RESEARCH ARTICLE OPEN ACCESS

16-Bit RISC Processor Design for Convolution Applications

Ramana K, Mrs. G. Sree Lakshmi and Dr. M. Ch. P. Jagdissh
Department of ECE, Malla Reddy Engg College-Hyderabad

Abstract
We propose a 16-bit non-pipelined RISC processor, which is used for signal processing applications. The processor consists of the blocks, namely, program counter, clock Control unit, ALU, IDU and registers. Advantageous architectural modifications have been made in the incrementer circuit used in program counter and carry select adder unit of the ALU in the RISC CPU core. Furthermore, a high speed and low power modified Wallace tree multiplier has been designed and introduced in the design of ALU. The RISC processor has been designed for executing 27-instruction set.
Convolution is an important signal processing application which is used in filter design. Many algorithms have been proposed in order to achieve an optimized performance of the filters by optimizing the convolution design. Modified Winograd algorithm is notable among them. They require 4 multiplication operations only, when compared to 6 for a 3*2 normal convolution methodology with an additional rise in the number of adders to 9 from 4.
Keywords: RISC, convolution, Wallace tree multiplier

I. INTRODUCTION

The trend in the recent past shows the RISC processors clearly outsmarting the earlier CISC processor architectures. The reasons have been the advantages, such as its simple, flexible and fixed instruction format and hardwired control logic, which paves for higher clock speed, by eliminating the need for microprogramming. The combined advantages of high speed, low power, area efficient and operation-specific design possibilities have made the RISC processor ubiquitous.

The main feature of the RISC processor is its ability to support single cycle operation, meaning that the instruction is fetched from the instruction memory at the maximum speed of the memory. RISC processors in general, are designed to achieve this by pipelining, where there is a possibility of stalling of clock cycles due to wrong instruction fetch when Jump type instructions are encountered. This reduces the efficiency of the processors. This paper describes a RISC architecture in which, single cycle operation is obtained without using a pipelined design. It averts possible stalling of clock cycles in effect.

The development of CMOS technology provides very high density and high performance integrated circuits. The performance provided by the existing devices has created a never-ending greed for increasingly better performing devices. This predicts the use of a whole RISC processor as a basic device by the year 2020. However, as the density of IC increases, the power consumption becomes a major threatening issue along with the complexity of the circuits. Hence, it becomes necessary to implement less complex, low power processor designs.

In this work, we have designed and developed a 16-bit single cycle non-pipelined RISC processor. In order to improve the performance, modification on incrementer circuit and carry select adder circuit have been done and modified structure has been integrated into the design and the performance is validated. A multiplier structure has been developed and modified Winograd algorithm is executed in order to validate our claim.

II. INSTRUCTION SETS OF COMPUTER

A. RISC: The instruction set is the hardware "language" in which the software tells the processor what to do. Surprisingly, reducing the size of instruction set. Eliminating certain instructions based upon a careful quantitative analysis, and reducing these seldom used instructions to be emulated in software can lead to higher performance.

Most of the CISC systems are micro programmed; because of the flexibility that micro programming offers the designer. Different instructions usually have micro routines of different lengths. This means that each instruction will take a number of different cycles to execute. This contradicts the principle of a uniform, streamlined handling of all instructions. An exception to this rule can be made when each instruction has a one-to-one correspondence with a single microinstruction. That is, each micro routine consists of a single control word, and still let the designer benefit from the advantages of microprogramming. However, contemporary CAD tools allow the designer of hardwired control units almost as easy as micro programmed ones. This enables the single cycle rule to be enforced, while reducing transistor count.

In order to facilitate the implementation of most instruction as register-to register operations, a sufficient amount of CPU general purpose registers

has to be provided. A sufficiently large register set will permit temporary storage of intermediate results, needed as operands in subsequent operations, in the CPU register file. This, in turn, will reduce the number of memory accesses by reducing the number of load/store operations in the program, speeding up its run time. A minimal number of 32 general purpose CPU registers has been adopted, by most of the industrial RISC system designers.

B. CISC: In early days, computers had only a small number of instructions and used simple instruction sets, forced mainly by the need to minimize the hardware used to implement them. As digital hardware become cheaper, computer instructions tended to increase both in number and complexity. These computers also employ a variety of data types and a large number of addressing modes. A computer with a large number of instructions, are known as complex instruction set computer, abbreviated CISC. Major characteristics of CISC architecture are:

1. A large number of instructions – typically from 100 to 250 instructions
2. Some instructions that perform specialized tasks and are used infrequently
3. A large variety of addressing modes – typically from 5 to 20 different modes
4. Variable-length instruction formats
5. Instructions that manipulate operands in memory

III. DESIGN OF 16-BIT RISC PROCESSOR

In this project, we propose a 16-bit non-pipelined RISC processor, which is used for signal processing applications. The processor consists of the blocks, namely, program counter, clock control unit, ALU, IDU and registers.

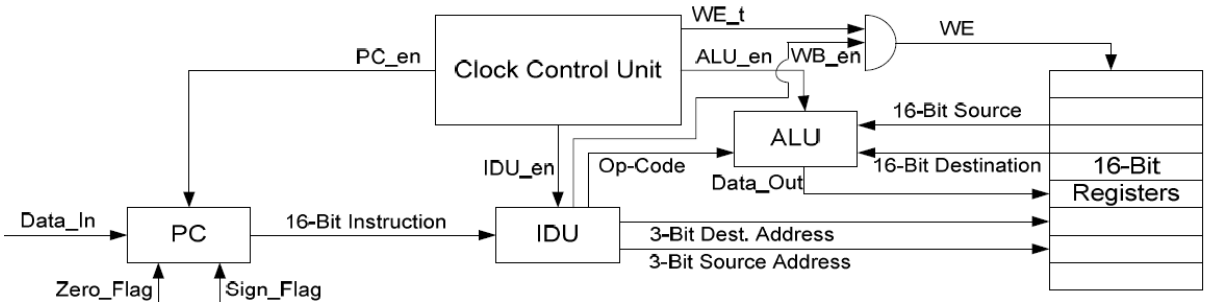

Fig.2 16-bit Non Pipelined Risc Processor

unit of the ALU in the RISC CPU core. Furthermore, a high speed and low power modified Wallace tree multiplier has been designed and introduced in the design of ALU.

A. Architecture

The architecture of the proposed RISC CPU is a uniform16-bit instruction format, single cycle non-pipelined processor. t has a load/store architecture, where the operations will only be performed on registers, and not on memory locations. It follows the classical von-Neumann architecture with just one common memory bus for both instructions and data. A total of 27 instructions are designed as a first step in the process of development of the processor. The instruction set consists of Logical, Immediate, Jump, Load, store and HALT type of instructions.

B. Logic Blocks:

The proposed RISC CPU consists of five blocks, namely, Arithmetic and Logical Unit (ALU), Program Counter (PC), Register file (REG), Instruction Decoder Unit (IDU) and Clock Control Unit (CCU). The data-path of the proposed CPU in Fig. 1 is explained as follows.

1) Program Counter: The Program Counter (PC) is a 16-bit latch that holds the memory address of location, from which the next machine language instruction will be fetched by the processor. The proposed PC is the largest sub-block and second to the control unit in complexity. It controls the flow of the instructions execution and it ensures the logical operation flow of the processor. It performs the two operations, namely, incrementing and loading. For most instructions, the PC is simply incremented in preparation for the following instruction or the following instruction nibbles. In this design, we employ a 6-bit pointer to indicate the instruction memory. It additionally uses a 6-bit pointer to point to the data memory, which will be used only when a Load/Store instruction is encountered for execution.

2) Arithmetic and Logic unit: The arithmetic and logic unit (ALU) performs arithmetic and logic operations. It also performs the bit operations such as rotate and shift by a defined number of bit positions. The proposed ALU contains three sub-modules, viz. arithmetic, logic and shift modules.

The arithmetic unit involves the execution of addition operations and generates Sign flag and Zero flag as per the result shown in the process. In order to reduce the complexity of the adder circuits used in the arithmetic unit of the RISC CPU, a very fast and low power carry select adder circuit has been introduced. The ALU also consists of a modified Wallace tree multiplier, which uses compressor

circuits to achieve low power and improved speed of operation. The multiplier is designed to execute in a single cycle.

The shift module is used for executing instructions such as rotation and shift operations. The shift module is mandatory for signal processing applications, which needs division by 2. This is achieved by a single right shift operation. The logic unit is used to perform logical operations, such as, Ex-or, OR, and AND. The Data out of each ALU operation is written back into the corresponding destination register, along with the flags updated. In order to maintain simplicity of the design, the carry out of the ALU is not taken into consideration.

3) Register File: The register file consists of 8 general purpose registers of 16-bits capacity each. These register files are utilized during the execution of arithmetic and data-centric instructions. It is fully visible to the programmer. It can be addressed as both source and destination using a 3-bit identifier. The register addresses are of 3-bit length, with the range of 000 to 111. The load instruction is used to load the values into the registers and store instruction is used to retrieve the values back to the memory to obtain the processed outputs back from the processor. The Link register is used to hold the addresses of the corresponding memory locations.

4) Instruction Decoder Unit: Our instruction set is limited yet comprehensive. Since our data bus is only 5 bits wide, it was decided to keep the number of instructions supported within 32 for easier implementation. At present, only 27 instructions have been implemented. The rest have been reserved for porting digital processing applications into our processor. The decoder units decodes the instruction and gives out the 3-bit source and destination addresses respectively, depending on the op-code's operation and it also decides whether the writeback circuit has to be enabled or not.

In case of Load/Store instructions, the IDU updates the Link register. In case of Jump instructions, if the conditions are satisfied, the IDU updates the PC register with the new address from where the next instruction has to be retrieved rather than the normal incremented value. Figure 2(a) shows the instruction format followed by Logical instructions and Data transfer instructions, such as, MOV, AND, OR, XOR, ADD, SUB, SL (Shift Left), RL (Rotate Right), SR (Shift Right), RR (Right Rotate), SWAP and Multiply instructions. Fig. 2(b) shows the instruction format followed by Immediate instructions. This type has the data vested into the instructions, such as LHI (Load 8-bit value into the eight higher significant bits of the given register), LLI (Load 8-bit value into given register's 8 least significant bits), ANDI, ORI, XORI, ADDI, SUBI. Figure 2(c) depicts the format of Load instruction. The instruction format for Store instruction is given in Fig. 2(d). Fig. 2(e) depicts the instruction format for JUMP, JZ (Jump if Zero), JNZ (Jump if not Zero), JP (Jump if positive), JN (Jump if Negative) instructions. Fig. 2(f) shows the format for the HALT command.

5) Clock Control Unit: Efficient phase scheduling is required to optimize the throughput and the energy consumption of the processor. In this paper, we propose a clock control unit (CCU) which is tasked with efficient phase scheduling, to select the various blocks of the processor.

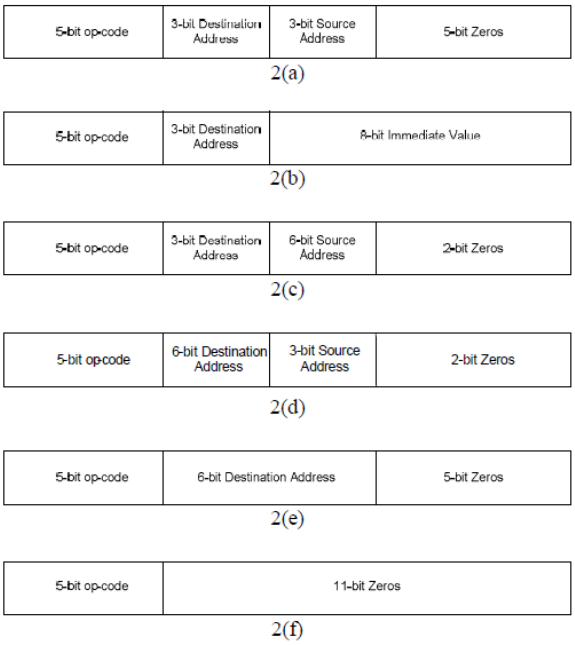

Fig. 2 (a) to (f) 16-Bit Instruction Format

IV. WINOGRAD CONVOLUTION ALGORITHM

To describe the functionality of the processor towards its use for signal processing applications, we have executed a 3*2 modified Winograd algorithm by implementing its pneumonic The application instructions are given below:

1. Load x0
2. Load x1
3. Load x2
4. Mov x2
5. Add x0, x2
6. Mov x1
7. Add (5), x1
8. Sub (5), x1
9. Mov Reg D to Reg E
10. Load H0
11. Load H1
12. Load H2
13. Load H3
14. Mul H0, X0
15. Mul H1, X1
16. Mul H2, X2
17. Mul H3, X3
18. SR (13) gives S0

19. SR (16) gives S3
20. Add (16), (17)
21. Sub (20), (15)
22. SR (21) gives S1
23. Add (15), (16)
24. Sub (14), (15)
25. SR (24) gives S2
26. HLT

The respective op-codes are initially stored in the instruction memory of the processor. The inputs are stored in the corresponding data memory, which should, in this case lie beyond the memory location 26. The maximum size of inputs x0, x1, x2 can each be of 6 bits width, while the values of h0 and h1 can each be of 3 bits width.

V. EXPERIMENTAL RESULTS

The RISC processor described above is designed using Verilog HDL and is simulated using Modelsim 9.2i. The proper functioning of the processor is validated. The simulation result shows that the processor is capable of implementing the given instruction in single clock cycle, thereby satisfying the basic requirements of the RISC processor. synthesis software were used to map the proposed processor on a target library. The target library includes the generic technology mapping information. The tool employed for mapping the Verilog-HDL components to the cell library is the Synopsys design compiler. It mapped the Verilog-HDL components to a SAED 90nm ASIC standard cell library. The power dissipation report produced by the design compiler is shown in Table I below.

Table 1: Power dissipation

Cell Internal Power	52.1345nW
Net Switching Power	9.127nW
Total Dynamic Power	60.235nW
Cell Leakage Power	327.2878µW
Total Power dissipation	327.3µW

VI. CONCLUSION

The design of a single cycle 16-Bit non-pipelined RISC processor for its application towards convolution application has been presented. Novel adder and multiplier structures have been employed in the RISC architecture. The processor has been designed for executing the instruction set comprising of 27 instructions in total. It is shown expandable up to 32 instructions, based on the user requirements. The processor design promises its use towards any signal processing applications.

REFERENCES

[1] Robert S. Plachno, VP of Audio "A True Single Cycle RISC Processor without Pipelining". ESS Design White Paper – RISC Embedded Controller.

[2] Youngjoon Shin, Chanho Lee, and Yong Moon, "A Low Power 16-Bit RISC Microprocessor Using ECRL Circuits", ETRI Journal, Volume 26, Number 6, December 2004.

[3] Yasuhiro Takahashi, Toshikazu Sekine, and Michio Yokoyama, "Design of a 16-bit Non-pipelined RISC CPU in a Two Phase Drive Adiabatic Dynamic CMOS Logic," International Journal of Computer and Electrical Engineering, Vol. 1, No. 1, April 2009 1793-8198.

[4] V. B. Saambhavi and V. S. Kanchana Bhaaskaran, A 16-Bit RISC Microprocessor Using DCPAL Circuits. International Journal of Advanced Engineering and Technology (IJAET), E-ISSN-0976-3945, Vol.II, Issue I, January-March 2011, pp. 154-162

[5] J.S. Denker, "A Review of Adiabatic Computing," IEEE Symp. Low Power Electronics, 1994, pp. 94-97.

[6] H. Mahmoodi-Meinnand, A. Afzali-Kusha, and M. Nourani, "Adiabatic Carry Look-Ahead Adder with Efficient Power Clock Generator," IEEE Proc., vol. 148, 2001, pp. 229-234.

Ramana k
Pursuing final year M.Tech with the specialization of digital systems &computer electronics in the department of ECE in Malla Reddy Engineering College-Hyderabad, India. He has completed his B.Tech in Warangal. His interests are in VLSI and Embedded Systems

Ganti Sree Lakshmi
Is working as an associate professor in the department of ECE in Malla Reddy Engg College-Hyderabad, India. She is having 9 years of teaching experience in VLSI and digital signal processing. She is pursuing Ph. D in the field of VLSI technology.

Dr. M. C.J Prasad
Is presently working as a head of the Department of ECE, MREC, Hyderabad, India. He is having 15 years of teaching experience. His areas of interests are communication systems, digital systems, image processing, digital signal processing, advanced DSP systems.