Enhanced Closed-loop Trim with I²C Control for Platform Manager Devices

December 2010 **Reference Design RD1078**

Introduction

Modern microprocessors and FPGAs require accurate power supply voltages, often more accurate than the tolerances provided by commodity low-dropout regulators (LDOs) and DC-to-DC power supplies. IR voltage drops incurred along the PCB traces as well as ground shifts also contribute to voltage errors, and also vary as a function of instantaneous current demand. This combination of errors can make it exceedingly difficult to implement a power distribution system with untrimmed power components, or even with end-of-line trim and calibration.

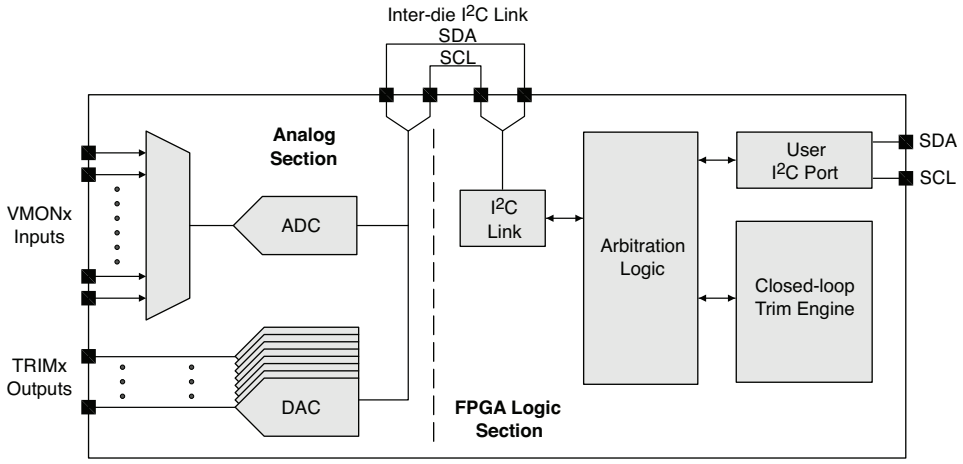
Closed-loop trim is a technique that solves many of the problems associated with providing accurate power by measuring the actual voltage provided by LDO or DC-to-DC converter, and providing a control signal that dynamically and continuously adjusts the output voltage to a desired target value.

While the Platform Manager™ products provide a highly effective hardware closed-loop trim engine built into the analog section of the device, there are situations that require additional levels of flexibility. Specifically, it is often desirable to be able to control trim functions dynamically from a host microcontroller, with the ability to enable or disable individual trim channels or change the trim targets on-the-fly. This reference design was developed to address these advanced requirements.

Theory of Operation

Figure 1 shows the top-level organization of the closed-loop trim reference design as implemented on a Platform Manager IC. The control and communications logic implemented in the FPGA communicates to the ADC and DAC hardware functions in the analog section over an I²C link bus optimized for that purpose.

Figure 1. Platform Manager Closed-Loop Trim Reference Design Top-Level Diagram



Some of the key features of this reference design are:

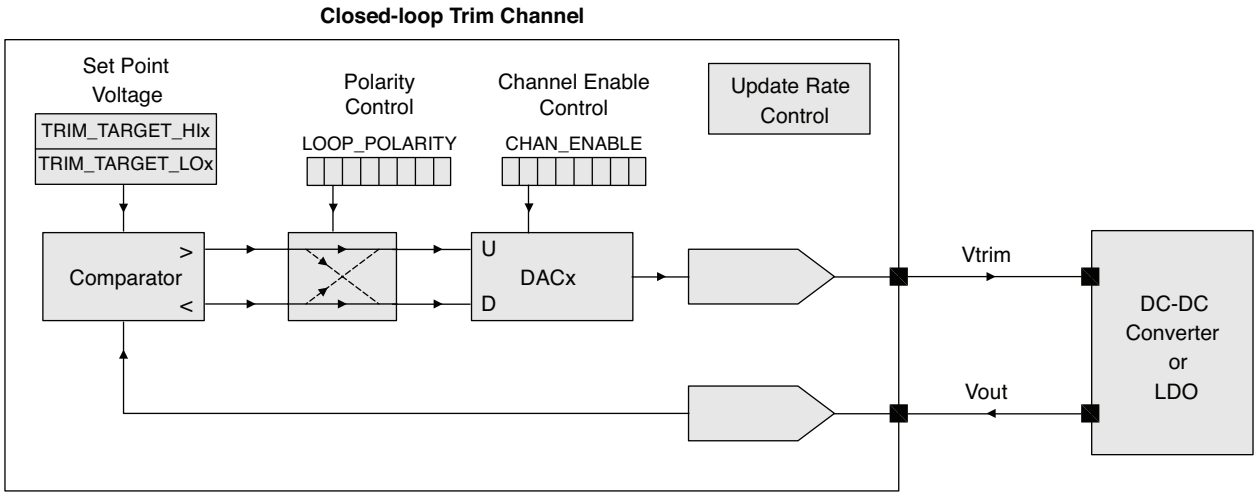
1. Trim channels may be independently enabled.
2. Trim channel target voltages may be modified on-the-fly.
3. I²C user control interface.

Closed-loop Trim Engine

The purpose of closed-loop trim is to be able to precisely adjust the voltage output of a power supply module to compensate for manufacturing and environmental variations. This function can be extremely useful in high-performance systems where supply voltage drops and ground shift can make open-loop techniques ineffective.

Figure 2 shows a functional block diagram of a power supply module being controlled through the closed-loop trim technique. The output voltage (Vout) of a power supply module, typically either a DC-to-DC converter or linear regulator, is measured by an ADC converter. The resultant value is then compared to a pre-determined 'set point' voltage. If the measured value is different than the set point voltage, a trim voltage supplied to the power supply module (Vtrim) is adjusted. The adjustment is made on the basis of the difference between set point voltage, measured voltage and polarity of the power supply module's response to a trim signal. For example, assume the power supply module's output voltage increases in response to an increasing trim voltage. In this case, if Vout is less than the set point voltage, one would increment the DAC count register and increase the Vtrim voltage. This process would be repeated until the Vout voltage equaled the set point voltage. If Vout were higher than the setpoint voltage, the DAC count would be decremented, and Vtrim reduced, until Vout equaled the set point voltage.

Figure 2. Closed-Loop Trim Detail



The IP core provides up to eight independent channels. Each channel is associated with a single, fixed DAC and ADC input (VMON) input. For example, Closed-loop trim channel 1 reads Vout through VMON1 and adjusts Trim DAC1.

All closed-loop trim operations are configured and controlled through commands sent through the user I²C port.

Configuration Parameters

To realize this reference design requires the user to set a number of configuration parameters through HDL definition statements (e.g. `define in Verilog). These parameters appear near the top of the HDL file and are set to defaults in the design files provided in the reference design that are compatible with trimming the Platform Manager Evaluation Board's LDO and DC-to-DC converter. Listing 1 shows the parameters and their default values.

Listing 1. Configuration Parameters

```
`define   ANALOG_DIE_ADDR        7'h20
`define   I2C_ADDR               7'h20

`define   TRIM_UPDATE_RATE       10

`define   LOOP_POLARITY_1        0
`define   LOOP_POLARITY_2        0
```

```
`define   LOOP_POLARITY_3        0
`define   LOOP_POLARITY_4        0
`define   LOOP_POLARITY_5        1      // Set to '1' to accomodate
`define   LOOP_POLARITY_6        1      // Eval Board LDO and DC-DC polarity
`define   LOOP_POLARITY_7        0
`define   LOOP_POLARITY_8        0

`define   INIT_VALUE_DAC_1       128
`define   INIT_VALUE_DAC_2       128
`define   INIT_VALUE_DAC_3       128
`define   INIT_VALUE_DAC_4       128
`define   INIT_VALUE_DAC_5       128
`define   INIT_VALUE_DAC_6       128
`define   INIT_VALUE_DAC_7       128
`define   INIT_VALUE_DAC_8       128
```

The function of each of the configuration parameters is as follows:

- **ANALOG_DIE_ADDR** – This is the I²C device address for the analog die. This value must also be programmed into the analog section using the I²C Configuration dialog (see Miscellaneous Configuration section)
- **I2C_ADDR** – This parameter sets the I²C device address for the user I²C interface logic (located in the FPGA fabric).
- **TRIM_UPDATE_RATE** – This parameter specifies the time interval (in milliseconds) at which enabled trim channels will be updated. 10msec is a recommended lower bound.
- **LOOP_POLARITY_x** – These parameters specify the feedback sense of the external circuitry to be controlled by the trim loop. A '0' tells the trim engine that the VMON input voltage for a trim channel can be expected to increase with increasing TrimDAC output voltage, while a '1' tells the trim engine that the VMON input voltage for a trim channel can be expected to decrease with increasing TrimDAC output voltage.
- **INIT_DAC_VALUE_x** – These parameters specify the initial value to be used for the TrimDACs when a trim channel is first enabled. Selecting an initial DAC value close to the final value can dramatically reduce the time required for a trim channel to settle.

Control Registers

This reference design is controlled and monitored through writes and reads with respect to a group of data and control registers. Table 1 lists these registers. Detailed descriptions of the operation of the different registers can be found following the table.

Table 1. Closed-loop Trim Engine Registers

Name	Addr (Hex)	Width	Access	Description
Analog Section Virtual Registers				
VMON_STATUS0	0x00	8	R	Writes to these register addresses are passed through to the corresponding registers in the analog section of Platform Manager, and reads from these register addresses read from the corresponding analog section registers. For further details on the operation of these registers, refer to the Platform Manager Data Sheet.
VMON_STATUS1	0x01	8	R	
VMON_STATUS2	0x02	8	R	
OUTPUT_STATUS0	0x03	8	R	
OUTPUT_STATUS1	0x04	8	R	
OUTPUT_STATUS2	0x05	8	R	
INPUT_STATUS	0x06	8	R	
ADC_VALUE_LOW	0x07	8	R	
ADC_VALUE_HIGH	0x08	8	R	
ADC_MUX	0x09	8	RW	
UES_BYTE0	0x0A	8	R	
UES_BYTE1	0x0B	8	R	
UES_BYTE2	0x0C	8	R	
UES_BYTE3	0x0D	8	R	
GP_OUTPUT1	0x0E	8	RW	
GP_OUTPUT2	0x0F	8	RW	
GP_OUTPUT3	0x10	8	RW	
INPUT_VALUE	0x11	8	RW	
RESET	0x12	8	W	
TRIM1_TRIM	0x13	8	RW	
TRIM2_TRIM	0x14	8	RW	
TRIM3_TRIM	0x15	8	RW	
TRIM4_TRIM	0x16	8	RW	
TRIM5_TRIM	0x17	8	RW	
TRIM6_TRIM	0x18	8	RW	
TRIM7_TRIM	0x19	8	RW	
TRIM8_TRIM	0x1A	8	RW	

Table 1. Closed-loop Trim Engine Registers (Continued)

Name	Addr (Hex)	Width	Access	Description
Closed-loop Trim Control Registers				
TRIM_TARGET_HI1	0x20	8	RW	Trim Target and ADC Range Settings for each channel.
TRIM_TARGET_LO1	0x21	8	RW	
TRIM_TARGET_HI2	0x22	8	RW	
TRIM_TARGET_LO2	0x23	8	RW	
TRIM_TARGET_HI3	0x24	8	RW	
TRIM_TARGET_LO3	0x25	8	RW	
TRIM_TARGET_HI4	0x26	8	RW	
TRIM_TARGET_LO4	0x27	8	RW	
TRIM_TARGET_HI5	0x28	8	RW	
TRIM_TARGET_LO5	0x29	8	RW	
TRIM_TARGET_HI6	0x2A	8	RW	
TRIM_TARGET_LO6	0x2B	8	RW	
TRIM_TARGET_HI7	0x2C	8	RW	
TRIM_TARGET_LO7	0x2D	8	RW	
TRIM_TARGET_HI8	0x2E	8	RW	
TRIM_TARGET_LO8	0x2F	8	RW	
DAC1	0X30	8	RW	Current Closed-loop trim Output DAC settings for each channel.
DAC2	0X31	8	RW	
DAC3	0X32	8	RW	
DAC4	0X33	8	RW	
DAC5	0X34	8	RW	
DAC6	0X35	8	RW	
DAC7	0X36	8	RW	
DAC8	0X37	8	RW	
CHAN_ENABLE	0x38	8	RW	Enable Closed Loop trim for each channel.
- unused -	0x39	8	R	Reserved for future use. Reads as 0x00.
LOOP_POLARITY	0x3A	8	RW	Loop Polarity control for each trim channel.

Analog Virtual Registers

Read and write operations performed on the registers between addresses 0x00-0x1A map to identical operations on the corresponding physical registers in Platform Manager's analog section. When directly accessing the analog hardware it is important to be aware of the possibility of contention between user accesses and accesses by the FPGA-based reference design logic. For example, if one has enabled a closed-loop trim channel and attempts to write to the associated DAC register, the value will be quickly overwritten by the FPGA-based logic.

TRIM_TARGET_HIx & TRIM_TARGET_LOx Register Descriptions

Figure 3. Trim Target Register Pairs

TRIM_TARGET_HIx

ATTEN	X	X	X	D11	D10	D9	D8
b7	b6	b5	b4	b3	b2	b1	b0

TRIM_TARGET_LOx

D7	D6	D5	D4	D3	D2	D1	D0
b7	b6	b5	b4	b3	b2	b1	b0

The trim target registers (Figure 3) are used to control the target setpoint of each trim loop referenced to the ADC measurement code. A 12-bit value is written to these registers indicating the trim target for the channel associated with the register pair. Additionally, the ADC attenuation is set here for each channel, with '1' indicating an ADC range of 0-2V and '0' indicating an ADC range of 0-6V. Note that regardless of the ATTEN bit setting, all ADC measurements are mapped into the 0-6V range. Setting the ATTEN bit to '0' increases the effective resolution when measuring low voltages.

DACx Register Descriptions

Figure 4. DACx Registers

D7	D6	D5	D4	D3	D2	D1	D0
b7	b6	b5	b4	b3	b2	b1	b0

The DAC registers (Figure 4) are used primarily for monitoring purposes, as they are updated internally by the trim control loop. They can be set to a preferred initial value prior to initiating a closed-loop trim operation, but should not be set while the trim system is enabled for that channel as doing so may cause abrupt changes in the controlled power supply voltage.

CHAN_ENABLE Register Description

Figure 5. Chan_Enable Register

CHEN8	CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1
b7	b6	b5	b4	b3	b2	b1	b0

In the Channel enable register (Figure 5) setting the bit corresponding to a closed-loop trim channel to '1' initiates the trim operation on that channel. Resetting the bit to '0' stops the stops the trim operation, leaving the trim voltage at the level it was at when the trim operation ceased.

LOOP_POLARITY Register Description

Figure 6. LOOP_POLARITY

POL8	POL7	PO6	POL5	POL4	POL3	POL2	POL1
b7	b6	b5	b4	b3	b2	b1	b0

The bits of the LOOP_POLARITY register (Figure 6) control the feedback polarity for the corresponding trim channel. A '0' bit indicates that the trim engine should assume positive gain between the DAC and ADC for that channel, and the DAC count is incremented if the ADC measurement is less than the trim target value, and decremented if the ADC measurement is greater than the trim target value. In contrast, a '1' bit indicates that the trim engine should assume negative gain between the DAC and ADC for that channel, and the DAC count is decremented if the ADC measurement is less than the trim target value, and incremented if the ADC measurement is greater than the trim target value.

Configuring a Trim Channel

This reference design is intended to be used with an external microcontroller that is capable of communicating through an I²C port. In its startup configuration, all trim targets are set to the lowest voltage (0.000V) and no trim channels are active. The general procedure for configuring a given trim channel requires the following series of configuration steps to be performed through I²C write commands:

1. Write the desired trim target value and ADC range setting to channel's TRIM_TARGET_HI and TRIM_TARGET_LO registers.
2. If not configured in HDL header, write loop polarity information to channel's bit in LOOP_POLARITY register.
3. If not configured in HDL header, write initial DAC value to channel's DAC register (optional).
4. Write '1' to the channel's associated bit in the CHAN_ENABLE register. (starts trim process)

For some concrete examples, let us consider how to perform closed-loop trim on the LDO and DC-to-DC converter provided on the Platform Manager Evaluation Board through I²C commands. The following examples assume that no configuration has been defined in the HDL file other than suitable I²C device addresses.

LDO Example

The sequence of I²C writes shown below can be used to set up the LDO to trim to a 2.5V target. This requires setting up trim channel 6 with inverse polarity, setting the attenuator to the 0-6.144V range, setting the 2.5V target (0x4E2).

1. Write 0x20 to register 0x3A (Set up polarity control)
2. Write 0x80 to register 0x35 (Initial DAC value)
3. Write 0x84 to register 0x2A (ADC attenuator set and high nibble of target 0x4E2)
4. Write 0xE2 to register 0x2B (Low byte of target value 0x4E2)
5. Write 0x20 to register 0x38 (Enable trim channel 6)

DC-to-DC Example

A similar sequence shown below can be used to configure the DC-to-DC converter connected to trim channel 5 to a target of 1.25V (0x271):

1. Write 0x10 to register 0x3A (Set up polarity control)
2. Write 0x80 to register 0x34 (Initial DAC value)
3. Write 0x02 to register 0x28 (ADC attenuator set and high nibble of target 0x271)
4. Write 0x71 to register 0x29 (Low byte of target value 0x271)
5. Write 0x10 to register 0x38 (Enable trim channel 5)

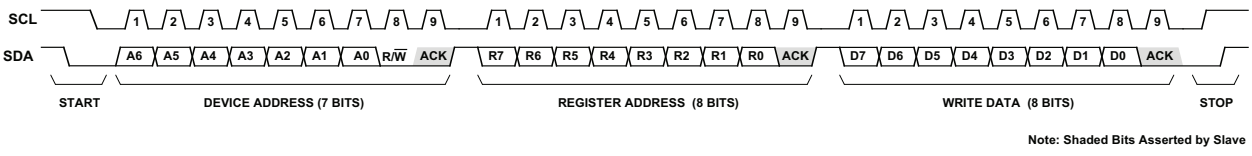
Note that if you wish to simultaneously trim both converters, then the data written to the polarity and enable registers must be modified so as to manage both channels.

User I²C Interface and Protocol

The I²C protocol used by this reference design is based on, and similar to, that implemented in the Platform Manager's analog section I²C interface.

To write data to a register, the Closed-loop Trim requires a 3-byte write operation where the first byte specifies the IP's device address, the second byte specifies the register address to be written to, and the third and final byte specifies the data to be written. Figure 7 shows the SDA and SCL waveforms for a register write operation.

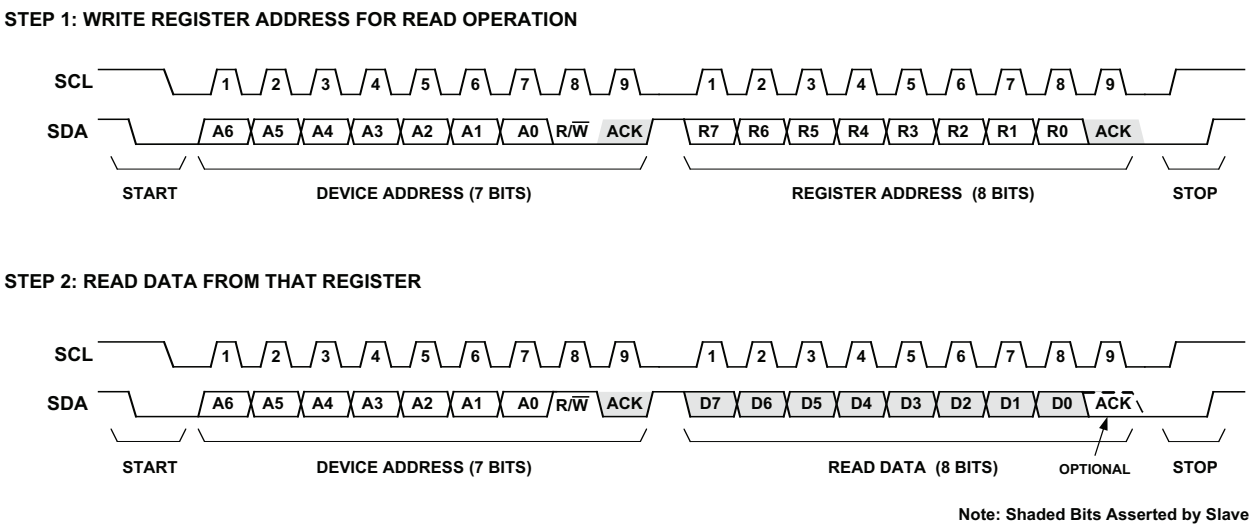
Figure 7. I²C Register Write Waveforms



The reference design FPGA logic communicates its readiness through the I²C ACK bit. If the user transmits a write command through the I²C port and the FPGA logic is not ready to accept the command, it will not assert the ACK bit. This behavior must be taken into account when designing software to communicate to the FPGA logic (e.g. planning for re-try operations).

Reading a register through the I²C port requires two separate I²C data frames. In the first frame, two bytes are transmitted as a 'write' operation with the device address and the register address. In the second frame, two more bytes are transmitted as a 'read' operation with the device address in the first byte, and the returned data transmitted by the FPGA logic in the second byte. Figure 8 shows the details of these two transmission cycles.

Figure 8. I²C Register Read Waveforms



In the case of a read operation, the ACK bit is only asserted when the logic successfully accepts a communication attempt. Because the logic relays certain register read requests to the Platform Manager hardware, there is a finite time elapsed between finishing step 1 of the read operation and when data becomes available in the logic to be read in step 2. If the data is not available, the logic will not assert the ACK bit in response to the step 2 read request. The user can repetitively poll the logic until data becomes available.

A comprehensive overview of I²C communications is a complex subject that is beyond the scope of this document. For further details, the reader should consult the I²C-Bus Specification and User Manual published by NXP Semiconductors.

Design Details

The FPGA logic section of the reference design is realized by instantiating the Closed_Loop_Trim module. As all communications in and out of this module are in serial form (I²C), relatively few I/O connections are required. Listing 2 shows the Verilog module I/O definition, and Table 2 describes the pin assignments that are suggested for evaluation on the Platform Manager Evaluation Board.

Listing 2. Top-level Module Definition

```
module Closed_Loop_Trim (

   // User Serial Communication Ports

   input       I2C_Clk,       // User I2C CLK  (input)
   inout       I2C_Sda,       // User I2C SDA  (bidirectional)


   // Connections to analog die & reset

   input       ADX_Clk,       // 8 MHz clock from analog die (input)
   input       ADX_Rstb,      // Reset signal (input, active low)
   output      ADX_Scl,       // I2C CLK to analog die (output)
   inout       ADX_Sda        // I2C SDA to analog die (bidirectional)

);
```

Table 2. Pin Assignments

Pin	Signal	Function	Notes
A2	ADX_Clk	8 MHz clock (input)	Connected to pin B11
F4	ADX_Rstb	Reset (input, active low)	Connected to push-button S1
D1	ADX_Sda	Analog die I²CSDA (bi-directional)	Connect to pin A12
D2	ADX_Scl	Analog die I²C SCL (input)	Connect to pin B12
D4	I2C_Clk	User I²C SCL (input)	Connected to header pin SSCL (J15.23)
D3	I2C_Sda	User I²C SDA (bi-directional)	Connected to header pin SSDA (J15.21)
B11	MCLK	8 MHz clock (output)	Master clock source, connect to pin A2
A12	SDA	Analog die I²C SDA (bi-directional)	Connect to pin D1
B12	SCL	Analog die I²C SCL (input)	Connect to pin D2

Using the Reference Design with PAC-Designer®

Although the reference design's HDL may be implemented with any Lattice synthesis flow that supports Platform Manager, it is primarily intended for use with PAC-Designer. This section outlines the configuration steps required to incorporate the HDL in a PAC-Designer design.

The following procedure can be used to import the HDL part of the reference design into PAC-Designer:

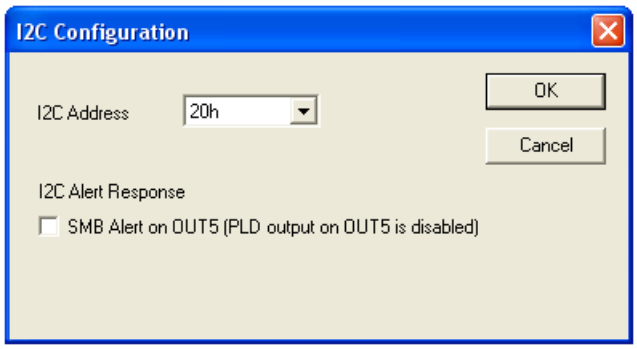
1. Start PAC-Designer 6.0
2. Create a **New Platform Manager Design** or open an existing design
3. Click on **FPGA Logic Window** opening LogiBuilder
4. Select **File > Import Sub Module > Configure Sub Module** from the main menu
5. Open Port Configuration dialog by clicking on the **<Port Mapping>** button in the Module Definition dialog
6. Assign signal names to port names. When all ports are assigned, click **<OK>**.
7. Click **<OK>** in the Module Definition dialog.

In addition to importing the reference design HDL and assigning I/O pins, the CPLD and analog sections of the Platform Manager device must be configured. Specifically:

1. The analog die I²C address must be set to match the ANALOG_DIE_ADDR parameter.
2. The Master Clock must be set to provide an externally available clock.
3. TrimCells intended for closed-loop trimming must be suitably configured.
4. A minimal amount of logic must be implemented in the CPLD to enable compilation of the design.

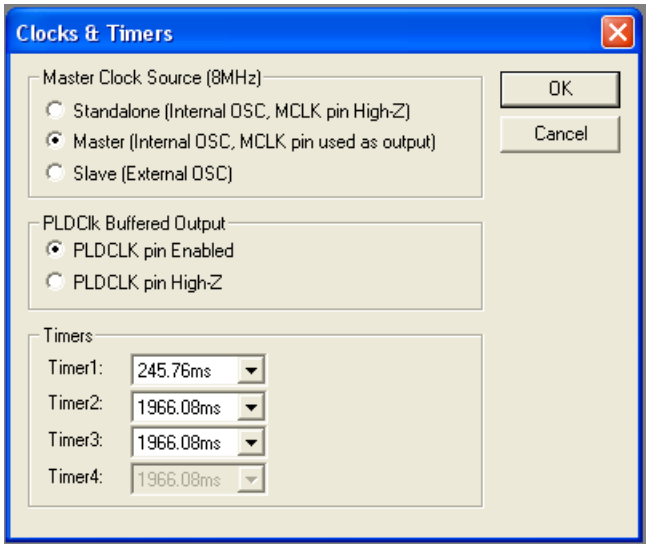
The device address for the analog die must be set so it matches the ANALOG_DIE_ADDR parameter in the HDL source file. This enables communications between the FPGA and analog die. This is accomplished through the PAC-Designer I²C Configuration dialog, shown in Figure 9. Please note that this address need not be the same as the I²C address defined for the user I²C interface implemented in the Platform Manager's FPGA section.

Figure 9. Analog Die I²C Configuration Dialog



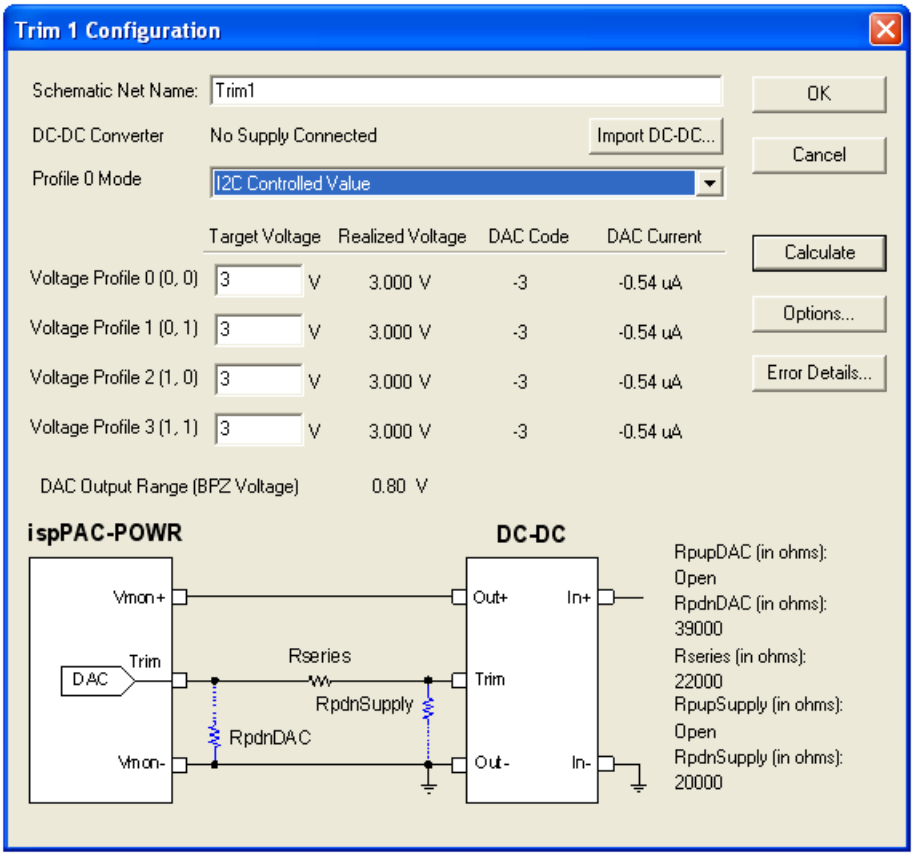
The logic in the FPGA section requires an 8MHz clock to function. Normally this clock can be provided from Platform Manager's MCLK clock output. To use this resource requires that the MCLK pin be configured so as to provide a clock output. This is done by setting the Master Clock Source option to **Master** using the Clocks & Timers dialog, as shown in Figure 10.

Figure 10. Master Clock Source Configuration Dialog



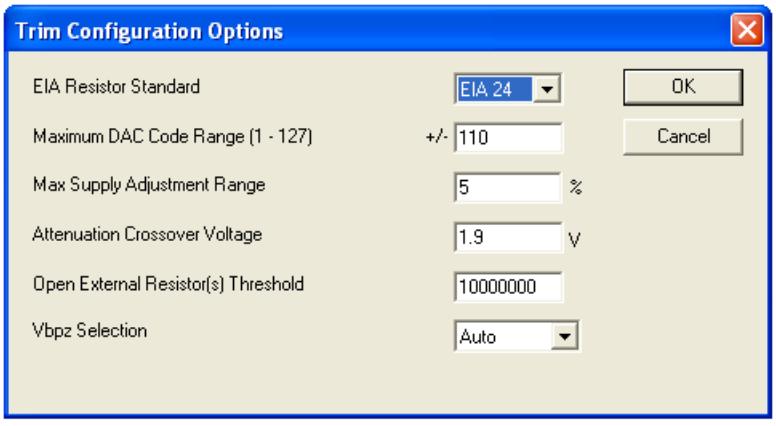
For each trim channel you wish to control through this reference design you will need to configure its associated analog circuitry. This is accomplished through the Trim Configuration dialog (Figure 11). The first thing that needs to be done is to set the 'Profile 0 Mode' option to **I²C Controlled Value**. This allows the DAC hardware to be controlled through the analog I²C port instead of the analog section's internal trim control logic.

Figure 11. Trim Configuration Dialog



The second item that must be configured is the Vbpz offset. If you use the Trim Configuration dialog to help design the external interface to the LDO or DCC-to-DC converter you are using, this will be automatically set. Alternately, you can also click on the **<Options…>** button to open the Trim Configuration Options dialog (Figure 12) and manually set this value.

Figure 12. Trim Configuration Options Dialog



For more detail on how to use the Trim Configuration dialog and how to design the external circuitry needed to successfully interface to external power components, please refer to application note AN6074, Interfacing the Trim Output of Power Manager II Devices to DC-DC Converters.

Finally, a minimal amount of CPLD logic is required to use this reference design. Specifically, it is necessary to ensure that the internal PLD_VPSx signals are set to '0' to permit the TrimDACs to be controlled from the I²C port. This easiest way to do this is by adding two Supervisory Equations to the CPLD logic as shown in Figure 13.

Figure 13. Controlling PLD_VPS0 and PLD_VPS1 through Supervisory Equations

Equation	Supervisory Logic Equation	Macrocell Configuration
EQ 0	PLD_VPS0 = 0	Output, combinatorial [non-registered]
EQ 1	PLD_VPS1 = 0	Output, combinatorial [non-registered]
	<end-of-supervisory-logic-table>	

Additionally, when using the LDO or DC-to-DC converter on the Evaluation Board, it is necessary to enable them. This is most easily done by issuing a LogiBuilder OUTPUT command to set OUT11 and OUT12 to a low state (=0).

Validation

Because this reference design was designed for implementation on the Platform Manager Evaluation Board, it is straightforward to validate in hardware. The default configuration is designed to provide closed-loop trim functions for the Platform Manager Evaluation Board's on-board LDO (hardwired to the TRIMDAC6/VMON6 trim channel) and DC-to-DC converter (hardwired to the TRIMDAC5/VMON5 trim channel). To exercise the design requires the use of an external I²C master with appropriate software.

Implementation

Device	FPGA LUTs	FPGA Slices	FPGA I/O	CPLD Macrocells	CPLD CTimers	CPLD Product Terms	CPLD I/O	VMONs
LPTM10-12107	507	254	6	n/a	n/a	n/a	n/a	Note 1

1. While this design reads VMON voltages, it does not prevent the VMON inputs from being used by the CPLD logic.

References

- DS1036, Platform Manager Data Sheet
- I²C-Bus Specification and User Manual, Rev. 03, 19 June 2007, NXP Semiconductors
- EB58, Platform Manager Development Kit User's Guide
- AN6074, Interfacing the Trim Output of Power Manager II Devices to DC-DC Converters

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
December 2010	01.0	Initial release.