Fault Logging and Monitoring Using Platform Manager Devices

October 2010 Reference Design RD1077

Introduction

For systems using microprocessors or computers there are usually numerous power supplies. If a power supply fails the power manager circuits may, as a minimum, force a shutdown. For maintenance and troubleshooting it is very desirable to know which power supply failed and the type of failure condition (over-voltage or under-voltage). This reference design presents a solution that records the supply fault condition in non-volatile memory so the fault(s) can read back at some later time. This solution is fast, reliable, and cost effective because, it is based on the Platform Manager™ and non-volatile SPI Flash memory.

Theory of Operation

This Fault Logging and Monitoring reference design uses a Lattice Platform Manager device to monitor the voltage levels in the system. The Platform Manager device is designed to monitor and control different power supplies within a system and has an on-board analog-to-digital (ADC) converter. The user can set high and low voltage alarm points within the device and these can then be used to initiate different control actions of the user's choosing. For this design, a voltage alarm will cause a fault status output to be driven high and the status of all the voltage monitor channels to be output on three data status lines.

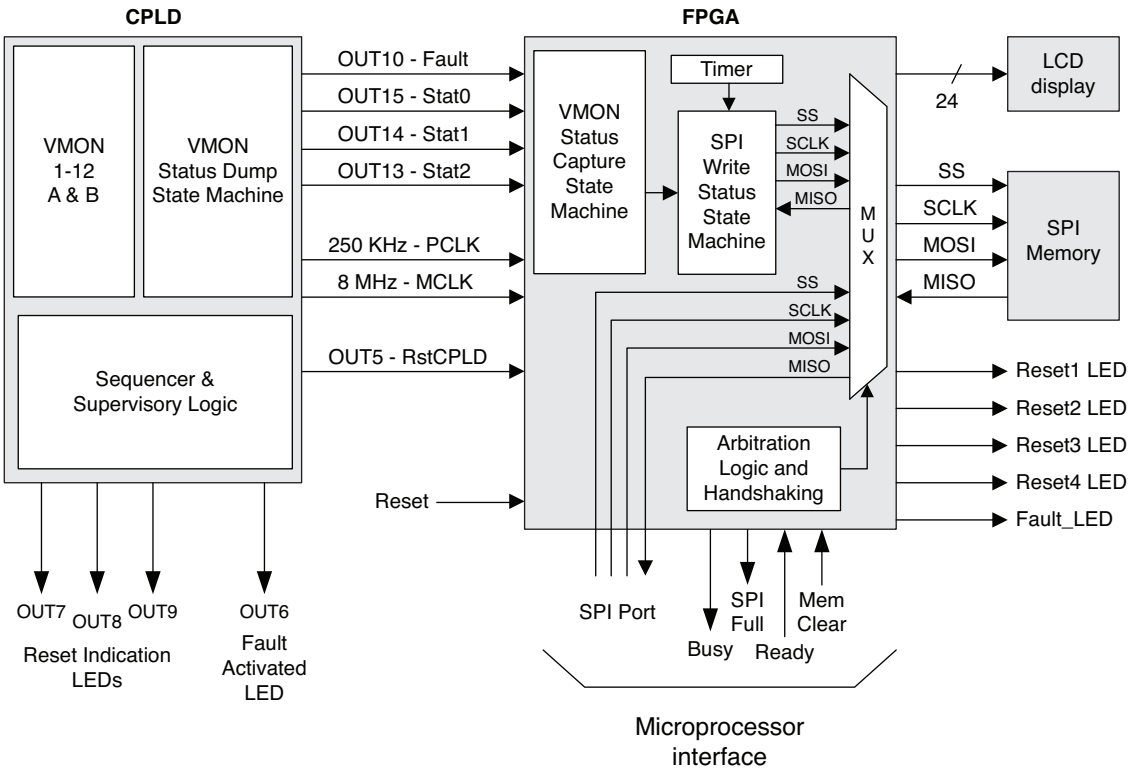
The fault status output and the three data status lines are connected to the FPGA section of the Lattice Platform Manager device which then captures the fault data, formats it, and writes the formatted data to a SPI Flash memory for later retrieval.

The sequence of events can be summarized as follows:

1. The Platform Manager detects a fault on one or more of the VMON inputs and dumps the VMON and I/O status to the FPGA section.
2. The dump of VMON and I/O status is implemented using supervisory logic equations and happens automatically using outputs and clock pins of the Platform Manager CPLD section.
3. The FPGA section implements a receiver state machine to capture the VMON and I/O status.
4. The FPGA section adds a time stamp to the VMON and I/O status information and sends the commands and writes the data to a SPI Flash memory device.

A block diagram of this Fault Monitoring and Logging Reference Design is shown in Figure 1.

Figure 1. Block Diagram of the Fault Monitoring and Logging Reference Design



Design Details

As shown in Figure 1, the fault logging process is divided between the CPLD and the FPGA portions of the Platform Manager device. The CPLD detects and dumps the faults to the FPGA that captures the faults. The FPGA then writes them to a standard non-volatile SPI memory. The FPGA also provides arbitration logic, a timer, and a MUX interface to the SPI memory for microprocessor support. This reference design also sequences the reset signals for seven devices; three from the CPLD and four from the FPGA section.

Figure 3 shows the details of the VMON and I/O status dump from the CPLD section of the Platform Manager to the FPGA section and Figure 4 shows the transfer of data from the FPGA section to the SPI Flash memory. The VMON and I/O status dump is generated within the CPLD section using an internal 250 KHz clock (fixed) which also drives the PCLK signal. The logic of the CPLD section is clocked from the PCLK signal by default. For this reference design the PCLK signal is used to drive the VMON Status Capture state machine inside the FPGA section to synchronize with the data output of the CPLD section.

The heart of this design is contained in the supervisory logic equations in the LogiBuilder window of PAC-Designer® (see Listing 1). One set of equations is used to implement a 4-bit binary up-counter and the second set of equations is used to dump all the VMON and I/O status values to the FPGA. When the counter is active the 3-bit bus of output pins is updated with the VMON status values on the rising edge of the PCLK signal. The rising edge of the Fault output (OUT10) is used to trigger the FPGA to capture the VMON and I/O status.

Equations 0 to 7 implement a 4-bit binary up-counter. This counter is reset to a value of 7 and then counts up using the sequence; 8, 9, 10, 11, 12, 13, 14 ,15, 0, 1, 2, 3, and 4 at the rate of the PCLK signal. The counter is enabled when NODE5 (named FAULT_ DUMPn) is set low in the main sequence (see Listing 2). When the counter reaches a value of 4 it sets FAULT_ DUMPn high which resets the counter and prevents multiple status dumps.

Equations 8 to 10 decode the 4-bit binary counter and combine the VMON and I/O status to provide the fault reporting to the FPGA. Equations 8 to 10 are combinatorial and provide a 3-bit data bus to the FPGA with the

VMON and I/O status. Each OR branch of the equation decodes the counter value and combines a unique VMON or I/O status so that all 24 VMON and the I/O status are transferred to the FPGA in 13 PLD clocks (52 µs). The FPGA uses the negative edge of the PCLK to latch the VMON and I/O status presented on the 3-bit bus (see Figure 3).

Equation 11 simply maps the most significant bit of the counter (Cntr3) to the output OUT10. This output transitions from low to high to trigger the FPGA to capture the status dump. The status lines and fault signal are shown in Figure 3 with the PCLK signal.

Listing 1. LogiBuilder Supervisory Equations

```
//  Equations 0 through 3 provide a 4-bit binary counter enabled by
//  FAULT_DUMPn
EQ 0     Cntr0 =  NOT Cntr0
EQ 1     Cntr1 =  ( Cntr0 AND NOT Cntr1 ) OR ( NOT Cntr0 AND Cntr1 )
EQ 2     Cntr2 =  ( NOT Cntr2 AND Cntr0 AND Cntr1 ) OR ( Cntr2 AND NOT
            ( Cntr0 AND Cntr1 ) )
EQ 3     Cntr3 =  ( NOT Cntr3 AND Cntr2 AND Cntr1 AND Cntr0 ) OR
            ( Cntr3 AND NOT ( Cntr2 AND Cntr1 AND Cntr0 ) )

//  The binary counter is reset to a count of 7 when the FAULT_DUMPn flag is
//  true (=1) and counts 8, 9, 10, 11, 12, 13, 14, 15, 0, 1, 2, 3, 4 when the
//  flag is low.
EQ 4     Cntr0.ap =  FAULT_DUMPn
EQ 5     Cntr1.ap =  FAULT_DUMPn
EQ 6     Cntr2.ap =  FAULT_DUMPn
EQ 7     Cntr3.ar =  FAULT_DUMPn

//  Equations 8 - 10 are combinatorial logic that decode the counter
//  value to select the VMON status to output. The three bits of VMON
//  status are updated on the rising edge of PCLK.

//  These status bits are decoded to fault line STAT0.
EQ 8     OUT15 =
       ( VMON1_A AND NOT Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON1_B AND     Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON2_A AND NOT Cntr0 AND     Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON2_B AND     Cntr0 AND     Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON3_A AND NOT Cntr0 AND NOT Cntr1 AND     Cntr2 AND     Cntr3 ) OR
       ( VMON3_B AND     Cntr0 AND NOT Cntr1 AND     Cntr2 AND     Cntr3 ) OR
       ( VMON4_A AND NOT Cntr0 AND     Cntr1 AND     Cntr2 AND     Cntr3 ) OR
       ( VMON4_B AND     Cntr0 AND     Cntr1 AND     Cntr2 AND     Cntr3 ) OR
       ( HVOUT1  AND NOT Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
       ( HVOUT2  AND     Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
       ( HVOUT3  AND NOT Cntr0 AND     Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
       ( HVOUT4  AND     Cntr0 AND     Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
       ( OUT5    AND NOT Cntr0 AND NOT Cntr1 AND     Cntr2 AND NOT Cntr3 );

//  These status bits are decoded to fault line STAT1.
EQ 9     OUT14 =
       ( VMON5_A AND NOT Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON5_B AND     Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON6_A AND NOT Cntr0 AND     Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON6_B AND     Cntr0 AND     Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
       ( VMON7_A AND NOT Cntr0 AND NOT Cntr1 AND     Cntr2 AND     Cntr3 ) OR
```

```
      ( VMON7_B AND     Cntr0 AND NOT Cntr1 AND     Cntr2 AND     Cntr3 ) OR
      ( VMON8_A AND NOT Cntr0 AND     Cntr1 AND     Cntr2 AND     Cntr3 ) OR
      ( VMON8_B AND     Cntr0 AND     Cntr1 AND     Cntr2 AND     Cntr3 ) OR
      ( OUT6    AND NOT Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( OUT7    AND     Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( OUT8    AND NOT Cntr0 AND     Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( OUT9    AND     Cntr0 AND     Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( OUT11   AND NOT Cntr0 AND NOT Cntr1 AND     Cntr2 AND NOT Cntr3 )

//  These status bits are decoded to fault line STAT2.
EQ 10   OUT13 =
      ( VMON9_A  AND NOT Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
      ( VMON9_B  AND     Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
      ( VMON10_A AND NOT Cntr0 AND     Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
      ( VMON10_B AND     Cntr0 AND     Cntr1 AND NOT Cntr2 AND     Cntr3 ) OR
      ( VMON11_A AND NOT Cntr0 AND NOT Cntr1 AND     Cntr2 AND     Cntr3 ) OR
      ( VMON11_B AND     Cntr0 AND NOT Cntr1 AND     Cntr2 AND     Cntr3 ) OR
      ( VMON12_A AND NOT Cntr0 AND     Cntr1 AND     Cntr2 AND     Cntr3 ) OR
      ( VMON12_B AND     Cntr0 AND     Cntr1 AND     Cntr2 AND     Cntr3 ) OR
      ( OUT12    AND NOT Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( IN1      AND     Cntr0 AND NOT Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( IN2      AND NOT Cntr0 AND     Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( IN3      AND     Cntr0 AND     Cntr1 AND NOT Cntr2 AND NOT Cntr3 ) OR
      ( IN4      AND NOT Cntr0 AND NOT Cntr1 AND     Cntr2 AND NOT Cntr3 )

//  Equation 11 is also combinatorial but only outputs the value of
//  the most significant bit of the counter. The rising edge of this
//  output is used to trigger the FPGA to capture the VMON status and
//  thus the faults. Based on the simple logic; Cntr3 is optimized
// by the fitter to OUT10. So the Cntr3 signal is missing
// from the simulation and fitter report.
EQ 11   OUT10 =  Cntr3
```

The main sequence (see Listing 2) is a very simple monitoring loop. Steps 0 and 1 provide the start-up and initialization of the outputs in a safe state and set the FAULT_ DUMPn flag high. Step 2 waits for the monitored supplies to reach a normal value to prohibit false indications of faults before the system is fully powered up. This step can be modified as required to meet the needs of the design being considered. Step 3 is a NOP which is inserted only to provide an additional comment in the design and can be removed if desired. Steps 4 to 6 provide reset sequencing for three devices and to the FPGA section of the Platform Manager. Step 7 monitors the status of VMON 8, VMON9, and VMON4 and Step 8 controls the FAULT_ DUMPn flag. Step 9 waits for both VMON8 and VMON9 to drop below their lower trip points before the design resets itself back to the beginning of the sequence. This step is mainly used to insure that only a single fault is logged for each event when demonstrating this reference design on the Platform Manager Evaluation Board. This step in the sequence should be modified to shut the system down or take other actions as appropriate for the design being considered.

Listing 2. LogiBuilder Sequence

```
//  CPLD Logic reset.
Step 0      Begin Startup Sequence

//  Wait for analog calibration. Set FAULT_DUMPn = 1.
Step 1      Wait for AGOOD

(Outputs)   HVOUT1 = 0, HVOUT2 = 0, HVOUT3 = 0, HVOUT4 = 0, OUT5 = 1,
            OUT6 = 1, OUT7 = 1, OUT8 = 1, OUT9 = 1, OUT11 = 1, OUT12 = 1,
            FAULT_DUMPn = 1

//  Wait for monitored supplies to be OK.
Step 2      Wait for VMON8_A AND VMON8_B AND VMON9_A AND VMON9_B

//  For comment only. VMON8 and VMON9 are on slider pots on demo board.
Step 3      NOP

//  Turn on LED D22 - first reset using timer 1 condition then wait for timer.
Step 4      Wait for 1048.58ms

(Outputs)   OUT7 = 0,

//  Turn on LED D23 - 2nd reset condition using timer 2 then wait for timer.
Step 5      Wait for 1572.86ms

(Outputs)   OUT8 = 0,

//   Turn on LED D24 - 3rd reset condition. Turn on reset to FPGA.
Step 6      OUT5 = 0, OUT9 = 0,

//  Wait here for fault condition. Can add additional fault conditions as
//  required.
Step 7      Wait for NOT VMON8_A OR NOT VMON8_B OR NOT VMON9_A OR NOT VMON9_B
            OR NOT VMON4_A OR NOT VMON4_B

//  Start Fault Dump sequence. Wait for counter to indicate dump step13 then
//  reset FAULT_DUMPn output.
Step 8      If NOT Cntr3 AND Cntr2 AND NOT Cntr1 AND NOT Cntr0
            Then Goto 9 with {FAULT_DUMPn = 1, }
            Else Goto 8

(Outputs)   FAULT_DUMPn = 0,

//  Turn Off resets. Then wait until both VMON 8 & 9 are below lower trip
//  point.
Step 9      Wait for NOT VMON8_A AND NOT VMON8_B AND NOT VMON9_A AND NOT
            VMON9_B

(Outputs)   OUT5 = 1, OUT6 = 0, OUT7 = 1, OUT8 = 1, OUT9 = 1,

//  Begin sequence over - Restarts Demo.
Step 10     Go to step 1
```

```
//  A shutdown sequence is not used in this design. One could be added if
//  desired.
Step 11     Begin Shutdown Sequence
Step 12     Halt (end-of-program)
```

When VMON8, VMON9, and VMON4 are all between the trip points the system is considered to be running normally. When one of these VMONs is outside the windowed trip points the FAULT_ DUMPn flag is activated and the system begins logging the status of all the VMON inputs, the digital inputs, the HVOUTS, and the digital outputs. The fault record does not include OUT10 or OUT13 through OUT15 since these are used by this reference design to log the faults. OUT16 is reserved for use by the long timer function so it is not logged either.

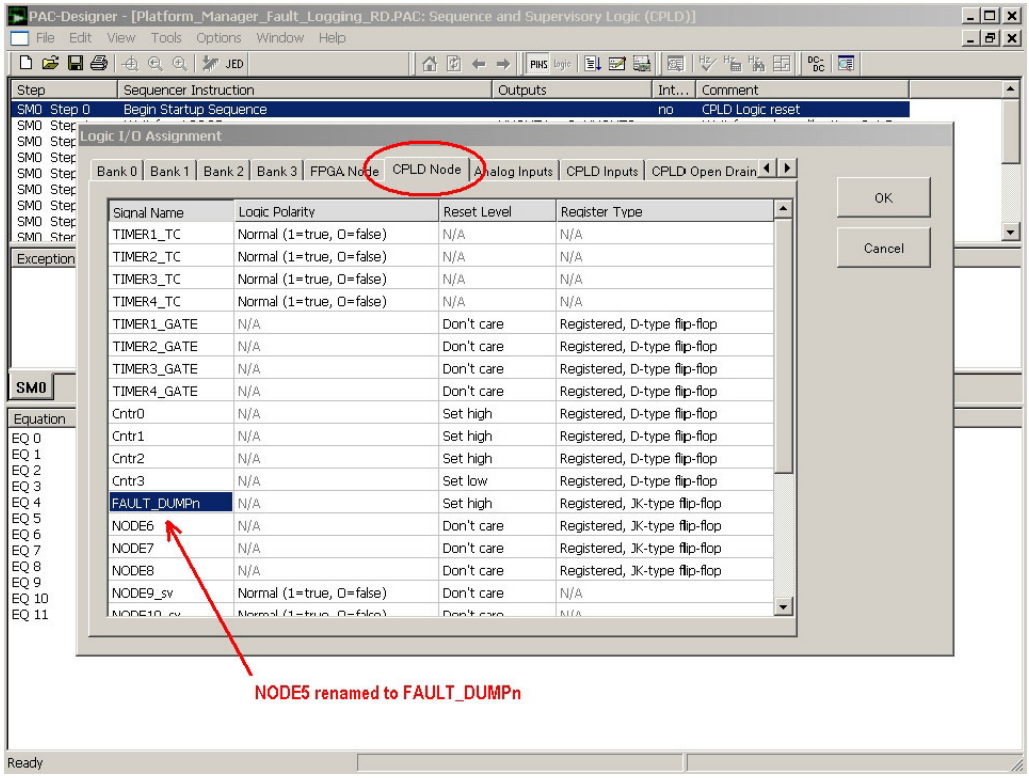
In the Analog Input Settings dialog (from the main schematic window) VMON 8 and VMON9 are configured in Window mode with the lower trip point of 1.5V and the upper trip point of 2.5V. On the Platform Manager Evaluation Board VMON8 and VMON9 are connected to the slide potentiometers to simulate either an under- or over-voltage situation. VMON4 is used to monitor the 5V supply voltage on the Platform Manager Evaluation Board and its lower trip setting is set to 4.5V while its upper trip setting is 5.55V. This allows this reference design to log a fault if the board power is removed while the system is running.

In this reference design the other analog input settings are all set to predefined limits which are applicable to the Platform Manager Evaluation Board. These settings are used to insure that the faults which are logged while using this reference design with the Platform Manager Evaluation Board produce a known fault record for comparison purposes, based upon the voltages being monitored on the evaluation board. The known fault record is shown in Table 3. All these analog input settings can be changed by the user to values applicable to their design when applying this reference design.

This design is easily modified to include any number of other VMON or input conditions to trigger the fault dump by adding to the Boolean logic in Step 7. Other sequences and controls could also be added before Step 7 based on the design requirements.

A key element to this design is the flag or internal signal FAULT_ DUMPn. It is through this signal that the main sequence tells the equations to dump the VMON and I/O status. This node is named using the Pin Definition dialog (see Figure 2) from LogiBuilder's PINS window. In this design the main sequence monitors VMON8 and VMON9 (the slide potentiometers on the Platform Manager Evaluation Board) and sets FAULT_ DUMPn low if the voltage is in error.

Figure 2. Naming Nodes in the Logic I/O Assignment Window



Another key element to this reference design is the SPI-MUX and arbitration logic. This allows the Platform Manger to share the SPI Flash Memory with any embedded or stand alone microprocessor or microcontroller. The arbitration logic prevents the two SPI masters from communicating to the SPI Flash memory at the same time.

The arbitration logic uses two outputs, the BUSY and MEM_FULL signals, to provide a status signal to the outside microprocessor or microcontroller. The BUSY signal is high when the FPGA section is writing a fault to the SPI memory or waiting to write a fault. The MEM_FULL signal is high when the SPI memory has faults written in each page and the FPGA section cannot write any additional faults to the memory. For this reference design the fault limit is set at 250 faults.

The arbitration logic also uses two inputs, the READY and MEM_CLEAR signals, to allow the microprocessor or microcontroller to signal the Platform Manager that it is communicating with the SPI memory directly. When the microprocessor wishes to communicate with the SPI memory directly it should check that the BUSY signal is low. If BUSY is high the microprocessor should wait for it to go low. Once the BUSY signal is low the microprocessor should set the READY signal low to switch the MUX to allow it to communicate with the SPI memory. When the microprocessor is done communicating with the SPI memory it must set the READY signal high again to allow the Platform Manager to log faults to the SPI memory. As long as the READY signal is low the Platform Manager cannot log a fault to the SPI memory. If a single fault condition occurs while the READY signal is low, the Platform Manager will capture that fault log and wait until the READY signal is high to write the log to the SPI memory. Additional faults while READY is low would be lost if they occur prior to the first fault being written to the SPI memory.

The microprocessor should pulse the MEM_CLEAR signal low when it is erasing the SPI memory to signal the Platform Manager that it should reset the memory address counter. This will allow the Platform Manager to begin logging faults at the beginning of the SPI memory address again. After the SPI memory has been cleared the MEM_CLEAR signal should be driven high again (or released as there is a internal pull-up on the signal) to wait for the next memory clear command.

For convenience, this reference design uses the 8 MHz MCLK signal from the CPLD section to drive the SPI memory write operations. (The MCLK signal of the Platform manager is a fixed rate clock.) A user design could use a faster clock signal if desired.

The entire operation takes less than 120us to store the fault record to the SPI Flash memory as shown in Figures 3 and 4.

Figure 3. Fault Logging Status Dump Waveforms

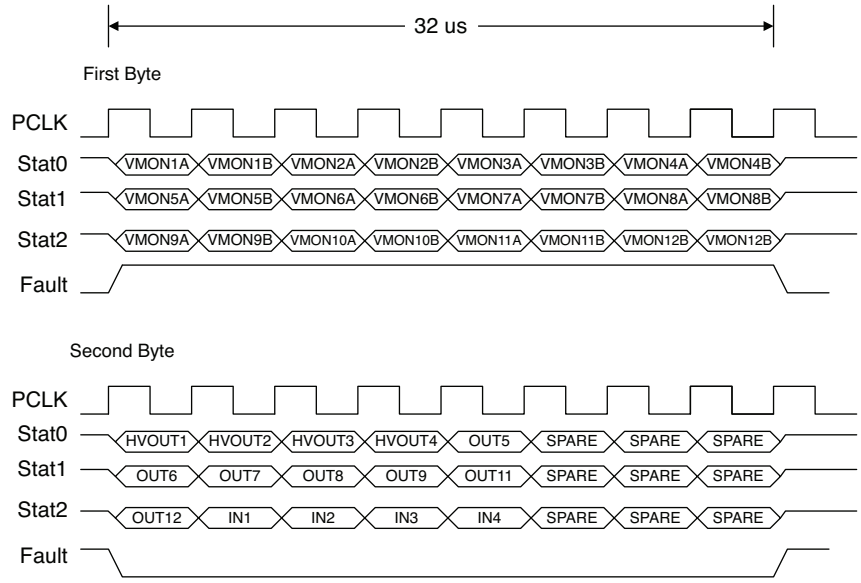
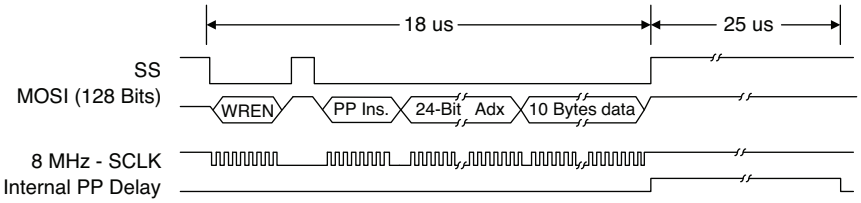


Figure 4. Fault Logging SPI Page Program Waveforms



To simplify the design and minimize the SPI write time, each page within the SPI Flash is dedicated to a single event log. The first byte of each page is used as a Fault Flag. If the page is erased the Fault Flag will have a value of 0xFF. If the page has any fault data recorded then the Fault Flag will have a value of 0xC3. This allows both the fault logger and an external fault reader (the microprocessor) to know which pages have valid data. For this reference design 10 bytes are used to log a fault and the organization of the bytes within a page is shown in Tables 1 and 2.

Table 1. SPI Page Memory Map byte arrangement

Byte Address	Byte Description	Comments
0x00	Fault Flag	0xFF = No fault, page empty : 0xC3 = Fault log, page written
0x01	VMON1 - VMON4	See Table 2 for details of fault order
0x02	VMON5 – VMON8	See Table 2 for details of fault order
0x03	VMON9 – VMON12	See Table 2 for details of fault order
0x04	OUT1 – OUT8	See Table 2 for details of fault order
0x05	OUT9, OUT11, OUT12, IN1-IN4	See Table 2 for details of fault order
0x06	Seconds_0	Least significant byte – long integer
0x07	Seconds_1	
0x08	Seconds_2	
0x09	Seconds_3	Most significant byte – long integer

Table 2. VMON Fault Memory Map Bit Arrangement

Byte Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x01	VMON4_B	VMON4_A	VMON3_B	VMON3_A	VMON2_B	VMON2_A	VMON1_B	VMON1_A
0x02	VMON8_B	VMON8_A	VMON7_B	VMON7_A	VMON6_B	VMON6_A	VMON5_B	VMON5_A
0x03	VMON12_B	VMON12_A	VMON11_B	VMON11_A	VMON10_B	VMON10_A	VMON9_B	VMON9_A
0x04	OUT8	OUT7	OUT6	OUT5	HVOUT4	HVOUT3	HVOUT2	HVOUT1
0x05	IN4	IN3	IN2	IN1	SPARE	OUT12	OUT11	OUT9

In this reference design the memory map bytes 0x06 through 0x09 contain a 32-bit binary representation of the timer value in seconds. The timer can count up to a value of 2^{32} -1 seconds which is equivalent to over 136 years. The timer will be reset when the device is powered down or a reset is issued.

Start-up and Reset Condition

When the design is powered up or a reset is issued, the reference design will begin by reading the SPI Flash memory to determine if any faults are stored in the memory at the current time. The presence of a fault in a page of the SPI Flash is indicated by byte address 0x00 as shown in Table 1. If a fault is present in a page the design will increment the page address counter to the next page address and re-check for a fault in the new page. This will continue until an empty SPI page is found and then the next fault will be logged into this location. This allows any existing faults to remain logged into the SPI Flash memory until the microprocessor issues a SPI Flash memory erase command. At this time the microprocessor will also set the MEM_CLEAR status bit low so the reference design will know that a memory clear has occurred. When the MEM_CLEAR status bit is received the reference design will reset the address counter to begin at page 00. Because of this design feature, the faults will be logged into the SPI Flash memory in sequential order.

Operation

To use this reference design with the Platform Manager Evaluation Board, follow the steps listed below.

1. Move both slider pots to the lower end of their range (toward DIP switches SW9 and SW10).
2. Install a jumper between D15 and the SMBA pins on header J15.
3. Power-on the board and insure that DIP SW2 is turned ON (all four positions).
4. Download the JEDEC file using ispVM™ System.
5. After the download has completed successfully, press the reset switch, S1 (next to DIP SW9).
6. Move both slider pots toward the middle of their range. When LED D22 turns on (located above S1) the slider pots are in their normal operating range.

7. The two LEDs (D23 and D24) above D22 should sequence on after D22 turns on.
8. The LEDs D20, D19, D15, and D13 should sequence on after D24 is turns on.
9. The number of faults stored in the SPI memory should be displayed on the LCD at this point; on a new board, zero will be displayed. If there are one or more faults stored in the SPI memory then LED D3 will blink also.
10. Once LEDs D22, D23, & D24 are all illuminated the fault logging pattern is ready to log a fault condition.
11. Move one of the slider pots to the minimum or maximum position to generate a fault condition.
12. When the fault is logged the LCD display should increment and LED D3 will begin to blink. LEDs D22, D23, D24, D20, D19, D15, and D13 will turn off and LED D21 will turn on.
13. Move BOTH slider pots to their minimum position to reset the sequence. LED D21 will turn off when this is done.
14. To log additional faults, move both slider pots to the middle of their operating range and repeat Steps 7-14.
15. The fault record in the SPI memory can can be read back and displayed using the PlatformManager_12107_I2C_Utility which is provided with PAC-Designer 6.0. This utility can be found in the “Design Utilities” item listed under the “Tools” menu.

Table 3. Standard Fault Record for Operation with the Platform Manager Evaluation Board[1, 2]

VMON or I/O Status	Value1
VMON1	A = 0, B = 0
VMON2	A = 0, B = 0
VMON3	A = 1, B = 1
VMON4	A = 1, B = 1
VMON5	A = 0, B = 0
VMON6	A = 0, B = 0
VMON72	A = 1, B = 1
VMON8	per slider
VMON9	per slider
VMON10	A = 1, B = 1
VMON11	A = 0, B = 0
VMON12	A = 0, B = 0
HVOUT1	0
HVOUT2	0
HVOUT3	0
HVOUT4	0
OUT5	0
OUT6	1
OUT7	0
OUT8	0
OUT9	0
OUT11	1
OUT12	1
IN1	1
IN2	1
IN3	1
IN4	1

1. A VMON value of 00 represents and under voltage condition. A VMON value of 11 represents a good reading (using the Windowed mode). A VMON value of 01 represents an over voltage condition. A value of 10 is an invalid reading.
2. VMON7 monitors the Thermister voltage on the board. This value could change depending upon the ambient temperature of the board.

Implementation

See the Readme.txt file in the zip file for instructions on implementing this reference design using PAC-Designer 6.0.

Table 4. Performance and Resource Utilization[1]

Device Family	FPGA LUTs	CPLD Macrocells	CPLD Product Terms	VMONs	I/Os	Timers	HVOUTs
Platform Manager	400	31	147	3[2]	60[3]	2	0

1. Performance and utilization characteristics are generated using LPTM10-12107, with Lattice PAC-Designer 6.0 and ispLEVER 8.1 Starter software. The performance and utilization may vary when using this design in a different device or software versions.
2. The VMON inputs used for this design are only needed to demonstrate the design on the Platform Manager Evaluation Board. All 12 VMON inputs are available for the user when implementing this design.
3. The number of I/Os includes 49 I/Os for the FPGA and 11 CPLD outputs. This is for the Fault Monitoring and Logging Reference Design with a SPI port interface which is available for a microprocessor or microcontroller. The external port for the microprocessor uses 4 I/Os for the SPI port plus another 4 I/Os for the arbitration logic. The external SPI port for the memory uses only 4 I/Os. There is also an LCD display which shows the number of faults stored in the SPI memory which uses 24 I/Os.

References

- DS1036, Platform Manager Data Sheet

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.