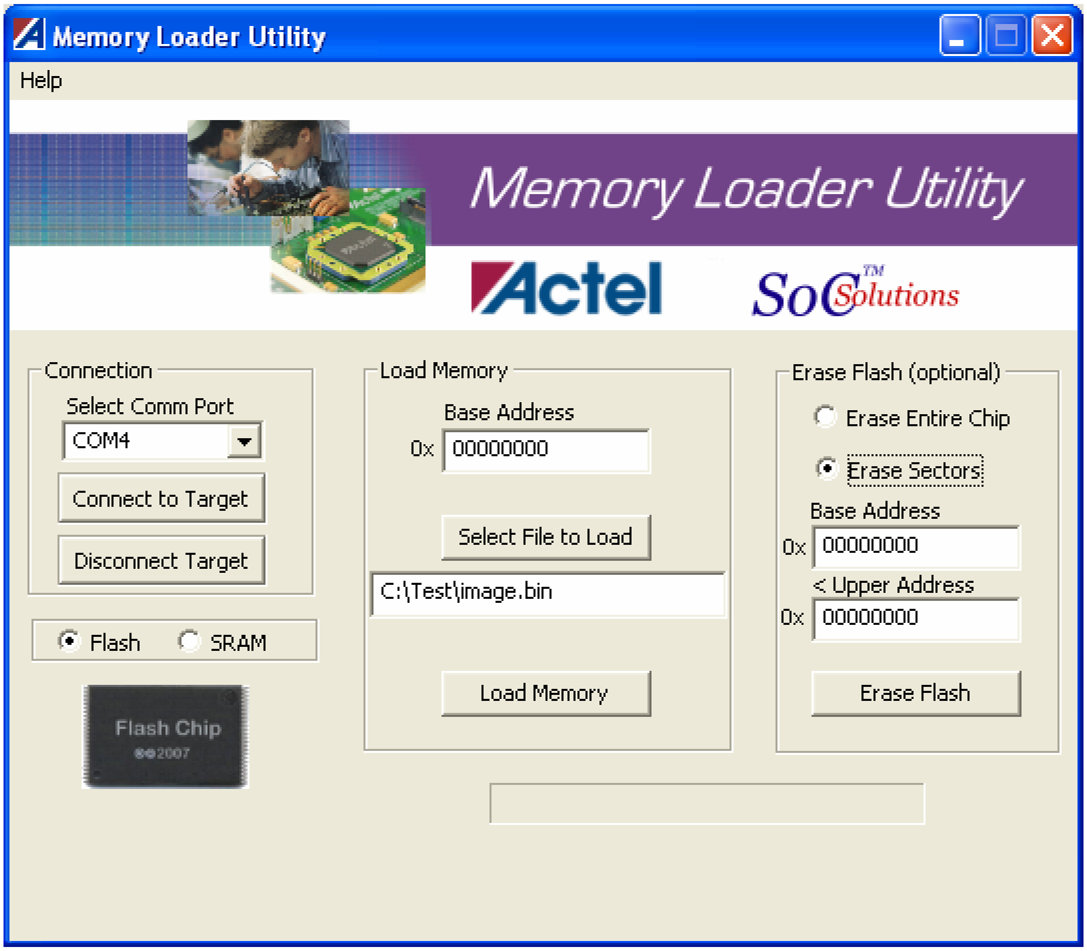
Help Version 1.0

Overview

The Memory Loader Utility is an application for the PC which allows Actel Development Kit users to program on-board FLASH memory. Supported development boards include the M1A3P, M7A3P, M1AFS, M7AFS and M1AGL boards. It is anticipated that users of the various Actel Development Kits will want to program the FLASH with an image that can be executed by the embedded Cortex-M1 or ARM7 Processor. This is the flow described in this document.

Figure 1 shows the Memory Loader Graphical User Interface (GUI).



Select Comm Port: COM4

● Flash ○ SRAM

Base Address: 0x 00000000

C:\Test\image.bin

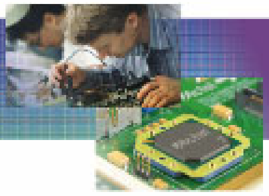
○ Erase Entire Chip

● Erase Sectors

Base Address: 0x 00000000

< Upper Address: 0x 00000000

Figure 1 - Memory Loader Utility GUI



Memory Loader Utility

Actel SoC Solutions™

Help Version 1.0

The Memory Loader Utility works in conjunction with a specific load for the Actel FPGA and an embedded (ARM7 or Cortex-M1) program to load FLASH memory. The Memory Loader Utility sends commands and data through a PC communications (COM) port to the development board; the embedded program on the development board listens for commands from the PC, and executes the requested transactions, e.g. FLASH Erase, FLASH Program. The Memory Loader Utility must be used in conjunction with:

1. The provided example hardware design loaded into the Actel FPGA.

   Alternatively, any CoreConsole design that contains the following may be used:

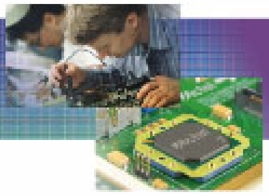
   A. An ARM processor configured to use the FlashPro3 debugging interface.

   B. CoreConsole CoreMemCtrl with SRAM at AHB slot 0 or 1, FLASH at AHB slot 1 or 0 (slot 0 = address 0, slot 1 = address 0x10000000).

   C. A Remap switch to toggle the address associated with AHB slots 0 and 1. This allows an embedded executable (Cortex-M1 or ARM7) programmed to FLASH with the Memory Loader Utility to run from FLASH after a remap operation.

   D. A CoreConsole CoreUartApb at a valid address (address is 0xA1000000 in the example hardware design) configured to operate at a rate of 57600 bits/second.

2. The provided embedded MemoryLoaderMx software program running on the development board through the SoftConsole debugger. Note that the user has the ability to change the following in the MemoryLoaderMx program for adaptation to different address maps/clock rates:
   A. UART base address - in case the CoreUartApb module is desired to be at a different address.
   B. UART baud rate - in case the hardware design uses a different master clock.



General instructions for programming the on-board FLASH

1. **Program the FPGA.**

    A. Ensure that the Actel Libero suite (including FlashPro) is installed on the PC. Also, ensure that the disk containing this file has been installed on the target PC.

    B. Connect the mini-USB cable between the PC and the board (USB PROG).

    C. If necessary, go through the driver installation procedure for the FlashPro3 drivers. *See FlashPro Users Guide for more details.*

    D. Use FlashPro to load the programming file (socTop.pdc) for the provided example design for the UJTAG-style debugger (e.g. M1A3P_Example_UJTAG) into the FPGA. *See FlashPro Users Guide for more details.*

    E. Connect the other mini-USB cable between the PC and the board (USB SERIAL). Note the resultant COM port in Windows Hardware Device Manager (Start->Settings->Control Panel->Hardware->Device Manager->Ports (COM & LPT)). The COM port should appear as (for example) "SFE USB to RS232 Controller (COM5)". The particular COM port listed here will be used to configure the Memory Loader Utility.

    F. Consult the documentation for the provided example design and ensure that switches are set so that SRAM is at 0.

2. **Produce an ARM executable image file (binary format) to be loaded.**

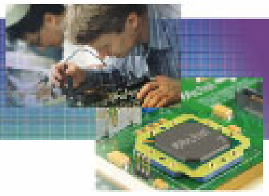
    A. Ensure that SoftConsole is installed on the PC.

    B. Run SoftConsole, and load the provided example project, e.g. TrafficLightM1.

    C. Build the project, and note the presence of the post-build step by selecting Project->Properties->Build Steps from the menu. The post-build step invokes the utility arm-eabi-none-objopy to convert the compiled executable format (.axf) to a binary format used by the Memory Loader Utility. There should now exist a file, e.g. TrafficLight.bin. This file will be used to program the FLASH.

3. **Run the embedded executable FlashProgramMx via the SoftConsole debugger.**

    A. Run SoftConsole, and load the provided project FlashProgramMx



B. Set up and run the companion debugger tool (external tools) for the M7/M1 processor. *For more detail, refer to User Guide that came with this Development Kit.*

C. Download the executable to SRAM by selecting FlashProgramMx from the Debug menu. *For more detail, refer to User Guide that came with this Development Kit.*

D. Run the executable by selecting run from the debug menu.

4. Use the Memory Loader Utility on the PC to program FLASH.

A. Run the Memory Loader Utility by double-clicking it.

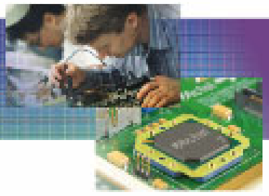
B. Select the proper COM port (from step 1-E) from the drop-down menu, and click "Connect"

C. If all the steps have been completed correctly, there will be a message to indicate that the connection is successful, and the other configurations on the Memory Loader Utility will become enabled. If there is a problem with one of the previous steps, the Memory Loader Utility will indicate that the connection attempt was unsuccessful.

D. In the Load Memory section, enter the address where the file will be loaded. In the example design, to have the image run from FLASH after a remap, the file needs to be loaded at the base address of FLASH: 0x10000000. In general, it is recommended that the address entered here correspond to a start of a FLASH sector, as a sector erase is performed before the file is loaded.

E. Click the Select File to Load button. This brings up a standard file open dialog. Browse to and select the (binary) file that should be loaded, and click OK to close the dialog. Note that the binary file format is a 32-bit little-endian binary format. This format is the result of running a post-build utility (objcopy) provided with SoftConsole to convert the ARM executable image (.axf) to a strictly binary format.

F. Click the Load Memory button. This action erases the necessary FLASH sectors, then programs FLASH, starting with the specified address, with the data from the specified file.



Memory Loader Utility Control Information

General Configuration Controls

A. **Select Comm Port:**
The Memory Loader Utility uses a PC Communications (Comm) port to communicate with the target board. The "Select Comm Port" control configures which port is used for PC-to-board communication.

B. **Connect to Target:**
The "Connect to Target" button opens the communications channel with the target board and sends a test message. If a valid response to the test message is received, the erase and load controls become enabled.

C. **Disconnect Target:**
The "Disconnect Target" button closes the communication channel with the target board and disables the erase and load controls.

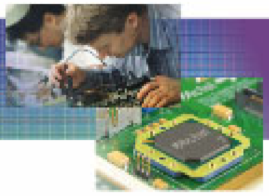
D. **Flash/SRAM Radio button:**
The "Flash" and "SRAM" radio buttons configure the type of memory that is to be accessed. Click "Flash" to program or erase on-board FLASH; Click "SRAM" to program on-board SRAM.

Load Memory Controls

A. **Base Address Edit Box:**
The Base Address Edit Box should contain a hexadecimal value from 0 to FFFFFFFF. This address is the start address for the memory load. It is highly recommended that this address be aligned with the base address of a FLASH sector. In most instances, this address will be the base address of the FLASH memory in the address map of the system.

B. **Select File to Load Button:**
The "Select File to Load" button invokes a standard Windows "File Open" dialog. Use this dialog to browse to the binary (*.bin) file. The selected file will be programmed into memory. The binary file format is not plain binary, but instead is a 32-bit little-endian format. This is the format that is produced by the SoftConsole post-build utility "objcopy".

C. **Load Memory Button:**
The "Load Memory Button" first erases the selected memory, then programs the selected memory. This is because FLASH memory must be erased before it can



be successfully programmed. If SRAM is selected, the erase starts with the base address. If FLASH is selected, the erase starts with the base sector address of the sector which contains the desired base address. If the file to load is larger than one FLASH sector, multiple sectors are erased. After the erasure, programming commences. During a Memory Load procedure, all controls are disabled.

Erase Flash (Optional) Controls

The Erase Flash (Optional) Controls are only enabled when the "Flash" radio button is selected.

A. **Erase Entire Chip / Erase Sectors Radio buttons:**
These buttons determine if the erase is to be done for the entire chip or just for certain sectors.

B. **Base Address Edit Box:**
The "Base Address" Edit Box should contain a hexadecimal value from 0 to FFFFFFFF. This address is the start address for the memory erase. It is highly recommended that this address be aligned with the address of a FLASH sector.

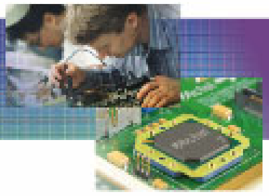
C. **Upper Address Edit Box (non-inclusive, only visible if Erase Sectors is selected):**
The "Upper Address" Edit Box should contain a hexadecimal value from 0 to FFFFFFFF. The Upper Address should be greater than the Base address. If the upper address is aligned with the base address of a FLASH sector, that sector is NOT included in the erase.

D. **Erase Flash Button:**
The "Erase Flash" button initiates the desired Erase procedure. During a Flash Erase procedure, all controls are disabled.

Status Controls

A. **Progress Bar:**
The Progress Bar gives progress for erase and file loading procures.

B. **Status Text:**
Status text appears in the lower left of the Memory Loader Utility main dialog window. This text gives additional information and status about the current activity of the application.



Menu Controls

A basic menu exists mainly to connect the user to the existing "help" facilities.

A. **Help->Help;**
This menu selection will invoke a separate window that shows this document.

B. **Help->About:**
This menu selection will show an "About" dialog that contains versioning information about the Memory Loader Utility application.