A Low-Power 16-bit RISC Microcontroller

Li Guangcai, Lu Tiejun, and Zong Yu
Beijing Microelectronics Technology Institute, Beijing, China
Email: {hi_lgc, Lu_Tie_Jun}@163.com, zongyv@sohu.com

***Abstract*—Low-power, single-chip integrated systems are prevailing in sensor applications due to the increasing need of long battery life. The aim of this paper is to describe the implementation of a low-power, 16-bit RISC microcontroller that will act as the core of a sensor system. It is designed using 0.35-μm CMOS process from XFAB. The power consumption of the microcontroller is 0.36mW operating at 1MHz with a 1.8V power supply.**

***Index Terms*—low-power, 16-bit RISC, microcontroller, sensor system**

I. Introduction

Battery life is now a key product differentiator in the consumer world. Especially for applications like gas, heat, or water measuring, a low power MCU which can ensure much longer battery life and save the customers from the trouble of changing the battery often, is very desirable. A low power MCU designed for sensor system is presented in this paper. To work in a sensor system, it can capture analog signals, convert them to digital values, process the data and transmit them to a host system. Using a comparator and timers it is suitable for gas, heat, and water measuring applications. With a 16-bit RISC CPU core and system architecture specially designed for low energy consumption, the MCU can consume relatively small power while working.

At first, the paper describes the architecture, giving a brief introduction of the instruction set used by the microprocessor and the system structure. Then the paper shows some important cells and modules, operating modes changing of the microprocessor and explains why the MCU is low power in details. At last, the design and verifying procedure is shown. The MCU is designed in the Verilog Language, synthesized and optimized by Design Compiler and loaded into FPGA to accomplish the hardware verification.

II. Architecture

A. The Instruction Set

The RISC instruction set architecture contains 27 instructions supporting 4 addressing modes and byte or word arithmetic, shift, logical, and control-flow operations[1]. With a orthogonal architecture every instruction is usable for every addressing mode. Instructions in the custom ISA were carefully chosen to minimize decode complexity and power without sacrificing function. The processor has thirteen general purpose registers and three special purpose registers including program counter(PC), stack pointer(SR), and status register(SR).

Followings are some of power reducing features of the CPU. The total sixteen large 16-bit registers reduce fetch to the memory. Data can transfers directly from memory-to-memory without intermediate register holding. The CPU support byte addressing and instruction formats when word operation is not necessary.

B. Architectural Components

Traditionally, most power optimization methods have focused at gate-level and physical level optimizations.[2] However, major power reductions are only possible by addressing power at the RTL and system architecture levels. The microcontroller discussed in this paper has involved a lot of consideration about architecture power reduction. The microcontroller architecture has three main parts, CPU, peripherals and memory, as shown in Fig. 1. CPU includes Decode Unit, Execution Unit, Debug Unit and BUS Arbitration; Peripherals includes Comparator, Clock Module, Timers, and some other peripherals. Below gives brief description about the important ones.

- Decode Unit: This module contains the execution state machine and instruction state machine, the two machines interactively call each other to finish instruction fetch and decode task and to give execution unit the control signals.
- Execution Unit: This module contains ALU and register files and executes the current decoded instruction according to the execution state. This unit contains a lot high switching rate registers which have been clock-gated to reduce dynamic power.
- BUS Arbitration: This module performs a simple arbitration between the Decode Unit and Execution Unit for ROM, RAM, and peripheral memory access.
- Clock Module: With frequency dividers, this module can provides various frequencies for low power application, i.e. low frequency for energy conservation and time keeping, high frequency for fast reaction to events.
- Comparator and Timers: They make the MCU's configurations ideal for gas, heat, and water meters.

Manuscript received November 25, 2012; revised February 16, 2013.

doi: 10.12720/ijeee.1.1.48-51

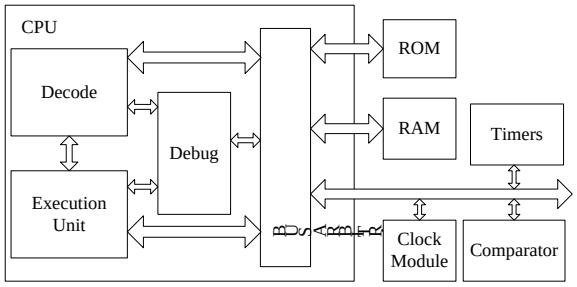

Figure 1. System architecture.

III. IMPORTANT CELLS OR MODULES

A. Address Decode Cell

The addresses visited by programs include program memory address, data memory address and peripherals address. The three of them share the same address space though they are physically separated. And the each space size is scalable to more flexibility to programming and expanding peripherals. For they share the same address space, they has the same address format. Then how to assign the address to physical address is a problem. The design uses a specially designed address decode cell, as shown in Fig. 2 to deal with it.

The current program address is stored in PC, address logic is an arithmetic which gets the needed address with address offset and data under the control of address control signal.[3]

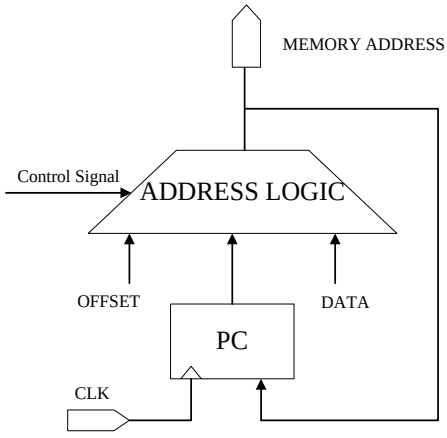

Figure 2. Address decode cell

B. Clock Module and Clock Gating

The clock module supports low system cost and low power consumption. Using two clock signals, the user can select the best balance of performance and power consumption. Fig. 3 shows the basic structure of the clock module. Low frequency clock source is from 32768-Hz watch crystals and high frequency clock source is from standard crystals in the 450-kHz to 8-MHz range. Different clock signals is chose under different work modes. For example, in the active mode, MCLK, i.e. the main clock, is enabled by the corresponding control signals to give clock signals for CPU operation. When the MCU changes from active mode to sleep mode or other low power mode, MCLK is shut down and ACLK, i.e. the auxiliary clock is active to provide clock signals for peripherals to work. With two dividers as shown in the figure, user can choose the most suitable clock frequency to balance the performance and power consumption, as we all that low frequency results in low power consumption.

Besides the two clock gate[4] used in the clock module, there are more than twenty clock gates inserted into the design by hand where clock toggle rates are very high. Comparing with inserting clock gates by EDA tools, this will reduce power more effectively.

Following is the Verilog code of the clock gate. After inserting the clock gates, the clock network has reduced about 70% power consumption.

```
module  clock_gate (
gclk,                  // Gated clock
clk,                   // Clock
enable,                // Clock enable
);
output        gclk;
input         clk;
input         enable;
// LATCH the enable signal
reg     enable_latch;
always @(clk or enable)
if (~clk)
enable_latch <= enable;
assign  gclk      =  (clk & enable_latch);
endmodule
```

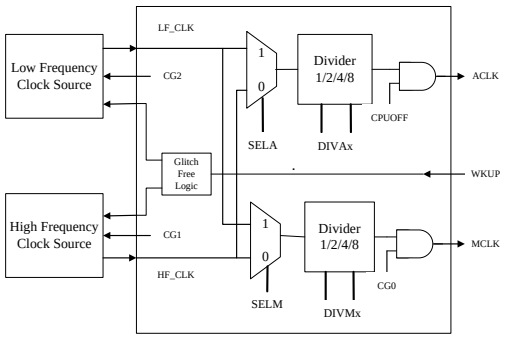

Figure 3. Clock module

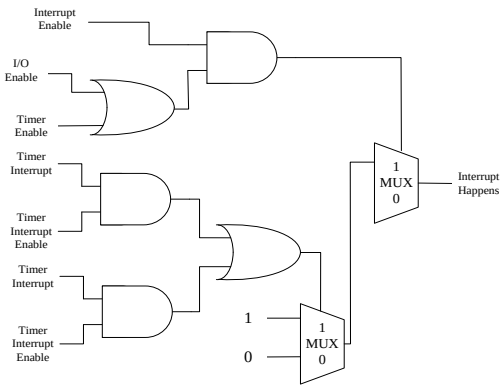

Figure 4. Interrupt judge circuit.

C. Interrupt System

The MCU has two interrupt sources, timer interrupt and I/O interrupt. Interrupt vectors are stored in the

beginning address space of ROM. The interrupt address stored in the ROM is actually the interrupt program address. It is only an JMP(jump unconditionally) instruction. The real interrupt program is stored in other parts of the ROM.

The MCU has two step to deal with interrupts, interrupt judge and interrupt execution[5]. The interrupt judge circuit is shown in Fig. 4.

The interrupt mechanism is placed outside CPU to give convenience to peripherals extension. There are two kinds of interrupts sources, timer interrupt and I/O interrupt. Each has its own enable bits. And only if the general interrupt enable signal is on and the corresponding enable bits of each interrupt source is on, interrupt in judged to happen and provided to CPU. At the same time, the interrupt vector address is provided to CPU too.

CPU response to the interrupt by the following steps:

- Wait until the currently executing instruction is completed.
- The current PC is pushed onto the stack.
- The SR is pushed onto the stack.
- The SR is cleared and the interrupt vector is handed to the PC. The interrupt program executes.
- When the interrupt program is completed, the RETI(return from interrupt) instruction end the interrupt and recover the original execution program.

IV. OPERATION MODE

A. Figures and Tables

Operating mode strategy is an essential problem in analyzing and designing power electronic circuits. Effective operating modes changing can result in much power saving[6]. For the application field of the MCU described in this paper is sensor system, it's calculation time is relative small. Most of time the MCU is waiting i.e. sleeping. And reducing the switching activity to a minimum according to the following average power estimation equation for CMOS is very good for low power consumption.

$$P_{av} = C_L V_{DD}^2 f \quad (1)$$

where is operating frequency[7].

For the two reasons above, the MCU is designed three operating mode. They are the active mode and two low power mode. The two low power mode are the real time clock mode and the sleep mode. In the active mode, the CPU and system main clock MCLK is working and this mode is usually used to do data computing. real time clock mode, the CPU of MCU is shut off.

The working mode transferring between each other is shown is Fig. 5. The two low-power modes are configured with the CPUOFF, OSCOFF bits in the SR(status register). When CPUOFF is positive CPU of the processor is shutdown. And a positive OSCOFF shut down the crystal oscillator thus there will be no clock source for the MCU. Including the CPUOFF, OSCOFF mode-control bits in the SR can make the present operating mode saved onto the stack when an interrupt happens. Program may return to the previous operating mode if the saved SR value is not changed during the interrupt. Program can return to a different operating mode by manipulating the saved SR value on the stack during the interrupt operation. The mode-control bits and the stack can be accessed with any instruction. After setting the mode-control bits, the selected operating mode takes effect immediately.

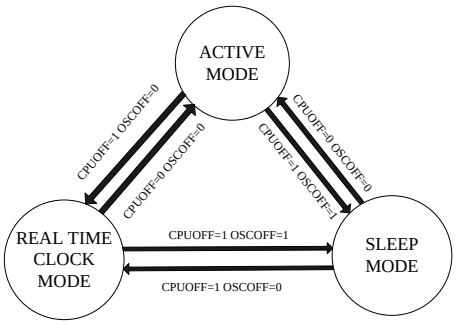

Figure 5. Operating modes

V. DESIGN FLOW AND TESTING

A. Design for Testing

Except the work done with synopsys tool DFT compiler, the MCU described in this paper is also manually designed to be fully scan friendly. Two scan control signals, scan_mode and scan_enalbe, are inserted originally into the design. During production, the Automatic Test Equipment controls the MCU through the scan_mode and scan_enable signals. The scan_mode port is always asserted during scan testing and is used to switch between functional and scan mode. To make scan function working, all internal resets, clock gates and clock muxes are connected with scan_enable or scan_mode signal. Take resets as an example, when in scan_mode, all internal resets of the MCU are connected to the general reset input port.

B. Design Flow and Result

The MCU described in this paper is implemented in the Verilog language on the RTL level. A test bench is created to do full verification and simulation of the MCU's function by the NC-Verilog simulator of Cadence. The simulation covered all the 27 instructions and 4 addressing modes. Stimulus are also created to verify the operating modes changing. Then it was synthesized and optimized by Design Compiler of Synopsys using a 0.35μm CMOS process from XFAB. The gate level netlist produced by Design Compiler passed function simulation and static timing analysis. Finally, power consumption is estimated through timing based power analysis with Synopsys' Prime Time-PX. Fig. 6 is the power report from Prime Time. It is produced under the condition of 1.8V and 1MHz. The MCU has a very competitive total power which is 0.36mW. At the same time the leakage power which representing the sleep mode power consumption is only 0.24μW.

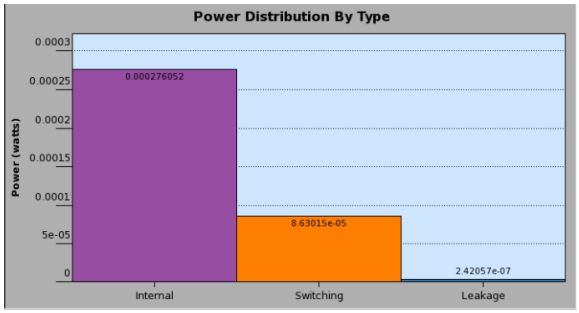

Figure 6. Power Report

VI. CONCLUSION

In this paper, a low-power microcontroller for a sensor system is presented. The microcontroller's instruction set and system architecture is specially designed to meet the need of low power sensor applications like gas, water meters. The clock module together with the interrupt system gives much flexibility for the users to choose the most suitable operating mode combination to gain the lowest power consumption.

REFERENCES

[1] R. K. S. Parihar, and S. Reddy, "A report on design of 16-bit RISC processor," *Birla Institute of Technology and Science*, May 2006.

[2] Mathur, A. Calypto, and Q. Wang, "Power reduction techniques and flows at RTL and system level," presented at VLSI Design, 2009 22nd International Conference on, January 5-9, 2009.

[3] P. K. Nizar and M. W. Williams, "Method and apparatus for memory address decode in memory subsystems supporting a large number of memory devices," U.S. Patent6252821, Jun 26, 2001.

[4] F. Emnett and M. Biegel, "Power reduction through RTL clock gating," *Synopsys Users Group*, San Jose, 2000.

[5] Y. Matsubara, "Hierarchical scheduling for integrating real-time applications with interrupt routines," presented at SoC Design Conference International, Nov. 22-24, 2009.

[6] Y. Kuroe, "A computer aided method for determining operating-mode boundaries in power electronic circuits," presented at Power Electronics Specialists Conference, 28th Annual IEEE, June 22-27, 1997.

[7] J. M. Rabaey, A. Chandrakasan, and B. Nikolie, "CMOS Invertor", in *Digital Integrated Circuits: A Design Perspective*, 2nd. Zhou Runde, Ed. 2004, ch.5, pp. 185.

Li Guangcai, born in Harbin, Heilongjiang, China, in June 6, 1986. He graduated from Harbin Institute Technology in 2010 with a bachelor degree in microelectronics. He is now studying in Beijing Microelectronics Institute to get a Master degree in microelectronics.

Lu Tiejun, research associate in Beijing Microelectronics Institute. He works mainly in the area of system architectural and low power design technology.

Zong Yv, research associate in Beijing Microelectronics Institute. He works mainly in the area of low power design technology. He also has a lot of experience in USB interface design field.