

Low Power RISC Core Using Clock Gating

B. Raj Kumar,
Associate Professor,
Dept., of ECE,

Yogananda Institute of Technology & Science, Tirupathi.

Dr. G. Chenchu krishnaiah
Professor and Head,
Dept., of ECE,

Gokula Krishna college of Engineering, Sullurpet.

ABSTRACT: Power has become an important aspect in the design of general purpose processors. The conventional RISC processors consume too much power as compared with other processors. The power reduction in these processors is done in the fabrication step itself. But this is a complex process. If we can implement the techniques for power reduction in front end process then we can easily design the low power processors without any complexity. In this paper we are proposing low power design in front end process. There are lot of techniques to reduce the power. Low power consumption helps to reduce the heat dissipation, lengthen battery life and increase device reliability.

This RISC processor is designed using pipelined architecture; through this we can improve the speed of the operation. In this we are using 5-stage pipelining. The 5 stages are Fetch, Decode, Execute, Memory and Write Back. During the design process we are including various low power techniques in architectural level also. The main power reducing method that has been explored in this architecture is clock gating. The proposed methods is more efficient than Back-end low power reduction techniques. Low power embedded processors are used in a wide variety of applications including cars, phones, digital cameras, printers, and other such devices. The reason for their wide use is that they are small; therefore, they do not take up much die area and are cost effective to fabricate.

Keywords: Architectural level power reduction, pipelining, clock gating.

I. INTRODUCTION

the destination registers. The bits from 10-6 represent the number of bits to be shifted. The last six bits, 5-0, represent the function code that represents the ALU function that is to be performed. If the opcode bits, 31-26, do not indicate an ALU function, then the function bits are ignored.

Low power has emerged as a principle theme in today's electronics industry. The power dissipation has become as important consideration as performance and area. This processor will follow the RISC architecture because it supports a predefined set of instructions. In this all the instructions have same length. RISC processors, first developed in the eighties. Embedded processors demand instruction set suitable for the specific applications, various and fast interrupt handling, low power consumption and so on. For this properties RISC is more efficient in comparison with micro programmed than CISC. This paper describes a 32-bit RISC processor designed for embedded and portable application. The features of this processor are it consumes less power and it operates in high speed. Other features of RISC are Uniform instruction format, identical general purpose registers, Simple addressing modes, Few data types in hardware.

II. INSTRUCTION SET

There are three basic types of instructions supported by this processor. These are the Register type, Jump type, and the Immediate type. The instruction format is shown in the below figure.

Register type: In this format bits 31-26 represents the opcode. Bits 25-21 represent the address of the first source register. Bits 20-16 give the address of the second source register. Bits 15-11 represents the address of

Field size	5 bits	5 bits	5 bits	5 bits	5 bits	6 bits
R-type	Opcode	Rs	Rt	Rd	shift	Function
I-type	Opcode	Rs	Rt	Immediate value (16 bits)		
B-type	Opcode	Branch offset (26 bits)				

Figure 1. Instruction format

Immediate type: As with the Register Type instruction, bits 31-26 represents the opcode. Bits 25-21 represent the address of the first source register. Bits 20-16 give the address of the second source register, and Bits 15-0 of this instruction type represent an 16-bit immediate value given in 2's complement form. When the opcode represents a unary operation, the value in this immediate field is used as the operand.

Jump type: Bits 31-26 of this instruction format represents the type of branch operation to be performed. The remaining 26 bits, 25-0, represent the branch offset in 2's complement format. This number is added to the value of the PC to obtain the branch target address.

The most significant feature of instruction set is that all of instruction may be executed conditionally according to the condition flags. A condition field within all instructions is compared with the condition flag in program status register. And the result of the comparison determines whether

The instruction whether the instruction can be executed or not. This reduces the need for forward branches and allows very dense in-line code to be written. In addition, shift operation is performed in serial with ALU in all data path cycles. This enables data processing instructions to execute

various arithmetic operations using shift operation in one clock cycle. Block data transfer instructions move all 16 available register to/from memory. The multiplication instruction uses Booth's algorithm to perform integer multiplication. They may be useful for digital signal processing.

IV. FUNCTIONAL BLOCK DIAGRAM

The architecture consists of five stage pipelining. Instruction fetch, Instruction decode, Execute, Memory and Write Back. Also, we added a Data Forward and Hazard detection unit to maintain proper data flow through the pipeline stages.

The architecture of the RISC processor is shown in the below figure.

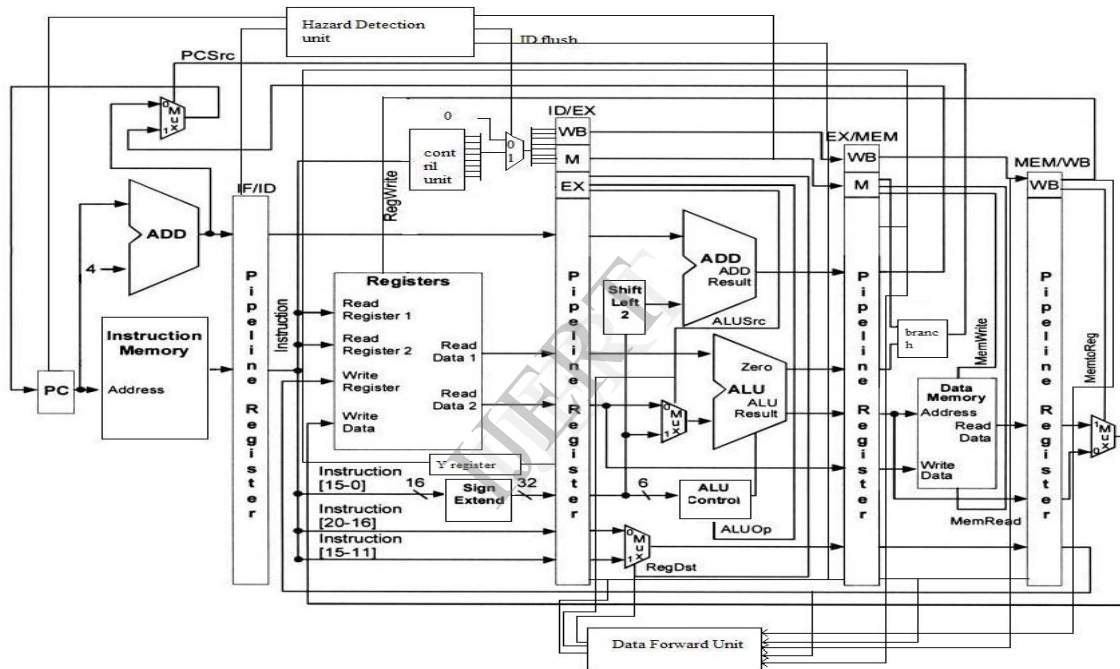


Figure 2. RISC Architecture

A. Instruction Fetch:

This stage consists of Program counter, Instruction Memory, and the Branch Decode unit.

Program Counter: the program counter (PC) contains the address, of the instruction that will be fetched from the Instruction memory during the next clock cycle. Normally the PC is incremented by one during each clock cycle unless a branch instruction is encountered, the PC is incremented/decremented by the amount indicated by the branch offset. The PC Write input of the PC serves as an enable signal. When PC Write signal is high, the contents of the PC are incremented during the next clock cycle, and when it is low, the contents of the PC remain unchanged.

Instruction Memory: The Instruction Memory contains the instructions that are executed by the processor. The input to

this unit is a 32-bit address from the program counter and the output is 32-bit instruction word.

Branch Decide Unit: The Branch Decide Unit is responsible for determining whether a branch is to take place or not based on the 2-bit Branch signal from the control Unit and the Zero flag from the Arithmetic and Logic Unit (ALU). The output of this unit is a 1-bit value which is high when a branch is to take place, and otherwise it is low. This output controls a multiplexer which in turn controls whether the PC gets incremented by one or by the amount indicated by the branch offset.

B. Instruction Decode:

This stage consists of the Control Unit, Register File, Y-Register, and the Sign Extend Unit.

Control Unit: The control unit generates all the control signals needed to control the coordination among the entire component of the processor. The input to this unit is the 4-

bit opcode field of the instruction word. This unit generates signals that control all the read and write operation of the register file, Y-Register and the Data memory. It is also responsible for generating signals that decide when to use the multiplier and when to use the ALU, and it also generates appropriate branch flags that are used by the Branch Decide unit.

Register file: This is a two port register file which can perform two simultaneous read and one write operation. It contains thirty two 32-bit general purpose registers. The registers are named R0 through R31. R0 is a special register which always contains the value zero and any write request to this register is always ignored. When the Reg_Write signal is high, a write operation is performed to the register.

Y-Register: The Y-Register is a special 32-bit register that is used to store the upper 32 bits (bits 32-63) of the result generated by the multiplier. When the Y_Write signal is high new value is written to this register, otherwise the currently stored value is outputted.

Sign Extend Unit: The input to this unit is an 16-bit immediate value provided by all the immediate type instructions. This unit sign extends the 16-bit value to a 32-bit value signed value.

C. Execute: This stage consists of the Branch Adder, Multiplier, Arithmetic Logic Unit (ALU), and the ALU Control Unit.

Branch Adder: The branch adder adds the 26-bit signed branch offset with the current value of the PC to calculate the branch target. The 26-bit offset is provided by the branch instruction. The output of this unit goes to the PC control multiplexer which updates the PC with this value only when a branch is to be taken.

Multiplier: The high-level block diagram of the multiplier is shown below. It consists of four distinct components. They are the Booth Encoder, Partial Product Generator, Carry Save Adder, and the Carry Look ahead Adder. This architecture employs two main techniques to increase the speed of the multiplication process. First technique is to reduce the number of partial products and the second is to increase the speed at which the partial products are added.

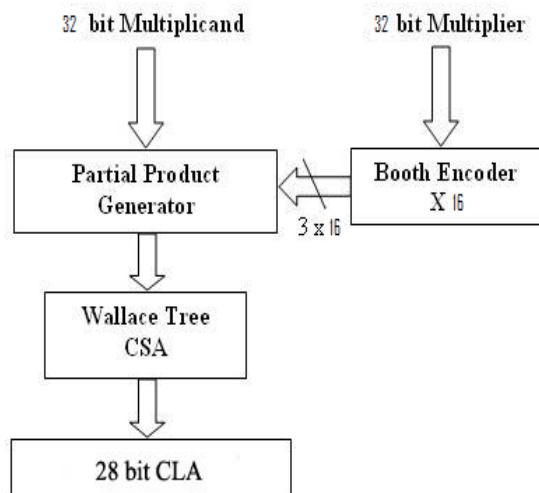


Figure 3. Multiplier Unit

Booth Encoder: This module encodes the 32-bit multiplier using radix 4 Booth's algorithm. Radix 4 encoding reduces the total number of multiplier digits by a factor of two, which means in this case the number of multiplier digits will reduce from 32 to 16. There are 32 input bits, there will be a total of 16 Booth encoder modules in the overall multiplier architecture.

Partial Product Generator (PPG): The output from the Booth encoder is used in this module to generate the partial products. Since there are 16 Booth encoders there will be a total of 16 partial products. The multiplication by two is implemented by shifting the multiplicand left one bit and the negation is implemented by taking the two's complement of the multiplicand.

Wallace Tree: This module is responsible for adding the partial products that were generated in the PPG module. This module uses 3 to 2 carry save adders (CSA) to implement the Wallace Tree. The individual CSAs are nothing more than full adders with the exception that the carry ins and the carry-outs are handled in a special way. Each column of numbers in the partial product is added using this method. The carry-outs generated in each stage of addition are transferred to the Wallace Tree of the column of bits of partial products on the left and the carry-ins comes from the column to the right.

Carry Look ahead Adder (CLA): This unit is used to add the final sum and carry vectors generated by the Wallace Trees for each column of bits from the partial products. Only a 60-bit CLA is needed, instead of a full 64 bits, because some of the bits of the final result are already available from the Wallace Trees.

Arithmetic Logic Unit (ALU): The ALU is responsible for all arithmetic and logic operations that take place within the processor. These operations can have one operand or two, with these values coming from either the register file or from the immediate value from the instruction directly. The operations supported by the ALU include add, subtract, compare, and, or, not, xor, rotate, logical shift, and arithmetic shift. The output of the ALU goes either to the data memory or through a multiplexer back to the register file. All operations are done according to the control signal coming from ALU control unit.

ALU Control Unit: This unit is responsible for providing signals to the ALU that indicates the operation that the ALU will perform. The input to this unit is the 6-bit opcode and the 6-bit function field of the instruction word. It uses these bits to decide the correct ALU operation for the current instruction cycle. This unit also provides another set of output that is used to gate the signals to the parts of the ALU that it will not be using for the current operation.

D. Memory:

This stage consists of the Data Memory module.

Data Memory: This module supports up to 64 words of 32-bit data words. The Load and Store instructions are used to access this module. When Mem_Write signal is high then write operation is take place otherwise a read operation is performed.

E. Write Back: This stage consists of some control circuitry that forwards the appropriate data, generated by the ALU/MAC or read from the Data Memory, to the register files to be written into the designated register.

F. Data Forward Unit: This unit is responsible for maintaining proper data flow to the ALU and the Multiplier. The primary function of this unit is to compare the destination register address of the data waiting in the Memory and Write Back pipeline registers to be written back to the register file with the current data needed by the ALU or the Multiplier and forward the most up-to-date data to these units.

G. Hazard Detection Unit: This unit detects conditions under which data forwarding is not possible and stalls the pipeline for one or two clock cycles in order to make sure that instructions are executed with the correct data set. When it detects that a stall is necessary, it disables any write operation in the instruction decode pipeline registers, stops the PC from incrementing, and clears all the control signals generated by the control unit. By taking these steps it can delay the execution of any instruction by one clock cycle. It can do this as many times as necessary to ensure proper execution of instructions.

IV. LOW POWER TECHNIQUES

The power reduction technique used in this is Clock gating. Clock gating is a method where the clock signal is prevented from reaching the various modules of the processor. The absence of the clock signal prevents any register and/or flip-flop from changing their value. As a result of this, the input to any combinational logic circuit remains unchanged, and thus no switching activity takes place in those circuits. Since, in CMOS circuits, most of the power dissipation results from switching activity, clock gating greatly reduces the overall power consumption.

V. HARDWARE IMPLEMENTATION

This processor is described and verified by using Verilog. The design is simulated using the MODELSIM simulator. This design is implemented using Xilinx Spartan synthesizer. The assembly codes are made and compiled to generate the machine codes. These instruction codes are used to verify the operations of each functional unit. The logic and timing verifications are performed. The implementation is done on Xilinx spartan FPGA and verified the operations of the processor. The power dissipation will reduce in this design.

VI. RESULTS

The simulation and implementation results are shown below

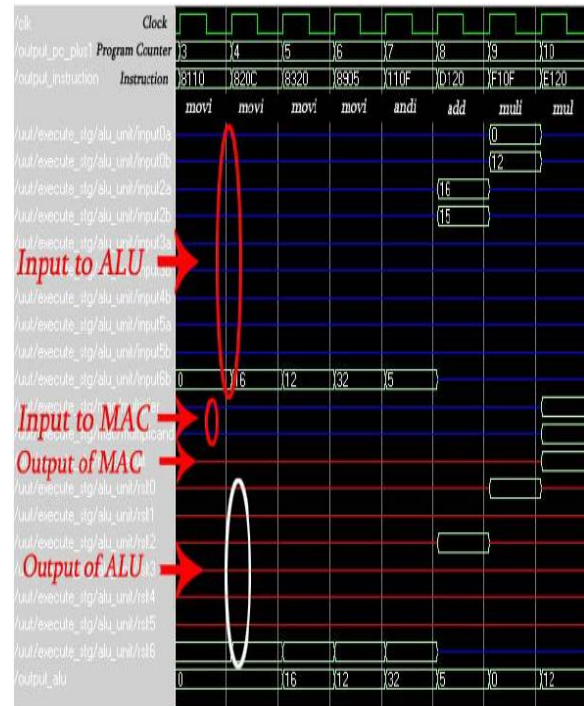


Figure 4. Simulation result

VII. CONCLUSION

The RISC processor for embedded and portable application has been designed with 5-stage pipeline and verified it. This processor was designed with high speed and it is performed in low power. We can easily reduce the power dissipation in front end process as compared with the back end process.

VIII. REFERENCE

- [1] Stephen B. Furber, "VLSI RISC Architecture and Organization", Marcel Dekker, Inc, 1989
- [2] Alex van Someren, Carol Atack, "The ARM RISC Chip, A Programmers Guide", 1993.
- [3] Y. Takahashi, Y. Fukuta, T. Sekine, and M. Yokoyama, .2PADCL: Two phase drive adiabatic dynamic CMOS logic., in *Proc. IEEE Asia-Paci_c Conf. Circuits and Systems*, Singapore, Dec. 4.7, 2006, pp. 1486.1489.
- [4] Y. Takahashi, T. Sekine, and M. Yokoyama, .VLSI implementation of a 4_4-bit multiplier in a two phase drive adiabatic dynamic CMOS logic., *IEICE Trans. Electron.*, vol. E90-C, no. 10, Oct. 2007, in press.
- [5] Qiao Ying-xu, Design of Wireless Sensor Networks Node Based On TinyOS Operating System. The 3th International Conference on Computer Science and Education [C] 2008.7 1201-1204.
- [6] Jilin Li, "Status and Development Trend of Coal Mine Safety Monitoring System", *Journal, Coal Technology*, Harbin, 2008(11), pp. 4- 5.
- [7] D. Koenig, M. S. Chiaramonte, A. Balbinot "Wireless

Network for Measurement of Whole-Body Vibration”, J. Sensors, vol. 8, pp. 3067-81, 2008.

[8]D. Egan, “The Emergence of ZigBee in building automation and industrial controls”, J. IEE Computing & Control Engineering, vol. 16(2), pp. 14-19, 2005.

[9]Hu Huo. Kjl3 Mine Monitoring System Software Design and Implementation. Computer Development & Applications,2008,14, (5).

[10] Y. Takahashi, T. Sekine, and M. Yokoyama, .VLSI implementation of a 4_4-bit multiplier in a two phase drive adiabatic dynamic CMOS logic., *IEICE Trans. Electron.*, vol. E90-C, no. 10, Oct. 2009.

AUTHORS PROFILE:



B. Raj Kumar,
Associate Professor,
Dept., of ECE,

Yogananda Institute of Technology & Science, Tirupathi.



Dr.G.Chenchu krishnaiah
Professor and Head,
Dept., of ECE,

Gokula Krishna college of Engineering, Sullurpet.

IJERT