

# FM577

## Firmware guide

4DSP Inc, 955 S Virginia Street, Suite 214, Reno, NV 89502, USA

4DSP bv, Crown Business Centre, Leidse Schouw 2, 2408 AE Alphen a/d Rijn, Netherlands

Email: [support@4dsp.com](mailto:support@4dsp.com)

This document is the property of 4DSP Inc. and may not be copied nor communicated to a third party without the written permission of 4DSP Inc.

© 4DSP Inc. 2006 - 2008

---

Table of Contents

1	Introduction.....	3
2	Cyclone III Device A registers.....	4
3	Create and write an FPGA configuration file to the Flash .....	6
4	Reconfigure FPGA Device B on the fly .....	6
5	Flash memory map.....	7

# 1 Introduction

The Firmware guide describes the software interface to the reference Firmware that comes with the API functions, as well as a short description of its architecture. The software interface is done through the PCI bus and using a number of specific control registers. A program example is provided along with an FPGA VHDL firmware example on the CD and this can be used by the user as a starting point to develop more complex algorithms.

Note: Information within this document is subject to change.

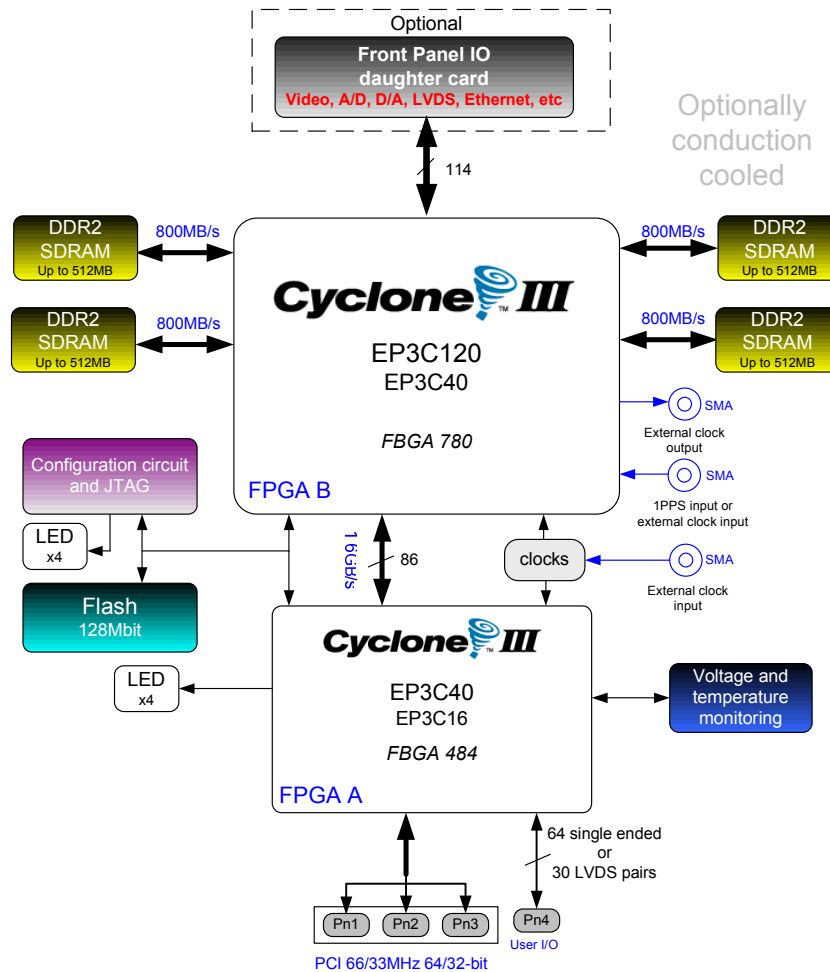


Figure 1: Architecture Block Diagram

## 2 Cyclone III Device A registers

The registers embedded in the Cyclone III Device A FPGA are accessible from both the program running on the host and the Cyclone III Device B FPGA. They facilitate message passing and give a simple and powerful framework for users wishing to implement complex firmware designs on the FM482.

Table 1: Device A register description.

HOST	FPGA_B	Register Name	Addr	Description
Write	No access	Reset	0x00	Reset=0x00000001. Auto clearing bit after write
Reserved	Reserved	Reserved	0x01	Reserved
Reserved	Reserved	Reserved	0x02	Reserved
Reserved	Reserved	Reserved	0x03	Reserved
Reserved	Reserved	Reserved	0x04	Reserved
Read	No access	Board_diagnostics5	0X05	Bits 9 down to 0: FPGA B temperature Bits 19 down to 10: not used Bits 29 down to 20: 5V deviceB (value*(1000+249)/249)
Reserved	Reserved	Reserved	0x06	Reserved
Write	No access	Interrupt_FM482	0x07	Interrupt generated by the FPGA_B user application = 0x00000002. Auto clearing bit when Mailbox_FPGA_B register is read. If this register is 0x00000003, it means that both the DMA controller and the FPGA_B generated an interrupt
reserved	reserved	reserved	0x08	reserved
reserved	reserved	reserved	0x09	reserved
Read	No access	Board_diagnostics4	0x0A	Bits 9 down to 0: not used Bits 19 down to 10: 3v3 on board voltage rail deviceB Bits 29 down to 20: monitor device B temperature
Write	No access	Mailbox_host	0x0B	Mailbox associated with Interrupt_Host
Write	Read	Interrupt_Host	0x0C	Interrupt generated by the host = 0x00000001. Auto clearing bit when Mailbox host is read
Read	Write	Mailbox_FPGA_B	0x0D	Mailbox associated with Interrupt_FM577
Read	No access	Board_diagnostics3	0x0E	Bits 9 down to 0: Pn4 IO voltage deviceA Bits 19 down to 10: -12V deviceA $=((value*(1000+100)-3v3*(1000))/100)$ Bits 29 down to 20: not used
Write	No access	Source_destination	0x0F	Defines the source or destination before starting a DMA. 0x00000000: Source or destination is the FPGA_A 0x00000001: Source or destination is the FPGA_B 0x00000010: Load FPGA_A bitstream to the flash 0x00000100: Load FPGA_B bitstream to the flash 0x00001000: Load FPGA_B bitstream directly to the C-III
Reserved	Reserved	Reserved	0X10	Reserved

Read	No access	Board_diagnostics1	0X11	Bits 9 down to 0: 3.3V on board voltage rail deviceA Bits 19 down to 10: monitor device A temperature Bits 29 down to 20: FPGA A temperature
Read	No access	Board_diagnostics2	0X12	Bits 9 down to 0: not used Bits 19 down to 10: +12V deviceA =(value *(1000+249)/249 Bits 29 down to 20: 3v3 deviceA
Read	No access	FPGA A Firmware information	0x13	Bits 7 downto 0: FPGA A PCI interface firmware revision number xxxx.xxxx e.g., 0010.01010Rev 2.5) Bits 11 downto 8: FPGA_A minor revision Bits 15 downto 12: FPGA_A major revision Bits 21 downto 16: FPGA_A customer number Bits 30 downto 22: FPGA_A FW type bit 31: extended FW information available
Write	No access	Board_information	0x14	Bits 7 downto 0: CPLD firmware revision number Bits 15 downto 8: FPGA_B device type Bits 31 downto 16: FM577 serial number
Write/Read	No access	User_ROM_register	0x15	ROM register programmed by the user and stored in the flash device. Can contain for example an FPGA firmware revision number.
reserved	reserved	reserved	0x16	reserved
Reserved	Reserved	Reserved	0x17	Reserved
Read/write	No access	FPGA A Clock_frequencies	0x18	Bits 15 downto 0 = clock counter Bit 18 downto 16 = clock source select
Reserved	Reserved	Reserved	0x19	Reserved
Reserved	Reserved	Reserved	0x1A	Reserved
Read	No access	Board_diagnostics6	0X1B	Bits 9 down to 0: 0V9 deviceB Bits 19 down to 10:Front panel Voltage deviceB Bits 29 down to 20: 1V2 deviceB
Read	No access	Board_diagnostics6	0X1C	Bits 9 down to 0: 0V9 deviceB Bits 19 down to 10:Front panel Voltage deviceB Bits 29 down to 20: 1V2 deviceB
Reserved	Reserved	Reserved	0X1D	
Reserved	Reserved	Reserved	0X1E	
Reserved	Reserved		0X1F	
Write	Read Write	Custom_register_0	0X20	Custom register 0 for user application
...	...	...	....	...
Write	Read Write	Custom_register_31	0X3F	Custom register 31 for user application

### 3 Create and write an FPGA configuration file to the Flash

The FPGA Device A and Device B configuration bit streams stored in the flash can be updated by the user. FPGA B can also be reconfigured on the fly directly from the (Refer to section 7 of this document).

1. In the Quartus project make sure that for the device and pin options you choose to have a raw binary file (.rbf) as programming file format.
2. Compile your design with the Quartus tools to generate a valid .rbf file.
3. To write the configuration file to the flash use the `_4FM_LoadFPGA` function from your C program or use the `_4FM_upload_firmware` in a terminal or command prompt. A third option is to use the 4FM control Gui to upload the firmware.

For a Device A configuration file type: `_4FM_upload_firmware FM577 0 -a 'file_name'`

For a Device B configuration file type: `_4FM_upload_firmware FM577 0 -b 'file_name'`

4. Give a power cycle to the computer for the FPGAs to be reconfigured from the flash.

### 4 Reconfigure FPGA Device B on the fly

The Cyclone III Device B on the FM577 can be reconfigured from the host system at any time without the need to reboot or give a power cycle to the system.

To reconfigure the FPGA Device B, please perform the following steps:

1. In the Quartus project make sure that for the device and pin options you choose to have a raw binary file (.rbf) as programming file format.
2. Compile your design with the Quartus tools to generate a valid .rbf file.
3. To reconfigure the Cyclone III Device B use the `_4FM_LoadFPGA` function from your C program or use the `_4FM_upload_firmware` in a terminal or command prompt.  
Type in a terminal: `_4FM_upload_firmware FM577 0 -b,direct 'file_name'`
4. Reset the board before performing any other action.

## 5 Flash memory map

### 128Mbit, 256 sectors of 64kbytes each

#### **Sector 0**

Byte 0: Flash status

Byte 1, 2, 3, 4: 32-bit user ROM (User\_ROM register)

#### **Sector 1 to Sector 43**

Cyclone III Device A safety configuration bitstream that can be loaded to the Cyclone III Device A in the case the user configuration is corrupted or does not allow to reprogram the flash anymore.

#### **Sector 44 to Sector 86**

Cyclone III Device A standard configuration bitstream

#### **Sector 87 to Sector 166**

Cyclone III Device B configuration bitstream

#### **Sector 166 to Sector 254**

Empty

#### **Sector 255 (protected sector)**

Board\_information register containing the Cyclone III Device B type and the 4FM serial number