Real-Time Calendar Applications in Actel Fusion® Devices

Introduction

The Actel Fusion Programmable System Chip is the world's first mixed-signal FPGA, integrating configurable analog, a real-time counter (RTC), Flash memory blocks, and clock generation circuits—all in a single chip. The embedded RTC can be used to create a power-cycle timer, an alarm clock, a calendar, or any applications that require time and date stamp information. This application note provides detailed information and reference HDL code for implementing a real-time calendar, which can be demonstrated using the Fusion Starter Kit (AFS-EVAL-KIT).

Design Description

Using the Fusion device's RTC, this application provides a count of seconds, minutes, hours, day of the week, day of the month, month, and year. The month-ending date is automatically adjusted for months with less than 31 days, including corrections for leap years. Moreover, the FPGA core can be powered off to save power, and the RTC will continue to keep track of the time and date information. Figure 1 shows the simplified block diagram of this reference design.

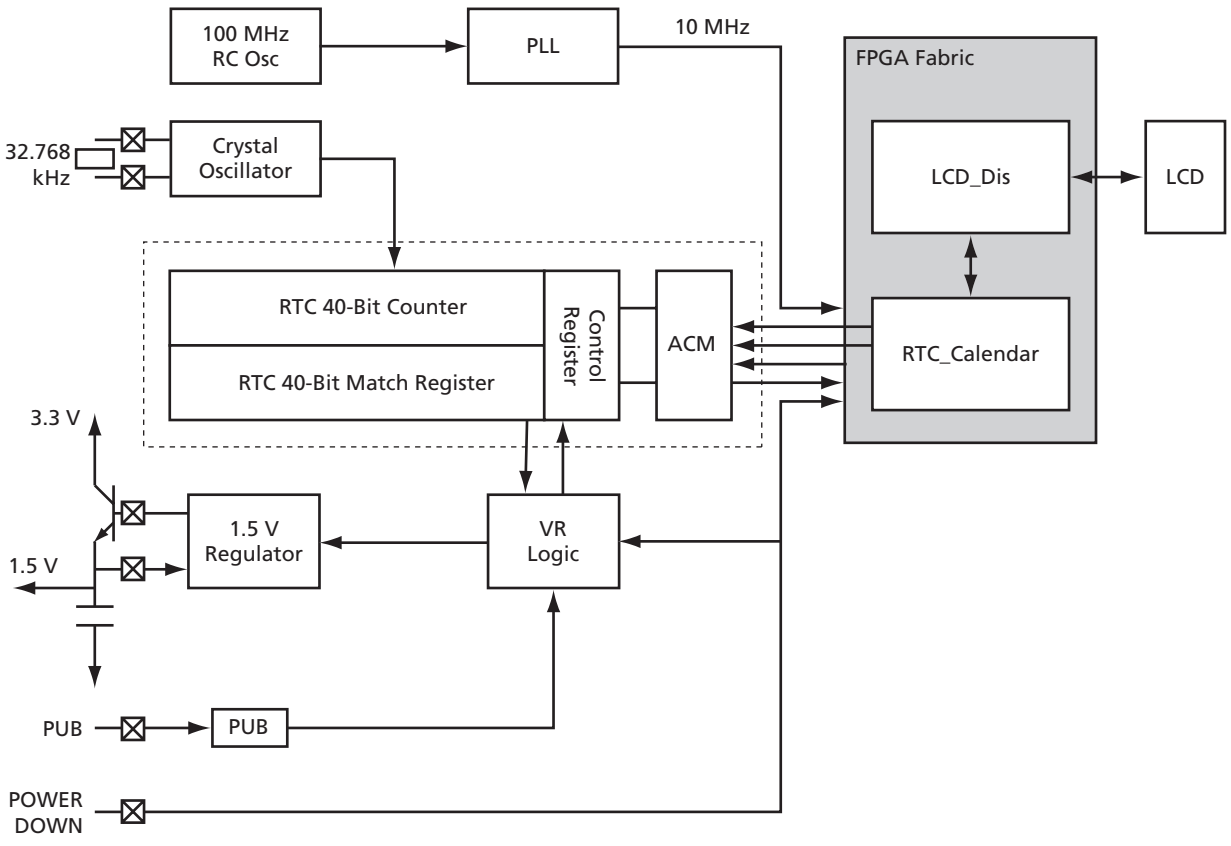

Figure 1 • **Real-Time Calendar – Top-Level Block Diagram**

RTC and Crystal Oscillator

The RTC is an embedded hardware block in Fusion devices. It is a 40-bit counter with a 40-bit match register and match flag accessible from the FPGA core. Based on the match register, the match flag, or other conditions, the FPGA core can be powered up or down via the internal 1.5 V regulator to reduce power consumption while the RTC remains active to keep track of the time. In this type of self-timed wake-up or restart operation, the only power supply required for device opeation is the 3.3 V supply to the analog system. In this reference design, the RTC is used to keep track of elapsed time so the corresponding time, day of the week, day of the month, month, and year can be calculated.

When using the RTC system, if a 32.768 kHz external crystal is connected to the embedded crystal oscillator pad of the device, the prescaler block of the crystal oscillator divides the frequency by 128, which gives bit 7 of the RTC a period of 1 second.

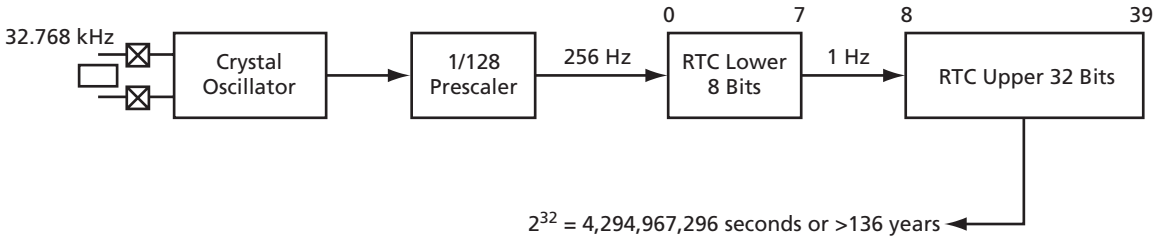

Figure 2 • **Using the Crystal Oscillator with the RTC**

Voltage Regulator (VR) Logic

Using the 3.3 V power supply and an external pass transistor, the Fusion FPGA core can be powered from the on-chip 1.5 V voltage regulator, which is then routed externally to the device's 1.5 V V_{CC} pins. Recommended transistors for use with the voltage regulator are PN2222A or 2N2222A transistors.

To use the 1.5 V embedded voltage regulator, create the VR Logic block using the SmartGen Voltage Regulator Power Supply Monitor Core (Figure 3). The active low PUB (Power Up Bar) signal is connected to a dedicated PUB pin on the device and is used to wake up the FPGA core from standby mode. On the other hand, the VRPU power-down signal of the VR_logic block is used to power down the FPGA core to enter standby mode. In this reference design, PUB and VRPU are connected to external switches to demonstrate how the time and date can be tracked while the FPGA core is off. Figure 4 on page 3 shows the ports of the VRPSM macro, and Table 1 on page 3 gives a description of each port.

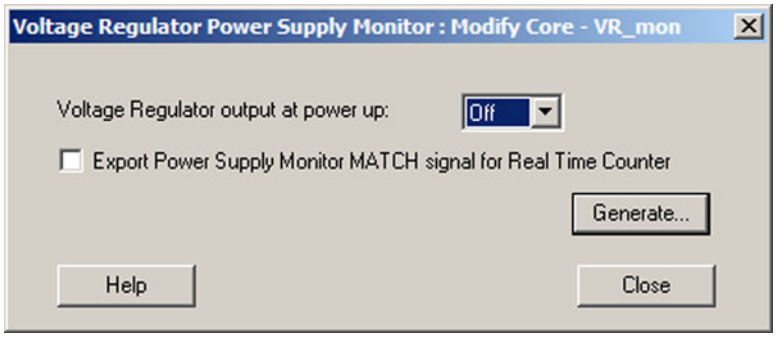

Figure 3 • **SmartGen Voltage Regulator Power Supply Monitor**

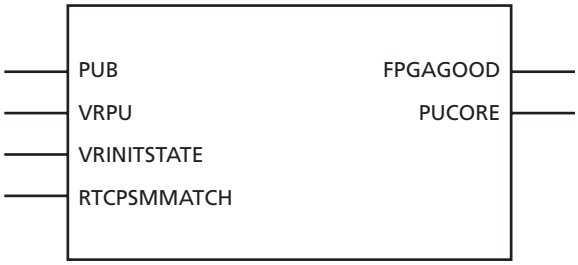

Figure 4 • **Voltage Regulator Power Supply Monitor (VRPSM)**

Table 1 • **VRPSM Macro Signals**

Port Name	Number of Bits	Direction	Function
PUB	1	Input	Active low signal to power up the FPGA core via the 1.5 V regulator. In this reference design, PUB is on the top level connected to an external switch.
VRPU	1	Input	When this pin is at logic '1', the FPGA core will be turned off via the voltage regulator.
VRINISTATE	1	Input	This feature is not used in this reference design and is not shown in the macro generated by SmartGen. If used, the signal enables you to set your voltage regulator output at power-up (ON or OFF).
RTCPSMMATCH	1	Input	This feature is not used in this reference design. If used, this active high signal is driven by the RTC's match signal to indicate that the RTC counter value matches with the pre-defined Match Register value set in SmartGen.
FPGAGOOD	1	Output	Logic '1' indicates that FPGA is logically functional.
PUCORE	1	Output	This signal is the inverse of PUB signal for the FPGA core.

RTC Analog System and Flash Memory System

In the Fusion design flow, RTC is considered one of the available peripherals of the analog system. Depending on the peripherals used in the analog system, SmartGen Analog System Builder will generate an appropriate netlist (*Analog_RTC.vhd*) for the analog system (Figure 5). In this reference design, only the RTC peripheral is used. Moreover, the analog system netlist generated from SmartGen includes an interface IP module called Analog Configuration MUX (ACM), which is the interface between the FPGA, the Analog Block configurations, and the RTC.

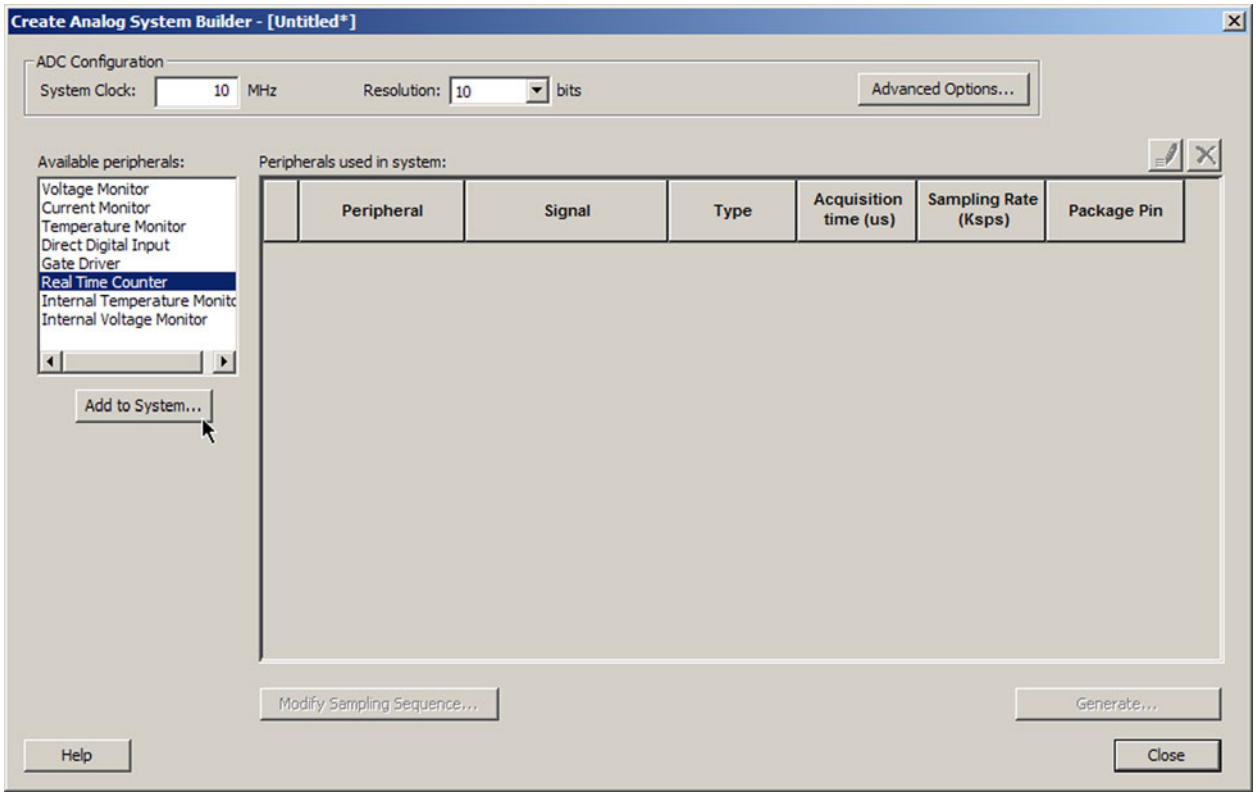

Figure 5 • **Creating the RTC Analog System Using SmartGen**

Whenever the analog system is used, a Flash Memory System is needed to initialize the analog system during power-up. During device power-up (when the 3.3 V power supply reaches a threshold value), the Flash Memory System initializes the analog system through the ACM. The Flash Memory System (*NVM_analog.vhd*) is used to store the configuration bits for all analog system peripheral settings. Users are required to connect the corresponding ports between the analog system and Flash Memory System, as described in Table 2 on page 5. The Flash Memory System is created using the SmartGen Flash Memory System Builder (Figure 6 on page 5). The use of the Flash memory in this way does not reduce any of the linear address space available for user storage that the user may need for other applications; initialization of the Analog Block uses a special auxiliary page in the Flash memory, thereby maintaining the full storage capacity for the user.

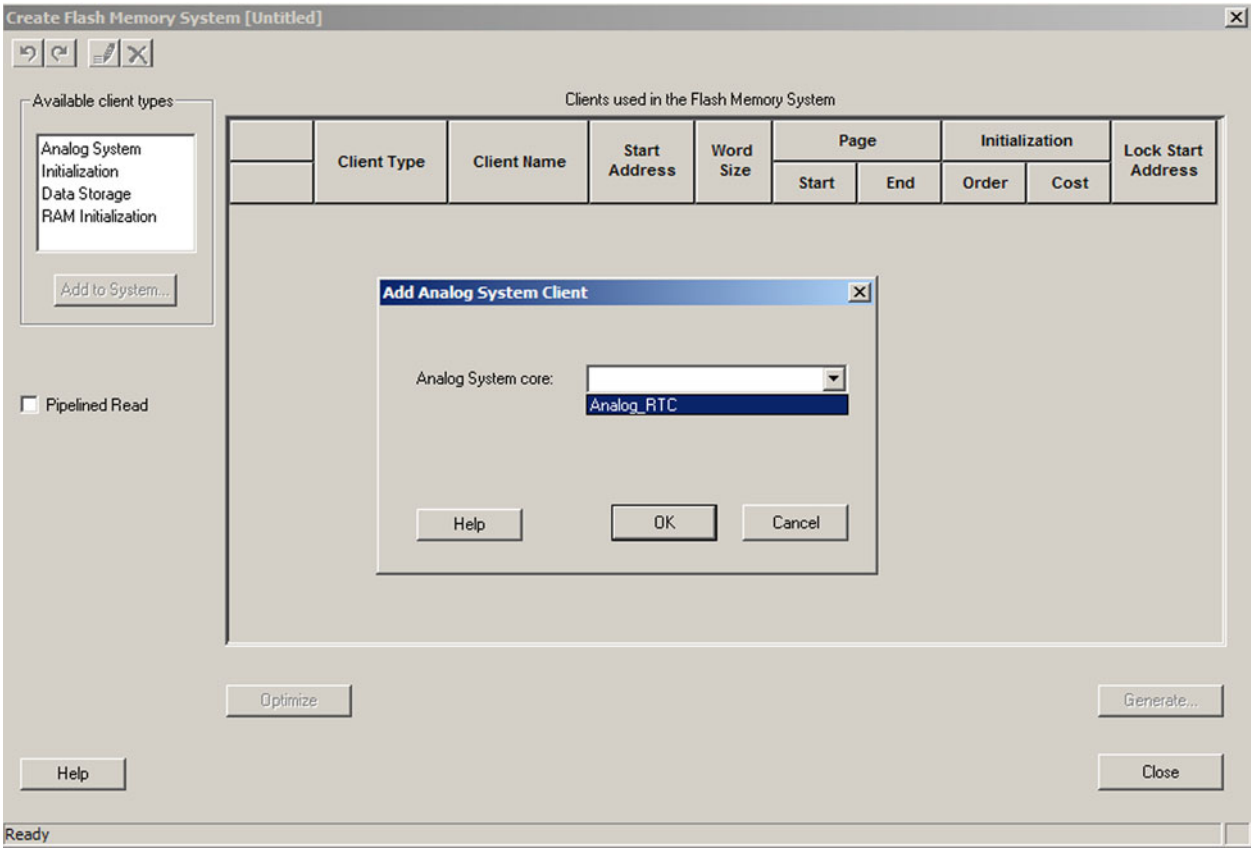

Figure 6 • **Creating a Flash Memory System for the RTC Analog System**

Table 2 • **Interface Signals between RTC Analog System and Flash Memory System**

Flash Memory System (from)	RTC Analog System (to)	Number of Bits	Description
INIT_ADDR	INIT_ADDR	9	Initialization address from Flash Memory Block. Maximum address location required by the largest of the analog system RAMs. In this design, it is controlled by the application.
INIT_DATA	INIT_DATA	9	Initialization data from Flash Memory Block. In this design, it is controlled by the application.
INIT_DONE	INIT_DONE	1	When the initiation is completed for the RTC Analog system, INIT_DONE = '1'.
INIT_ACM_WEN	INIT_ACM_WEN	1	ACM Write Enable. In this design, it is controlled by the application.
INIT_ASSC_WEN	INIT_ASSC_WEN	1	Enable the ASSC for initialization from Flash Memory Block.
INIT_EV_WEN	INIT_EV_WEN	1	Enable the SMEV for initialization from Flash Memory Block.
INIT_TR_WEN	INIT_TR_WEN	1	Enable the SMTR for initialization from Flash Memory Block.

Table 3 • **NVM_analog Signals**

Flash Memory System	Number of Bits	Direction	Description
INIT_CLK	1	Input	10 MHz system clock from the output of the PLL.
SYS_RESET	1	Input	Active low system reset signal
INIT_POWER_UP	1	Input	Set to '1'. Indicates the system is ready for Flash Memory Block initialization. Active high.
INIT_ADDR	9	Output	Initialization address from Flash Memory Block. In this design, the analog system initialization is controlled by RTC_Calendar.
INIT_DATA	9	Output	Initialization data from Flash Memory Block. In this design, the analog system initialization is controlled by RTC_Calendar.
INIT_DONE	1	Output	When initialization is completed for the RTC analog system, INIT_DONE = '1'.
INIT_ACM_WEN	1	Output	Active high ACM Write Enable. In this design, this is not used. The analog system ACM Write Enable is controlled by RTC_Calendar.
INIT_ASSC_WEN	1	Output	Enable the ASSC for initialization from Flash Memory Block – active high.
INIT_EV_WEN	1	Output	Enable the SMEV for initialization from Flash Memory Block – active high.
INIT_TR_WEN	1	Output	Enable the SMTR for initialization from Flash Memory Block – active high.
INIT_ACM_RTC_WEN	1	Output	Enable the RTC section of the ACM for initialization from the Flash Memory System Builder. In this design, it is not used and is left floating.

Table 4 • **RTC Block Signals**

Port Name	Number of Bits	Direction	Function
SYS_CLK	1	Input	System clock. In this design, it is a 10 MHz system clock from the PLL.
SYS_RESET	1	Input	Active low system reset signal
VAREF	1	Bidirectional	Voltage Reference. Connects to external voltage for external voltage reference. Returns the internal V_{REF} in the case of internal voltage reference. Must be connected to a top-level port without any I/Os.
DATAVALID	1	Output	Indicates that the data from ADC is valid.
ASSC_DONE	1	Output	Indicates that ASSC finished processing the current sample.
ASSC_WAIT	1	Output	Wait signal – not used in the design.
ASSC_CHSAT	1	Output	Sampled Channel Saturated
ASSC_CHLATD	1	Output	Channel Selector Latched
INIT_ADDR	9	Input	Initialization address from Flash Memory Block. Maximum address location required by the largest of the analog system RAMs. In this design, this is controlled by the RTC_Calendar module.

Table 4 • **RTC Block Signals (Continued)**

Port Name	Number of Bits	Direction	Function
INIT_DATA	9	Input	Initialization data from Flash Memory Block. In this design, this is controlled by the RTC_Calendar module.
INIT_ACM_WEN	1	Input	ACM Write Enable. In this design, this is controlled by RTC_Calendar.
INIT_ACM_RTC_WEN	1	Input	Enable the RTC section of the ACM for initialization from the Flash Memory System Builder. In this design, it is set at constant '0'. Active high.
INIT_ASSC_WEN	1	Input	Enable the ASSC for initialization from Flash Memory Block.
INIT_EV_WEN	1	Input	Enable the SMEV for initialization from Flash Memory Block.
INIT_TR_WEN	1	Input	Enable the SMTR for initialization from Flash Memory Block.
INIT_DONE	1	Input	This is connected to INIT_DONE on the NVM_analog block. When initialization is done, INIT_DONE = '1'.
ACMCLK	1	Input	Clock for writing to ACM. Used with INIT_ADDR and INIT_DATA to update the RTC Match Register and Counter. It is exposed when the user selects access to the ACM bus. This signal needs to be connected to a clock slower than 10 MHz. In this design, it is connected to the 10 MHz system clock from the PLL.
ACMRDATA_I	8	Output	ACM Read Data Bus. Used with ACMCLK and INIT_ADDR to read the contents of ACM. It is exposed when the user selects access to the ACM bus.
RTCCLK	1	Input	RTC Clock. Must be driven by CLKOUT on XTAL_RTC.
RTCXTLSEL	1	Output	Crystal Oscillator Select. Must be connected to XTLSEL on the XTLOSC macro.
RTCXTLMODE_I	2	Output	Crystal Oscillator Mode Select. This signal must be connected to RTCXTLMODE on the XTLOSC macro.
RTCMATCH	1	Output	When the RTC value matches the match register value, this signal goes high. This is not used in this design.

Calendar

RTC_Calendar is the core of this real-time calendar application (Figure 7 on page 8). It handles the initialization and all calculations based on the value of the RTC. The time and date are calculated using BCD (binary-coded decimal) counters. One of the key features of this application is that the 1.5 V FPGA core can be powered down to save power while 3.3 V is supplied so the RTC can keep track of the elapsed time.

When the external switch used in this demo design is driven low (POWER_down), the registers used to keep track of the current time and date are stored in the match register of the RTC, which is still active as long as the 3.3 V power supply is provided. This context-saving process will take seven clock cycles. At the same time, the RTC begins to count the elapsed time from the time the FPGA core is turned off to the time the FPGA core is powered on again.

In this application, the FPGA core is turned on using an external switch (PUB), which is connected to the PUB port of the VR_mon module. Once the voltage regulator enables the FPGA core, the information saved in the match register is restored and, together with the elapsed time recorded by the RTC, the current time and date are calculated and displayed on the demo board LCD. By default, the design has a default setting of Tuesday, Feb 28, 2006 at 11:59:45 P.M. Table 5 on page 8 lists the signals for this functional block.

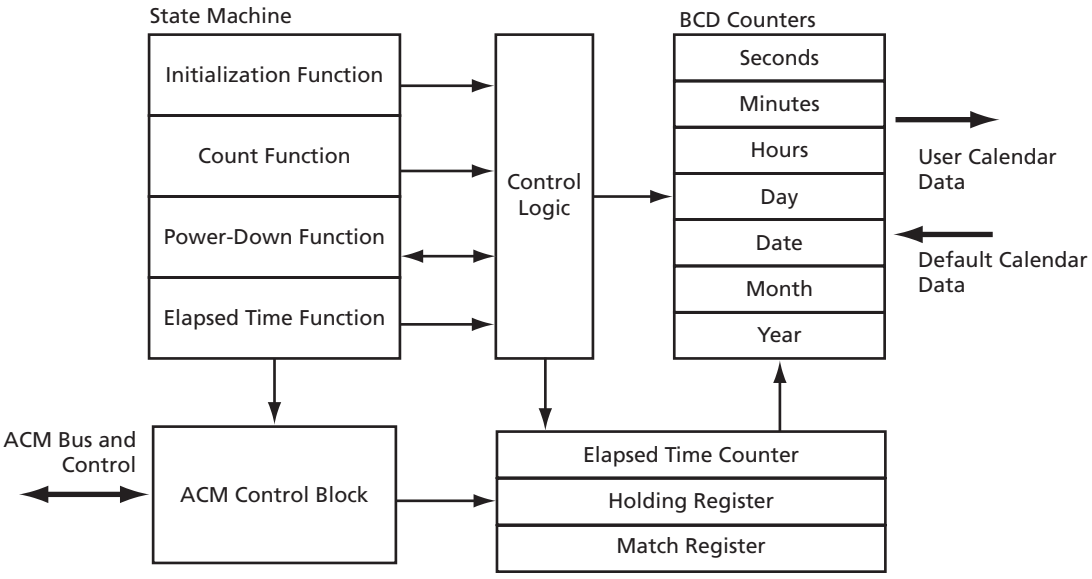

Figure 7 • **RTC_Calendar Functional Diagram**

Table 5 • **RTC_Calendar Signals**

Port Name	Number of Bits	Direction	Function
SYS_CLOCK	1	Input	System clock. In this design, it is a 10 MHz clock from the PLL.
SYS_RESET	1	Input	Active low system reset signal
INIT_DONE	1	Input	This instance is activated when INIT_DONE = '1'.
POWER_down	1	Input	Connected to the external pin. When pressed (logic '0'), the FSM in this block saves the content to the RTC register and powers down the FPGA core voltage to save power.
clk1hz	1	Input	A 15-bit counter, RTCcount, is used to divide the 32768 Hz clock by 32768 to generate a 1 Hz reference clock signal for RTC_Calendar.
ACMRDATA	8	Input	Connected to ACMRDATA_I on the RTC block. This is used to access the counter and match register values of the RTC.
ACMADDR	8	Output	RTC Memory Map Address. 0x48–0x4C is the address of the match register. 0x40–0x44 is the memory map for the counter value.
ACMWDATA	8	Output	Write data. Store the current time data in the match register.
ACMWEN	1	Output	Write enable signal for the RTC system.
LOAD_rtc	1	Input	Default value: '0'
LOAD_year	8	Input	Default value: "06" = year 2006
LOAD_month	8	Input	Default value: "02" = February.
LOAD_date	8	Input	Default value: "28" = 28th day of month
LOAD_day	8	Input	Default value: "02" = Tuesday
LOAD_hours	8	Input	Default value: "71" = 11 hours
LOAD_minutes	8	Input	Default value: "59" = 59 minutes
LOAD_seconds	8	Input	Default value: "45" = 45 seconds
RTC_act	1	Output	Debug out signal to indicate that the RTC is active.

Table 5 • **RTC_Calendar Signals (Continued)**

Port Name	Number of Bits	Direction	Function
RTC_year	8	Output	The year data is connected to the RTC_year port on LCD_DIS.
RTC_month	8	Output	The month data is connected to the RTC_month port on LCD_DIS.
RTC_date	8	Output	The date data is connected to the RTC_date port on LCD_DIS.
RTC_day	8	Output	The day data is connected to the RTC_day port on LCD_DIS.
RTC_hours	8	Output	The hours data is connected to the RTC_hours port on LCD_DIS.
RTC_minutes	8	Output	The minutes data is connected to the RTC_minutes port on LCD_DIS.
RTC_seconds	8	Output	The seconds data is connected to the RTC_seconds port on LCD_DIS.

LCD Block

LCD_DIS.vhd is the LCD driver, which continuously displays the current time (in *hh:mm:ss* format) and date (in *mm-dd-yy* format). By default, the LCD will display the time. SW2 is connected to the external switch, and when pressed, the LCD will be switched to display the date. Table 6 lists the signals for this function block.

Table 6 • **LCD_DIS Signals**

Port Name	Number of Bits	Direction	Function
CLK	1	Input	10 MHz clock from the output of the PLL
RS_LCD	1	Output	Debug signal. Register select input. High for data input and low for instruction register.
DATA_LCD	4	Output	Debug signal. Data output to the LCD to display the time and date information.
EN_LCD	1	Output	Debug signal. Active high signal. Enable signal for data read/write.
R_nW_LCD	1	Output	Debug signal. Read/write signal. High for read mode and low for write mode.
RTC_year	8	Input	Year data from RTC_Calendar
RTC_month	8	Input	Month data from RTC_Calendar
RTC_date	8	Input	Date data from RTC_Calendar
RTC_hours	8	Input	Hours data from RTC_Calendar
RTC_minutes	8	Input	Minutes data from RTC_Calendar
RTC_seconds	8	Input	Seconds data from RTC_Calendar
SW1	1	Input	Active high. Resets the LCD display.
SW2	1	Input	Active high. When asserted, the LCD displays the date in mm-dd-yy format.

Utilization

In this application, one of the goals is to implement all functions using the minimum possible FPGA logic tile resources. With power saving mode, the displayed time and date only utilize approximately 17% of the FPGA core logic tiles in an ASF600 device (Table 7). The reference code provided with this application note can easily be modified to fit any application that needs calendar functionality.

Table 7 • **AFS600 Reference Design Device Utilization**

Functional Block	Tiles	Percentage
RTC_Calendar	700	5%
LCD_DIS	250	1.8%
NVM_analog	230	1.6%
Analog_RTC*	1120	8.1%
Other miscellaneous logic	73	0.5%
Total	2373	17%

Note: **If an analog system is already instantiated in the design, the incremental resources required by the analog system for the RTC are negligible.*

Demonstration

The Actel Fusion Starter Kit (order code: AFS-EVAL-KIT) can be used to demonstrate and validate this application. By default, the LCD displays the default time set in the design, and it can be switched to display the date. Table 8 lists the functions of the switches on the Fusion Starter Kit used in this application.

Table 8 • **Switch Functions**

Function	Port Name	Pin Number	Switch
Date/time switch	SW2	A11	SW2
Power down	POWER_down	C11	SW3
System reset	SYS_RESET	C10	SW4
Power up	PUB	R15	SW7

To demonstrate the power-down mode, press SW3 to shut down the FPGA core via the voltage regulator. At this time, the RTC will begin counting how long the FPGA core has been shut off. Press SW7 to wake up the FPGA core. The design will display the current time based on the elapsed time from the RTC and the shut-off time in the RTC match register. On the Fusion Starter Kit, make sure JP28 is connected between pins 1 and 2 and a 470 Ω resistor is used to tie the TRST pin to ground.

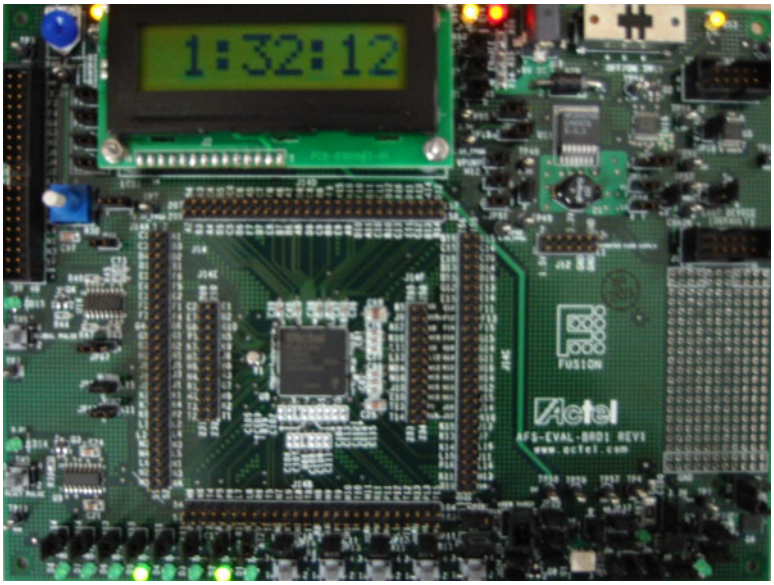

Figure 8 • **Fusion Starter Kit Demo – Time**

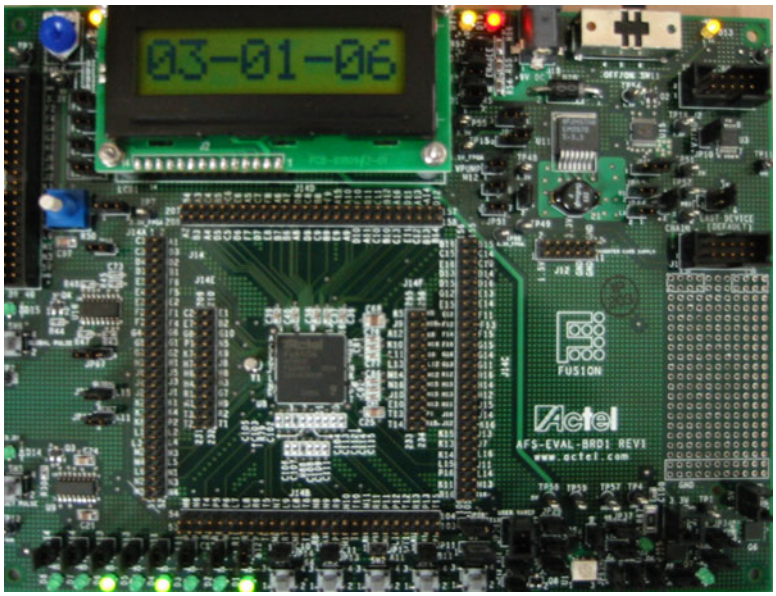

Figure 9 • **Fusion Starter Kit Demo – Date**

Conclusion

This real-time calendar reference design shows how easily this feature can be added to a Fusion design. The ability to shut down the FPGA core to save power is important in most handheld or power-saving applications. The RTC block in Fusion FPGAs allows designers to integrate a complete real-time clock application into the FPGA with the desired set of features.

References

Fusion Family of Mixed-Signal Flash FPGAs datasheet
http://www.actel.com/documents/Fusion_DS.pdf
Real-Time Clock in Actel Fusion FPGAs
http://www.actel.com/documents/Fusion_RTC_AB.pdf

Appendix – Design Files Summary

The full design can be downloaded at http://www.actel.com/documents/RTC_Demo.zip. Table 9 gives descriptions of key design source files.

Table 9 • **Key VHDL Source Code**

File	Functionality
RTC_demo.vhd	Top-level wrapper
RCO.vhd	RC oscillator generated by SmartGen
PLL_100_10.vhd	PLL generated by SmartGen with 100 MHz input and 10 MHz output configuration
VR_mon.vhd	Voltage monitor power supply management
Shift_reg.vhd	8-bit shift register. When the power-down signal is activated, it takes seven cycles to save the calendar data into the RTC match register. The shift register is used to delay the actual shut-down of the FPGA core.
RTC_calendar.vhd	Provides time and date calendar functionality. It handles the power-down and power-up context switching activities.
LCD_DIS.vhd	LCD driver to display time and date information
XTAL_RTC.vhd	Crystal oscillator block generated by SmartGen
NVM_analog.vhd	NVM analog system to provide the configuration settings for the analog RTC system
Analog_RTC.vhd	Analog RTC system

www.actel.com

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.

Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Phone +44 (0) 1276 401 450
Fax +44 (0) 1276 401 490

Actel Japan
www.jp.actel.com

EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong
www.actel.com.cn

Suite 2114, Two Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488

51900139.0/6.06