Temperature Monitor Using Platform Manager Devices

October 2010 Reference Design RD1080

Introduction

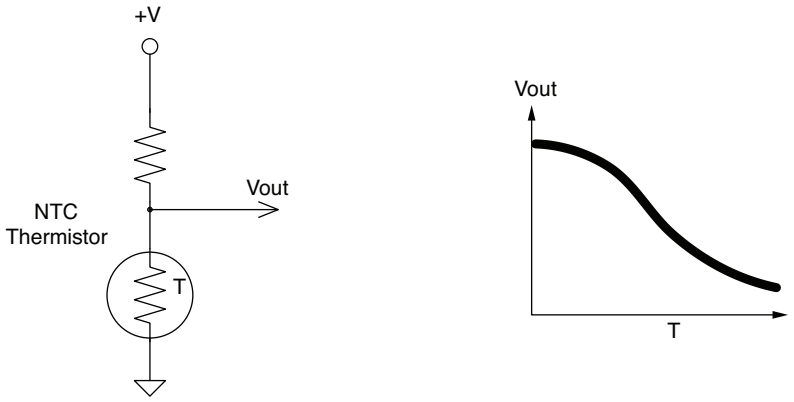
The PAC-Designer® LogiBuilder design language provides Platform Manager™ devices with the ability to monitor the binary status of analog voltage inputs with respect to a predetermined threshold and use that information to make conditional decisions. For more advanced user functions it may be desirable to be able to process the actual voltage value. To do so requires controlling the Platform Manager's on-chip analog-to-digital converter (ADC) and transferring the results of conversions to one or more registers in the device's FPGA fabric. This reference design provides a simple, but complete example of how to access the Platform Manager's ADC. This reference design is designed to use sources such as thermistors and slider pots to provide analog input signals, and a 3-digit LCD to display the measured values. While intended for general-purpose use, this reference design has been implemented to operate on the Platform Manager Evaluation Board and utilize its hardware resources. The modular implementation of this reference design makes it straightforward to adapt to other hardware configurations or as part of larger designs.

Theory of Operation

To monitor temperature requires the use of a temperature sensor. There are a wide range of technologies from which to choose, but for most on-board applications, the choices will typically be between diode junctions, monolithic sensors (which include on-board signal processing), and thermistors. For cost-sensitive application where extreme accuracy is not required, thermistors will often be the best choice, offering moderate precision (+/- 2°C), and simplicity of use for less than US$0.05 in high volume.

Another advantage of thermistors as temperature sensors is that when put into a divider circuit with a resistor, they can provide a voltage output which does not require pre-amplification before being fed into an ADC (Figure 1). Although the resulting output voltage is a non-linear function of temperature, when the output voltage is near +V/2, the output response is usefully linear over a limited temperature range.

Figure 1. Thermistor Temperature Measurement Circuit and Response



Platform Manager devices incorporate an on-chip ADC converter that can measure voltages on any of the VMON input channels, as well as the voltages present on the PVCC and PVCCINP power supply pins. This ADC can be accessed externally through a serial I^2C bus that appears on the SCL and SDA pins.

The ADC conversion process is managed through a series of reads and writes to registers through the I^2C bus. Three registers, shown in Figure 2, are involved in performing a conversion.

Figure 2. ADC Control and Data Registers

0x07 - ADC_VALUE_LOW (Read Only)

D3	D2	D1	D0	1	1	1	DONE
b7	b6	b5	b4	b3	b2	b1	b0

0x08 - ADC_VALUE_HIGH (Read Only)

D11	D10	D9	D8	D7	D6	D5	D4
b7	b6	b5	b4	b3	b2	b1	b0

0x09 - ADC_MUX (Read/Write)

X	X	X	ATTEN	SEL3	SEL2	SEL1	SEL0
b7	b6	b5	b4	b3	b2	b1	b0

A write to the ADC_MUX register (0x09) simultaneously performs three functions:

1. Selects the input channel for the conversion (SEL3..SEL0 bits)
2. Sets the ADC range (ATTEN bit)
3. Initiates the conversion

When the conversion starts, the DONE bit in ADC_VALUE_LOW (0x08) is set to '0', and when the conversion finished, this bit is set to '1'. This bit can be polled through repeated register read cycles to determine when a conversion is complete. Alternatively, a conversion cycle requires a maximum of 200us, so it is also possible to simply start the conversion and wait for this interval before reading back data. When the conversion is finished, valid data is available in the ADC_VALUE_HIGH (0x08) and ADC_VALUE_LOW (0x09) registers. For further details on how the ADC operates, refer to the Platform Manager Data Sheet.

The ADC provides a user-programmable range feature that is specified using the ATTEN bit (register 0x09) on a conversion-by-conversion basis. When the ATTEN bit is set to '0', the effective input range is 0-2V, and when the ATTEN bit is set to '1', the effective input range is 0-6V. Both of the conversions are mapped uniformly onto a 12-bit result, so that a single input voltage will provide approximately the same result code regardless of range setting (assuming the voltage falls within the lower 0-2V range).

The channel on which the conversion should be performed is also selected on a conversion-by-conversion basis. Table 1 lists the input channels that are selected when a given set of SEL[3:0] bits is written (register 0x09).

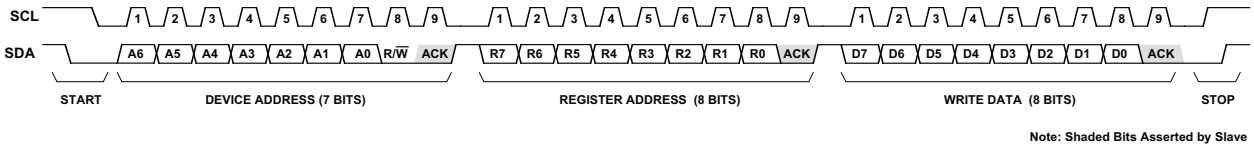
Table 1. ADC Input Channel Selection

SEL3	SEL2	SEL1	SEL0	Input Channel (ExternalPpin)
0	0	0	0	VMON1
0	0	0	1	VMON2
0	0	1	0	VMON3
0	0	1	1	VMON4
0	1	0	0	VMON5
0	1	0	1	VMON6
0	1	1	0	VMON7
0	1	1	1	VMON8
1	0	0	0	VMON9
1	0	0	1	VMON10
1	0	1	0	VMON11
1	0	1	1	VMON12
1	1	0	0	PVCCA
1	1	0	1	PVCCINP

Although the register-level control of the Platform Manager is simple, it must be accessed over the I^2C serial link. Doing this effectively requires understanding the Platform Manager's I^2C protocol for reading and writing registers.

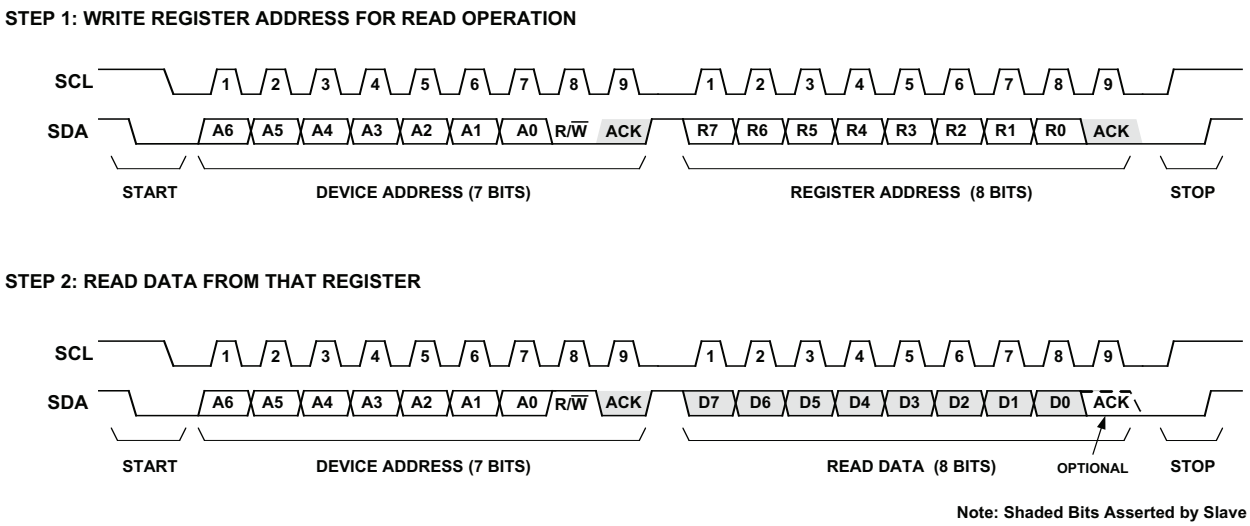
To write data to a register, the Platform Manager requires a 3-byte write operation where the first byte specifies the Platform Manager's device address, the second byte specifies the register address to be written to, and the third and final byte specifies the data to be written. Figure 3 shows the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

Figure 3. Platform Manager I^2C Register Write Waveforms



Reading a Platform Manager register through the I^2C port requires two separate I^2C 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 Platform Manager in the second byte. Figure 4 shows the details of these two transmission cycles.

Figure 4. Platform Manager I²C Register Read Waveforms

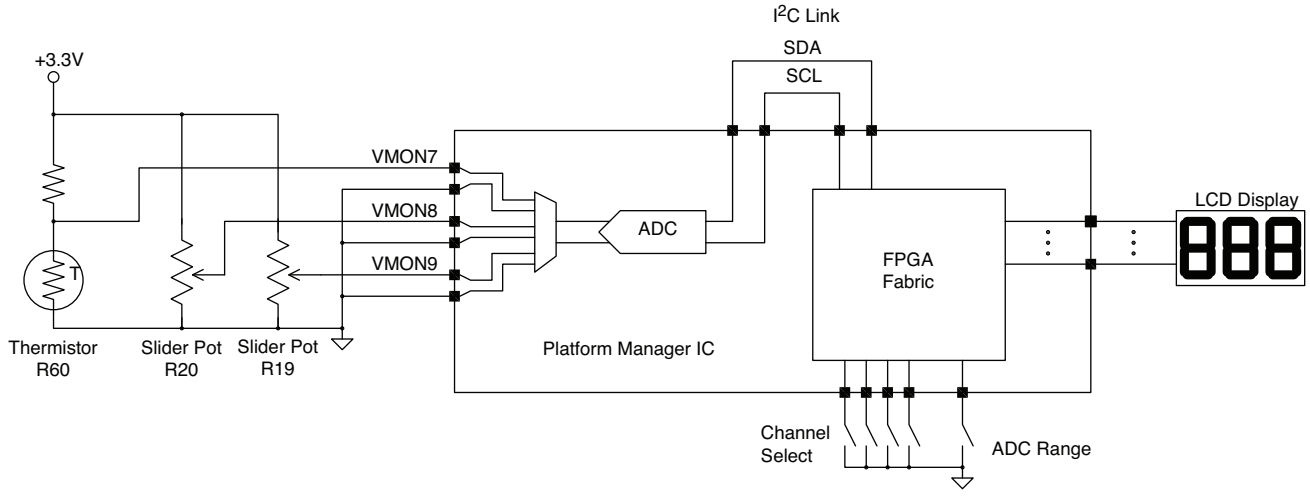


A comprehensive overview of I²C communications is a complex subject that is beyond the scope of this document. For further details, the reader should consult the I²C-Bus Specification and User Manual published by NXP Semiconductors.

Design Details

This reference design reads an ADC input and displays the converted value on a 3-digit 7-segment LCD display. Figure 5 shows a top-level block diagram of the system. The reference design is intended for operation on the Platform Manager Evaluation Board and uses the two slide pots and thermistor as user-variable signal sources and the LCD display as an output device. DIP switches SW9 and SW10 allow the user to select the ADC channel and input range. The SDA and SCL I/Os are connected to the I/O pins controlled by the FPGA fabric.

Figure 5. Top-Level Block Diagram of Reference Design



The top-level module definition for this reference design is shown in Listing 1.

Listing 1. Top-level Module Definitions

```
module DISPLAY_ADC (

      input clk_8mhz,// 8 MHz clock from Analog Section
      input reset_b,// External Reset signal (active low)
      outputscl,// I2C clock output
      inoutsda,// I2C Data I/O

      // Channel Select Switches (active low)

      input chsel0_b,
      input chsel1_b,
      input chsel2_b,
      input chsel3_b,

      // ADC Range Select - 0=[0V..2V], 1=[0V..6V]

      input adc_range,

      // LCD Display Drive outputs

      output bias,  // modulated backbias output
      outputs1a, s1b, s1c, s1d, s1e, s1f, s1g, // Segment drive, MSD
      outputs2a, s2b, s2c, s2d, s2e, s2f, s2g,
      outputs3a, s3b, s3c, s3d, s3e, s3f, s3g // Segment drive, LSD

      );
```

The evaluation board provides all of the external connections necessary to run this reference design without hardware modification. Table 2 lists the key connections needed to implement this design in a new hardware design. Note that in a new design, these functions may be reassigned to accommodate board layout requirements by generating a new LPF file.

Table 2. Pins Associated with FPGA Functions[1]

FPGA Pins	Signal	Function	Notes
A5	chsel0_b	Channel Select 0	Connected to dipswitch SW10
A7	chsel1_b	Channel Select 1	Connected to dipswitch SW10
A9	chsel2_b	Channel Select 2	Connected to dipswitch SW10
B5	chsel3_b	Channel Select 3	Connected to dipswitch SW10
B7	adc_range	ADC Range Select	Connected to dipswitch SW9
A2	clk_8mhz	8 MHz Clock Input	Connected to B11
F4	Reset_b	Reset	Connected to pushbutton S1
D1	sda	FPGA SDA	Connected to A12
D2	scl	FPGA SCL	Connected to B12
P1	bias	LCD bias	LCD display bias
G2	s1a	LCD digit 1, segment a	LCD Segment Drive
G3	s1b	LCD digit 1, segment b	LCD Segment Drive
M3	s1c	LCD digit 1, segment c	LCD Segment Drive
N2	s1d	LCD digit 1, segment d	LCD Segment Drive
N3	s1e	LCD digit 1, segment e	LCD Segment Drive
G1	s1f	LCD digit 1, segment f	LCD Segment Drive
F3	s1g	LCD digit 1, segment g	LCD Segment Drive
H3	s2a	LCD digit 2, segment a	LCD Segment Drive
H4	s2b	LCD digit 2, segment b	LCD Segment Drive
L2	s2c	LCD digit 2, segment c	LCD Segment Drive
L3	s2d	LCD digit 2, segment d	LCD Segment Drive
M1	s2e	LCD digit 2, segment e	LCD Segment Drive
H2	s2f	LCD digit 2, segment f	LCD Segment Drive
G4	s2g	LCD digit 2, segment g	LCD Segment Drive
J3	s3a	LCD digit 3, segment a	LCD Segment Drive
J4	s3b	LCD digit 3, segment b	LCD Segment Drive
K1	s3c	LCD digit 3, segment c	LCD Segment Drive
K2	s3d	LCD digit 3, segment d	LCD Segment Drive
K3	s3e	LCD digit 3, segment e	LCD Segment Drive
J2	s3f	LCD digit 3, segment f	LCD Segment Drive
J1	s3g	LCD digit 3, segment g	LCD Segment Drive

1. Pin assignments are for LPTM10-12107.

Pins associated with analog functions (Table 3), however, are fixed in hardware and must be connected to the indicated pin.

Table 3. Analog Function Pins

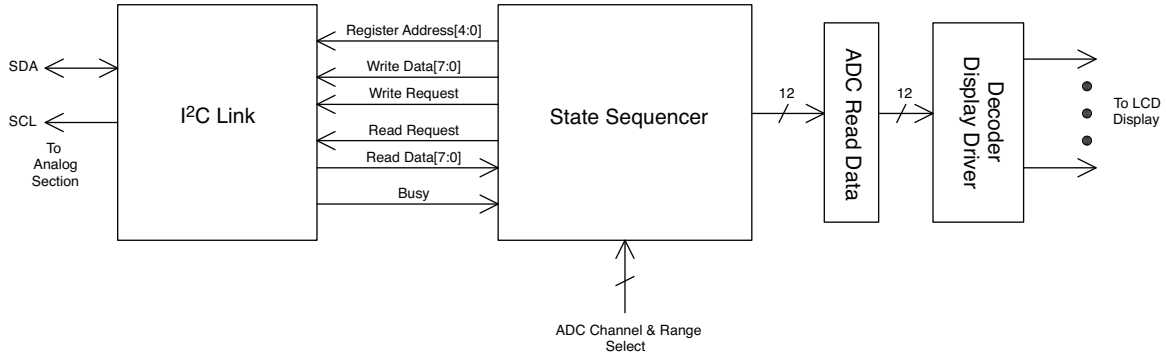
Analog Signal Pin	Function	Notes
B11	MCLK – 8MHz Clock	Master Clock Source
A12	SDA	I²C data – connect to D1
B12	SCL	I²C clock – connect to D2
J16	VMON7P	Connected to thermistor R60
H16	VMON8P	Connected to slide pot R20
G16	VMON9P	Connected to slide pot R19
J15, H15, G15	VMON7M, VMON8M, VMON9M	Must be connected to ground!

For more details on device connections employed in this reference design, as well as detailed schematics, refer to the Platform Manager Development Kit User's Guide.

FPGA Logic

The FPGA logic can be divided into three major functions, as shown in Figure 6. First, there is a state sequencer that controls the series of I²C reads and writes, and assembles and moves data as needed. Second, there is an I²C link that manages the details of the data transfers to and from the Platform Manager ADC. Finally, there is decoder and display logic that takes the value read from the ADC, formats it, and generates drive signals for the LCD display. Each of these functions will be discussed in the following sections.

Figure 6. FPGA Logic Organization



I²C Link

The I²C Link logic implements a specialized I²C bus master that is tailored to handling the subset of the bus protocol needed to interface with the Platform Manager's analog functions. Because all Platform Manager I²C transactions are either of the form 'write byte to register' or 'read byte from register', this provides the opportunity to hide a great deal of complexity and provide an easy-to-use interface. To write data to a Platform Manager register, the state sequencer logic needs to provide the register address, the data, and strobe the `rd` signal. The I²C link will raise a busy signal in response until it has completed the write transaction. Reading a register follows a similar protocol, where the state sequencer logic provides the register address, and strobes the `wr` signal. The analog link's `busy` line then goes high until the read operation has completed and valid data is available for output. Listing 2 shows the module definition for the analog link.

Listing 2. Module Definition for I²C Link

```
module I2C_LINK (

      input clk, rst,   // clock (8MHz) and Reset inputs
      input [6:0]devaddr, // device address (assumed fixed so not latched)
      input [4:0]regaddr,// register address(latched outside module)
      input [7:0]datain,// data to transmit (latched outside module)
      output [7:0]dataout,// output data (only valid when busy=0)
      input wr, rd, // read and write command signals
      output reg busy,// busy status
      output reg sclout,// scl output
      input sdain,// sda input
      output reg sdaout // sda output
);
```

Because SDA is a bidirectional line, the `sdain` and `sdaout` signals must be merged in the top-level module to the I/O pin. This is accomplished through the code in Listing 3, where `sda` is defined as 'inout' at the module top-level.

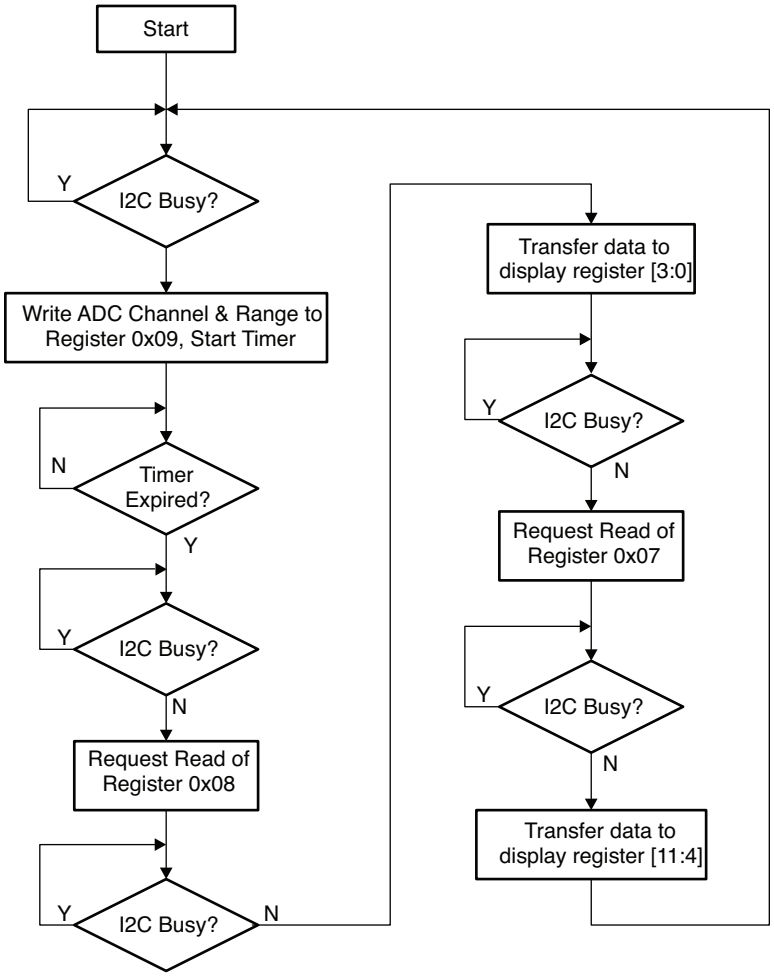
Listing 3. I²C SDA Bidirectional Merge Logic

```
      assign sda = sdaout ? 1'bZ : 1'b0;
      assign Sdain = sda;
```

State Sequencer

The state sequencer carries out a sequential 'program' that repetitively initiates an ADC conversion, waits until it is done, and then transfers the result data to a 12-bit register from which it can be conveniently displayed. Because the I²C link offloads much of the detailed logic needed for communicating with the Platform Manager register set, the program implemented by the state sequencer is relatively simple. Figure 7 shows a high-level flowchart of the sequence logic.

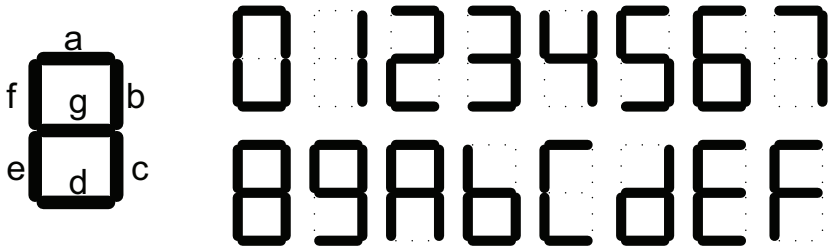
Figure 7. Flowchart of State Sequencer 'Program'



Decoder/Display Driver

The purpose of the Decoder/Display Driver logic is to take a 12-bit binary word and use it to drive a 3-digit, 7-segment LCD display. This requires two major functions to be performed. First, the binary data must be segregated into 4-bit nibbles and decoded into a 7-segment hexadecimal pattern. The segment decode patterns used in this design are shown in Figure 8.

Figure 8. Hexadecimal Decode Patterns

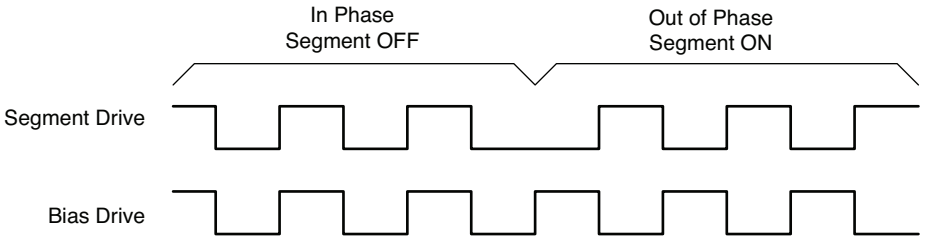


The decoder logic is replicated three times, once for each nibble of data to be decoded. Although this was an acceptable use of logic for the case of three digits, a time-multiplexing scheme might be more appropriate in cases where it is necessary to display longer strings.

Unlike an LED display, in which segments can be driven with DC signals, an LCD display requires a symmetric AC (no DC bias) signal to activate a given segment. When AC voltage is applied to a segment, it darkens, and when zero voltage is applied to a segment it clears. To simplify the generation of AC waveforms, a common 'bias' terminal is provided that can also be driven with an AC signal.

To generate the necessary drive signals, a counter divides the 8MHz clock down to 61Hz ($\div 2^{17}$) which is used to drive the LCD bias line. This same signal is then XOR'ed with each segment signal. This generates the bias and segment waveforms shown in Figure 9. When the bias and segment drive signals are in phase, zero voltage is applied to the segment. When the bias and segment drive signals are out of phase, this results in the application of a symmetric AC signal with zero net DC offset.

Figure 9. LCD Segment and Bias Drive Signals



Listing 4 shows the module definition for the LCD driver module.

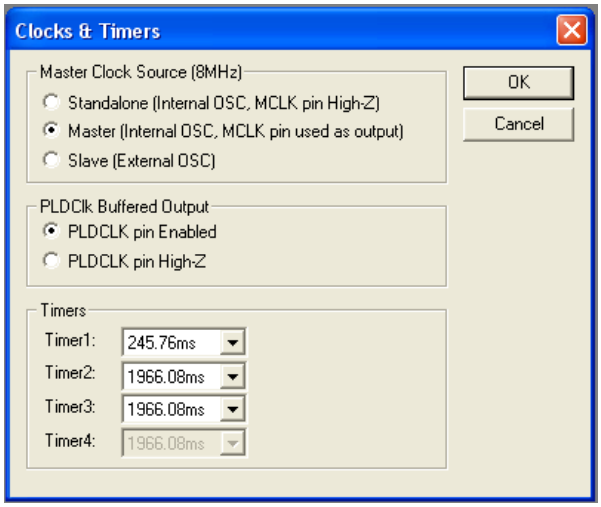
Listing 4. Module Definition for LCD Decoder/Driver

```
module LCD301_HexDisplay (data, ena, clk, bias,
        s1a, s1b, s1c, s1d, s1e, s1f, s1g,
           s2a, s2b, s2c, s2d, s2e, s2f, s2g,
           s3a, s3b, s3c, s3d, s3e, s3f, s3g);

      input [11:0] data;// 12 bit binary input vector
      input ena;  // overall enable
      inputclk;// input clock

      output  bias;  // modulated backbias output

      outputs1a, s1b, s1c, s1d, s1e, s1f, s1g; // modulated
      outputs2a, s2b, s2c, s2d, s2e, s2f, s2g;// segment
      outputs3a, s3b, s3c, s3d, s3e, s3f, s3g;// outputs
```

Miscellaneous Configuration

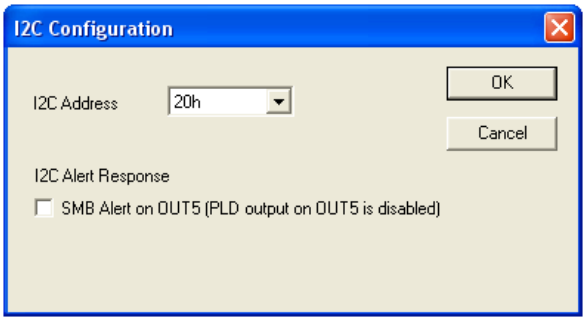
To implement this reference design, little analog configuration is required for the Platform Manager. This is done through the 'Clocks & Timers' configuration dialog shown in Figure 10.

Figure 10. Master Clock Source Configuration



It is also important to configure the Platform Manager's I^2C address so that it matches that specified in the reference design's Verilog code. Near the top of the Verilog file the device address is defined through the ANALOG_ADDR constant, and is set to 0x20 (hexadecimal) in the reference design. To allow the FPGA logic to communicate to the ADC, the corresponding address must be set identically in PAC-Designer through the 'I²C Configuration' dialog in PAC-Designer (Figure 11), which can be opened by clicking on the I^2C block in the main schematic.

Figure 11. I^2C Address Configuration



Finally, there must be LogiBuilder code for the CPLD that sets or resets at least one output. While this code does not affect the operation of the FPGA or the ADC, it is necessary for the compilation process. For this reference design, the CPLD is programmed with a short program that blinks LED D21 at ~2 Hz, a quick and useful check that the design has properly compiled and downloaded.

Validation

Because this reference design was designed for implementation on the Platform Manager Evaluation Board, it is straightforward to validate in hardware. When properly synthesized and downloaded to the Platform Manager Evaluation Board, the board's LCD will display a hexadecimal code ranging from 0 to FFF. As previously mentioned, LED D21 will also blink at approximately 2Hz as a result of the dummy program implemented in the CPLD logic. Depending on the SW10 DIP switch settings, changes to these inputs will be dynamically displayed on the LCD. Table 4 lists the switch settings required to display the evaluation board's analog signal sources.

Table 4. DIP Switch Settings for Evaluation Board Signal Sources

SW10.1	SW10.2	SW10.3	SW10.4	Function
ON	OFF	OFF	OFF	VMON9 – Slide pot R19
OFF	ON	ON	ON	VMON8 – Slide pot R20
OFF	ON	ON	OFF	VMON7 – Thermistor R60
OFF	ON	OFF	ON	VMON6 – LDO Output
OFF	ON	OFF	OFF	VMON5 – DC-DC Module output

DIP switch SW9.1 controls the ADC range. When the switch is turned ON, the ADC goes into its 0-2V range and will display codes from 0x000 – 0x3FF. When the switch is turned OFF the ADC goes into its 0-6V range and is capable of displaying codes over the full range of 0x000 – 0xFFF.

Implementation

Device	FPGA LUTs	FPGA Slices	FPGA I/O	CPLD Macrocells	CPLD CTimers	CPLD P.T.	CPLD I/O	VMONs
LPTM10-12107	210	105	31	n/a	n/a	n/a	n/a	Note 1

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

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
October 2010	01.0	Initial release.