FMC-IPMI
Software Guide

Version 1.01

AVNET®
electronics marketing

Revision History

Version	Description	Date
0.01	Preliminary Version	Nov 27, 2009
0.03	Add description of EEPROM contents for each FMC module Fix GA[1:0] error, change to GA[0:1]	Dec 02, 2009
0.04	Modify for use with new FMC-IIC software library	Jan 11, 2010
1.01	Release (no content changes)	Jun 23, 2010

References

1. FMC Specification
   http://www.vita.com/fmc.html
2. Intelligent Platform Management Interface Specifications
   www.intel.com/design/servers/ipmi/spec.htm
3. Platform Management FRU Information Storage Definition V1.0
   http://download.intel.com/design/servers/ipmi/FRU1011.pdf
4. FMC-IIC Software Guide
   FMC-IIC_Software_Guide_1_02.pdf
5.

Table of Contents

Table of Figures

Table of Tables

Introduction

The FMC-IPMI software library is implemented for use in the EDK development environment.

Avnet's new hardware portfolio is based on the FMC (FPGA mezzanine card) standard. Since Avnet is making available a variety of kits which combine different FMC carriers and FMC modules together, it is important to validate the presence of the correct FMC modules for each of the reference designs.

The FMC specification [1] requires each FMC module to have an EEPROM whose content is defined by the IPMI (Intelligent Platform Management Interface) specifications [2]. More specifically, the EEPROM is defined as a FRU (Field Replaceable Unit). The contents of the FRU's EEPROM is defined in the FRU specification [3]

Features

The FMC-IPMI software library provides the following functionality:

- Accessing FRU content from the IPMI EEPROM
    - Common Header
    - Board Information
    - Multi-Record (not supported yet)
        - Power Supply Information (not supported yet)
        - DC Load (not supported yet)
        - DC Output (not supported yet)
        - FMC Information (not supported yet)
- Detection and Identification of FMC modules
    - Avnet FMC-IMAGEOV
    - Avnet FMC-DVI/DP
    - Avnet FMC-ISMNET
- Enabling of logic driving outputs to FMC modules

The FMC-IPMI software library supports the following FMC carriers:

- Avnet Virtex-6 LX130T development kit (TBD)
- Avnet Spartan-6 LX150T development kit
- Avnet Spartan-5 LX45T co-processing kit (TBD)
- Avnet Spartan-6 LX16 low cost kit (TBD)
- Xilinx ML605 (TBD)
- Xilinx SP601 (TBD)
- Xilinx SP605 (TBD)

Use Case - Targeted Design Platforms

The typical use case of the FMC-IPMI software library will be to detect the presence of hardware components that make up a TDP (Targeted Design Platforms). The reference designs created for a TDP should successfully identify the required FMC modules before driving signals to the FMC connector.

Avnet TDP - Industrial Ethernet Kit

This kit consists of the following hardware components:

- Avnet Spartan-6 LX150T development kit
  - FMC1 : Avnet FMC-ISMNET
  - FMC2 : *{vacant}*

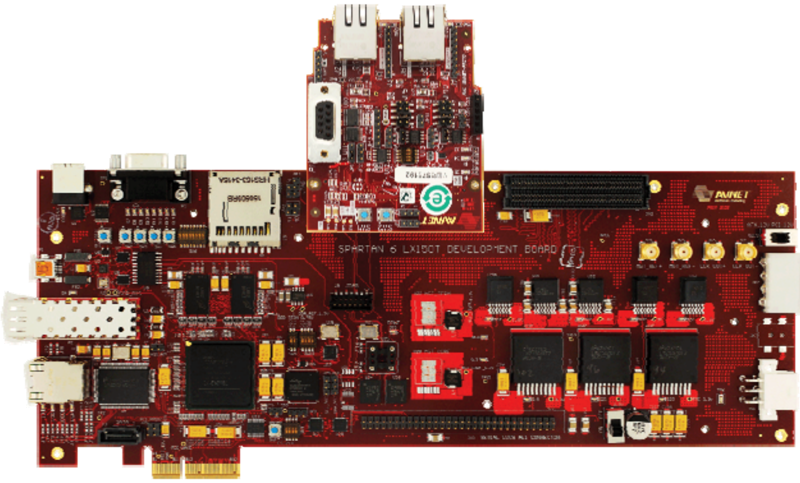

Figure 1 – Industrial Networking Kit

Avnet TDP - Industrial Video Processing Kit

This kit consists of the following hardware components:

- Avnet Spartan-6 LX150T development kit
  - FMC1 : Avnet FMC-IMAGEOV
  - FMC2 : Avnet FMC-DVI/DP

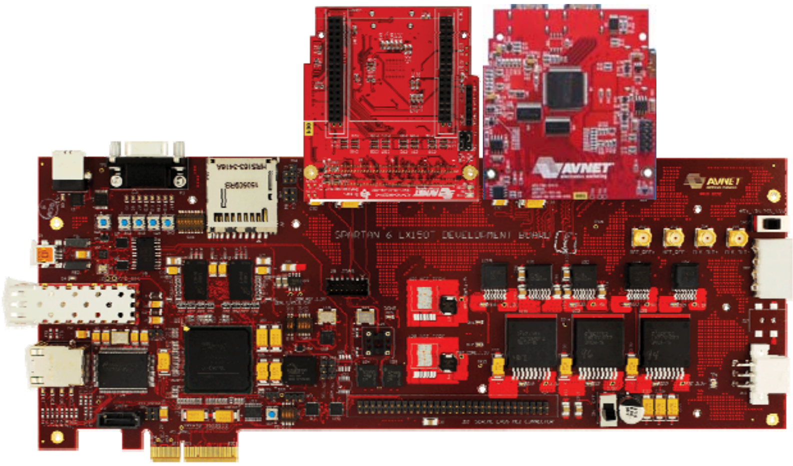

Figure 2 – Industrial Video Processing Kit

Software Library

Functional Description

The FMC-IPMI software library will be used as the first validation step of a program executing on the MicroBlaze processor in the FPGA of the FMC carrier board.

The library is implemented to be used with the XPS_IIC pcore, as illustrated in the following figure.

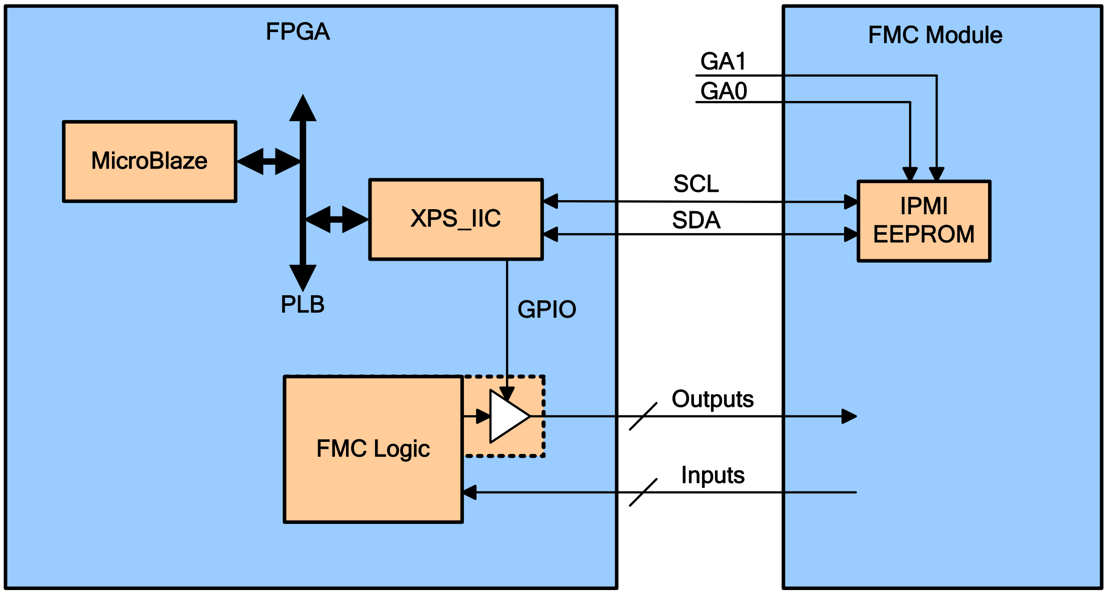

Figure 3 – FMC-IPMI – FPGA Implementation with XPS_IIC

The XPS_IIC pcore performs the I2C transactions which access the content of the IPMI EEPROM. The GA[1:0] bits determine the address of the IPMI EEPROM. For a carrier having more than one FMC slot, these bits will be different for each FMC slot.

Table 1 – GA[0:1] Mapping – I2C EEPROM Addresses

GA[0:1]	I2C EEPROM address
00b	0xA0
01b	0xA2
10b	0xA4
11b	0xA6

The following table lists the various FMC carriers and how they implement the GA[0:1] mapping of each of their FMC slots differently.

Table 2 – GA[0:1] Mapping – Overview of FMC carriers

	GA[0:1] mapping	
FMC Carrier	**FMC slot 1**	**FMC slot 2**
Avnet Virtex-6 LX130T development kit	JT2:JT1 (default=00b)	▨
Avnet Spartan-6 LX150T development kit	00b	01b
Avnet Spartan-6 LX16 low cost	JT5:JT4 (default=00b)	▨
Avnet Spartan-6 LX45T co-processing kit	JP1:JP2 (default=00b)	
Xilinx ML605	00b	01b
Xilinx SP601	10b	▨
Xilinx SP605	10b	▨

The optional GPIO outputs of the XPS_IIC pcore can be used to generate enable (EN) signals to the FPGA logic driving the outputs to the FMC pins.

By default, these (EN) signals will be “OFF”. When the MicroBlaze successfully identifies the FMC module(s) expected by the reference design, it will enable the output buffers driving signals to the FMC connector.

The recommended usage of the XPS_IIC’s GPIO outputs is:

Table 3 – FMC-IPMI – Recommended usage of the XPS_IIC GPIO

GPIO	Usage
0	Enable FMC slot 1
1	Enable FMC slot 2
2..N	Available to user design

Programming API

The FMC-IPMI software library is implemented in two layers.

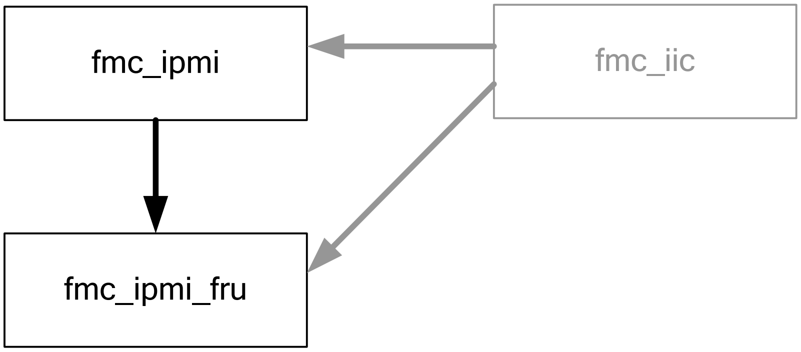

Figure 4 – FMC-IPMI – Implementation Layers

Each layer can be used by an application. A typical application will only use the top layer to identify the presence of an FMC module.

Each layer is described in the following sections.

FMC_IPMI Layer

The FMC_IPMI layer provides the minimum functionnality that an application needs to identify an FMC module and enable/disable the logic driving signals to the FMC connector.

Table 4 – FMC_IPMI – Function Overview

Function	Description
detect_ipmi_address	Detect the I2C address of the IPMI EEPROM.
fmc_ipmi_detect	Detect the presence of an FMC module.
fmc_ipmi_enable	Enable the logic driving the FMC module.
fmc_ipmi_disable	Disable the logic driving the FMC module.

Each of the previous functions has an argument called fmcId which can be any one of the following values.

Table 5 – FMC_IPMI – FMC_ID Values

Value	Description
FMC_ID_SLOT1	Detect IPMI EEPROM in following address range: 0xA0
FMC_ID_SLOT2	Detect IPMI EEPROM in the following address range: 0xA2, 0xA4
FMC_ID_ALL	Detect IPMI EEPROM in following address range: 0xA0, 0xA2, 0xA4, 0xA6

The FMC_ID_ALL value can be used for an FMC carrier having only 1 FMC slot.

For an FMC carrier having two FMC slots, the FMC_ID_SLOT1 and FMC_ID_SLOT2 values can used to distinguish between the two slots.

The FMC_ID_SLOT2 can adapt to an incorrect GA[1:0] mapping caused by flipped bits. This is indeed the case for the FMC_IMAGEOV and FMC_DVI/DP modules.

FMC_IPMI_FRU Layer

The FMC_IPMI_FRU layer provides the ability to access the FRU content in the IPMI EEPROM. Since a typical application will only read the FRU content, the write capability is only enabled for production (using a #define directive).

The following table lists the functions are available in this software layer

Table 6 – FMC_IPMI_FRU – Function Overview

Function	Description
fmc_ipmi_fru_get_common_info	Read the FRU Common Header from the IPMI EEPROM
fmc_ipmi_fru_get_board_info	Read the FRU Board Info Area from the IPMI EEPROM
fmc_ipmi_fru_set_common_info	Write the FRU Common Header to the IPMI EEPROM
fmc_ipmi_fru_set_board_info	Write the FRU Board Info Area to the IPMI EEPROM

The grayed functions are only enabled for production using a #define directive.

Example Usage

The following code excerpt describes how to use the FMC-IPMI software library to detect the presence of a FMC module.

```
#include "fmc_iic.h"
fmc_iic_t ipmi_iic;

#include "fmc_ipmi.h"
#define FMC_IPMI_I2C_CONTROLLER      XPAR_XPS_IIC_0_BASEADDR

...

// FMC-IIC Initialization
if ( !fmc_iic_xps_init( &ipmi_iic, "IPMI IIC Chain", XPAR_XPS_IIC_0_BASEADDR ) )
{
   xil_printf( "ERROR : Failed to initialize IPMI IIC Chain\n\r" );
   exit(0);
}

// FMC Module Validation
if ( fmc_ipmi_detect( &ipmi_iic, "FMC-ISMNET", FMC_ID_ALL ) )
{
   fmc_ipmi_enable( &fmc_ipmi_iic, FMC_ID_ALL );
}
else
{
   fmc_ipmi_disable( &fmc_ipmi_iic, FMC_ID_ALL );
   exit(0);
}
```

If there is no FMC module, the output of FMC Module Validation code will be:

```
FMC Module Validation
[fmc_ipmi_get_common_info] FRU_I2C_ERROR (idx=0)
ERROR : No FMC module detected
```

If there is an FMC module, but it does not have valid FRU content in its IPMI EEPROM, the output of FMC Module Validation code will be:

```
FMC Module Validation
[fmc_ipmi_get_common_info] FRU_CHECKSUM_ERROR
ERROR : FMC module does not have valid FRU content in its IPMI EEPROM
```

If the incorrect FMC module is present, the output of FMC Module Validation code will be:

```
FMC Module Validation
Board Information:
        Manufacturer    = Avnet
        Product Name    = FMC-DVI/DP
        Serial Number   = proto
        Part Number     = AES-FMCDVI-G
ERROR : Unexpected FMC-DVI/DP module, Expected FMC-ISMNET module
```

If the correct FMC module is present, the output of FMC Module Validation code will be:

```
FMC Module Validation
Board Information:
        Manufacturer    = Avnet
        Product Name    = FMC-ISMNET
        Serial Number   = proto
        Part Number     = AES-FMCISMNET-G
SUCCESS : Detected FMC-ISMNET module!
```

Known Limitations

The following limitations exist with the FMC-IPMI software library.

Table 7 – FMC-IPMI – Known Limitations

Layer	Limitations
fmc_ipmi	
fmc_ipmi_fru	Board Information support: - manufacturing date/time not supported yet
	Multi-Record support: - not supported yet

FMC IPMI Content

The Avnet FMC modules are programmed with the following content:

Table 8 – FMC IPMI EEPROM Content – Board Information

Content	**FMC-ISMNET**	**FMC-IMAGEOV**	**FMC-DVI/DP**
Board Information			
- Manufacturer Date/Time	-	-	-
- Manufacturer	Avnet	Avnet	Avnet
- Product	FMC-ISMNET	FMC-IMAGEOV	FMC-DVI/DP
- Serial	-	-	-
- Part Number	AES-FMC-ISMNET-G	AES-FMC-IMAGEOV-G	AES-FMC-DVI-G
- FRU File ID	-	-	-