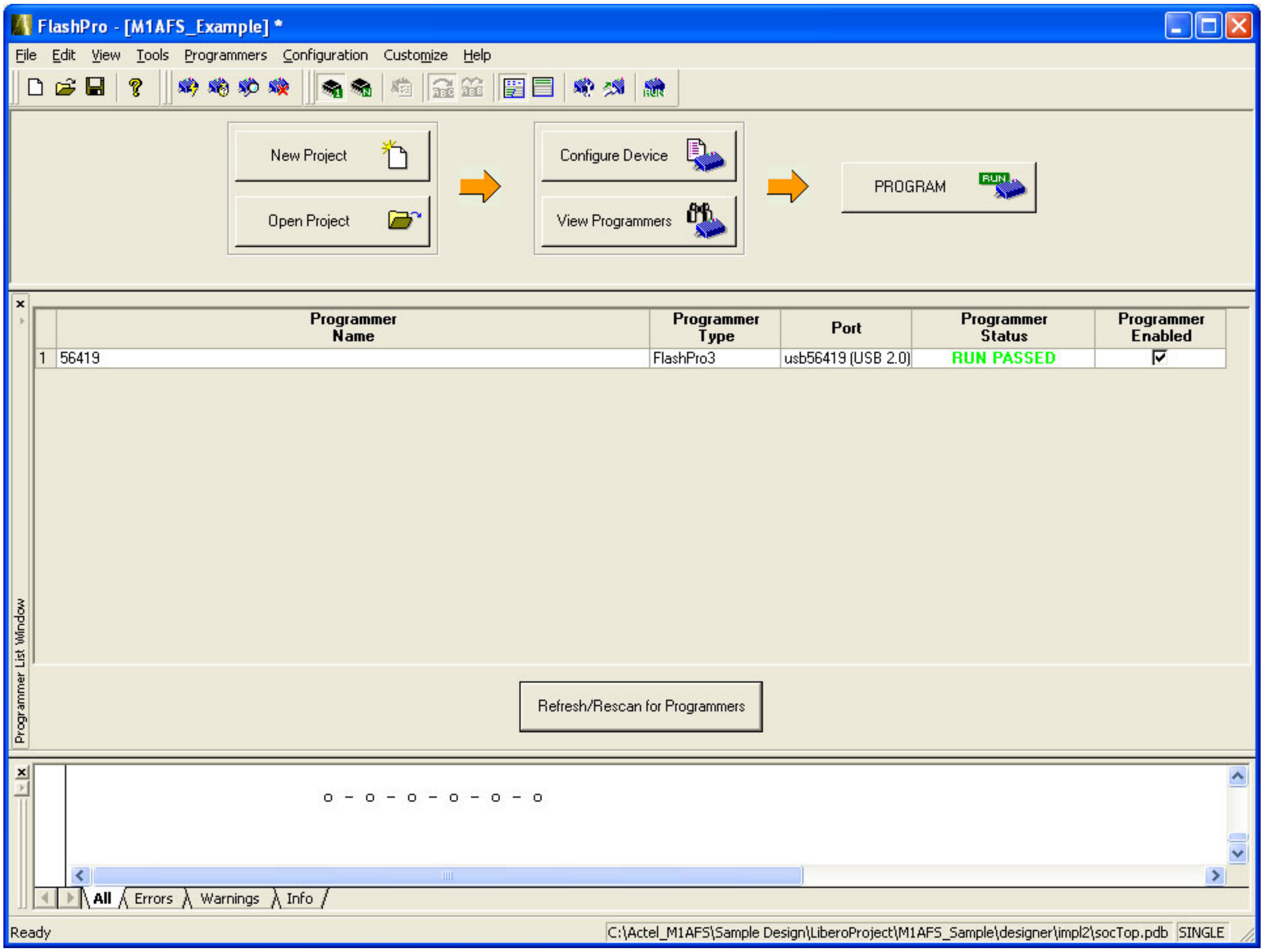
Apply power to the board by connecting one end of the external 5-Volt power supply to the J3 connector on the M1AFS board and the other end to a power outlet.

Next, connect one end of a supplied USB cable to a USB port (connector) on your PC or Laptop. Connect the other end to M1AFS connector J1 (PROG). After a few seconds, you should see the big yellow "ON" LED at the top right of the board illuminate.

Make sure that switch SW1, positions 0-8 are set to ON and position 9 is set to OFF.

FlashPro may either be run stand-alone or through the Libero Project Manager. For stand-alone usage, follow the FlashPro User Guide to program the M1AFS600. The socTop.pdb programming file is in the <install folder>\Sample Design\LiberoProject\M1AFS_Sample\designer\impl2 folder.

Run the FlashPro Programmer Libero Project Manager by clicking on the Programming block in the design flow pane. Use the FlashPro window to program the FPGA as shown in Figure 4.5.



	Programmer Name	Programmer Type	Port	Programmer Status	Programmer Enabled
1	56419	FlashPro3	usb56419 (USB 2.0)	RUN PASSED	☑

o - o - o - o - o - o

Ready C:\Actel_M1AFS\Sample Design\LiberoProject\M1AFS_Sample\designer\impl2\socTop.pdb SINGLE

Figure 4.5 –FlashPro3 Programmer Application

SoftConsole Designs

There are two SoftConsole embedded software designs provided with the development kit: a simple Analog Serial Display example and a memory loader utility for programming on-board Flash.

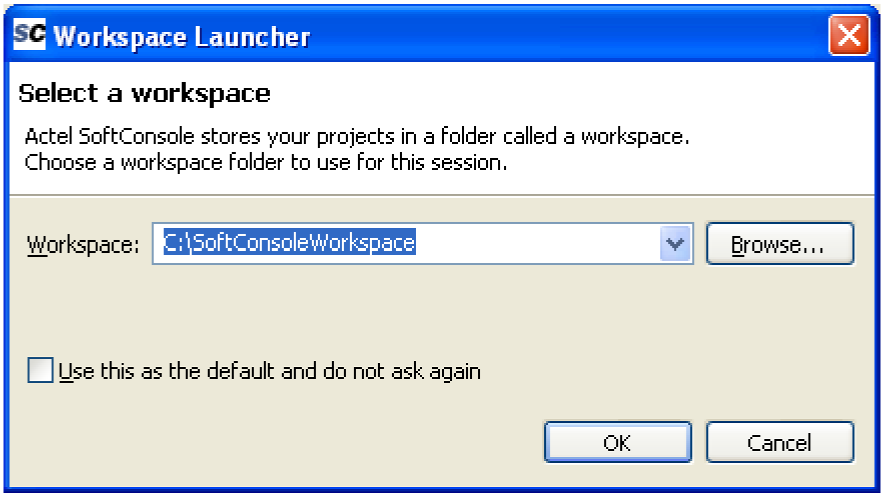
The Analog Serial Display sample design will demonstrate the Cortex-M1 system executing a program to read and display analog measurements. The display is implemented as ASCII characters through a UART that gets converted to USB on the board and can be seen on a HyperTerminal display on the PC. This design will demonstrate the basics of creating a Cortex-M1 based system FPGA design using CoreConsole and the embedded software to run the on the hardware using SoftConsole.

The memory loader utility provides a convenient method for programming the on-board Flash.

Both designs execute software instructions from external SRAM via the CoreMemCtrl block and AHB Lite bus. The external SRAM will contain the software program, data variables, and software stacks.

Importing and compiling the SoftConsole designs

Start SoftConsole by selecting Actel SoftConsole v2.x -> Actel SoftConsole… from the Start Menu. Browse to your SoftConsole "workspace" folder as shown in Figure 4.6. Click OK.



Workspace Launcher

Select a workspace

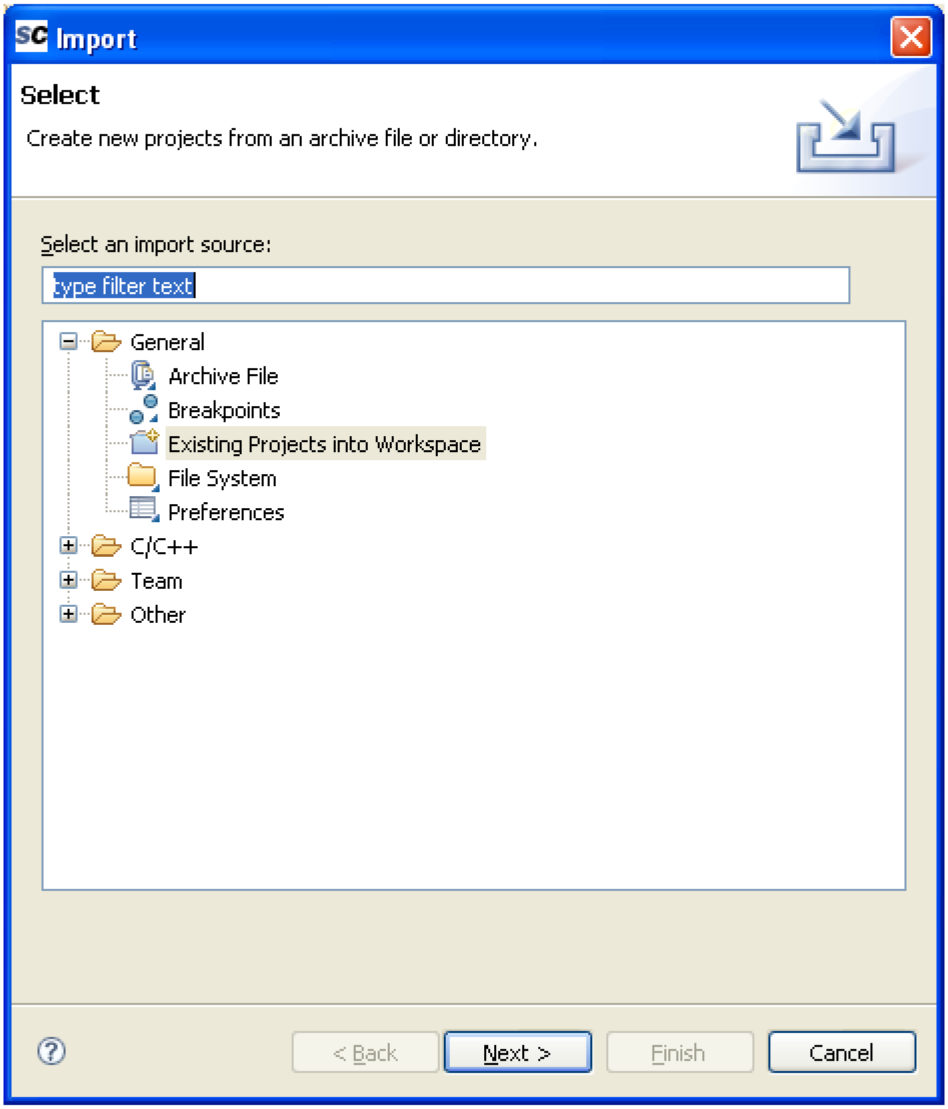
Actel SoftConsole stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\SoftConsoleWorkspace

Figure 4.6 - SoftConsole Workspace Launcher

Now import the Analog Serial Display sample design and the Memory Loader Utility project into SoftConsole by selecting File -> Import… item from the menu.

Figure 4.7 shows the resulting window.



Import

Select

Create new projects from an archive file or directory.

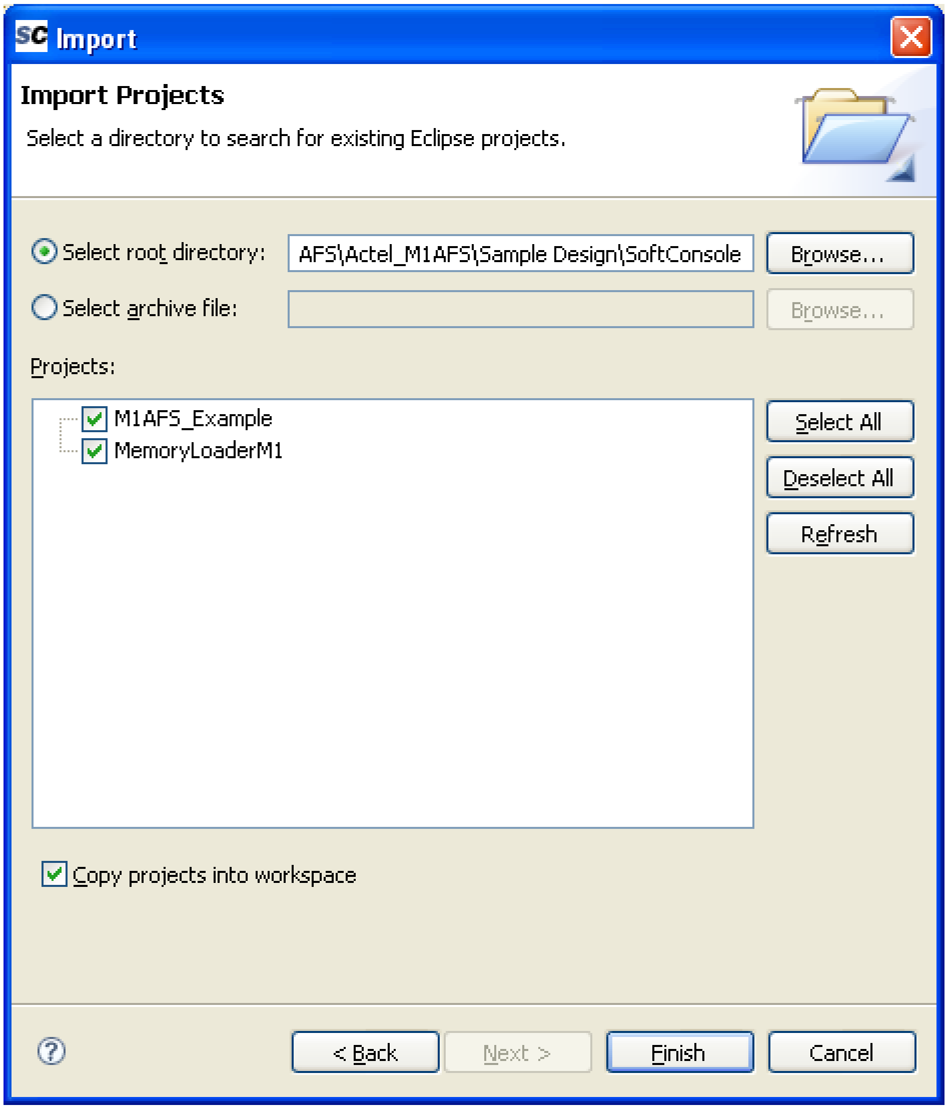
Select an import source:

type filter text

Figure 4.7 - SoftConsole Import Window

Select “Existing Projects into Workspace” in the SoftConsole Import Window and then click “Next”.

Figure 4.8 shows the next window.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: AFS\Actel_M1AFS\Sample Design\SoftConsole

○ Select archive file:

Projects:

☑ M1AFS_Example

☑ MemoryLoaderM1

☑ Copy projects into workspace

Figure 4.8 - Next SoftConsole Import Window

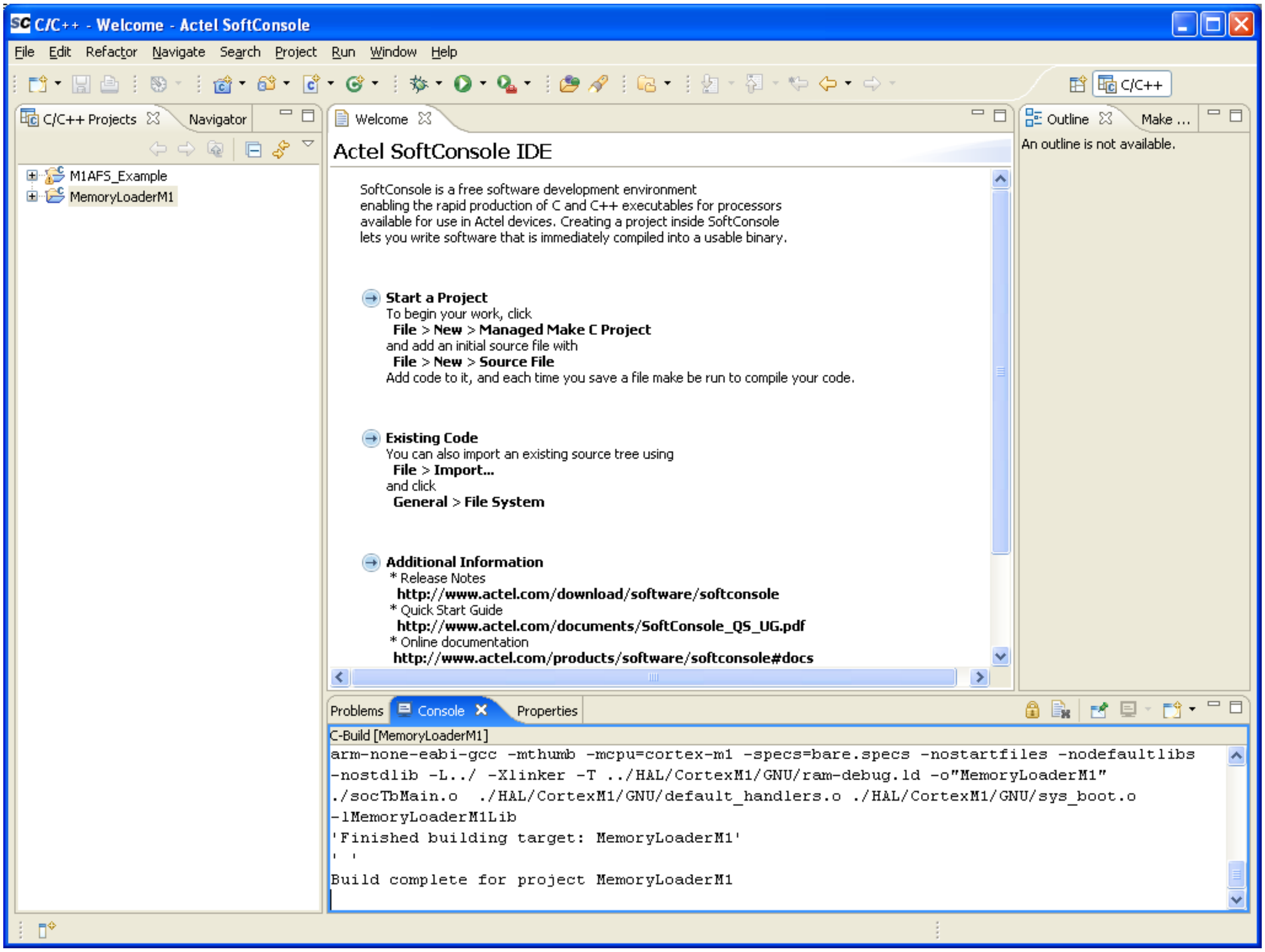
Click the "Select root directory" radio button and browse to the <install folder> \Sample Design\SoftConsole folder.

Make sure the MemoryLoaderM1 and M1AFS_Example checkboxes are checked.

Click "Copy projects into workspace" checkbox and then click "Finish".

By clicking Finish, the projects will be copied into the "Workspace" and then compiled automatically.

Figure 4.9 shows the SoftConsole “Workspace” after these projects have be imported and compiled.



Actel SoftConsole IDE

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Actel devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click
File > **New** > **Managed Make C Project**
and add an initial source file with
File > **New** > **Source File**
Add code to it, and each time you save a file make be run to compile your code.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Quick Start Guide
http://www.actel.com/documents/SoftConsole_QS_UG.pdf
* Online documentation
http://www.actel.com/products/software/softconsole#docs

```
arm-none-eabi-gcc -mthumb -mcpu=cortex-m1 -specs=bare.specs -nostartfiles -nodefaultlibs
-nostdlib -L../ -Xlinker -T ../HAL/CortexM1/GNU/ram-debug.ld -o"MemoryLoaderM1"
./socTbMain.o  ./HAL/CortexM1/GNU/default_handlers.o ./HAL/CortexM1/GNU/sys_boot.o
-lMemoryLoaderM1Lib
'Finished building target: MemoryLoaderM1'
' '
Build complete for project MemoryLoaderM1
```

Figure 4.9 - SoftConsole "Workspace" with new projects added

Debugging the Analog Serial Display sample design

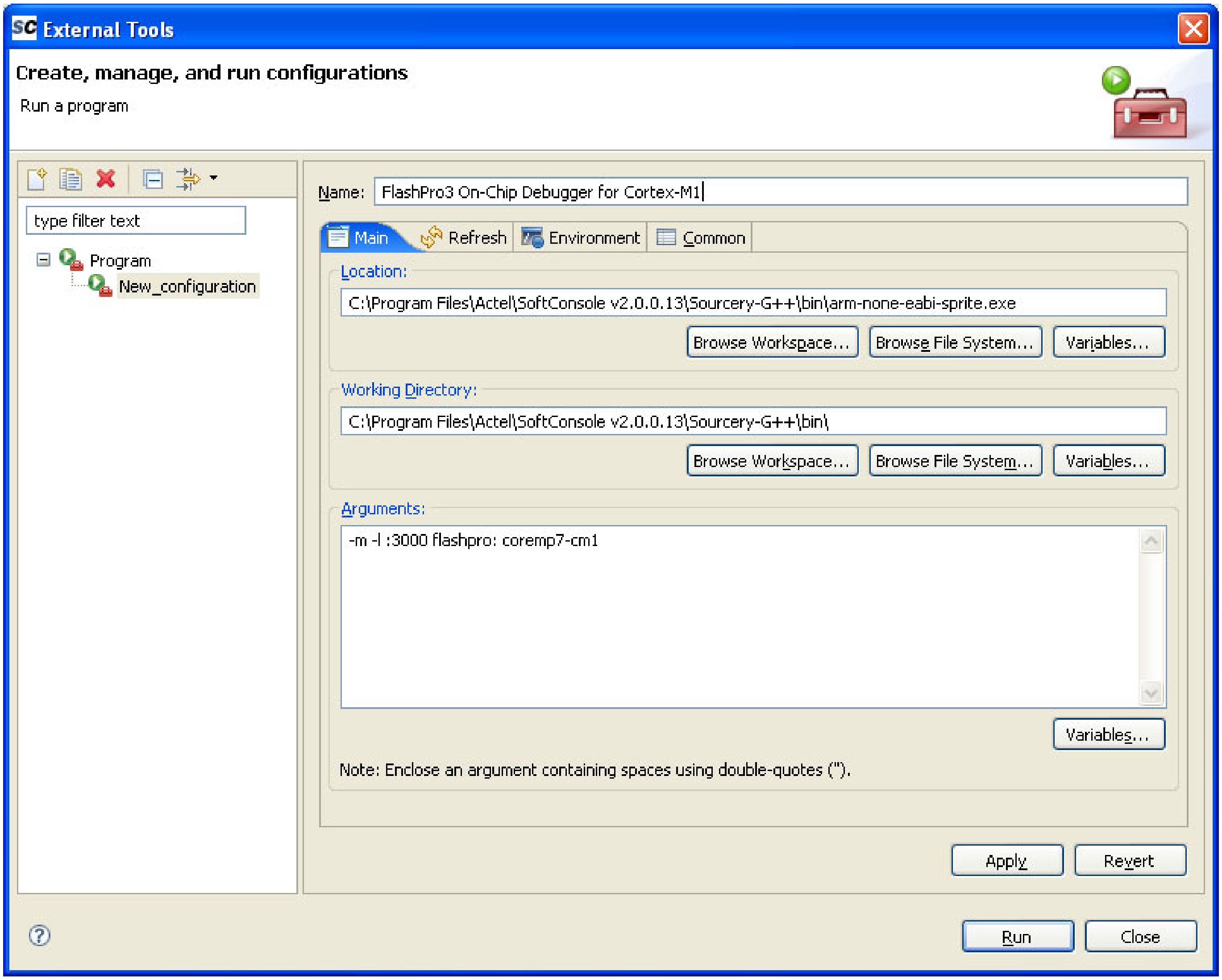
Set up the Cortex-M1 Companion Debugger by selecting Run->External Tools->External Tools… from the menu. Create a new configuration and fill in the fields on the External Tools (Main tab) as shown in Figure 4.10.

Name: FlashPro3 On-Chip Debugger for Cortex-M1

Location: <SoftConsole install folder>\Sourcery-G++\bin\arm-none-eabi-sprite.exe

Working Directory: <SoftConsole install folder>\Sourcery-G++\bin

Arguments: -m -l :3000 flashpro: coremp7-cm1



External Tools

Create, manage, and run configurations

Run a program

Name: FlashPro3 On-Chip Debugger for Cortex-M1

Location: C:\Program Files\Actel\SoftConsole v2.0.0.13\Sourcery-G++\bin\arm-none-eabi-sprite.exe

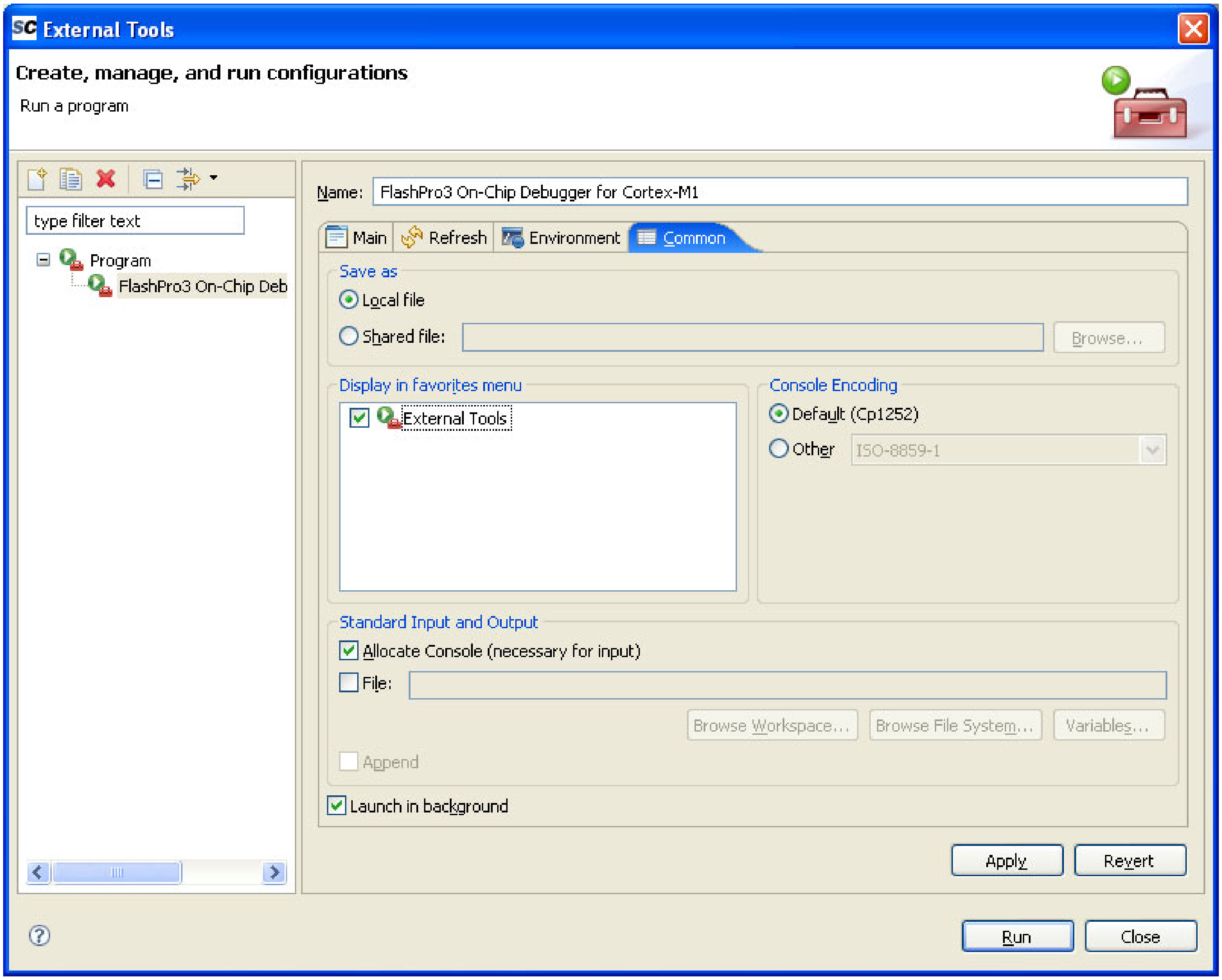
Working Directory: C:\Program Files\Actel\SoftConsole v2.0.0.13\Sourcery-G++\bin\

Arguments: -m -l :3000 flashpro: coremp7-cm1

Note: Enclose an argument containing spaces using double-quotes (").

Figure 4.10 - Companion Debugger External Tools Window

Click the “Common” tab and click the “External Tools” checkbox in the “Display in favorites menu” section as shown in Figure 4.11. This creates a shortcut to run the configured “FlashPro3 On-Chip Debugger for Cortex-M1” directly from the Run->External Tools menu.



External Tools

Create, manage, and run configurations

Run a program

type filter text

⊟ Program

FlashPro3 On-Chip Deb

Name: FlashPro3 On-Chip Debugger for Cortex-M1

Save as

● Local file

○ Shared file:

Display in favorites menu

☑ External Tools

Console Encoding

● Default (Cp1252)

○ Other ISO-8859-1

Standard Input and Output

☑ Allocate Console (necessary for input)

☐ File:

☐ Append

☑ Launch in background

Figure 4.11 – SoftConsole External Tools Window

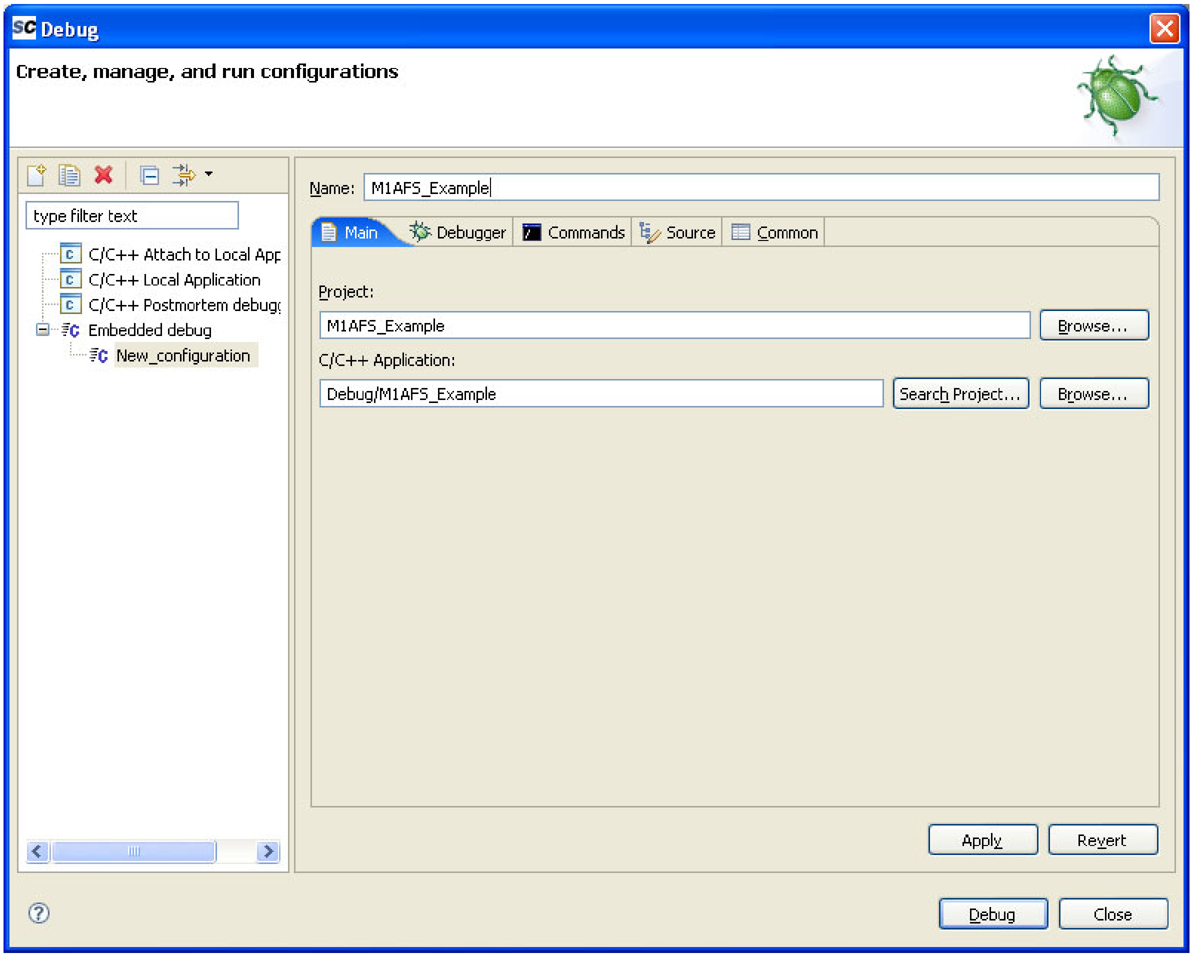
Click Apply to save these settings, then click Run to Run the FlashPro3 Debugger for the Cortex-M1.

Note: This companion debugger must be launched before starting any debugging session. The companion debugger is available from the Run->External Tools->FlashPro3 Debugger for the Cortex-M1 menu.

Configure the Debug session for the Analog Serial Display sample design by selecting Run->Debug… from the menu. Create a new configuration and configure the settings on the "Main" tab as shown in Figure 4.12:

Name: M1AFS_Example

Project: M1AFS_Example C/C++ Application: Debug/ M1AFS_Example



Debug

Create, manage, and run configurations

Name: M1AFS_Example

Project:
M1AFS_Example

C/C++ Application:
Debug/M1AFS_Example

Figure 4.12 SoftConsole Debug Window

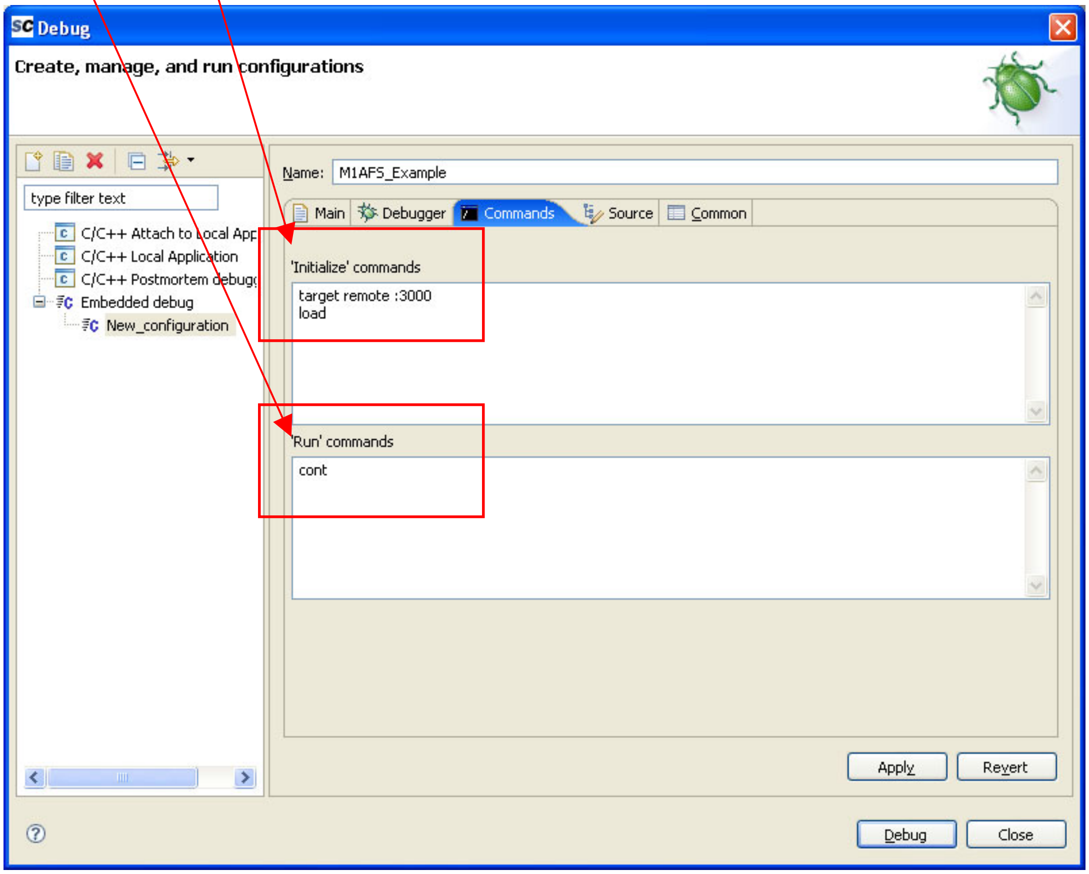
Click the “Commands” tab, and configure the settings as shown in Figure 4.13:

‘Initialize’ commands:

```
target remote :3000
load
```

‘Run’ commands:

```
cont
```



Name: M1AFS_Example

'Initialize' commands

```
target remote :3000
load
```

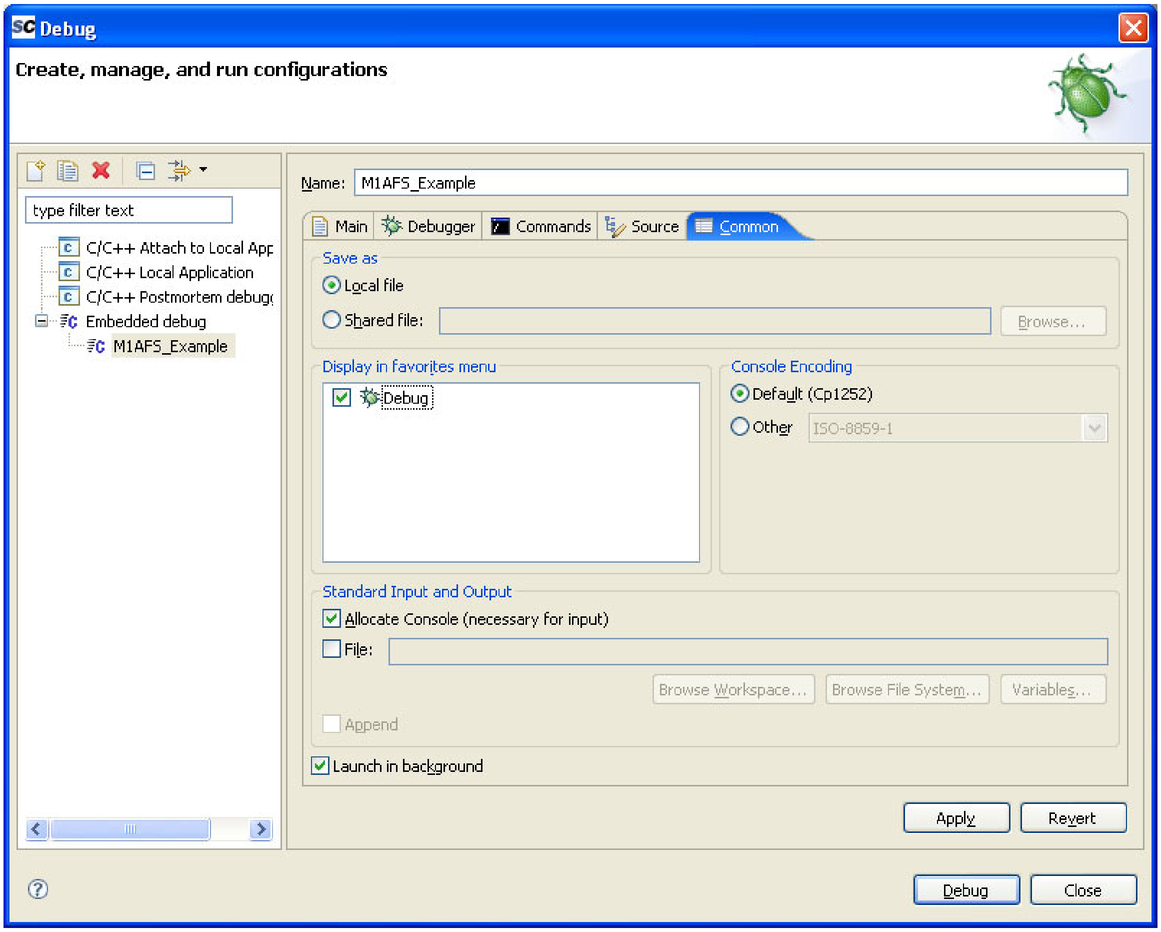
'Run' commands

```
cont
```

Figure 4.13 – SoftConsole Debug Window – Commands Tab

Click the “Common” tab, and configure the settings as shown in Figure 4.14:

Ensure the “Debug” checkbox in the “Display in favorites menu” section is checked.



Name: M1AFS_Example

Save as
● Local file
○ Shared file:

Display in favorites menu
☑ Debug

Console Encoding
● Default (Cp1252)
○ Other ISO-8859-1

Standard Input and Output
☑ Allocate Console (necessary for input)
☐ File:
☐ Append

☑ Launch in background

Figure 4.14 – SoftConsole Debug Window – Common Tab

Click Apply to save these settings, then click Debug to start the debugging session for the Analog Serial Display sample design. Note that the companion debugger must already be running before starting any debugging session. The M1AFS_Example debug session is available from the Run->Debug menu.

Now open a HyperTerminal window. HyperTerminal is a serial communications program that is included in MicroSoft Windows installations under Start -> Programs -> Accessories -> Communications -> HyperTerminal. For the COM port, use the same one that was selected above. Communications settings should be:

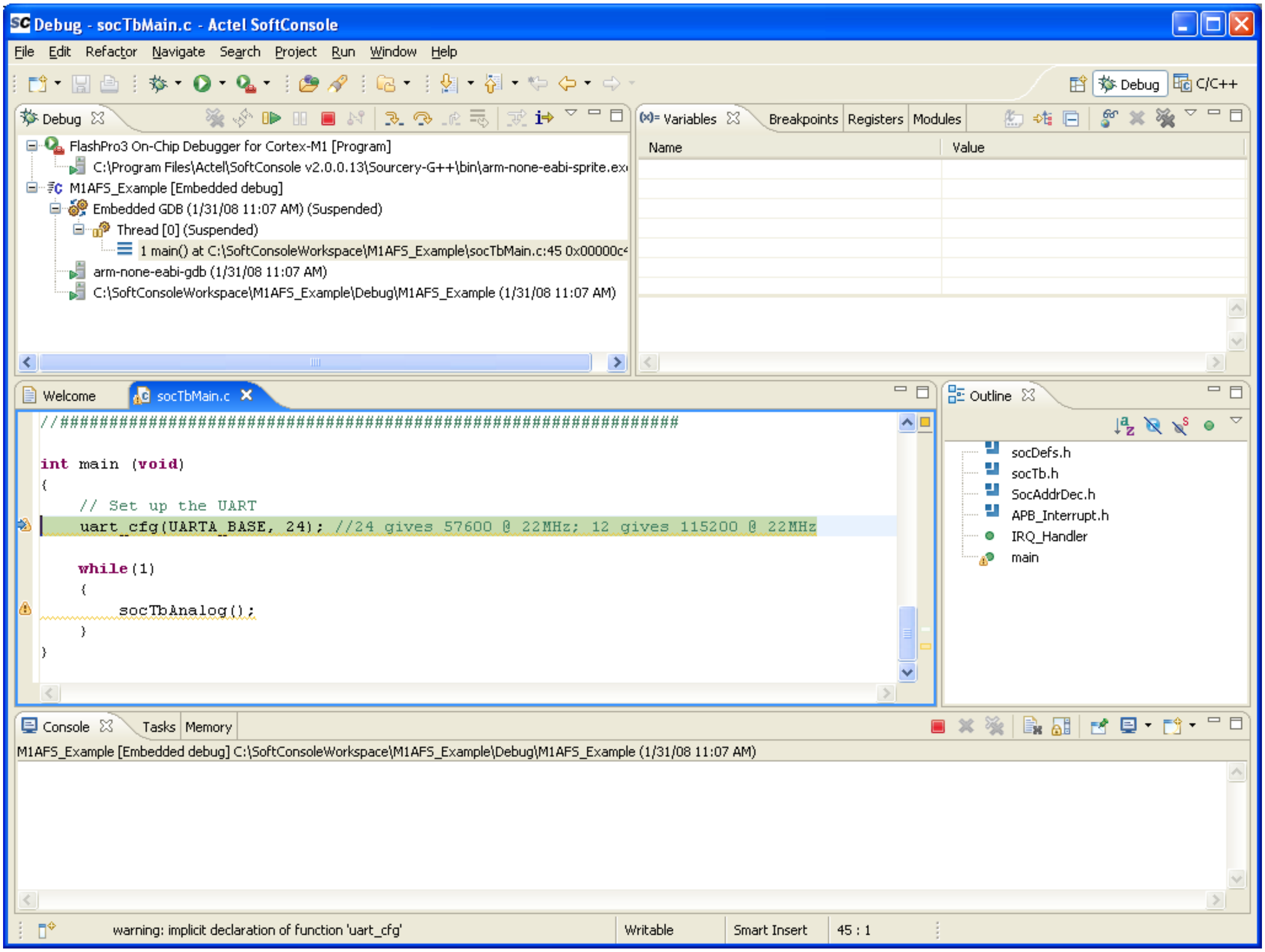
Bits per Second: 57600

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None



FlashPro3 On-Chip Debugger for Cortex-M1 [Program]
C:\Program Files\Actel\SoftConsole v2.0.0.13\Sourcery-G++\bin\arm-none-eabi-sprite.ex
M1AFS_Example [Embedded debug]
Embedded GDB (1/31/08 11:07 AM) (Suspended)
Thread [0] (Suspended)
1 main() at C:\SoftConsoleWorkspace\M1AFS_Example\socTbMain.c:45 0x00000c
arm-none-eabi-gdb (1/31/08 11:07 AM)
C:\SoftConsoleWorkspace\M1AFS_Example\Debug\M1AFS_Example (1/31/08 11:07 AM)

```
//################################################################

int main (void)
{
    // Set up the UART
    uart_cfg(UARTA_BASE, 24); //24 gives 57600 @ 22MHz; 12 gives 115200 @ 22MHz

    while(1)
    {
        socTbAnalog();
    }
}
```

socDefs.h
socTb.h
SocAddrDec.h
APB_Interrupt.h
IRQ_Handler
main

M1AFS_Example [Embedded debug] C:\SoftConsoleWorkspace\M1AFS_Example\Debug\M1AFS_Example (1/31/08 11:07 AM)

warning: implicit declaration of function 'uart_cfg'

Figure 4.15 – SoftConsole Main Debug Screen

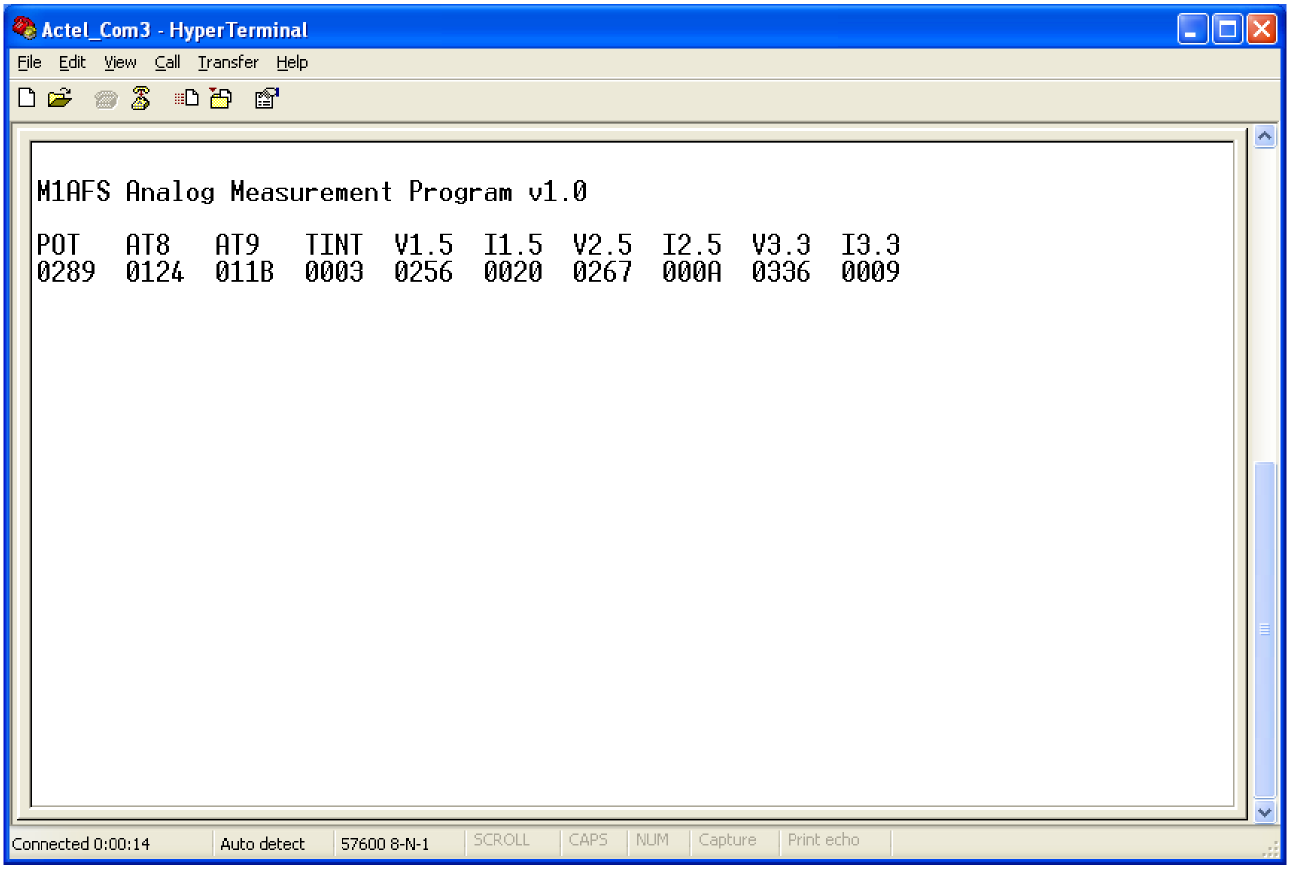
Program execution stops automatically at the beginning of the "main()" function. It is now possible to set breakpoints, single-step through the program or simply run the program.

The Analog Serial Display sample design program has a forever loop inside main() function which executes the following:

- takes analog measurements on several A/D channels and
- then formats and outputs the measurements to the UART.

This loop is executed as fast as possible.

Ensure the USB Serial cable at J2 is connected to the PC. Reference the "User Tests" section in this user guide for determining which COM port to use. Figure 4.16 shows a screen capture of the Analog Serial Display in HyperTerminal.



```
M1AFS Analog Measurement Program v1.0

POT   AT8   AT9   TINT  V1.5  I1.5  V2.5  I2.5  V3.3  I3.3
0289  0124  011B  0003  0256  0020  0267  000A  0336  0009
```

Connected 0:00:14 | Auto detect | 57600 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

Figure 4.16 – Analog Serial Display in HyperTerminal

The following table describes the measurements shown above.

Measurement Label	Analog Signal	Description
POT	AV8	Vary potentiometer R6 to change the POT value. Full scale voltage is 5V.
AT8	AT8	Temperature sensor Q4 on board.
AT9	AT9	Temperature sensor Q3 on board.
TINT	internal	Temperature sensor internal to FPGA.
V1.5	AV7	Voltage measurement for selected (JP8) internal or external 1.5V power supply.
I1.5	AC7	Current measurement for 1.5V power supply.
V2.5	AV9	Voltage measurement for selected (JP8) internal or external 2.5V power supply.
I2.5	AC9	Current measurement for 2.5V power supply.
V3.3	AV6	Voltage measurement for selected (JP8) internal or external 3.3V power supply.
I3.3	AC6	Current measurement for 3.3V power supply.

Table 5.1 – Analog Measurements

For further exploration of this simple software application please see the socTbTests.c and APB_Analog.c files that are included in the SoftConsole project. Note that the P1 header is designed to accept an analog daughter card. The function socTbAnalog can easily be modified to take measurements on any of the signals on the P1 header.

To exit, close HyperTerminal and terminate the SoftConsole debugging session.

Using the Memory Loader Utility

Ensure that the Cortex-M1 companion debugger is running. The companion debugger is available from the Run->External Tools->FlashPro3 Debugger for the Cortex-M1 menu.

Create a SoftConsole debug session similar to the one created for the Analog Serial Display sample design by selecting Run->Debug… from the menu. Create a new configuration and configure the settings on the "Main" tab as shown in Figure 4.17:

Name: MemoryLoaderM1

Project: MemoryLoaderM1

C/C++ Application: Debug/MemoryLoaderM1

Click the "Commands" tab and configure the settings as follows:

'Initialize' commands:

target remote :3000

load

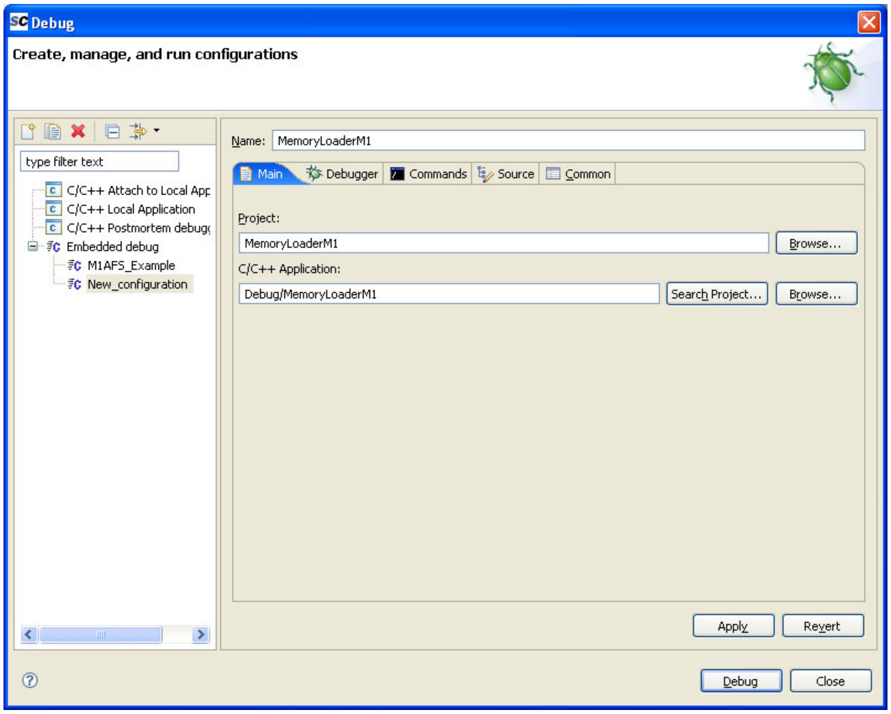
'Run' commands:

cont

Click the "Common" tab and configure the settings as follows:

Ensure the "Debug" checkbox in the "Display in favorites menu" section is checked.

Click Apply to save these settings, then click Debug to start the debugging session for the MemoryLoader example. Note that the companion debugger must already be running before starting any debugging session. The MemoryLoaderM1 debug session is available from the Run->Debug menu.



Name: MemoryLoaderM1

Project: MemoryLoaderM1

C/C++ Application: Debug/MemoryLoaderM1

Figure 5-17 – Memory Loader Utility SoftConsole setup

Run the MemoryLoaderM1 embedded application by selecting Run->Resume from the menu. The MemoryLoaderM1 application configures the UART in the example system, then enters a forever loop listening for commands from the MemoryLoader PC application.

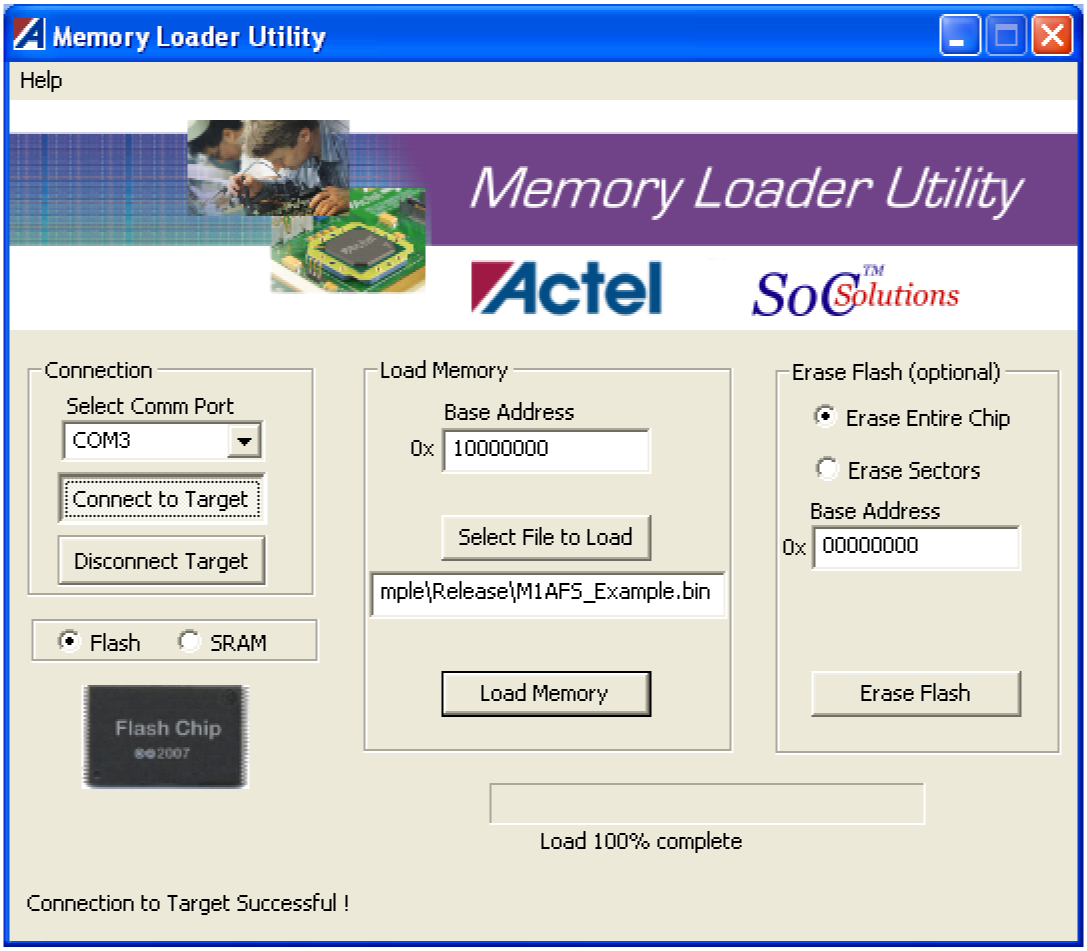
Now we are ready to execute the MemoryLoader PC Application by running Memory_Loader.exe from the <install folder>\Sample Design\PC_Software\MemoryLoader folder. Figure 4.18 shows the Memory_Loader PC Application.

Ensure the USB Serial cable at J2 is connected to the PC. Reference the "User Tests" section in this user guide for determining which COM port to use. Select the appropriate COM port from the drop-down box, and click the "Connect to Target" button to connect to the M1AFS board. If the connection to the M1AFS board is successful, a status message will appear in the lower left of the MemoryLoader Utility.

In the Load Memory section, set the Base Address to 10000000 (Hex), and click the "Select File to Load" button. Browse to the file M1AFS_Example.bin located in the <SoftConsole workspace folder>\ M1AFS_Example\Release folder.

Note: M1AFS_Example.bin was created by a post-build step for the Release version of the M1AFS_Example SoftConsole project.

Click the "Load Memory" button to erase and load the external Flash with the M1AFS_Example image for the Cortex-M1.



Memory Loader Utility

Connection

Select Comm Port: COM3

● Flash ○ SRAM

Load Memory

Base Address 0x 10000000

mple\Release\M1AFS_Example.bin

Erase Flash (optional)

● Erase Entire Chip

○ Erase Sectors

Base Address 0x 00000000

Load 100% complete

Connection to Target Successful !

Figure 4.18 – Memory Loader Application

For more information about the Memory Loader Utility for the PC, select Help->Help from the menu.

Close the Memory Loader Utility, terminate the SoftConsole debugging session, and terminate the companion debugger for the Cortex-M1.

Rebooting the board to run the Analog Serial Display software from Flash

The Release version of the Analog Serial Display sample design software has been loaded into external Flash. To run the M1AFS_Example application, toggle SW1 #9 to the ON position. This remaps external Flash to memory address 0x00000000. Then press and release the system reset button at SW3 to start program execution. Now observe the analog measurements update as in the debug session for the M1AFS_Example example.

Restoring factory default settings

The completion of the procedure documented in this chapter restores factory default settings for the M1AFS FPGA and the on-board FLASH.

FG484 Package for the M1AFS FPGAs

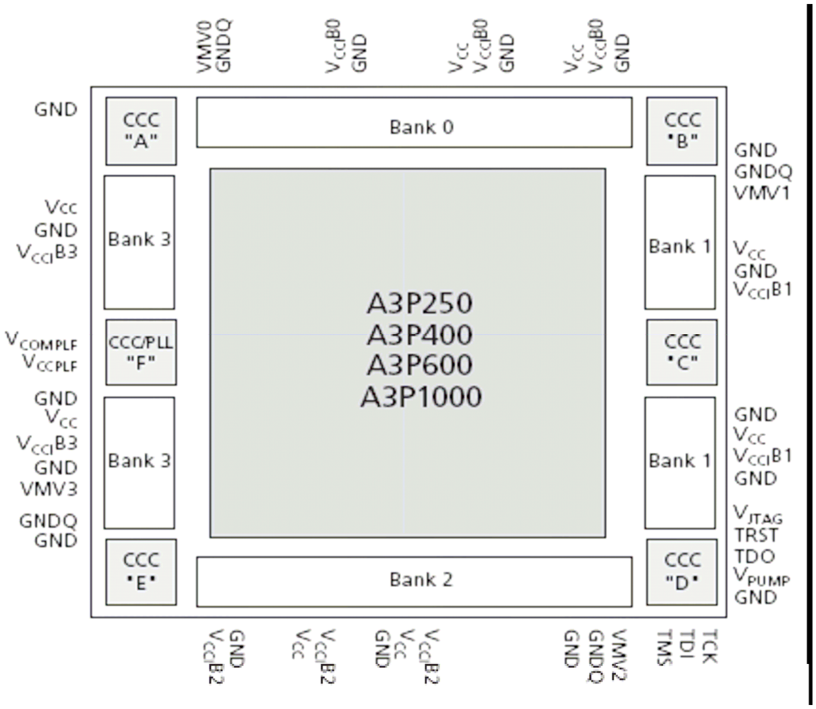


Figure A.1- M1AFS600 Layout

Figure A.2 shows the bottom view of the 484-Pin FBGA.

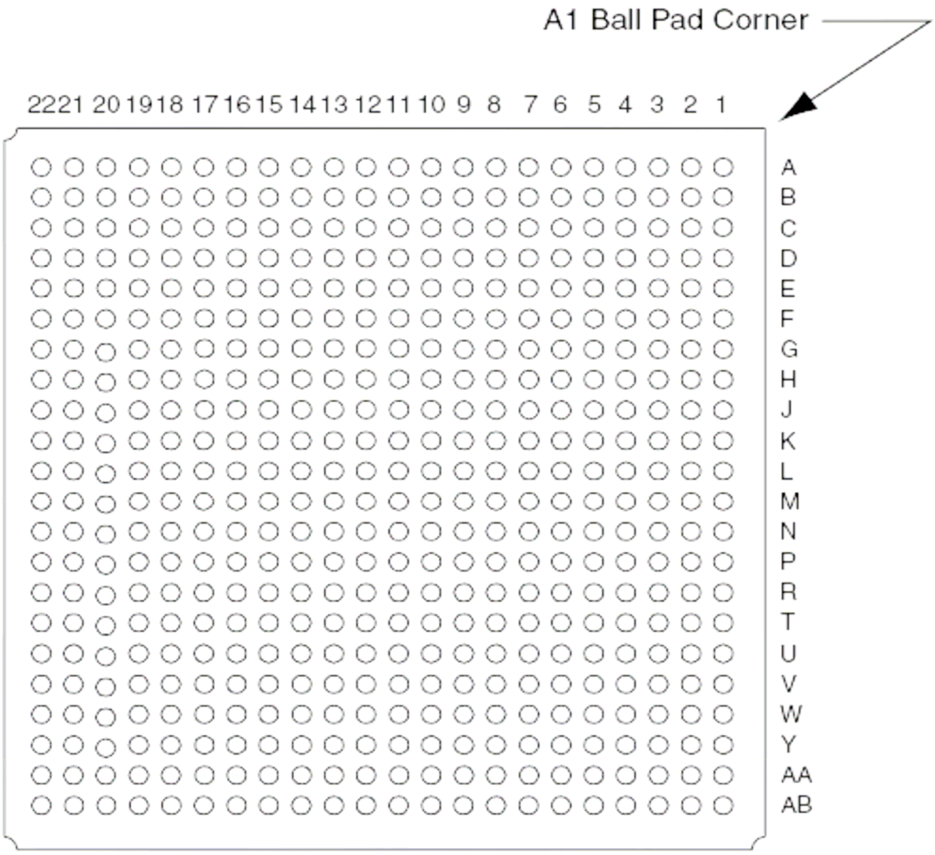


Figure A.2- 484 Pin FBGA Package

Note: For package manufacturing and environmental information, visit the Packaging Solutions page: http://www.actel.com/products/solutions/package/default.aspx.

Due to the comprehensive and flexible nature of M1AFS FPGA device user I / Os, a naming scheme is used to show the details of the I / O. The name identifies to which I / O bank it belongs, as well as the pairing and pin polarity for differential I / Os.

I / O Nomenclature	=	Gmn / IOuxwBy
Gmn is only used for I / Os that also have CCC access – i.e., global pins.		
G	=	Global
m	=	Global pin location associated with each CCC on the device: A (northwest corner), B (northeast corner), C (east middle), D (southeast corner), E (southwest corner), and F (west middle)
n	=	Global input MUX and pin number of the associated Global location m, either A0, A1, A2, B0, B1, B2, C0, C1, or C2
u	=	I / O pair number in the bank, starting at 00 from the northwest I / O bank in a clockwise direction
x	=	P (Positive) or N (Negative) for differential pairs, or R (Regular – single-ended) for the I / Os that support single-ended and voltage-referenced I / O standards only. U (Positive-LVDS only) or V (Negative-LVDS only) restrict the I / O differential pair from being selected as LVPECL pair.
w	=	D (Differential Pair) or P (Pair) or S (Single-Ended). D (Differential Pair) if both members of the pair are bonded out to adjacent pins or are separated only by one GND or NC pin; P (Pair) if both members of the pair are bonded out but do not meet the adjacency requirement; or S (Single-Ended) if the I / O pair is not bonded out. For Differential (D) pairs, adjacency for ball grid packages means only vertical or horizontal. Diagonal adjacency does not meet the requirements for a true differential pair.
B	=	Bank
y	=	Bank number [0..3]. Bank number starting at 0 from the northwest I / O bank in a clockwise direction

Table A.1 - I/O Nomenclature

The following tables show the FPGA pin connections to the M1AFS Development Board signals and the Sample Design signals. The first table shows the digital signals and the second shows the analog signals.

FPGA Ball	Sample Design Signal	FPGA Signal	Schematic Signal	I / O	Dev. Kit Function
A8	pbRstN	IO10PDB0V1	PBRESET_N	I	Push Button Reset
A10	poRstN	IO14PDB0V1	PORESET_N	I	Power on Reset
G10	flashRstN	IO04PDB0V0	FLASH_RST_N	0	Reset to Flash Chips
A3	sysClk	GAA1/IO01PDB0V0	OSC_CLK	I	System Clock
A5	ICE_TCK	GAB1/IO02PDB0V0	PROC_TCK	I	JTAG Clock
C10	ICE_TDI	IO11PDB0V1	PROC_TDI	I	JTAG Data In
D10	ICE_TMS	IO11NDB0V1	PROC_TMS	I	JTAG Mode Select
F10	ICE_TDO	IO08PDB0V1	PROC_TDO	O	JTAG Data Out
A9	ICE_TRTCK	IO14NDB0V1	PROC_RTCK	I	JTAG sync
M22	rs232Atx	GCB1/IO44PDB2V0	RS232_TX	O	UART Transmit
M21	rs232Arx	GCB0/IO44NDB2V0	RS232_RX	I	UART Receive
B3	extAddr[2]	GAA0/IO01NDB0V0	MEM_ADDR2	O	SRAM / FLASH Address
J3	extAddr[3]	GFB2/IO74PDB4V0	MEM_ADDR3	O	SRAM / FLASH Address
K4	extAddr[4]	IO75NDB4V0	MEM_ADDR4	O	SRAM / FLASH Address
K3	extAddr[5]	IO74NDB4V0	MEM_ADDR5	O	SRAM / FLASH Address
P5	extAddr[6]	IO65NDB4V0	MEM_ADDR6	O	SRAM / FLASH Address
P4	extAddr[7]	IO65PDB4V0	MEM_ADDR7	O	SRAM / FLASH Address
P3	extAddr[8]	IO68NDB4V0	MEM_ADDR8	O	SRAM / FLASH Address
L7	extAddr[9]	GFB1/IO71PDB4V0	MEM_ADDR9	O	SRAM / FLASH Address
M7	extAddr[10]	GFB0/IO71NDB4V0	MEM_ADDR10	O	SRAM / FLASH Address
J4	extAddr[11]	GFA2/IO75PDB4V0	MEM_ADDR11	O	SRAM / FLASH Address
N3	extAddr[12]	IO68PDB4V0	MEM_ADDR12	O	SRAM / FLASH Address
P1	extAddr[13]	GFA1/IO70PDB4V0	MEM_ADDR13	O	SRAM / FLASH Address
P2	extAddr[14]	GFA0/IO70NDB4V0	MEM_ADDR14	O	SRAM / FLASH Address
R1	extAddr[15]	IO69PDB4V0	MEM_ADDR15	O	SRAM / FLASH Address
R2	extAddr[16]	IO69NDB4V0	MEM_ADDR16	O	SRAM / FLASH Address
R4	extAddr[17]	IO64PDB4V0	MEM_ADDR17	O	SRAM / FLASH Address
R5	extAddr[18]	IO64NDB4V0	MEM_ADDR18	O	SRAM / FLASH Address
T3	extAddr[19]	IO66PDB4V0	MEM_ADDR19	O	SRAM / FLASH Address
T4	extAddr[20]	IO66NDB4V0	MEM_ADDR20	O	SRAM / FLASH Address
U4	extAddr[21]	GEC0/IO63NDB4V0	MEM_ADDR21	O	SRAM / FLASH Address
W5	extAddr[22]	IO59NDB4V0	MEM_ADDR22	O	SRAM / FLASH Address
W4	extAddr[23]	GEB2/IO59PDB4V0	MEM_ADDR23	O	SRAM / FLASH Address
V4	extAddr[24]	GEA1/IO61PDB4V0	MEM_ADDR24	O	SRAM / FLASH Address
V5	extAddr[25]	GEA0/IO61NDB4V0	MEM_ADDR25	O	SRAM / FLASH Address
G3	extData[0]	IO78NDB4V0	MEM_DATA0	I / O	SRAM / FLASH Data
G4	extData[1]	IO78PDB4V0	MEM_DATA1	I / O	SRAM / FLASH Data
H4	extData[2]	IO79NDB4V0	MEM_DATA2	I / O	SRAM / FLASH Data
H5	extData[3]	IO79PDB4V0	MEM_DATA3	I / O	SRAM / FLASH Data
F1	extData[4]	IO80NDB4V0	MEM_DATA4	I / O	SRAM / FLASH Data
F2	extData[5]	IO80PDB4V0	MEM_DATA5	I / O	SRAM / FLASH Data
E1	extData[6]	IO81NDB4V0	MEM_DATA6	I / O	SRAM / FLASH Data
E2	extData[7]	IO81PDB4V0	MEM_DATA7	I / O	SRAM / FLASH Data
D1	extData[8]	IO82NDB4V0	MEM_DATA8	I / O	SRAM / FLASH Data

FPGA Ball	Sample Design Signal	FPGA Signal	Schematic Signal	I / O	Dev. Kit Function
C1	extData[9]	IO82PDB4V0	MEM_DATA9	I / O	SRAM / FLASH Data
D3	extData[10]	IO83NDB4V0	MEM_DATA10	I / O	SRAM / FLASH Data
F4	extData[11]	IO84NDB4V0	MEM_DATA11	I / O	SRAM / FLASH Data
E5	extData[12]	IO85NDB4V0	MEM_DATA12	I / O	SRAM / FLASH Data
D5	extData[13]	GAA2/IO85PDB4V0	MEM_DATA13	I / O	SRAM / FLASH Data
E4	extData[14]	GAB2/IO84PDB4V0	MEM_DATA14	I / O	SRAM / FLASH Data
D4	extData[15]	GAC2/IO83PDB4V0	MEM_DATA15	I / O	SRAM / FLASH Data
D6	extData[16]	GAC0/IO03NDB0V0	MEM_DATA16	I / O	SRAM / FLASH Data
D7	extData[17]	GAC1/IO03PDB0V0	MEM_DATA17	I / O	SRAM / FLASH Data
C3	extData[18]	IO00NDB0V0	MEM_DATA18	I / O	SRAM / FLASH Data
C4	extData[19]	IO00PDB0V0	MEM_DATA19	I / O	SRAM / FLASH Data
G9	extData[20]	IO04NDB0V0	MEM_DATA20	I / O	SRAM / FLASH Data
B5	extData[21]	IO05NDB0V0	MEM_DATA21	I / O	SRAM / FLASH Data
B6	extData[22]	IO05PDB0V0	MEM_DATA22	I / O	SRAM / FLASH Data
C6	extData[23]	IO06NDB0V0	MEM_DATA23	I / O	SRAM / FLASH Data
C7	extData[24]	IO06PDB0V0	MEM_DATA24	I / O	SRAM / FLASH Data
A6	extData[25]	IO07NDB0V1	MEM_DATA25	I / O	SRAM / FLASH Data
A7	extData[26]	IO07PDB0V1	MEM_DATA26	I / O	SRAM / FLASH Data
D8	extData[27]	IO09NDB0V1	MEM_DATA27	I / O	SRAM / FLASH Data
D9	extData[28]	IO09PDB0V1	MEM_DATA28	I / O	SRAM / FLASH Data
B8	extData[29]	IO10NDB0V1	MEM_DATA29	I / O	SRAM / FLASH Data
C9	extData[30]	IO13NDB0V1	MEM_DATA30	I / O	SRAM / FLASH Data
B9	extData[31]	IO13PDB0V1	MEM_DATA31	I / O	SRAM / FLASH Data
G11	flashHiCeN	IO12NDB0V1	FLASH_HCE_N	O	Flash Chip Enable (high chip)
G12	flashLoCeN	IO12PDB0V1	FLASH_LCE_N	O	Flash Chip Enable (low chip)
A4	flashWeN	GAB0/IO02NDB0V0	FLASH_WE_N	O	Flash Write Enable
F9	flashOeN	IO08NDB0V1	FLASH_OE_N	O	Flash Output Enable
E1	sramCeN	GFC2/IO73PDB4V0	SRAM_CE_N	O	SRAM Chip Enable
J2	sramBsN[0]	IO76NDB4V0	SRBS0_N	O	SRAM Byte Select 0
H2	sramBsN[1]	IO76PDB4V0	SRBS1_N	O	SRAM Byte Select 1
M5	sramBsN[2]	GFC0/IO72NDB4V0	SRBS2_N	O	SRAM Byte Select 2
L5	sramBsN[3]	GFC1/IO72PDB4V0	SRBS3_N	O	SRAM Byte Select 3
H1	sramWeN	IO77NDB4V0	SRAM_WE_N	O	SRAM Write Enable
G1	sramOeN	IO77PDB4V0	SRAM_OE_N	O	SRAM Output Enable
U1	ledOut[0]	IO67PDB4V0	LED0	O	Drives LED 0
U2	ledOut[1]	IO67NDB4V0	LED1	O	Drives LED 1
Y2	ledOut[2]	IO60NDB4V0	LED2	O	Drives LED 2
Y4	ledOut[3]	IO58NDB4V0	LED3	O	Drives LED 3
Y1	ledOut[4]	GEC2/IO60PDB4V0	LED4	O	Drives LED 4
U3	ledOut[5]	GEC1/IO63PDB4V0	LED5	O	Drives LED 5
V1	ledOut[6]	GEB1/IO62PDB4V0	LED6	O	Drives LED 6
V2	ledOut[7]	GEB0/IO62NDB4V0	LED7	O	Drives LED 7
Y3	ledOut[8]	GEA2/IO58PDB4V0	LED8	O	Drives LED 8
M16	ledOut[9]	IO48NDB2V0	LED9	O	Drives LED 9
P19	switchIn[0]	GDB1/IO53PDB2V0	SWITCH0	I	Switch Input 0
R19	switchIn[1]	GDB0/IO53NDB2V0	SWITCH1	I	Switch Input 1
T20	switchIn[2]	GDA2/IO55PDB2V0	SWITCH2	I	Switch Input 2
P18	switchIn[3]	GDA1/IO54PDB2V0	SWITCH3	I	Switch Input 3

FPGA Ball	Sample Design Signal	FPGA Signal	Schematic Signal	I / O	Dev. Kit Function
R18	switchIn[4]	GDA0/IO54NDB2V0	SWITCH4	I	Switch Input 4
U19	switchIn[5]	IO57NDB2V0	SWITCH5	I	Switch Input 5
P16	switchIn[6]	IO56NDB2V0	SWITCH6	I	Switch Input 6
T19	switchIn[7]	IO55NDB2V0	SWITCH7	I	Switch Input 7
P21	switchIn[8]	IO51PDB2V0	SWITCH8	I	Switch Input 8
P20	switchIn[9]	IO51NDB2V0	SWITCH9	I	Switch Input 9
C20	gpio0Out[0]	GBA2/IO30PDB2V0	GPIOA_0	O	GPIO Output
C22	gpio0Out[1]	GBB2/IO31PDB2V	GPIOA_1	O	GPIO Output
D20	gpio0Out[2]	IO30NDB2V0	GPIOA_2	O	GPIO Output
D22	gpio0Out[3]	IO31NDB2V0	GPIOA_3	O	GPIO Output
E21	gpio0Out[4]	IO32NDB2V0	GPIOA_4	O	GPIO Output
F19	gpio0Out[5]	IO33NDB2V0	GPIOA_5	O	GPIO Output
E19	gpio0Out[6]	IO33PDB2V0	GPIOA_6	O	GPIO Output
F21	gpio0Out[7]	IO34NDB2V0	GPIOA_7	O	GPIO Output
F20	gpio0Out[8]	IO34PDB2V0	GPIOA_8	O	GPIO Output
G22	gpio0Out[9]	IO35NDB2V0	GPIOA_9	O	GPIO Output
G20	gpio0Out[10]	IO36NDB2V0	GPIOA_10	O	GPIO Output
G19	gpio0Out[11]	IO36PDB2V0	GPIOA_11	O	GPIO Output
H21	gpio0Out[12]	IO37NDB2V0	GPIOA_12	O	GPIO Output
H22	gpio0Out[13]	IO37PDB2V0	GPIOA_13	O	GPIO Output
J18	gpio0Out[14]	IO38NDB2V0	GPIOA_14	O	GPIO Output
H18	gpio0Out[15]	IO38PDB2V0	GPIOA_15	O	GPIO Output
J19	gpio0Out[16]	IO39NDB2V0	GPIOA_16	O	GPIO Output
K16	gpio0Out[17]	IO40NDB2V0	GPIOA_17	O	GPIO Output
M19	gpio0Out[18]	GCA0/IO45NDB2V0	GPIOA_18	O	GPIO Output
L19	gpio0Out[19]	GCA1/IO45PDB2V0	GPIOA_19	O	GPIO Output
H19	gpio0Out[20]	GCA2/IO39PDB2V0	GPIOA_20	O	GPIO Output
J16	gpio0Out[21]	GCB2/IO40PDB2V0	GPIOA_21	O	GPIO Output
L21	gpio0Out[22]	GCC0/IO43NDB2V0	GPIOA_22	O	GPIO Output
L22	gpio0Out[23]	GCC1/IO43PDB2V0	GPIOA_23	O	GPIO Output
J20	gpio0Out[24]	GCC2/IO41PDB2V0	GPIOA_24	O	GPIO Output
K20	gpio0Out[25]	IO41NDB2V0	GPIOA_25	O	GPIO Output
K22	gpio0Out[26]	IO42NDB2V0	GPIOA_26	O	GPIO Output
J22	gpio0Out[27]	IO41PDB2V0	GPIOA_27	O	GPIO Output
M18	gpio0Out[28]	IO46NDB2V0	GPIOA_28	O	GPIO Output
L18	gpio0Out[29]	IO46PDB2V0	GPIOA_29	O	GPIO Output
N19	gpio0Out[30]	IO47NDB2V0	GPIOA_30	O	GPIO Output
R22	gpio0Out[31]	IO47PDB2V0	DIFFA1P	O	GPIO Output
N20	gpio0In[0]	IO47PDB2V0	GPIOA_31	I	GPIO Input
E12	gpio0In[1]	IO15PDB1V0	B1_15P	I	GPIO Input
E11	gpio0In[2]	IO15NDB1V0	B1_15N	I	GPIO Input
D12	gpio0In[3]	IO16PDB1V0	B1_16P	I	GPIO Input
D11	gpio0In[4]	IO16NDB1V0	B1_16N	I	GPIO Input
A11	gpio0In[5]	IO17PDB1V0	B1_17P	I	GPIO Input
B11	gpio0In[6]	IO17NDB1V0	B1_17N	I	GPIO Input
A12	gpio0In[7]	IO18PDB1V0	B1_18P	I	GPIO Input
B12	gpio0In[8]	IO18NDB1V0	B1_18N	I	GPIO Input
A14	gpio0In[9]	IO19PDB1V0	B1_19P	I	GPIO Input

FPGA Ball	Sample Design Signal	FPGA Signal	Schematic Signal	I / O	Dev. Kit Function
A13	gpio0In[10]	IO19NDB1V0	B1_19N	I	GPIO Input
C14	gpio0In[11]	IO20PDB1V0	B1_20P	I	GPIO Input
C13	gpio0In[12]	IO20NDB1V0	B1_20N	I	GPIO Input
B15	gpio0In[13]	IO21PDB1V0	B1_21P	I	GPIO Input
B14	gpio0In[14]	IO21NDB1V0	B1_21N	I	GPIO Input
F14	gpio0In[15]	IO22PDB1V0	B1_22P	I	GPIO Input
F13	gpio0In[16]	IO22NDB1V0	B1_22N	I	GPIO Input
D15	gpio0In[17]	IO23PDB1V0	B1_23P	I	GPIO Input
D14	gpio0In[18]	IO23NDB1V0	B1_23N	I	GPIO Input
A16	gpio0In[19]	IO24PDB1V0	B1_24P	I	GPIO Input
A15	gpio0In[20]	IO24NDB1V0	B1_24N	I	GPIO Input
A19	gpio0In[21]	IO25PDB1V0	B1_25P	I	GPIO Input
C16	gpio0In[22]	IO25NDB1V0	B1_25N	I	GPIO Input
A20	gpio0In[23]	IO29PDB1V0	B1_29P	I	GPIO Input
D16	gpio0In[24]	IO29NDB1V0	B1_29N	I	GPIO Input
B18	gpio0In[25]	GBA1/IO28PDB1V1	B1_28P_GBA1	I	GPIO Input
A18	gpio0In[26]	GBA0/IO28NDB1V1	B1_28N_GBA0	I	GPIO Input
D17	gpio0In[27]	GBB1/IO27PDB1V1	B1_27P_GBB1	I	GPIO Input
C17	gpio0In[28]	GBB0/IO27NDB1V1	B1_27N_GBB0	I	GPIO Input
B17	gpio0In[29]	GBC1/IO26PDB1V1	B1_26P_GBC1	I	GPIO Input
A17	gpio0In[30]	GBC0/IO26NDB1V1	B1_26N_GBC0	I	GPIO Input
R21	gpio0In[31]	IO50NDB2V0	DIFFA1N	I	GPIO Input
T22	gpio1Out10	GDC1/IO52PDB2V0	DIFFA2P	O	GPIO Output
U22	gpio1In10	GDC0/IO52NDB2V0	DIFFA2N	I	GPIO Input
P22	pwmOut1	IO49NDB2V0	PWMDAC1	O	PWM Output
N22	pwmOut2	IO49RDB2V0	PWMDAC2	O	PWM Output
L16	test[0]	IO48PDB2V0	TEST0	I/O	Test I/O (testpoint on bottom of board)
N16	test[1]	GDB2/IO56PDB2V0	TEST1	I/O	Test I/O (testpoint on bottom of board)
U20	test[2]	GDC2/IO57PDB2V0	TEST2	I/O	Test I/O (testpoint on bottom of board)

Table A.2- M1AFS FPGA Digital Signals

FPGA Ball	Sample Design Signal	FPGA Signal	Schematic Signal	I / O	Dev. Kit Function
W6	AV0	AV0	AV0	I	Analog Voltage Input
Y6	AC0	AC0	AC0	I	Analog Current Sensor Input
AA6	AG0	AG0	AG0	O	Analog Gate Driver Output
AB6	AT0	AT0	AT0	O	Analog Temperature Sensor Output
AB7	ATRTN0	ATRTN0	ATRTN0	I	Analog Temperature Sensor Return
W8	AV1	AV1	AV1	I	Analog Voltage Input
Y8	AC1	AC1	AC1	I	Analog Current Sensor Input
AA8	AG1	AG1	AG1	O	Analog Gate Driver Output
AB8	AT1	AT1	AT1	O	Analog Temperature Sensor Output
W9	AV2	AV2	AV2	I	Analog Voltage Input
Y9	AC2	AC2	AC2	I	Analog Current Sensor Input
AA9	AG2	AG2	AG2	O	Analog Gate Driver Output
AB9	AT2	AT2	AT2	O	Analog Temperature Sensor Output
AB10	ATRTN1	ATRTN1	ATRTN1	I	Analog Temperature Sensor Return
W11	AV3	AV3	AV3	I	Analog Voltage Input
Y11	AC3	AC3	AC3	I	Analog Current Sensor Input
AA11	AG3	AG3	AG3	O	Analog Gate Driver Output
AB11	AT3	AT3	AT3	O	Analog Temperature Sensor Output
T10	AV4	AV4	AV4	I	Analog Voltage Input
U10	AC4	AC4	AC4	I	Analog Current Sensor Input
V10	AG4	AG4	AG4	O	Analog Gate Driver Output
V11	AT4	AT4	AT4	O	Analog Temperature Sensor Output
V12	ATRTN2	ATRTN2	ATRTN2	I	Analog Temperature Sensor Return
T12	AV5	AV5	AV5	I	Analog Voltage Input
T13	AC5	AC5	AC5	I	Analog Current Sensor Input
U13	AG5	AG5	AG5	O	Analog Gate Driver Output
V13	AT5	AT5	AT5	O	Analog Temperature Sensor Output
W12	AV6	AV6	AV6	I	Analog Voltage Input
Y12	AC6	AC6	AC6	I	Analog Current Sensor Input
AA12	AG6	AG6	AG6	O	Analog Gate Driver Output
AB12	AT6	AT6	AT6	O	Analog Temperature Sensor Output
AB13	ATRTN3	ATRTN3	ATRTN3	I	Analog Temperature Sensor Return
W14	AV7	AV7	AV7	I	Analog Voltage Input
Y14	AC7	AC7	AC7	I	Analog Current Sensor Input
AA14	AG7	AG7	AG7	O	Analog Gate Driver Output
AB14	AT7	AT7	AT7	O	Analog Temperature Sensor Output
W15	AV8	AV8	AV8	I	Analog Voltage Input
Y15	AC8	AC8	AC8	I	Analog Current Sensor Input
AA15	AG8	AG8	AG8	O	Analog Gate Driver Output
AB15	AT8	AT8	AT8	O	Analog Temperature Sensor Output
AB16	ATRTN4	ATRTN4	ATRTN4	I	Analog Temperature Sensor Return
W17	AV9	AV9	AV9	I	Analog Voltage Input
Y17	AC9	AC9	AC9	I	Analog Current Sensor Input

AA17	AG9	AG9	AG9	O	Analog Gate Driver Output
AB17	AT9	AT9	AT9	O	Analog Temperature Sensor Output
AA18	N/A	VAREF	N/A	I/O	Internal Reference Voltage output or External Reference Voltage Input
AA5	N/A	PCAP	PCAP	N/A	Positive Terminal of Charge Pump Capacitor
Y5	N/A	NCAP	NCAP	N/A	Negative Terminal of Charge Pump Capacitor
U15	N/A	PUB	PUB	I	Power Up Bar (active low) Push button connection for external momentary switch used to turn on 1.5 V regulator
Y19	N/A	PTBASE	PTBASE	O	Pass Transistor Base The control signal for voltage regulator
AA20	N/A	PTEM	V1P5_INT	I	Pass Transistor Emitter The feedback input of the voltage regulator
M2	XTAL1	XTAL1	XTAL1	I	Input from external crystal circuit
L4	N/A	XTAL2	XTAL2	O	Output from external crystal circuit

Table A.3- M1AFS FPGA Analog Signals

Board Schematics

For detailed Schematic, please refer to the **"M1AFS Schematics.pdf"** in the **"Dev Kit Documentation"** folder.

There are 9 pages to the Schematics, titled as follows:

1. POWER SUPPLIES
2. PWR, CLOCK
3. SRAM & FLASH
4. RESET, TEST, USB
5. TEST CONN'S
6. FPGA
7. FPGA
8. ANALOG
9. FLASHPRO3

The schematic number is SOC-M1AFS-S-002. Please reference this number and the schematic sheet when contacting support@socsolutions.com for M1AFS Development Board and schematic support.

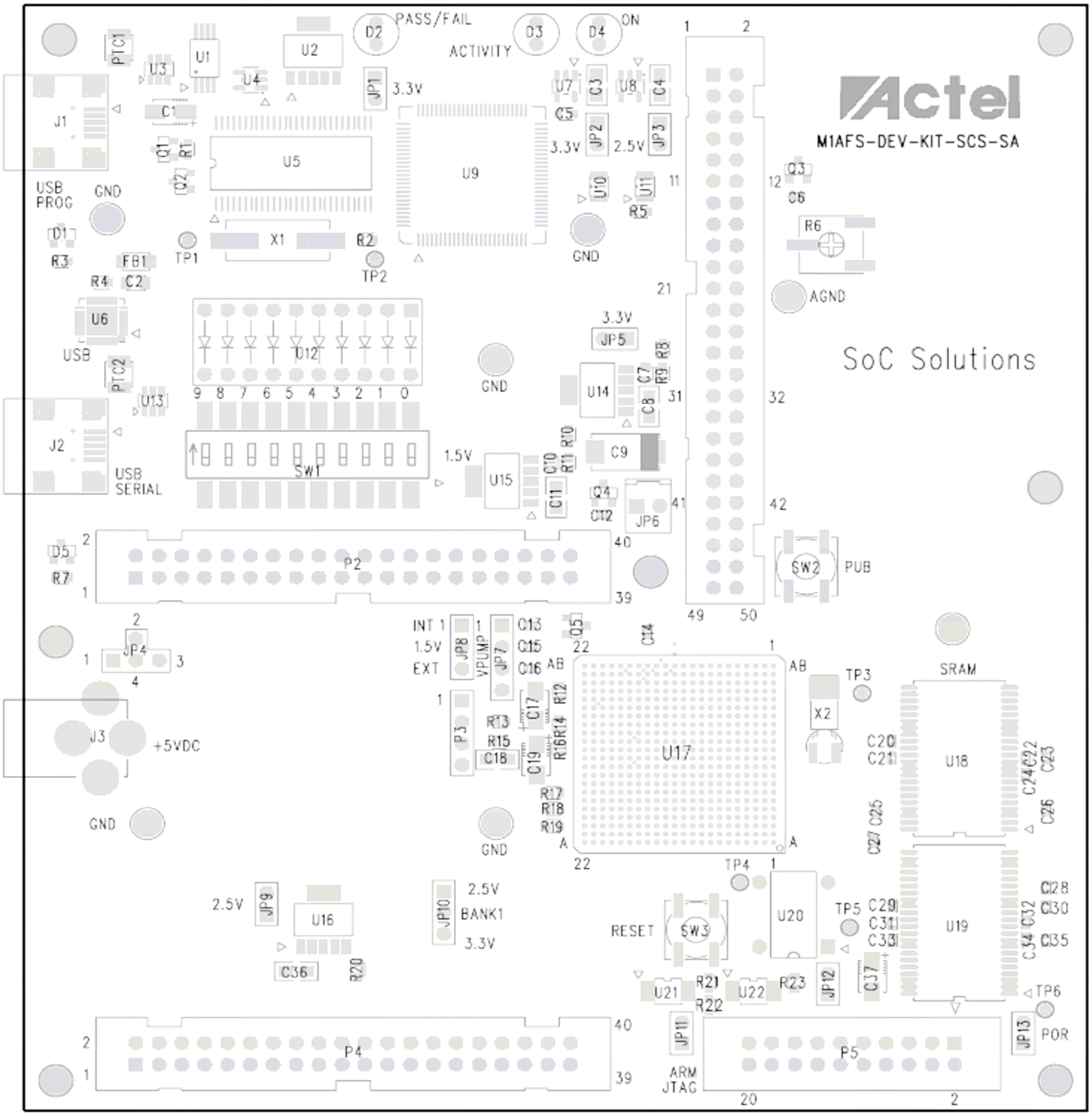


Figure B.1 - M1AFS Development Board Top Assembly Drawing

C

M1AFS Development Board Tests

This appendix outlines how to test the M1AFS Development Board.

Introduction

The Cortex-M1 Enabled Fusion Development Kit is supplied with the design and the procedure used for manufacturing tests. The following procedure is a subset of the actual procedure used by the manufacturer to fully test the M1AFS Development Board. This test procedure is useful for verifying the functionality of the board as well as observing some of the Cortex-M1 Enabled Fusion Development Kit features in action.

Board Configuration

Please refer to the *Chapter 3 – Initial M1AFS Development Board Configuration* section for configuring the M1AFS Development Board.

Also refer to the *Chapter 3 – Powering up the M1AFS Development Board* section for powering and cabling the M1AFS Development Board.

Load the "User Tests" design

Use the Actel FlashPro programming tool to program the FPGA with the embedded controller design (**socTop.stp** in the **Board Verification\Hardware** folder). The "User Tests" design architecture is described later in this appendix.

Procedure for running "User Tests"

1. Run the **M1AFS Board – User Tests** by navigating to the "Board Verification\PC_Software" folder and double clicking the M1AFS_Tests.exe file.

2. The first dialog box will be a COM configuration dialog. Reference the *Determining the Serial COM Port* section in *Chapter 3* in this user guide for determining which COM port to use. Select the appropriate COM port and click OK.

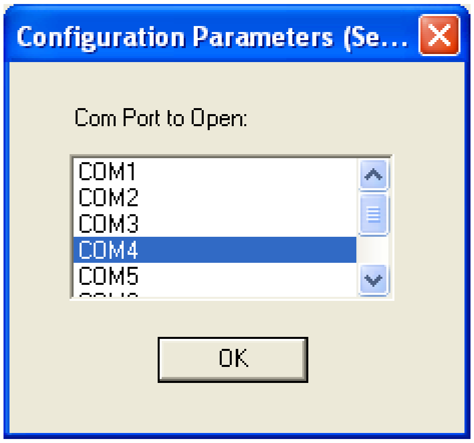
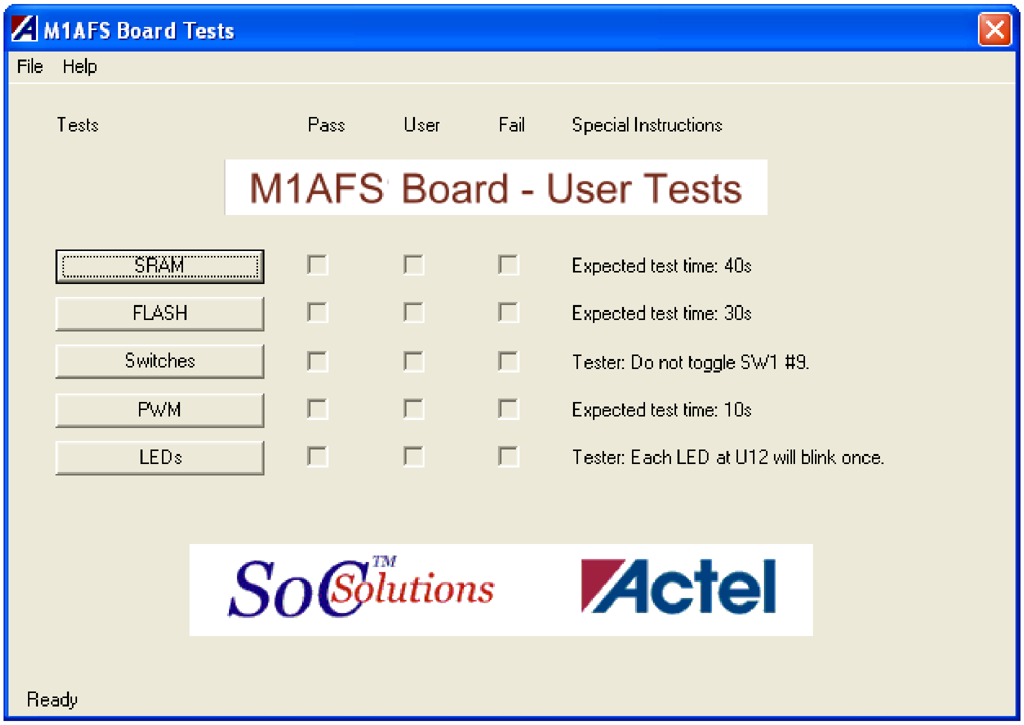


Figure C.1- COM Port Selection

3. Now run all the tests in the **M1AFS Board – User Tests** Application.



Tests	Pass	User	Fail	Special Instructions
SRAM	□	□	□	Expected test time: 40s
FLASH	□	□	□	Expected test time: 30s
Switches	□	□	□	Tester: Do not toggle SW1 #9.
PWM	□	□	□	Expected test time: 10s
LEDs	□	□	□	Tester: Each LED at U12 will blink once.

Figure C.2- M1AFS Board – User Tests Application

a. The individual tests can be run in any order.

- Clicking the **"SRAM"** button performs a complete check of the external SRAM chips at U18 and U19.
- Clicking the **"FLASH"** button performs a continuity check of the external FLASH chips at U200 and U201.
- Clicking the **"Switches"** button reads the state of the switch bank SW1 then outputs the result in a text format. IMPORTANT: Do not toggle SW1 #9 while running the board checkout program. (#9 is indicated on the board silkscreen.) Doing so will cause the board checkout system to function incorrectly.
- Clicking the **"PWM"** button will enable two PWM outputs to two of the LEDs at U12.
- Clicking the **"LEDs"** button will blink each of the 10 LEDs at U12 on and off individually.

b. If any test fails, then do the following sequence:

- Close down application.
- Power down the board by unplugging the external power supply from J3.
- Unplug USB cables at J1, J2.
- Verify the jumpers and switches correspond to factory defaults.
- Repeat steps 1-4 above.

c. If the test still fails, contact *SoC Solutions* at support@socsolutions.com

“User Tests” Design

The design used for board verification is very similar to the Sample Design. The only difference between the two designs is that the internal flash and external flash occupy different slots on the AHB bus, and thus different locations in the address map. In the figure below, the internal flash (CoreAhbNvm) occupies AHB slot 0, and external flash occupies AHB slot 2. Internal flash is loaded with a Cortex-M1 program that executes after the FPGA is programmed with the “User Tests” design.

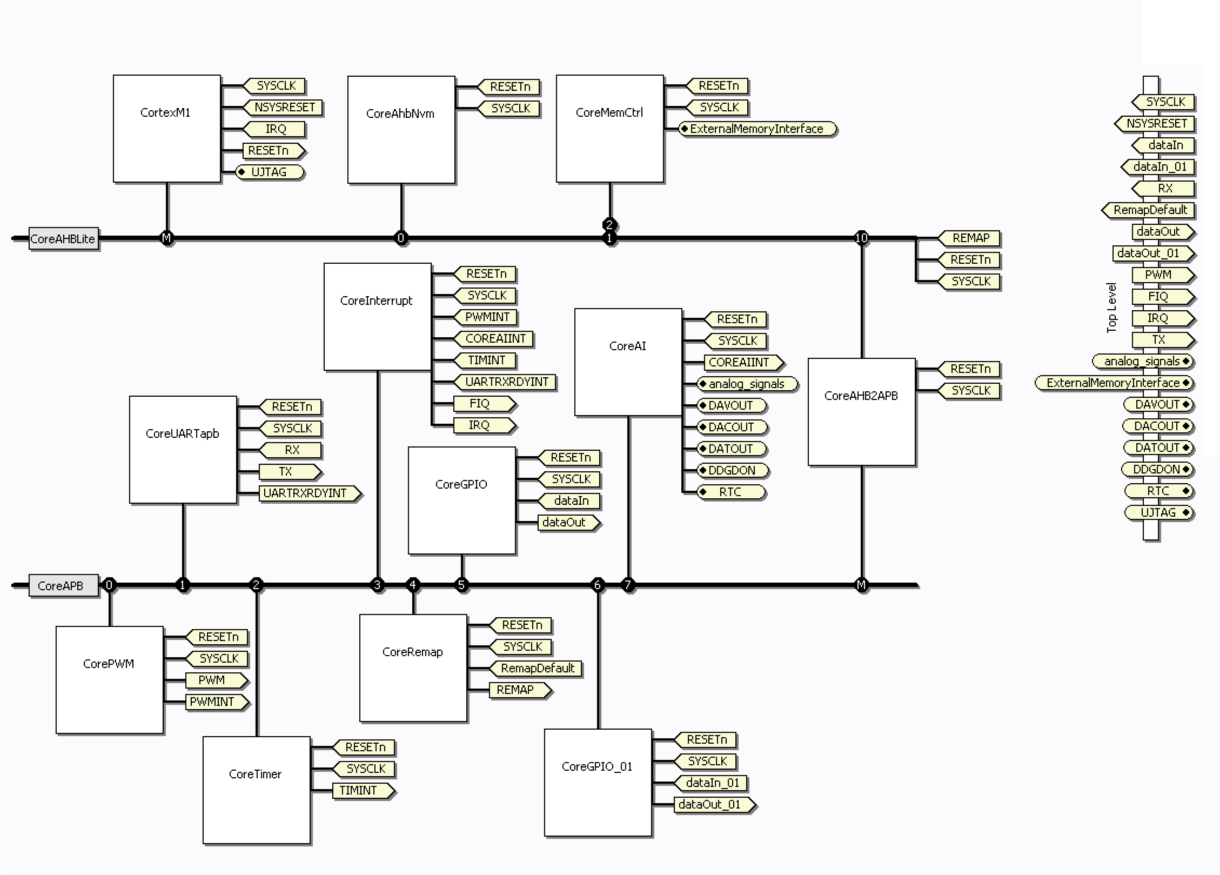


Figure C.3- “User Tests” Design

Product Support

SoC Solutions is providing technical and non-technical support for this product. The product is distributed through Actel and its distributors. For pricing information, please contact Actel. For all other support, please contact SoC Solutions by email or phone.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is support@socsolutions.com.

Phone

Our Support Staff can help you with hardware, software and installation issues. SoC Solutions will retrieve information, such as your name, company name, phone number and your question, and then issues a case number. The phone hours are from 10:00 A.M. to 6:00 P.M., Eastern Time, Monday through Friday. The Technical Support numbers are:

770-680-2500 Ask for technical support for Actel.

Customers needing assistance outside the US time zones can either contact technical support via email support@socsolutions.com.