Long Delay Timers Using Platform Manager Devices

October 2010 — Reference Design RD1079

Introduction

Over the last decade, the complexity of power management on circuit boards for wireless infrastructure, data-com, telecom, industrial control and entertainment has increased to a point where specialty power management devices require complex sequencing and timing solutions to manage their power systems. Platform Manager™ devices from Lattice handle many of these sequencing and control demands in a single device by integrating much of the functionality required to control such systems.

This reference design explores the "long timer" features of the Platform Manager. The Platform Manager architecture expands the timer functionality beyond the abilities of the Power Manager II devices by increasing both the number of timers as well as the number of time delay options. Like Power Manager II, these features use internal clocks but extend the range and number of timers used.

The Platform Manager device is divided into CPLD and FPGA sections. The CPLD portion is user-configurable logic that is tightly coupled (albeit user-defined) to the power management analog signaling of the Platform Manager. The FPGA section allows for more expansive and sophisticated, LUT-based control of the Platform Manager analog signaling as well as user-defined glue logic (which might or might not be unrelated to the task of power management). See the Platform Manager Data Sheet for detailed information.

Architecture

To provide maximum flexibility, the Platform Manager device offers an extended section of FPGA logic accessible to the user for sequencer and/or general I/O control. In order to expand the timers and allow for building very long delays, a prescaled, CPLD clock signal is brought out from a timer (Timer4), through OUT16, an open-drain output pin, and is then routed externally to the FPGA fabric as a base timing signal. This configuration allows for timer delays from seconds, minutes, or even hours. Note that Timer1, Timer2 and Timer3 are all still available for use within the CPLD section of the device and provide delays of 32us to ~2sec.

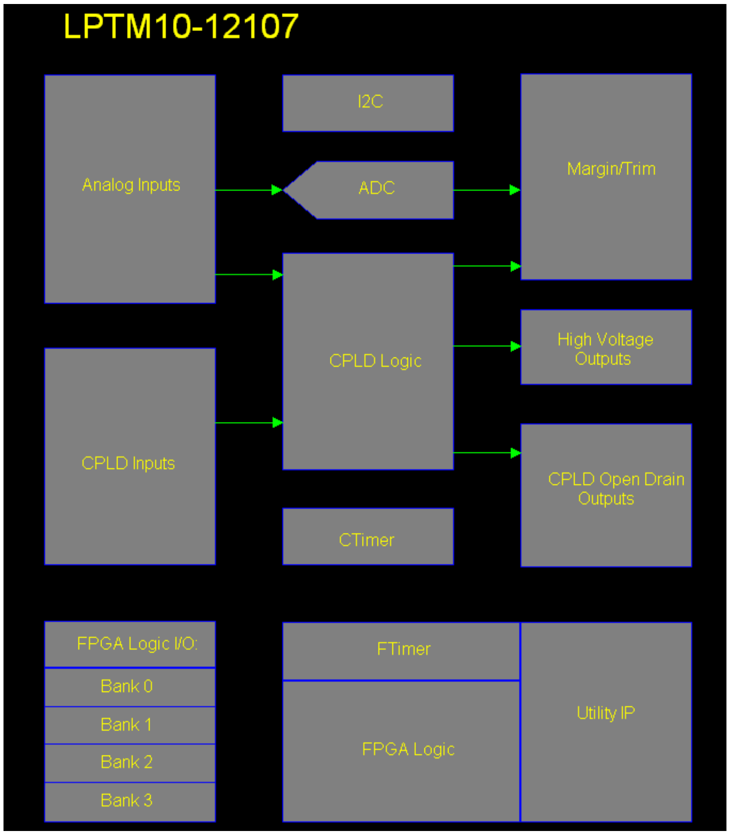
In addition to the slower Timer4 output clock source, the 250kHz CPLDCLK and 8Mz internal clocks are also available to the LUT-based FPGA. These are routed externally from pin to pin. The clocks can be used to build custom timers or other sequence logic for miscellaneous power supply control or display and interface functions.

Software Setup

The Long Delay Timers reference design uses PAC-Designer® software as the primary tool for design implementation. The PAC-Designer main software screen is shown in Figure 1. This is the user interface for entering equations, logic functions, timers, and voltage monitors. Each functional group is associated with a graphical block. Clicking on any block opens easy-to-use editing features, such as the CPLD logic, FPGA logic, CTimer block or the FTimer block. I/O control is set with simple pull-down menus that allow the user to change input or output names and set VMON voltage trip points.

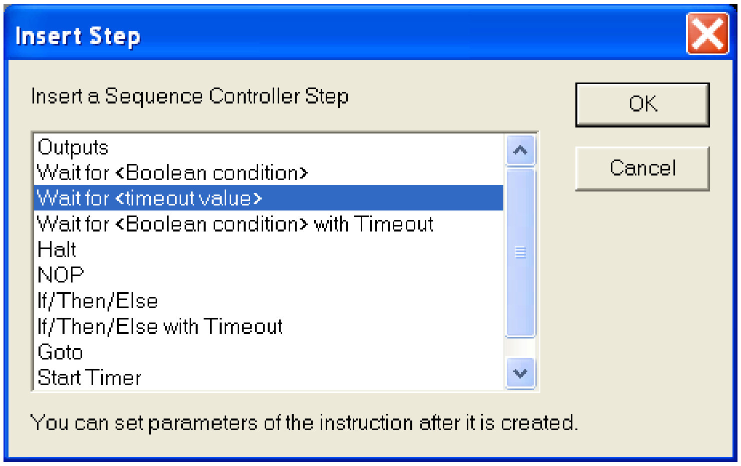
© 2010 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Figure 1. PAC-Designer Main Software Entry Screen



There is a functional block used for CTimers and another for FTimers. The Platform Manager CPLD section supports three timers that range from 32us to 1966ms. Simple pull-down menus allow the user to change all timer values that are available. To use a CPLD timer, a step is inserted in the LogiBuilder sequence with a "Wait For Timeout Value", as shown in Figure 2.

Figure 2. Inserting a Wait for Timeout

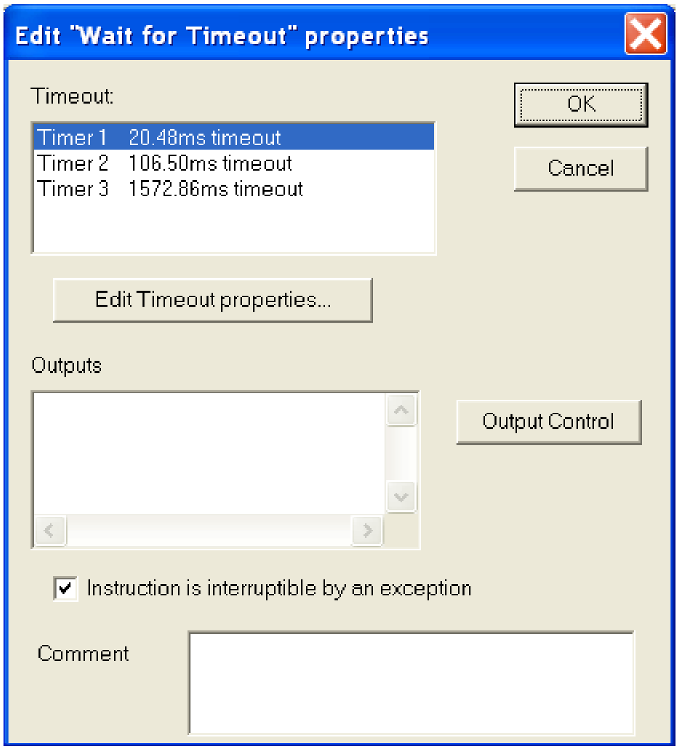

Insert Step
OK
Cancel
Outputs
Wait for <Boolean condition>
Wait for <timeout value>
Wait for <Boolean condition> with Timeout
Halt
NOP
If/Then/Else
If/Then/Else with Timeout
Goto
Start Timer

Insert a Sequence Controller Step

You can set parameters of the instruction after it is created.

Once a CPLD timer is assigned in the LogiBuilder sequence, its properties are set in the Edit Control for Timeout Properties shown in Figure 3.

Figure 3. Edit Control for CPLD Timeout Properties



Edit "Wait for Timeout" properties

Timeout:

Timer 1 20.48ms timeout
Timer 2 106.50ms timeout
Timer 3 1572.86ms timeout

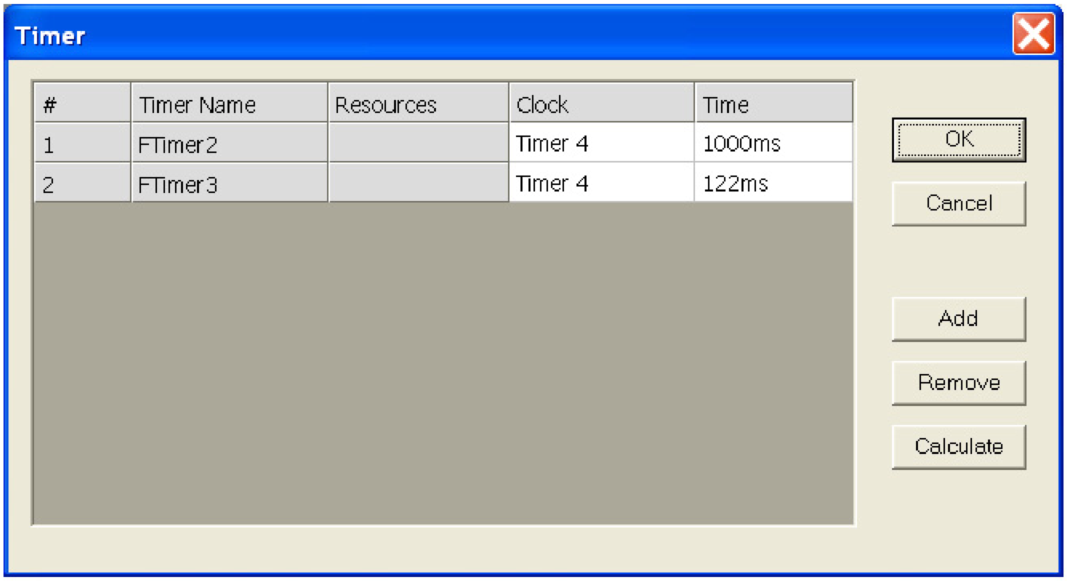
Outputs

☑ Instruction is interruptible by an exception

Comment

LogiBuilder can also be used to sequence outputs in the FPGA logic. A timer step is inserted in the sequence using the same method as for the CPLD, except when editing the properties, the FTimer dialog opens and allows multiple timers and values to be edited. The FTimer functional block allows the user to define timers from milliseconds to hours (see Figure 4).

Figure 4. FTimer Settings for the FPGA



Timer

#	Timer Name	Resources	Clock	Time
1	FTimer2		Timer 4	1000ms
2	FTimer3		Timer 4	122ms

There is no pre-defined limit to the number of FTimers since these are built in the FPGA and are therefore LUT-based and dependent on the available FPGA LUT resources (as opposed to the CPLD section where timers are hardware-limited to three with a fourth used as a time-base for the FTimers in the FPGA).

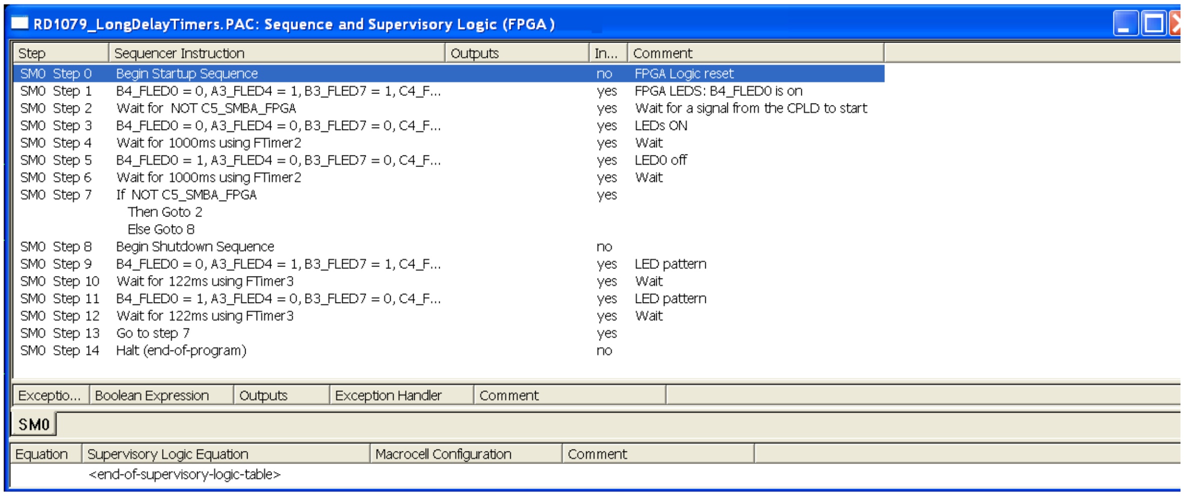
Once an FTimer is chosen in code, LogiBuilder places the Timer4 code needed in the CPLD supervisory equations. The equations will contain a D flip-flop used to drive OUT16.

PAC-Designer software prepares the code required for connecting the CPLD section of Timer4 output (driven to the OUT16 open-drain pin) to the FPGA fabric. The output from the Timer4 terminal-count drives OUT16. This pin requires an external pull-up resistor and an external connection with an FPGA input pin on Bank2.

Once the software entry is done, a compilation step generates a Verilog file. There is no need to write or debug code for building long delay timers. The code is “correct by design” and ready to synthesize. The features of the software allow simple edits with a click of the mouse to build timers. In most cases, a simple “Wait for Timeout” instruction is used in PAC-Designer. Complex designs with multiple timers and multiple delays, can be used to solve the power-up requirements of very large system-level boards.

Each FTimer is independent and uses Timer4 and OUT16 for the clock source. The step in the sequencer will start the timer and wait in that step until the time delay has expired. Figure 5 shows a LogiBuilder code example showing the use of the FTimer in a simple sequence.

Figure 5. Wait For FTimer



RD1079_LongDelayTimers.PAC: Sequence and Supervisory Logic (FPGA)

Step	Sequencer Instruction	Outputs	In...	Comment
SM0 Step 0	Begin Startup Sequence		no	FPGA Logic reset
SM0 Step 1	B4_FLED0 = 0, A3_FLED4 = 1, B3_FLED7 = 1, C4_F...		yes	FPGA LEDS: B4_FLED0 is on
SM0 Step 2	Wait for NOT C5_SMBA_FPGA		yes	Wait for a signal from the CPLD to start
SM0 Step 3	B4_FLED0 = 0, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LEDs ON
SM0 Step 4	Wait for 1000ms using FTimer2		yes	Wait
SM0 Step 5	B4_FLED0 = 1, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LED0 off
SM0 Step 6	Wait for 1000ms using FTimer2		yes	Wait
SM0 Step 7	If NOT C5_SMBA_FPGA Then Goto 2 Else Goto 8		yes	
SM0 Step 8	Begin Shutdown Sequence		no	
SM0 Step 9	B4_FLED0 = 0, A3_FLED4 = 1, B3_FLED7 = 1, C4_F...		yes	LED pattern
SM0 Step 10	Wait for 122ms using FTimer3		yes	Wait
SM0 Step 11	B4_FLED0 = 1, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LED pattern
SM0 Step 12	Wait for 122ms using FTimer3		yes	Wait
SM0 Step 13	Go to step 7		yes	
SM0 Step 14	Halt (end-of-program)		no	

Exceptio...	Boolean Expression	Outputs	Exception Handler	Comment

SM0

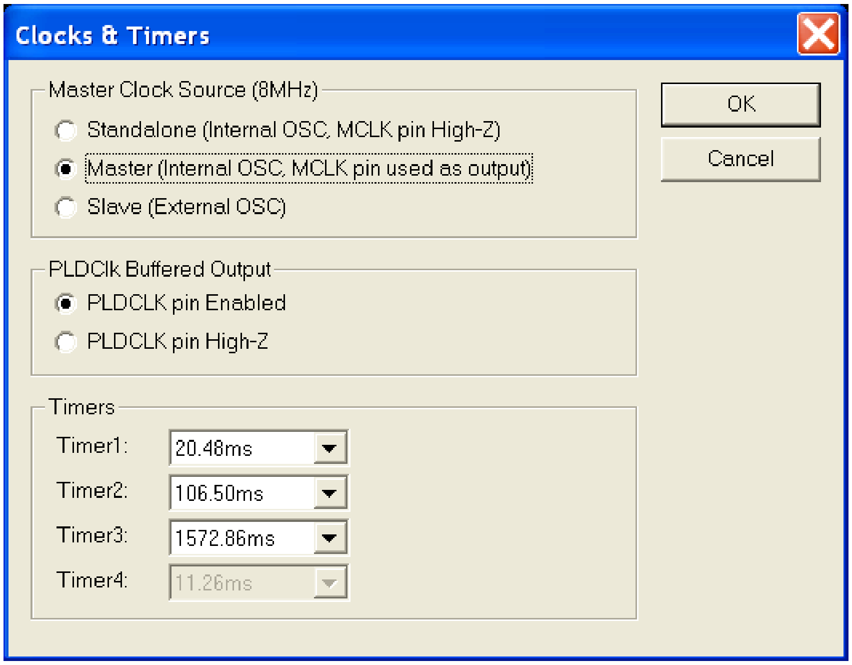
Equation	Supervisory Logic Equation	Macrocell Configuration	Comment
	<end-of-supervisory-logic-table>		

To edit the FTimer values, the user can simply double-click on the timer sequence step. The timeouts can be changed either before or after they have been added into a LogiBuilder sequence step.

Master Clock

The 8MHz internal MCLK is connected externally to the FPGA from the 8MHz MCLK output. This clock is used for communications and I/O as well as an internal main fabric clock. The Master mode is used to enable the 8MHz internal oscillator as shown in Figure 6.

Figure 6. Clock and Timer Dialog Box



Master Clock Source (8MHz)

○ Standalone (Internal OSC, MCLK pin High-Z)
● Master (Internal OSC, MCLK pin used as output)
○ Slave (External OSC)

PLDClk Buffered Output

● PLDCLK pin Enabled
○ PLDCLK pin High-Z

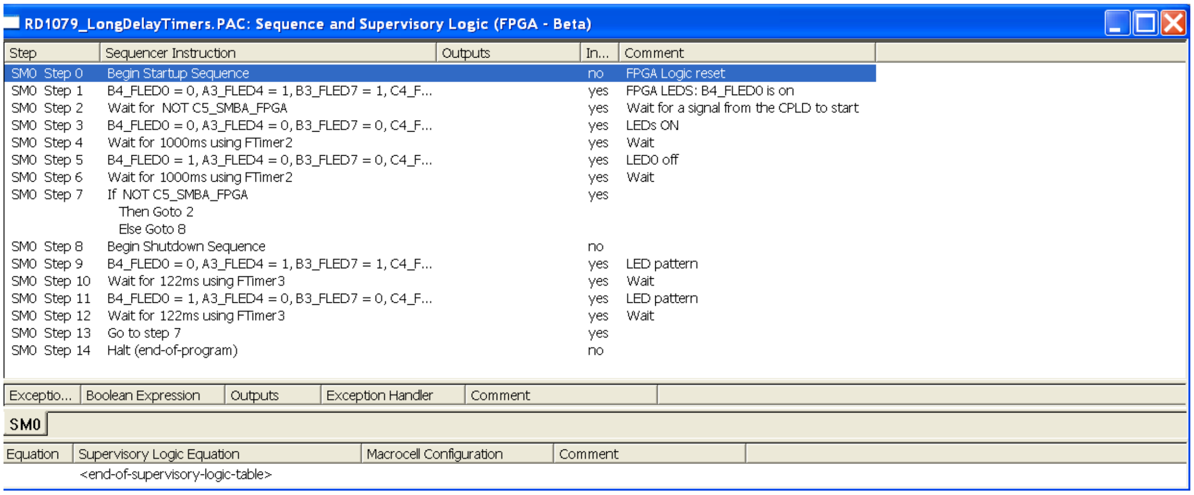
Timers

Timer1: 20.48ms
Timer2: 106.50ms
Timer3: 1572.86ms
Timer4: 11.26ms

Compiled Design

Once sequences are entered into both the CPLD and FPGA sections and all timers are set up, the design can be compiled. When compiled, the design will generate a Verilog file and a report file for the CPLD and FPGA sections. To generate a JEDEC file, click on the JED icon on the top toolbar (see Figure 7). For programming the devices, the JEDEC file can then be downloaded to the Platform Manager device using ispVM™ System software.

Figure 7. FPGA Sequencer Code



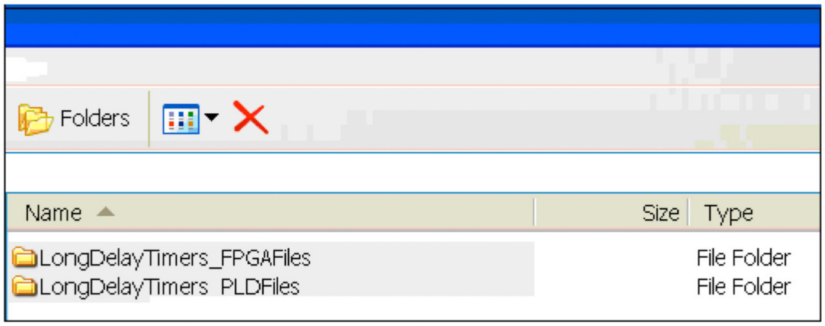
RD1079_LongDelayTimers.PAC: Sequence and Supervisory Logic (FPGA - Beta)

Step	Sequencer Instruction	Outputs	In...	Comment
SM0 Step 0	Begin Startup Sequence		no	FPGA Logic reset
SM0 Step 1	B4_FLED0 = 0, A3_FLED4 = 1, B3_FLED7 = 1, C4_F...		yes	FPGA LEDS: B4_FLED0 is on
SM0 Step 2	Wait for NOT C5_SMBA_FPGA		yes	Wait for a signal from the CPLD to start
SM0 Step 3	B4_FLED0 = 0, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LEDs ON
SM0 Step 4	Wait for 1000ms using FTimer2		yes	Wait
SM0 Step 5	B4_FLED0 = 1, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LED0 off
SM0 Step 6	Wait for 1000ms using FTimer2		yes	Wait
SM0 Step 7	If NOT C5_SMBA_FPGA Then Goto 2 Else Goto 8		yes	
SM0 Step 8	Begin Shutdown Sequence		no	
SM0 Step 9	B4_FLED0 = 0, A3_FLED4 = 1, B3_FLED7 = 1, C4_F...		yes	LED pattern
SM0 Step 10	Wait for 122ms using FTimer3		yes	Wait
SM0 Step 11	B4_FLED0 = 1, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LED pattern
SM0 Step 12	Wait for 122ms using FTimer3		yes	Wait
SM0 Step 13	Go to step 7		yes	
SM0 Step 14	Halt (end-of-program)		no	

Exceptio...	Boolean Expression	Outputs	Exception Handler	Comment

Equation	Supervisory Logic Equation	Macrocell Configuration	Comment
	<end-of-supervisory-logic-table>		

To view the generated Verilog code for the FPGA, open the folder with the same file name as the PAC-Designer file >filename_FPGAFiles.



Name	Size	Type
LongDelayTimers_FPGAFiles		File Folder
LongDelayTimers_PLDFiles		File Folder

Clock Hardware Setup

Since the clocks are routed externally, the user must route the appropriate signals on the circuit board for driving the clock lines. For the two package outlines, the pin interconnect requirements for timers, CPLDCLK and MCLK are listed in Tables 1-3. Figure 8 shows a schematic representation of the external connections.

Table 1. Pin Interconnection for Timers

Package	OUT16 Timer4	Route to	Bank
208-Ball ftBGA	OUT16 (R13)	PB9C (R9)	Bank2
128-Pin TQFP	OUT16 (Pin31)	PB3B (Pin39)	Bank2

To set up additional logic and sequencers in the FPGA with PAC-Designer, the 250kHz CPLDCLK is driven externally and routed to an FPGA pin as an input clock. Table 2 shows the connections required for the CPLDCLK.

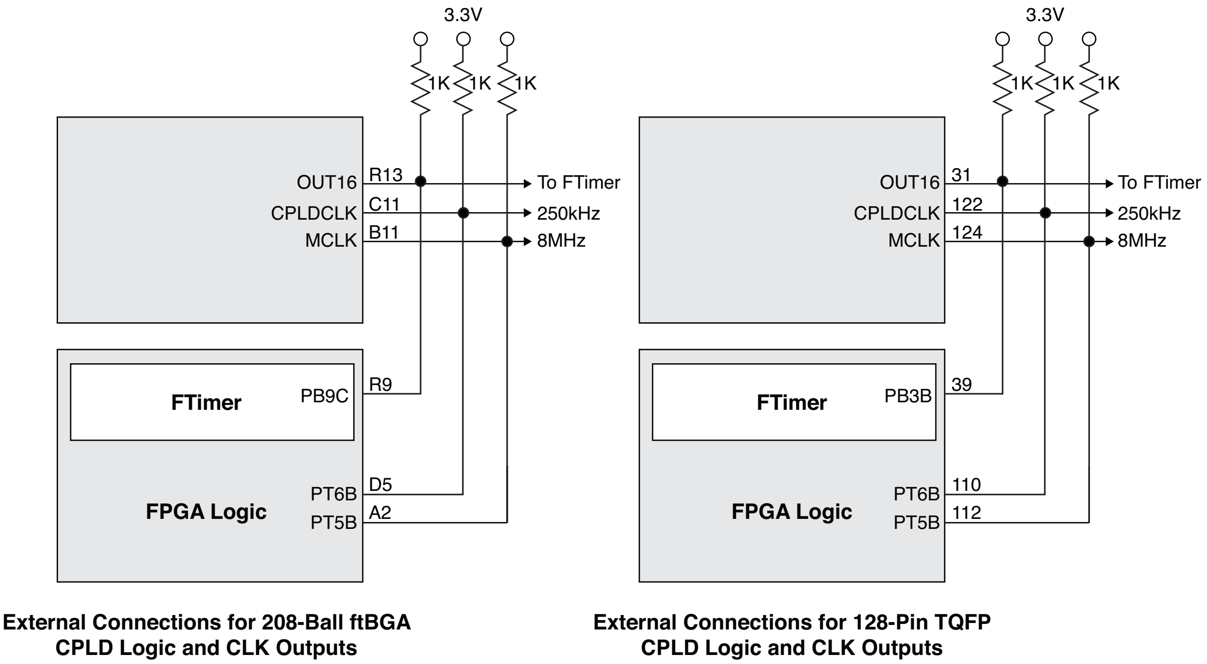
Table 2. Pin Interconnection for CPLDCLK

Package	CPLDCLK	Route to	Bank
208-Ball ftBGA	CPLDCLK (C11)	PT6B (D5)	Bank0
128-Pin TQFP	CPLDCLK (122)	PT6B (Pin110)	Bank0

Table 3. Pin Interconnection for MCLK

Package	MCLK	Route to	Bank
208-Ball ftBGA	MCLK (B11)	PT5B (A2)	Bank0
128-Pin TQFP	MCLK (124)	PT5B (112)	Bank0

Figure 8. External Connections for Clocks



Platform Manager Development Kit

To evaluate with this reference design in hardware, you can use the Platform Manager Development Kit along with the PAC-Designer and ispVM software. The Platform Manager Development Kit includes the Platform Manager Evaluation Board, which allows you to experiment with this reference design in hardware. With PAC-Designer, you can make changes to the design to suit you own needs, then download those changes to the evaluation board. You will need ispVM in order to download either the Long Delay Timers reference design or you own design to the evaluation board.

See the following documets for further information on the Platform Manager Development Kit:

- Platform Manager Development Kit User’s Guide
- Platform Manager Development Kit QuickSTART Guide

Figure 9. Platform Manager Development Kit

Long Delay Timer Operation with the Platform Manager Development Kit

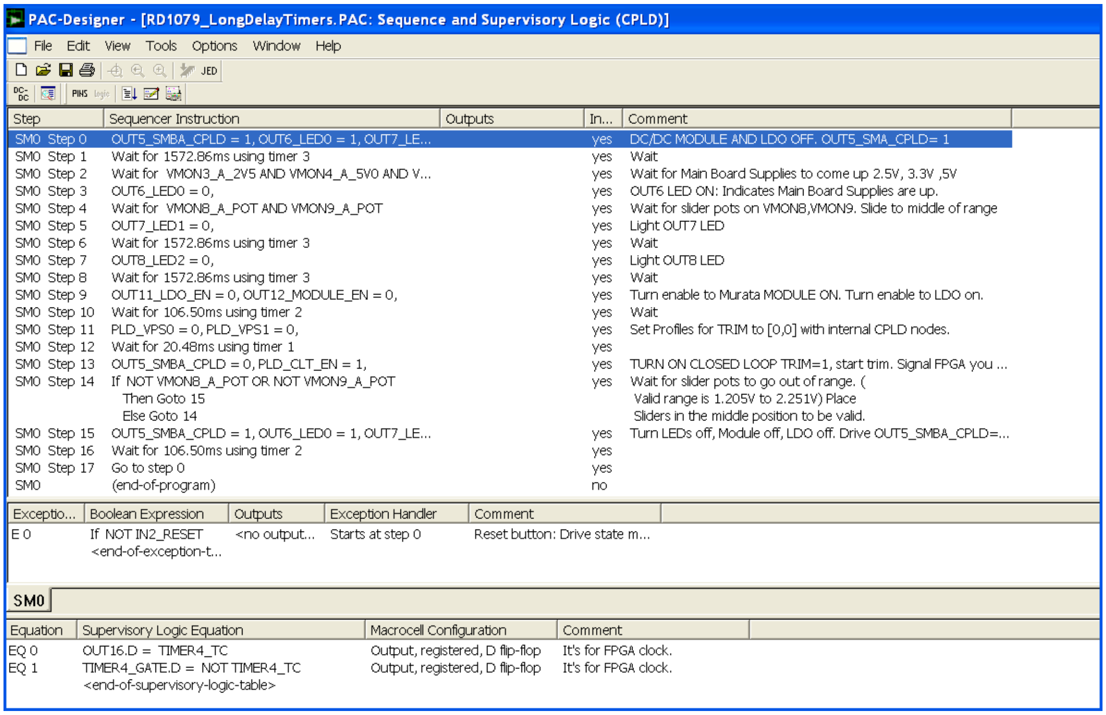
The Long Delay Timers Reference Design can be downloaded to the Platform Manager Development Kit to exercise the timers and other display features of the Platform Manager Evaluation Board. Simple changes can be made to the PAC-Designer source code, to familiarize the user with the timer operations, display control and I/Os functions.

Figure 10 shows the code that resides in the CPLD section of the PAC-Designer file.

The basic sequence waits for board power to become active and for the user to adjust the potentiometers, then continues with sequencing LEDs and the two power supplies on the board.

Once the CPLD section has checked all the supplies, it drives OUT5_SMBA_CPLD low, starting a sequence on the FPGA logic sequencer (see Figure 12).

Figure 10. Long_Delay_Timers CPLD Sequencer and Logic


PAC-Designer - [RD1079_LongDelayTimers.PAC: Sequence and Supervisory Logic (CPLD)]
File
Edit
View
Tools
Options
Window
Help
JED
DC-DC
PINS
logic
SM0

Step	Sequencer Instruction	Outputs	In...	Comment
SM0 Step 0	OUT5_SMBA_CPLD = 1, OUT6_LED0 = 1, OUT7_LE...		yes	DC/DC MODULE AND LDO OFF. OUT5_SMA_CPLD= 1
SM0 Step 1	Wait for 1572.86ms using timer 3		yes	Wait
SM0 Step 2	Wait for VMON3_A_2V5 AND VMON4_A_5V0 AND V...		yes	Wait for Main Board Supplies to come up 2.5V, 3.3V ,5V
SM0 Step 3	OUT6_LED0 = 0,		yes	OUT6 LED ON: Indicates Main Board Supplies are up.
SM0 Step 4	Wait for VMON8_A_POT AND VMON9_A_POT		yes	Wait for slider pots on VMON8,VMON9. Slide to middle of range
SM0 Step 5	OUT7_LED1 = 0,		yes	Light OUT7 LED
SM0 Step 6	Wait for 1572.86ms using timer 3		yes	Wait
SM0 Step 7	OUT8_LED2 = 0,		yes	Light OUT8 LED
SM0 Step 8	Wait for 1572.86ms using timer 3		yes	Wait
SM0 Step 9	OUT11_LDO_EN = 0, OUT12_MODULE_EN = 0,		yes	Turn enable to Murata MODULE ON. Turn enable to LDO on.
SM0 Step 10	Wait for 106.50ms using timer 2		yes	Wait
SM0 Step 11	PLD_VPS0 = 0, PLD_VPS1 = 0,		yes	Set Profiles for TRIM to [0,0] with internal CPLD nodes.
SM0 Step 12	Wait for 20.48ms using timer 1		yes	
SM0 Step 13	OUT5_SMBA_CPLD = 0, PLD_CLT_EN = 1,		yes	TURN ON CLOSED LOOP TRIM=1, start trim. Signal FPGA you ...
SM0 Step 14	If NOT VMON8_A_POT OR NOT VMON9_A_POT Then Goto 15 Else Goto 14		yes	Wait for slider pots to go out of range. (Valid range is 1.205V to 2.251V) Place Sliders in the middle position to be valid.
SM0 Step 15	OUT5_SMBA_CPLD = 1, OUT6_LED0 = 1, OUT7_LE...		yes	Turn LEDs off, Module off, LDO off. Drive OUT5_SMBA_CPLD=...
SM0 Step 16	Wait for 106.50ms using timer 2		yes	
SM0 Step 17	Go to step 0		yes	
SM0	(end-of-program)		no	

Exceptio...	Boolean Expression	Outputs	Exception Handler	Comment
E 0	If NOT IN2_RESET	<no output...	Starts at step 0	Reset button: Drive state m...
	<end-of-exception-t...			

Equation	Supervisory Logic Equation	Macrocell Configuration	Comment
EQ 0	OUT16.D = TIMER4_TC	Output, registered, D flip-flop	It's for FPGA clock.
EQ 1	TIMER4_GATE.D = NOT TIMER4_TC	Output, registered, D flip-flop	It's for FPGA clock.
	<end-of-supervisory-logic-table>		

CPLD Sequence

Step 0 – Step 3: CPLD Sequencer and Logic Functions

The CPLD begins by driving the following outputs to a known state in Step 0:

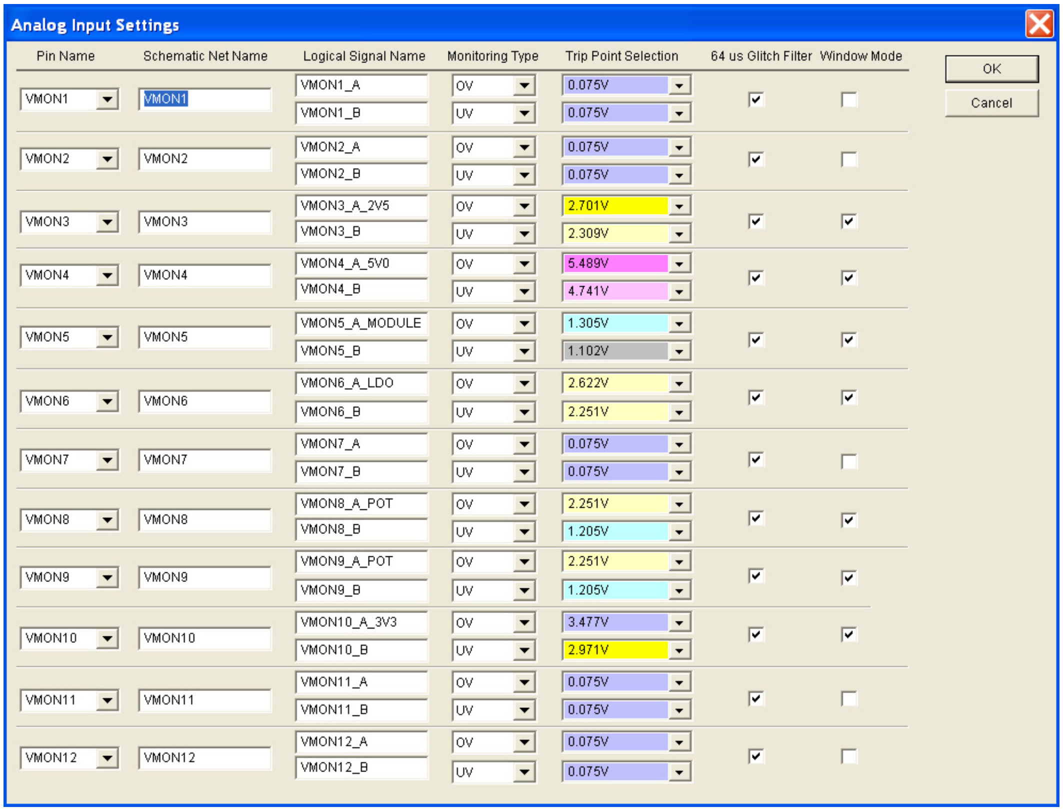
• OUT5_SMBA_CPLD=1	Drives an output signal from the CPLD to the FPGA
• OUT6_LED0=1	LED is OFF
• OUT7_LED1=1	LED is OFF
• OUT8_LED2=1	LED is OFF
• OUT9_LED3=0	LED is ON
• OUT11_LDO_EN=1	LDO User Power Rail is OFF, 0V
• OUT12_Module=1	Module User Power Rail is OFF, 0V

After Step 0, the OUT9_LED3 (D24 LED) should be ON. The sequencer then waits 1572ms and checks the 2.5V, 3.3V and 5V power supplies. Once all three of these are within the VMON window comparator range, the OUT6_LED0 (D21 LED) will turn ON.

Step 4 – Step 7: CPLD Sequencer and Logic Functions

Step 4 waits for the slider potentiometers to be within range. Setting both potentiometers to 1.65V or near the middle of the range, will trip the VMON window comparators. Both potentiometers are set to the same trip point levels and set up as a window compare. The VMON trip points are 1.205V to 2.251V as shown in Figure 11. Once both potentiometers are brought into range, OUT7_LED1 (D22) will turn ON, followed by a delay of 1572 ms, then OUT8_LED2 (D23) will turn ON.

Figure 11. Analog Input Settings



Pin Name	Schematic Net Name	Logical Signal Name	Monitoring Type	Trip Point Selection	64 us Glitch Filter	Window Mode
VMON1	VMON1	VMON1_A	OV	0.075V	☑	□
		VMON1_B	UV	0.075V		
VMON2	VMON2	VMON2_A	OV	0.075V	☑	□
		VMON2_B	UV	0.075V		
VMON3	VMON3	VMON3_A_2V5	OV	2.701V	☑	☑
		VMON3_B	UV	2.309V		
VMON4	VMON4	VMON4_A_5V0	OV	5.489V	☑	☑
		VMON4_B	UV	4.741V		
VMON5	VMON5	VMON5_A_MODULE	OV	1.305V	☑	☑
		VMON5_B	UV	1.102V		
VMON6	VMON6	VMON6_A_LDO	OV	2.622V	☑	☑
		VMON6_B	UV	2.251V		
VMON7	VMON7	VMON7_A	OV	0.075V	☑	□
		VMON7_B	UV	0.075V		
VMON8	VMON8	VMON8_A_POT	OV	2.251V	☑	☑
		VMON8_B	UV	1.205V		
VMON9	VMON9	VMON9_A_POT	OV	2.251V	☑	☑
		VMON9_B	UV	1.205V		
VMON10	VMON10	VMON10_A_3V3	OV	3.477V	☑	☑
		VMON10_B	UV	2.971V		
VMON11	VMON11	VMON11_A	OV	0.075V	☑	□
		VMON11_B	UV	0.075V		
VMON12	VMON12	VMON12_A	OV	0.075V	☑	□
		VMON12_B	UV	0.075V		

Step 8 – Step 12: CPLD Sequencer and Logic Functions

Step 8 waits for a delay of 1572 ms using Timer3. Once the timer reaches terminal count, the output enables for the Module and the LDO are turned on, with a logic low. Another timer, Timer2, is used for a delay of 106.5 ms, before PLD_VPS0 and PLD_VPS1 are both driven low [0,0] for the internal nodes used to trim/margin the LDO and DC/DC Module. After the PLD_VPS bits are set low, Timer1 is called for a delay of 20.48 ms.

Step 13 – Step 17: CPLD Sequencer and Logic Functions

In Step 13, the OUT5_SMBA_CPLD output is driven low as a flag signal to the FPGA. This same step is used to start the Closed Loop Trim by driving CLT_EN to a logic high.

Step 14 is a loop that monitors the two slider potentiometers for a valid window compare.

If either potentiometers falls outside of the range, it drops to Step 15 and shuts off OUT5_SMBA_CPLD=0, turns OFF all supplies and cycles back to Step 0.

Exception Condition: CPLD Exception

The Exception is used as a reset into the CPLD using IN2. When the reset push-button is pressed, the sequencer returns to Step 0 to start over.

- If NOT IN2_Reset, start at Step 0.

Generating an Error: VMON Slider Potentiometers

If either of the slider potentiometers are raised or lowered outside the window, an error will be generated. When the voltage error occurs, it shuts off the CPLD LEDs (D21-D24), sets OUT5_SMBA_CPLD = 1 and shuts off the user power supplies. At the shut-down, it waits 106.5 ms and then returns to Step 0 to start over.

CPLD Supervisory Equations

The supervisory equations include:

OUT16.d = Timer4_TC

Timer4_gate.d = Timer4_TC

These are used to generate a timer clock to drive FTimers of the FPGA. OUT16 is a slow clock prescaler used to drive multiple timers in the FPGA. The equation is built automatically by Logi-Builder when an FPGA Ftimer is configured. The value is based on the requests from the FTimer settings.

FPGA Sequence

Step 0 – Step 2

In this design, the FPGA sequencer starts when it receives a flag from the CPLD. This occurs when the C5_SMBA_FPGA input is driven low by the CPLD. In Step 1, the B4_LED0 is on, in Step 2, it is waiting for the input C5_SMBA_FPGA to go low. When the CPLD has successfully completed its sequence, it drives OUT5_SMBA_CPLD low, thus triggering the FPGA sequencer.

• Step 0 - Step1: Turn on B4_LED0,	Other FPGA LEDs OFF
• Step 2: Wait for NOT C5_SMBA_FPGA	Waiting for input to go low

Step 3 – Step 7

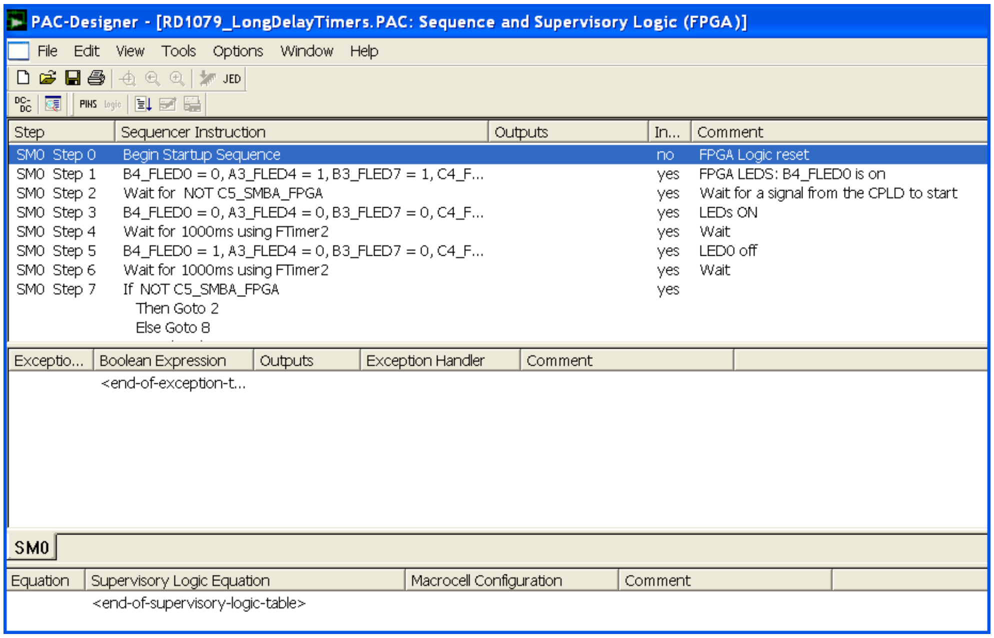
Steps 3-7 blink the FLED0 (B4_LED0) ON and OFF at 1000ms, using FTimer2. This timer is clocked by CPLD output OUT16, which is set up in the CPLD. The upper seven LEDs for the FPGA (FLED7..FLED) are ON.

Step 9 – Step 13

In Step 9, the FPGA LEDs blink ON and OFF. The upper four LEDs (FLED7..FLED4) and lower four LEDs (FLED3..FLED0) turn ON and OFF in groups of four, every 122ms when there is an error. The CPLD flag for this error signals the FPGA using OUT5_SMBA_CPLD.

Step 13 causes the sequence to return to Step 7 to check the C5_SMBA_FPGA input after each LED cycle. Reset signals the FPGA back to Step 0 to start the sequence again.

Figure 12. Long_Delay_Timers FPGA Sequencer and Logic Functions


PAC-Designer - [RD1079_LongDelayTimers.PAC: Sequence and Supervisory Logic (FPGA)]
File
Edit
View
Tools
Options
Window
Help
JED
DC-DC
PINS
logic

Step	Sequencer Instruction	Outputs	In...	Comment
SM0 Step 0	Begin Startup Sequence		no	FPGA Logic reset
SM0 Step 1	B4_FLED0 = 0, A3_FLED4 = 1, B3_FLED7 = 1, C4_F...		yes	FPGA LEDS: B4_FLED0 is on
SM0 Step 2	Wait for NOT C5_SMBA_FPGA		yes	Wait for a signal from the CPLD to start
SM0 Step 3	B4_FLED0 = 0, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LEDs ON
SM0 Step 4	Wait for 1000ms using FTimer2		yes	Wait
SM0 Step 5	B4_FLED0 = 1, A3_FLED4 = 0, B3_FLED7 = 0, C4_F...		yes	LED0 off
SM0 Step 6	Wait for 1000ms using FTimer2		yes	Wait
SM0 Step 7	If NOT C5_SMBA_FPGA Then Goto 2 Else Goto 8		yes	

Exceptio...	Boolean Expression	Outputs	Exception Handler	Comment
	<end-of-exception-t...			

SM0

Equation	Supervisory Logic Equation	Macrocell Configuration	Comment
	<end-of-supervisory-logic-table>		

The user can easily edit any of the sequencer steps in the CPLD or the FPGA to experiment with I/Os, LCD, LEDs and push-buttons. Any timer delay in the CPLD or FPGA can be edited by simply clicking on the timer line in the code. Once an edit is made, compile the design and generate a JEDEC file for download.

The JEDEC file is downloaded using Lattice ispVM programming software.

Implementation

Table 4. Performance and Resource Utilization[1]

Device Family	FPGA LUTs	CPLD Macrocells	CPLD Product Terms	VMONs	I/Os	Timers	HVOUTs
Platform Manager	38	20	100	5	22	4	0

1. Performance and utilization characteristics are generated using LPTM10-12107, with Lattice PAC-Designer 6.0 and ispLEVER® 8.1 SP1 Starter software. The performance and utilization may vary when using this design in a different device or software versions.

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.
October 2010	01.1	Added “Long Delay Timer Operation with the Platform Manager Development Kit” section.