Abstract – Pipeline processor is a processor that applies to the single cycle architecture. It executes several instructions simultaneously, improving throughput significantly. It must add logic to handle dependencies between simultaneously executing instructions. It requires non-architectural pipeline registers. All the commercial high performance processor use pipeline today.

Keywords – Processors, RISC, SISC, Pipeline

I. INTRODUCTION
First of all, we should know “what is processor?” In general, processor tells your computer what to do and when to do it, it decides which tasks are more important and prioritizes them to your computer’s needs. Technically, it is the logic circuitry that responds to and processes the basic instructions that drives a computer. The term processor has generally replaced the term central processing unit (CPU). The processor in a personal computer or embedded in small devices is often called a microprocessor.

There are two types of computer architecture on the basis of which processor have been designed:
RISC (Reduced Instruction Set Computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more than million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation. The RISC concept has led to a more thoughtful design of the microprocessor.

A CISC (Complex Instruction Set Computer) is a computer where single instructions can execute several low-level operations (such as a load from memory, an arithmetic operation, and a memory store) and/or are capable of multi-step operations or addressing modes within single instructions.

RISC Versus CISC: A RISC architecture has to execute more instructions performs same function performed by the CISC architecture. To compensate for this drawback, RISC architectures must use the chip area saved, by not complex instructions decoder in providing a large number of CPU registers, additional execution unit, instruction caches. The use of these resources leads to the reduction of traffic between the processor and memory. On the other hand, CISC architecture with a richer and complex instruction, CISC architecture has rich set of instructions than its RISC counterpart. However, CISC architecture requires complex decoding scheme and hence is subject to logic delays.

A. Features of RISC Machine
1. Relatively few instructions.
2. Relatively few addressing modes.
3. Memory access limited to load and store instructions.
4. All operations are done within the registers of the CPU.
5. Fixed-length and easily-decoded instruction format.
7. Efficient instruction pipeline.

B. Advantages
1. They can be used effectively by optimizing compilers.
2. Because of the smaller chip area needed for instruction handling and sequential control in RISC processors, more space is available for processors registers and on-chip caches.
3. Higher performances results because off-chip data and instruction access are reduced.
C. Micro-Architectures

1) Single Cycle Architecture
It executes entire instruction in one cycle. It’s easy to explain and it has simple control unit. Because it completes operation in one cycle, it does not require non-architectural state.

2) Multi-Cycle Architecture
Multi-cycle architecture executes instruction in a series of shorter cycle. Simpler instruction takes fewer cycles than complicated ones. The multi-cycle architecture executes only one instruction at a time, but each instruction takes multiple clock cycle.

3) Pipeline Architecture
This applies pipelining to the single cycle architecture. It executes several instructions simultaneously, improving throughput significantly. Pipelining must add logic to handle dependencies between simultaneously executing instructions. It also requires non-architectural pipeline registers. The added logic is worthwhile; all commercial high-performance processor use pipeline today. The processor which is been design is divided into five pipeline stage. Because each stage has only one-fifth of the entire logic, the clock frequencies almost five times faster. Hence, the latency of each instruction is ideally unchanged, but the throughput is ideally five times better. This micro-architecture executes millions of instructions per second, so throughput is more important than the latency. The five stages are fetching, decode, and execute, memory and write-back.

D. Advantages of Pipeline Architecture
- The cycle time of processor is reduced, thus increasing instruction issue ratio in most cases.
- Some combinational circuits such as adders or multipliers can be made faster by adding more circuitry. If pipelining is used instead, it can save circuitry vs. more complex combinational circuits.

E. Disadvantages of Pipeline Architecture
- A non-pipeline processor executes only a single instruction at a time. This prevents branch delays and problem with serial instruction being executed concurrently. Consequently the design is simpler and cheaper to manufacture.
- The latency is slightly lower than in pipeline equivalent.
- The performance of pipeline processor is much harder to predict and may very more widely between different programs.

F. Classification of Computer Architectures:

Von Neumann Machines
Perhaps the most significant characteristic of von-Neumann computer architecture is the use of a single program counter (PC) to control the flow of executing programs. Program instructions are executed in the same order as they appear in the main memory. Branching to subroutines or other programs is allowed. However, a return to the calling routine is usually made available. Generally we can call a computer a von-Neumann machine if it satisfies the following requirements:

1. It has three basic units:
   a) A CPU: Central Processing Unit
   b) A Main memory
   c) An I/O unit
2. Its programs are stored in the main memory. A program can manipulate its data which can reside in the main memory as well.
3. It executes its programs sequentially and one instruction is executed at any given time.

Harvard Architecture
Harvard architecture is a class of von-Neumann computer organization. Whereas in conventional von-Neumann computers, the same set of buses (address and data) is used for both program instructions and data, see Figure 1. In Harvard architecture, two separate sets of buses are used for program instructions and data. In such architecture, program instructions and data appear to be accessed simultaneously, see Figure 2.
Non Von-Neumann Machines

Beyond the memory bottleneck, the performance of computer systems based on the von Neumann architecture is limited by this architecture’s “one instruction at a time” execution paradigm. Executing multiple instructions simultaneously using pipelining can improve performance by exploiting parallelism among instructions. However, performance is still limited by the decode bottleneck since only one instruction can be decoded for execution in each cycle. To allow more parallelism to be exploited, multiple operations must be simultaneously decoded for execution.

The sequence of instructions decoded and executed by the CPU is referred to as an instruction stream. Similarly, a data stream is the corresponding sequence of operands specified by those instructions. Using these definitions, Flynn proposed the following taxonomy for parallel computing systems:

- SIMD (Single Instruction stream, Multiple Data stream)
- MISD (Multiple Instruction stream, Single Data stream)
- MIMD (Multiple Instruction stream, Multiple Data stream)

G. Introduction of RISC Machines

The dominant architecture in the PC market, the Intel IA-32, belongs to the Complex Instruction Set Computer (CISC) design. The obvious reason for this classification is the “complex” nature of its Instruction Set Architecture (ISA). The motivation for designing such complex instruction sets is to provide an instruction set that closely supports the operations and data structures used by Higher-Level Languages (HLLs). However, the side effects of this design effort are far too serious to ignore.

The decision of CISC processor designers to provide a variety of addressing modes leads to variable-length instructions. For example, instruction length increases if an operand is in memory as opposed to in a register. This is because we have to specify the memory address as part of instruction encoding, which takes many more bits. This complicates instruction decoding and scheduling. The side effect of providing a wide range of instruction types is that the number of clocks required to execute instructions varies widely. This again leads to problems in instruction scheduling and pipelining. For these and other reasons, in the early 1980s designers started looking at simple ISAs. Because these ISAs tend to produce instruction sets with far fewer instructions, they coined the term Reduced Instruction Set Computer (RISC). Even though the main goal was not to reduce the number of instructions, but the complexity, the term has stuck.

However, there is not any exact definition for RISCs. A computer can qualify to be a RISC if it can meet most of the following characteristics:

- Instruction set is simple.
- Instructions are of a uniform length.
- Instruction set uses few instruction formats.
- Little overlapping of instruction functionality.
- Instruction set implements few addressing modes.
- Few instructions move data to and from the main memory.
- All operate instruction manipulate only data from the register file.
- Instruction set supports a limited number of data types.

H. Survey of RISC Architectures

RISC architecture has been found in various RISC such as in the desktop, servers and embedded RISC. Desktop and server RISC are digital ALPHA, IBM and Motorola POWERPC, MIPS-I etc. while embedded RISC are ARM (Advanced RISC Machine), ARM thumb, MIPS-16 etc.

<table>
<thead>
<tr>
<th>Year announced</th>
<th>ALPHA</th>
<th>MIPS-I</th>
<th>POWER-PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1986</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1993</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction bits</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Addressing modes</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Integer register</td>
<td>31GPR×64 bits</td>
<td>31GPR×32 bits</td>
<td>31GPR×32 bits</td>
</tr>
</tbody>
</table>
Due to the need for executing instructions the design of an embedded RISC Interlock Pipeline Stages) using VHDL. It also architecture based on Reduced Instruction Set Computer (RISC) CPU. In [3], authors presented the design of a RISC optimization techniques. Memory, logical unit, arithmetic unit and design is based on program counter, instruction fetch, decode, execute, pipeline control and memory. The reduction in the power is achieved using HDL modification techniques. The design is based on program counter, instruction memory, logical unit, arithmetic unit and optimization techniques. In [3], authors presented the design of a RISC (Reduced Instruction Set Computer) CPU architecture based on MIPS (Microprocessor Interlock Pipeline Stages) using VHDL. It also describes the instruction set, architecture and timing diagram of the processor. Floating point number to fixed number conversion is the main task while working on this numbers, this conversion has been achieved by using Float to Fixed number converter module. In [4], authors implemented a 5-stage pipelined 32-bit High performance MIPS based RISC Core. MIPS (Microprocessor without Interlocked Pipeline Stages) is a RISC (Reduced Instruction Set Computer) architecture. A RISC is a microprocessor that had been designed to perform a small set of instructions, with the aim of increasing the overall speed of the processor. MIPS have 5 stages of pipeline viz. Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Memory Access (MEM) and Write Back (WB) modules. The various modules being used are Instruction Memory, Data Memory, ALU, Registers etc.

In [5], authors proposed 5 stage pipelined architecture of 32 bit RISC Processor (MIPS) using VHDL. The simulations are done with ModelSim simulator. A Reduced Instruction Set computer is a microprocessor that had been designed to perform a small set of instructions, with the aim of increasing the overall speed of the processor. In paper [6], the author proposed a 16-bit non-pipelined RISC processor, which is used for signal processing applications. 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. It is expandable up to 32 instructions, based on the user requirements. The processor has been realized using Verilog HDL, simulated using Modelsim 6.2 and synthesized using Synopsys. This paper extended the utility of the processor towards convolution application, which is one of the most important signal processing application. The simulations depict the total dissipated power by the processor to be approximately 329.3 μW with the total area of 65012 nm². Paper [7] details the design of an embedded RISC controller used for mixed signal audio integrated circuits. This processor replaced an existing 8 bit CISC embedded processor and obtained a performance improvement of about 6x. This performance improvement was entirely due to architectural improvements using the same input clock rate and external ROM IP block. A design for a single cycle RISC processor has been discussed that does not use pipelining. This operation is

### Table 1: Characteristics of embedded RISC

<table>
<thead>
<tr>
<th></th>
<th>ARM</th>
<th>THUMB</th>
<th>MIPS-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date announced</td>
<td>1985</td>
<td>1995</td>
<td>1996</td>
</tr>
<tr>
<td>Instruction bits</td>
<td>32</td>
<td>16</td>
<td>16/32</td>
</tr>
<tr>
<td>Addressing modes</td>
<td>6</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Integer register</td>
<td>15GPR×32 bits</td>
<td>8GPR×32 bits</td>
<td>8GPR×32/64 bits</td>
</tr>
</tbody>
</table>

Table 2 shows that embedded RISC have 8 to 16 general purpose registers and the length of instruction is 16 to 32 bits while Table 1 shows that desktop and server RISC have 32 general purpose registers and the length of instruction is fixed i.e.32 bits. It has been also observed that desktop and server RISC require less no. of addressing modes as compare to the embedded RISC.

**II. LITERATURE REVIEW**

Carrying out literature review is very significant in any research project. It clearly establishes the need of the work and the background development. It generates related queries regarding improvements in the study already done and allows unsolved problems to emerge and thus clearly define all boundaries regarding the development of the research project. This section discuss the previous work done on RISC processors.

In [1], authors implemented the 32-bit RISC MIPS processor on Spartan-6 FPGA. The project involves simulation and synthesis of a processor. A Reduced Instruction Set compiler (RISC) is a microprocessor that had been designed to perform a small set of instructions, with the aim of increasing the overall speed of the processor. The idea of this project was to create a RISC MIPS processor as a building block in Verilog HDL.

In [2], authors presented the design and implementation of 32 bit MIPS processor. The architecture with pipelined control RISC core consists of fetch, decode, execute, pipeline control and memory. The reduction in the power is achieved using HDL modification techniques. The design is based on program counter, instruction memory, logical unit, arithmetic unit and optimization techniques.

In [3], authors presented the design of a RISC (Reduced Instruction Set Computer) CPU architecture based on MIPS (Microprocessor Interlock Pipeline Stages) using VHDL. It also describes the instruction set, architecture and timing diagram of the processor. Floating point number to fixed number conversion is the main task while working on this numbers, this conversion has been achieved by using Float to Fixed number converter module. In [4], authors implemented a 5-stage pipelined 32-bit High performance MIPS based RISC Core. MIPS (Microprocessor without Interlocked Pipeline Stages) is a RISC (Reduced Instruction Set Computer) architecture. A RISC is a microprocessor that had been designed to perform a small set of instructions, with the aim of increasing the overall speed of the processor. MIPS have 5 stages of pipeline viz. Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Memory Access (MEM) and Write Back (WB) modules. The various modules being used are Instruction Memory, Data Memory, ALU, Registers etc.

In [5], authors proposed 5 stage pipelined architecture of 32 bit RISC Processor (MIPS) using VHDL. The simulations are done with ModelSim simulator. A Reduced Instruction Set computer is a microprocessor that had been designed to perform a small set of instructions, with the aim of increasing the overall speed of the processor.

In paper [6], the author proposed a 16-bit non-pipelined RISC processor, which is used for signal processing applications. 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. It is expandable up to 32 instructions, based on the user requirements. The processor has been realized using Verilog HDL, simulated using Modelsim 6.2 and synthesized using Synopsys. This paper extended the utility of the processor towards convolution application, which is one of the most important signal processing application. The simulations depict the total dissipated power by the processor to be approximately 329.3 μW with the total area of 65012 nm².

Paper [7] details the design of an embedded RISC controller used for mixed signal audio integrated circuits. This processor replaced an existing 8 bit CISC embedded processor and obtained a performance improvement of about 6x. This performance improvement was entirely due to architectural improvements using the same input clock rate and external ROM IP block. A design for a single cycle RISC processor has been discussed that does not use pipelining. This operation is...
obtained by folding the processor execution into the memory cycle. The RISC uses a ripple through latch style of design as opposed to a synchronous flip-flop style design.

Paper [8] presented the design of a 16-bit RISC processor core. The quasi-adiabatic DCPAL circuit design style is employed in the designs. The major blocks constituting the processor core are the Arithmetic and Logical Unit (ALU), Program Counter, Adiabatic Register file and Control Unit housing the instruction decoders. The schematic designs are drawn using the schematic edit tool and the extracted net lists are used in the simulations. The design is validated by developing and testing the equivalent CMOS circuit counterparts for the processor core design. The 250nm Technology model library from TSMC are used in the designs. Use of the identical design environment for the adiabatic and CMOS logic circuits ensures justifiable comparison among the circuits. The schematic design and HSPICE simulations are realized using the industry standard tool flow. Adiabatic gain values of up to 10 are realized in the circuits.

In paper [9], the author proposed a design of a 16-bit RISC CPU core using an adiabatic logic which is called a two phase drive adiabatic dynamic CMOS (2PADCL), in this paper. The proposed adiabatic RISC CPU is non-pipelined with a latency of three cycles, and also consists of six blocks; an arithmetic and logic unit (ALU), a program counter, a register file, an instruction decoder unit, a multiplexer and a clock control unit. Through the SPICE simulation, the 2PADCL CPU was evaluated for 0.35nm standard CMOS library and was compared with the CMOS CPU. The simulation results show that the power consumption of the adiabatic CPU is about 1/4 compared to that of the CMOS CPU.

In [10], authors focused on principles of adiabatic logic, its classifications and comparison of various adiabatic logic circuits. An attempt has been made in this paper to modify some adiabatic logic circuits to minimize total power consumption with respect to normal CMOS logic. This paper investigates the design approaches of low power adiabatic gates in terms of energy dissipation. A computer simulation using SPICE is carried out on several inverter circuits.

III. CONCLUSION
From the above literature survey it can be concluded that 5 stage pipelined structure can be implemented to achieve faster processors while there will be a trade-offs between processor size and power consumptions.

IV. REFERENCE