XILINX®

XAPP347 (v1.0) May 16, 2001

Decrease Processor Power Consumption using a CoolRunner CPLD

Summary

This application note describes system design techniques using a low power CoolRunner™ CPLD to reduce overall system power consumption. Utilizing a CoolRunner CPLD to off load operations from the system microprocessor keeps the processor in a power saving mode longer and contributes to significant power savings.

Introduction

One of the most critical factors in designing handheld, portable electronics today is reducing system power consumption. With increased consumer expectations, portable devices require longer battery life and higher performance. Even power reductions on the order of 10mW are crucial to portable system designers and manufacturers.

Several design techniques are used by designers today to significantly reduce overall system power consumption, such as:

- Reducing operating voltage
- Optimizing system and CPU clock frequency
- Eliminating spikes of large current consumption during the power up sequence
- Efficiently managing system battery operation
- Efficiently managing operating mode of system devices
- Minimizing bus activity
- Reducing bus capacitance
- Reducing switching noise

These are just a few examples of design techniques for reducing the power consumption in any end application.

One of the most important power saving techniques mentioned in this list is the ability to manage the operating mode of devices in the system. Many manufacturers today offer devices with power saving modes that temporarily suspend the device from its normal operation. These devices have the option to power down or transition to a non-functioning state if the device is not active for a specific amount of time. This feature is available on many of today's microprocessors and microcontrollers. By taking advantage and managing the operating mode of large power consumers on a PCB, such as the processor, the overall power consumption of the system can be reduced significantly.

Reducing power consumption not only involves correct management of the operating mode of a device, but designing a system to take advantage of the modes a device can operate within. Off loading operations of the microprocessor allows it to stay in its low-power state for a longer amount of time. One way to reduce system power is to allow a low-power device, such as a CoolRunner CPLD, to manage these off loaded operations. This paper will describe this possibility along with types of operations that allow a processor to remain in low power state longer, thereby reducing system power consumption.

Microprocessor Operating Modes

In some portable applications, the CPU can consume 30% of the overall system power. Figure 1 illustrates the typical power consumption of system components in a Web Pad application.

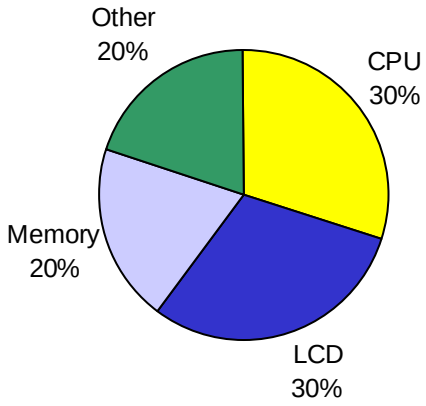

Figure 1: **Web Pad Power Consumption**

Microprocessor power consumption can range from 720 µW to 1W during normal operation. Microprocessor operating modes vary by part and manufacturer and include modes such as Normal, Run, Sleep, Suspend, Standby, Stop, and Idle operation. Operating modes can vary in power consumption as much as 230 mW between states. Normal operation of some low power microprocessors can be as little as 250 mW.

Example

> **NOTE:** The microprocessor reference provided is an example to illustrate the power consumption in different operating modes. Since no standard method exists to determine power consumption, the data provided in this document is based on data provided by the manufacturer and is not guaranteed. Please see "References" on page 9 for more information.

To illustrate the difference in power consumption of operating modes in a microprocessor, an example is provided. Figure 2 illustrates the power consumption of the Intel StrongARM SA-1110 microprocessor operating modes. The power dissipation numbers shown in Figure 2 are determined by operating at 206 MHz with a nominal external voltage supply of 3.3V and internal voltage supply of 1.8V.

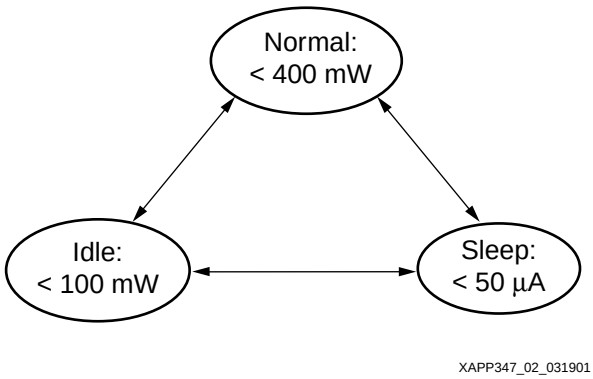

Figure 2: **Intel StrongARM SA-1110 Power Consumption**

Operating modes of the StrongARM processor include Normal, Idle, and Sleep. In Normal operation, the CPU is full-on, with the device fully powered and receiving active clocks. In Idle mode, even though power is applied to the CPU and other components, all clocks to the CPU

are stopped, with only clocks to peripheral devices active. In Sleep mode, power to the CPU and other peripheral components is disabled. Sleep mode disables all functions except the real-time clock, interrupt controller, power manager, and general purpose I/O.

Operating Mode Control

Microprocessors with power saving modes have an on-board power management controller. Operating modes allow the operating system or software application to temporarily suspend the CPU. The microprocessor executes a series of instructions to be placed into a power saving state. Once in a power down mode, several components of the microprocessor can still respond to system interrupts.

For example, the Idle mode of the StrongARM SA-1110 processor saves significant power, but certain modules remain powered, such as the LCD, memory and I/O controllers. Even though the clock to the CPU is stopped, peripheral modules are still active. The Idle mode can still consume a significant amount of power, on the order of 100 mW. By placing the processor into the Sleep mode, only active modules are powered to respond to interrupts and wake up signal requests. Sleep mode consumes even less power than Idle mode; current consumption can be less than 100 μA.

For a microprocessor to return to normal operation from a power down mode, an event must occur. The following events can wake up the processor, but vary based on manufacturer, part, and current operating mode:

- Hardware reset
- System interrupt
- General purpose I/O interrupt
- Real-time clock interrupt
- OS timer interrupt
- Peripheral interrupt
- External wake-up signal

Upon recognition of an enabled wake up event, the microprocessor will begin a series of steps to wake up from a power down state. Figure 3 illustrates the general flow for a processor waking up from a power down mode.

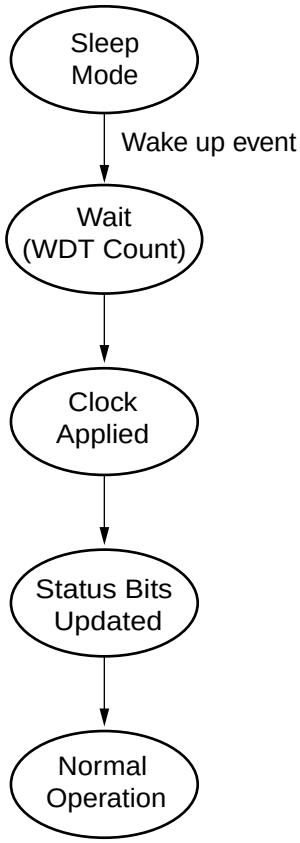

Figure 3: **Wake up Sequence**

CPLD Design

Operating modes are utilized when the microprocessor is idle for a specific amount of time. When a microprocessor receives an enabled interrupt, the processor will respond to the interrupt request. When the processor is responding to the interrupt, it will operate in its run or normal mode. **Reducing the number of interrupts to the processor will increase the time the processor is in a power saving state.** Ideally, if the microprocessor does not have any instructions to execute, it will remain in a power saving mode forever. Inserting an external device to respond and handle system interrupts can reduce the operations required of the processor. By allowing the microprocessor to stay in its power down mode as long as possible, significant power savings can be realized.

Utilizing a low power programmable logic device to supplement the microprocessor will save system power and increase system battery life. CoolRunner CPLDs simultaneously deliver high performance and low power consumption. Standby current of a CoolRunner CPLD is less than 100 µA. Figure 4 illustrates using a reprogrammable CPLD to interface to incoming

system interrupts. Utilizing an external data acquisition device to off load interrupt requests required of the microprocessor will reduce overall system power.

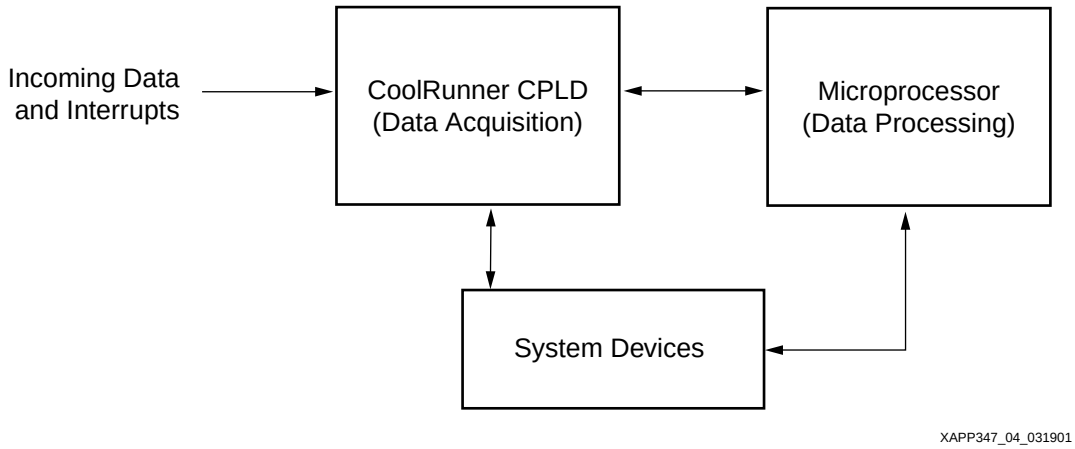

Figure 4: **System Block Diagram**

System Interrupts

Depending on the end application for the processor, a variety of external devices may interrupt the processor. These interrupts include both data acquisition and data processing requests. By separating data processing interrupts to the microprocessor, data acquisition interrupts can now be serviced by the external CPLD. Utilizing a CPLD to handle data acquisition interrupts will off load interrupt requests to the microprocessor and save power.

Catagorization of the type of data acquisition interrupts to the CPLD will depend on the end application. Peripheral devices or incoming data demanding a response to incoming data can be classified as data acquisition interrupt requests. Data acquisition interrupts include:

- Memory access interrupts
- Communication interfaces such as I2C, UART, SPI, or ISA
- General Purpose I/O interrupts
- LCD interface interrupts

This is not a complete list of interrupts that can be processed by the CPLD, but provides a starting point for the system design.

Operational Flow

Figure 5 illustrates the main operational flow for the design of the CoolRunner CPLD. Once a valid external interrupt is recognized by the CPLD, it will determine if it contains the functionality to process the interrupt. Once the CPLD has processed the interrupt, it can assert an interrupt to the processor for any data processing requests needed. If the CPLD is unable to process the

interrupt, the interrupt is passed to the processor. The CPLD also monitors the operating state of the processor.

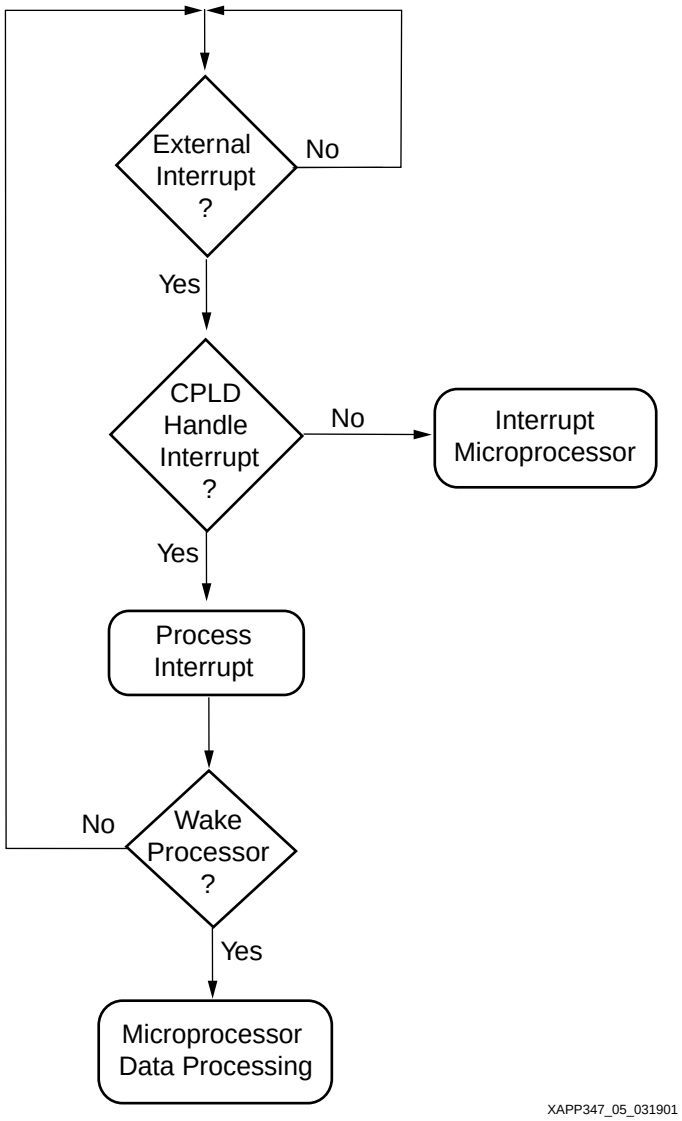

Figure 5: **CPLD Flow Diagram**

Functionality

The CoolRunner CPLD design consists of an interrupt interface and controller to handle interrupt requests, the functionality to process the interrupt and a processor interface. The main functions of the CoolRunner CPLD are described in more detail below and separated as follows:

- Interrupt interface for system devices
- Interrupt controller
- Interface with peripheral devices for interrupt processing
- Microprocessor interrupt interface
- Microprocessor operating mode interface

Interrupt Interface

The interrupt interface of the CPLD receives all external device interrupt requests previously recognized by the microprocessor. The interrupt interface determines if the CPLD is capable of

processing the interrupt request. The CPLD handles data acquisition interrupts that request data receiving and storage capabilities. If the CPLD is unable to process the interrupt, the interrupt is passed to the microprocessor.

The CPLD interrupt interface provides the masking capability for all interrupt sources and the ability to determine the interrupt source. Programmable logic provides flexibility to change the trigger mode, which includes a high or low level and falling or rising edge sensitivity. The CPLD interrupt control registers are similar to the registers in the microprocessor.

Interrupt Controller

The CPLD interrupt controller emulates the functionality that exists in the system microprocessor. The interrupt controller interprets from which device the data acquisition interrupt was received and initiates the processing of the interrupt. The CPLD processes the data acquisition interrupt request that would have otherwise interrupted the microprocessor.

The interrupt controller initiates the action to process the request. An example of this is an application where the CPLD is receiving data from a remote device. The device is requesting to write the data being sent into memory. The CPLD interrupt controller recognizes a valid interrupt and initiates the memory interface to interpret the data.

Peripheral Device Interfaces

The CPLD provides the interface to system devices that are needed in processing interrupt requests. Device interfaces that are needed are dependent on the end application. When an external device interrupts the CPLD to read or write data into a memory component, that particular memory interface is needed in the CPLD design. The types of interfaces needed can range from memories to LCD interfaces to communication interfaces such as PCI, UART, SPI and ISA.

Microprocessor Interrupt Interface

The CPLD, like any external device requesting services of the processor, has the capability to interrupt the microprocessor. The CPLD must be able to interrupt the microprocessor once a data acquisition operation is complete. The designer has the option to set the priority level of interrupt requests from the CPLD and whether or not interrupts received from the CPLD will wake the processor from a power down state.

Microprocessor Operating Mode Interface

Depending on the system microprocessor, the CPLD will be able to recognize the operation state of the processor. Some microprocessors provide external pins that represent the current operating mode. Depending on the CPLD and microprocessor design, the CPLD could recognize the current operating state of the processor and determine whether to assert an interrupt to the processor to execute a waiting interrupt. For example, if a low priority interrupt is received by the CPLD and the processor does not need to transition from its low power state, the CPLD can create a register indicating pending interrupts. Then when the processor wakes, the interrupt pending register can be read by the microprocessor.

Benefits

Figure 6 and 7 illustrate the power savings that may be realized in a typical battery operated device using a CoolRunner CPLD (Figure 7) versus a stand alone microprocessor design (Figure 6). The power requirements of the CoolRunner CPLD are minimal compared to the power savings realized by keeping the microprocessor in its low power modes for a longer amount of time. Standby current of the CoolRunner CPLD is on the order of 100uA. The operating power consumption depends on the application and clock frequency. For a 64-macrocell CPLD fully populated with 16-bit counters and a 50MHz clock, I_{CC} is around 10 mA.

Note that the actual power savings realized will depend on the system design including the type of microprocessor and the CPLD design.

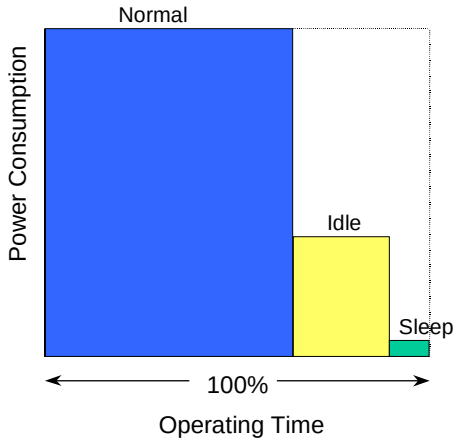

Figure 6: **Stand Alone Processor Power Consumption**

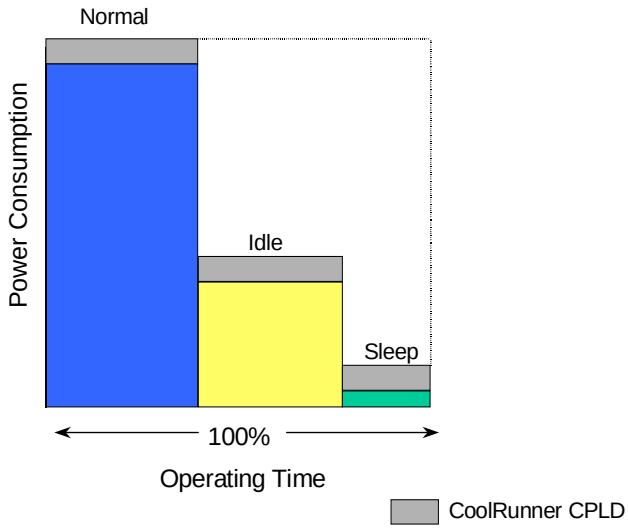

Figure 7: **Power Consumption of Processor and CoolRunner CPLD**

Along with power savings attained using a CoolRunner CPLD, interrupt response time is reduced. The peripheral device no longer has to wait the delay time for the microprocessor to wake from a power saving state. Additional design savings can be realized and include:

- Reducing the number of interruptions to the processor
- Reducing the number of processor wake up cycles over a length of time
- Reduction of clock frequency without impact on throughput
- Running processor at lower frequency for data processing operations
- Running CoolRunner CPLD at a higher frequency for data acquisition operations

Conclusion

Designing a power sensitive application involves not only using software for power management, but utilization of hardware design techniques. Designing a low-power CPLD to keep a microprocessor in a low power operating state longer can significantly reduce system power consumption. Xilinx CoolRunner CPLDs offer a flexible low power solution for any end application.

References

1. Intel StrongARM SA-1110 Microprocessor Developer's Manual. June 2000.
2. DragonBall Power Management. Motorola Semiconductor Application Note. Motorola, Inc. 1998.
3. Geode GX1 Processor Series Low Power Integrated x86 Solution. National Semiconductor. October 2000.
4. Turley, Jim. Microprocessors for Consumer Electronics, PDAs, and Communications. Microprocessor Report. September 1999.
5. EDN Access. 27th Annual Microprocessor/Controller Report. Sept. 2000.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/16/01	1.0	Initial Xilinx release.