



AN 742: PMBus SmartVID Controller Reference Designs

AN-742
2017-05-08

 **Subscribe**

 **Send Feedback**



Contents

1 PMBus SmartVID Controller Reference Designs.....	3
1.1 Adapter Logic.....	4
1.2 Supported Modes.....	4
1.2.1 Multi-Master Mode.....	5
1.3 Commands.....	5
1.4 Hardware and Software Requirements.....	5
1.5 Running the Designs.....	6
1.5.1 Running the Design in Hardware.....	6
1.5.2 Running the Design in Software.....	6
2 Document Revision History.....	9



1 PMBus SmartVID Controller Reference Designs

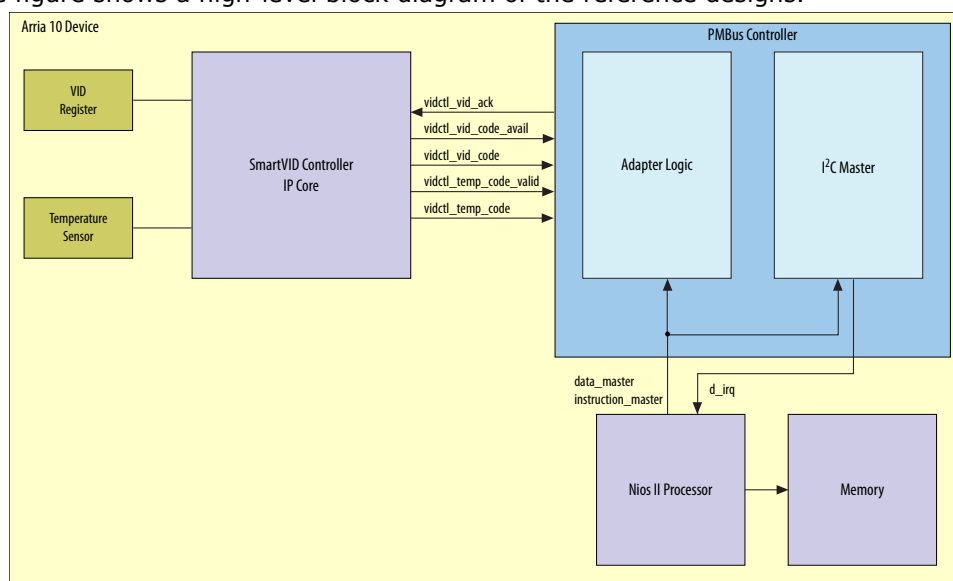
The PMBus* SmartVID Controller reference designs show you how to interface the SmartVID Controller IP core with the Power Management Bus or PMBus control system.

The reference designs demonstrate the following modes: PMBus Master, PMBus Slave, and PMBus Multi-Master.

The PMBus is an open standard protocol that provides a way to communicate with power conversion and other devices. In this reference design, the PMBus system is in accordance to the *PMBus Specification revision 1.2*.

Figure 1. PMBus SmartVID Controller Reference Design

The figure shows a high-level block diagram of the reference designs.



The PMBus SmartVID Controller reference design consists of both conduit and Avalon Memory-Mapped (Avalon-MM) interfaces.

- The SmartVID Controller IP core and the PMBus controller connect through a conduit signal interface.
- The PMBus controller and the Nios II processor communicate using the Avalon[®]-MM interface.

Related Links

[SmartVID Controller IP Core User Guide](#)

Provides more information about the SmartVID Controller IP core.

1.1 Adapter Logic

The Adapter Logic module accepts the SmartVID outputs, stores them in a control and status register (CSR), and passes on the outputs to the Nios II processor through the Avalon-MM interface.

Figure 2. Adapter Logic Module

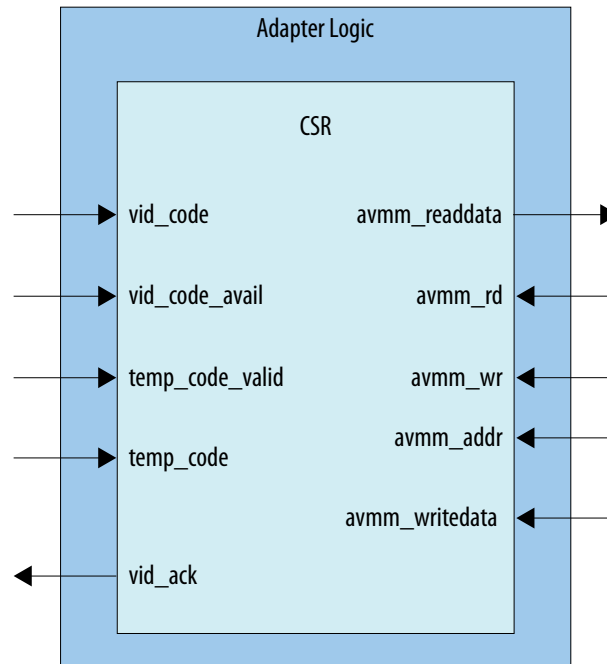


Table 1. Configuration Register for Adapter Logic Module

Bit	Register	Offset	Direction	Reset Value	Description
31:18	Reserved	0x4	RO	All 0	Reserved.
17:8	temp_code	0x3	RO	All 0	Temperature code—indicates the temperature code produced by the Temperature Sensor IP core.
7	vid_ack	0x2	WO	0	Receives a pulse from your controller when the vid_code signal is sampled and sent to the voltage regulator.
6:1	vid_code	0x1	RO	Default	Indicates the VID code produced by the SmartVID Controller IP core.
0	vid_code_avail	0x0	RO	0	Samples the VID code.

1.2 Supported Modes

In a PMBus system, the Intel® Arria® 10 device acts as a PMBus master or a PMBus slave depending on your design choice.



- PMBus Master Mode
 - In this mode, only one PMBus device acts as a master on the bus.
- PMBus Slave Mode
 - In this mode, the PMBus device receives or responds to a command.
- PMBus Multi-Master Mode
 - In this mode, multiple masters drive the bus.

Related Links

[PMBus Control System](#)

Provides more information about the supporting modes.

1.2.1 Multi-Master Mode

When multiple devices start to communicate at the same time, the device writing the most zeros to the bus or the slowest device wins the arbitration. The other devices immediately discontinue any operation on the bus.

When there is an ongoing bus communication, all devices must detect the communication and not interrupt it. The devices must wait for a stop condition to appear before starting communication to the bus.

In this mode, all masters must be multi masters in a multi-master system. Single-master systems may not understand the arbitration and busy detection mechanisms causing unpredictable results.

1.3 Commands

The reference design supports the command language stated in the *PMBus Specification revision 1.2*.

Table 2. PMBus SmartVID Controller Reference Design Commands

Command Code	Command Name	Transaction Type (Writing Data)	Transaction Type (Reading Data)	Number of Data Bytes
20'h	VOUT_MODE	Write Byte	Read Byte	1
21'h	VOUT_COMMAND	Write Word	Read Word	2
78'h	STATUS_BYTE	Write Byte	Read Byte	1
79'h	STATUS_WORD	Write Word	Read Word	2
8D'h	READ_TEMPERATURE_1	N/A	Read Word	2
03'h	CLEAR_FAULTS	Send Byte	N/A	0

1.4 Hardware and Software Requirements

The reference designs require the following hardware and software.



- SmartVID Controller IP core
- Temperature Sensor IP core
- Intel Quartus® Prime software version 15.1

1.5 Running the Designs

To run the designs, download the provided zip files.

You can obtain the PMBus SmartVID Controller reference designs from the following zip files:

- [Master Mode design](#)
- [Slave Mode design](#)
- [Multi-Master design](#)

The zip files contain all the files necessary to demonstrate the three different modes. Unzip the files in the working directory you designate for this project.

1.5.1 Running the Design in Hardware

To run the reference design in hardware, follow these steps:

1. Open the project file in the unzipped folder using the Quartus Prime software.
2. Compile the design. On the **Processing** menu, click **Start Compilation**.
3. Download the generated SRAM Object File (.sof) when compilation completes.

1.5.2 Running the Design in Software

To run the reference designs in software, follow these steps:

1. Launch Nios II Software Build Tools (SBT) for Eclipse.
2. Import the project files from the software folder.
 - a. In the Nios II SBT for Eclipse window, open **File ► Import ► Nios II Software Build Tools Project ► Import Nios II Software Build Tools Project**.
 - b. Select the required files.



- Master Mode
 - a10_pmbus
 - a10_pmbus_bsp
 - Slave Mode
 - a10_pmbus_master_slave0
 - a10_pmbusmaster_slave0_bsp
 - a10_pmbusmaster_slave1
 - a10_pmbusmaster_slave1_bsp
 - Multi-Master Mode
 - a10_pmbus_multimaster0
 - a10_pmbus_multimaster0_bsp
 - a10_pmbus_multimaster1
 - a10_pmbus_multimaster1_bsp
3. Regenerate the Board Support Package (BSP).
- a. Right click **<file_name>_bsp > Nios II > Generate BSP.**
 - Master Mode
 - a10_pmbus_bsp
 - Slave Mode
 - a10_pmbusmaster_slave0_bsp
 - a10_pmbusmaster_slave1_bsp
 - Multi-Master Mode
 - a10_pmbus_multimaster0_bsp
 - a10_pmbus_multimaster1_bsp
4. Open the following the files in the Nios II command shell.
- Master Mode
 - a10_pmbus
 - Slave Mode
 - a10_pmbus_slave0
 - a10_pmbus_slave1
 - Multi-Master Mode
 - a10_pmbus_multimaster0
 - a10_pmbus_multimaster1
5. Open the downloaded .sof and program the software.
- a. In the Nios command shell, type the following:



- Master Mode

- a10_pmbus:

```
nios2-download -g a10_pmbus.elf && nios2-terminal
```

- Slave Mode

- a10_pmbus_slave0:

```
nios2-download -g a10_pmbus_slave0.elf -i 0 && nios2-terminal -i 0
```

- a10_pmbus_slave1:

```
nios2-download -g a10_pmbus_slave1.elf -i 1 && nios2-terminal -i 1
```

- Multi-Master Mode

- a10_pmbus_multimaster0:

```
nios2-download -g a10_pmbus_multimaster0.elf -i 0 && nios2-terminal -i 0
```

- a10_pmbus_multimaster1:

```
nios2-download -g a10_pmbus_multimaster1.elf -i 1 && nios2-terminal -i 1
```




2 Document Revision History

The table shows the revision history for this application note.

Table 3. Document Revision History

Date	Version	Changes
May 2017	2017.05.08	<ul style="list-style-type: none">• Corrected typo: Slaver mode to Slave mode.• Rebranded as Intel.
November 2015	2015.11.02	Initial release.