International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Design of Instruction Set Architecture Based 16 Bit MIPS Architecture with Pipeline Stages

Kanaka Sai Hemanth Dogga[1], Chand Basha Shaik [2], Sreemukhi Muddusetty[3]

[1]*B. Tech, Electronics and Communication Engineering, Andhra University, Vizag,*
[2]*Director, Business Development at SION SEMICONDUCTORS PVT LTD, Bangalore, India*
[3]*B.Tech, Electronics and Communication Engineering, India Indian Institute of Information Technology Design and Manufacturing, Kurnool, India*

Abstract - *The MIPS instruction set architecture (ISA) is a RISC (Reduced Instruction Set Architecture) based micro-processor architecture. The Instruction set architecture (ISA) is of built-in data types especially integers, floating point numbers, fixed set of instructions, fixed set of on-processor variables, interface for reading/writing memory, mechanisms to do input/output. In this cutting-edge world, microchips are discovering their way into pretty much every field in this era, joining a component of quickness into customary gadgets. Microcontrollers are the need of the day. Energy productive, space effective are improved continuously and researchers are still working on it. Our paper demonstrates Instruction Set that is a subset of MIPS architecture. It demonstrates the benefits of MIPS like simplicity and fastness. In addition, it is a smartly optimized subset of MIPS. It is a faster version consisting of the most commonly required instructions. In this paper, the architecture based on 16-bit data width, consists 5 stage pipeline, and 8 general purpose registers are used.*

Key Words: ISA, RISC, MIPS, pipeline, general purpose registers, and data width.

1.INTRODUCTION

Processors are viewed as one of the most significant devices in our regular machines called PCs. Prior to we start, we need to comprehend what precisely processors are and their suitable executions. Processor is an electronic circuit that capacities as the CPU of a PC, giving computational control. Processors are likewise utilized in other progressed electronic frameworks, like PC printers, vehicles, and stream aircrafts, Mini-computers and so on.

Processors have been portrayed in various ways. They have been contrasted with the mind and the core of people. Their activity has been jumped at the chance to an exchanged board, and to the sensory system in a working system. They have frequently been called microcomputers. The first reason for the processor was to control memory. That is the thing that they were initially intended to do, and that is their main event today. In particular, a processor is a segment that carries out memory.

Normal processors fuse arithmetic and logic functional units just as the related control logic, guidance preparing hardware, and a segment of the memory pecking order. Bits of the interface logic for the input/output (I/O) and memory subsystems may likewise be implanted, permitting less expensive in general frameworks. While numerous processors and single-chip plans, some elite plans depend on a couple of chips to give various functional units and moderately huge stores.

Programmers have added guidance and information storage to the processors. A reserve is an uncommon kind of rapid smash where information and the location of the information is put away. At whatever point the processor attempts to peruse information from fundamental memory, the reserve is analyzed first. On the off chance that one of the addresses put away in the reserve coordinates with the location being utilized for the memory read, it is called a hit. The store will supply the information all things considered. Reserve is ordinarily multiple times quicker than main memory. The information will be there the following time we need it. Cache reserve is utilized to store often utilized guidelines. Information reserve is utilized to store much of the time utilized information.

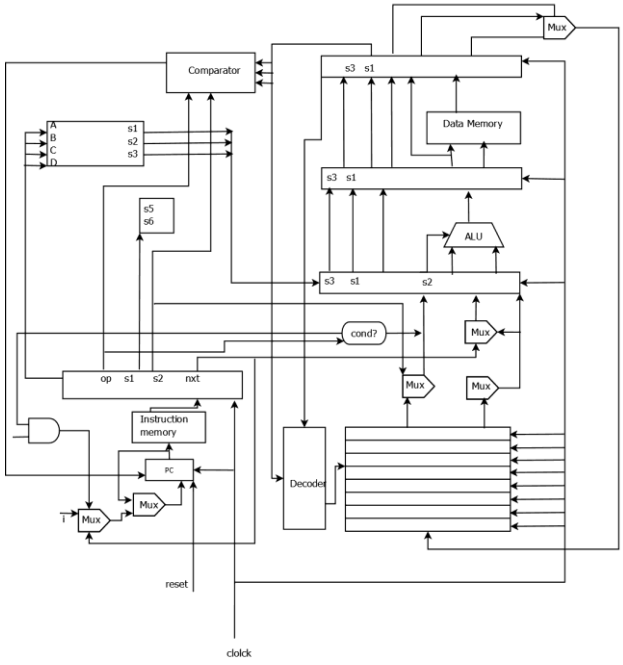
RISC normally has enormous arrangement of registers. The quantity of registers accessible in a processor can influence execution a similar way a memory access does. An intricate computation may require the utilization of a few information esteems. In the event that the information esteems all live in memory during the computations, numerous memory-elements gets to should be utilized to use them. On the off chance that the information esteems are put away in the inward registers of the processor all things being equal, their entrance during computations will be a lot quicker. It is acceptable then to have part of interior registers. Executing less guidelines and tending to modes on silicon diminishes the intricacy of the guidance decoder, the tending to rationale, and the execution unit. This permits the machine to be timed at a quicker speed, since less work needs to be done each clock period.

1.1 MIPS Architecture

Instruction set architecture (ISA) based MIPS is a RISC based microprocessor architecture that was developed by MIPS Computer Systems Inc. MIPS is now an industry standard and the performance leader within the embedded system industry. These designs are found in various products used in our everyday lives. It was estimated that one in three of all RISC chips produced was a MIPS-based design. Architecture of MIPS and RISC microprocessor includes a

specified length of straight forward decoded instruction format. RISC processors generally have a load-store architecture. This means there are two instructions for accessing memory. They are load instruction to load data from memory and a store instruction to write data to memory. It also means that no other instructions can access memory directly. For example, an instruction like add this byte from memory to register 1, from a CISC instruction set might require two instructions in a load-store architecture. So, in MIPS memory access is limited to load and store instructions. It has a hardwired control unit. It has a large general purpose register file so that all operations are done within the registers of the microprocessor. Architecture based on Instruction Set Architecture performs three basic operations. They are arithmetic/bitwise logic namely addition, subtraction, multiplication, right- shift, and xor etc. Data transfers between registers and memory, and control flow. Control flow include jump to an address in memory unconditionally, jump to an address, invoking function and return from function.

MIPS 16 BIT ARCHITECTURE



The execution of each instruction in a processor can be split up into a certain number of stages. The purpose of each stage is different for each processor design. The MIPS processor has 5 stages: Instruction Fetch, Instruction Decode, Execution stage, Memory stage, and Write Back stage.

1.2 Processor Stages

Instruction Fetch: The instruction is fetched from memory and placed in the instruction register (IR). The Program Counter (PC) address is incremented and then written back into the PC to be ready for the coming clock cycle. This next incremented address is saved in the IF pipeline register.

Instruction Decode: Each instruction bits are considered and are decoded into control signals. Operands are moved from respective registers to working registers. For each instruction, the condition is cross-checked and the address is computed. Control Logic is given in the below table.

OPcode	OP	S1	S2	S3	S4	S5	S6	ALU
0001	ADD	0	000	0	0	0	0	000
0010	SUB	0	001	1	0	0	0	001
0011	AND	0	010	1	0	0	0	010
0100	OR	0	011	1	0	0	0	011
0101	NOT	0	100	1	X	0	0	100
0110	SL	0	101	1	0	0	0	101
0111	SR	0	110	1	0	0	0	110

Execute Stage: In this stage, the instruction is executed. If the instruction is an arithmetic or logical operation, its results are computed. If it is a load-store instruction, the address is computed. All this is done by arithmetic-logical unit (ALU).

Memory Stage: If the instruction fetched is a load-store, the memory is read or written according to instruction.

Write Back Stage: The results of the operation are written to destination register.

2. Hazard Detection Unit

Hazard Detection Unit compare the source register of the instruction First in the Instruction Decode stage and the earlier instructions in the destination register. The hazard detection units assert pipeline stall signal if the source register is equal to any of the destination registers and not equal to zero. The pipeline stall signal will be canceled if the signal is sent to another stage. The signal is stored in the Instruction Fetch of Instruction Decode stage and alerts the Execution stage.

3. MIPS Assembly/Instruction Formats

R Instructions: R instructions are used when all the values of data used by the instruction are located in registers.

All R-type instructions have the format given below:

OP rd, rs, rt

Here "OP" is the mneumonic for the decided instruction. rs and rt are the source registers, and rd is the destination register.

Instruction Set R- Type Used:

ADD	**0001**
SUB	0010
AND	0011
OR	0100
XOR	0101

I Instructions: I instructions are used when the given instruction is operated on an immediate value and a register value. The Immediate values may be a maximum of 16 bits long. Larger numbers might not be manipulated by the immediate instructions.

I instructions are called in the following format

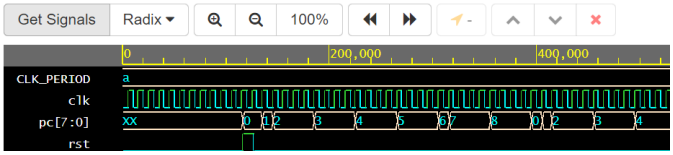
OP rt, rs, IMM

Where rt is the destination register, rs is the source register, and IMM is the immediate value.

Instruction Set I- Type Used:

ADDI	**1001**
LD	1010
ST	1011
BZ	1100

4.Simulation Results



The Program Counter is properly functioning as expected and the given assembly program showed up expected results

The instructions are given and the registers store the value according to the instruction. The instructions are mentioned in the figure. The values of registers are mentioned below.

```
0     1       3       6       9       9       0       15
  R0      R1      R2      R3      R4      R5      R6      R7
     16_Bit MIPS TESTING
     Program Counter:   0 ,Instruction: 9201
     Program Counter:   1 ,Instruction: 9442
     Program Counter:   2 ,Instruction: 9683
     Program Counter:   3 ,Instruction: 1898
     Program Counter:   4 ,Instruction: b842
     Program Counter:   5 ,Instruction: aa42
     Program Counter:   6 ,Instruction: 2d28
     Program Counter:   7 ,Instruction: c1b8
     Program Counter:   8 ,Instruction: 9fc5
     Program Counter:   0 ,Instruction: 9201
     Program Counter:   1 ,Instruction: 9442
     Program Counter:   2 ,Instruction: 9683
     Program Counter:   3 ,Instruction: 1898
     Program Counter:   4 ,Instruction: b842
     Program Counter:   5 ,Instruction: aa42
     Program Counter:   6 ,Instruction: 2d28
     Program Counter:   7 ,Instruction: c1b8
     Program Counter:   8 ,Instruction: 9fc5
```

REFERENCES

[1] https://en.wikipedia.org/wiki/MIPS_architecture

[2] Gautam P. Parthasarathy R, Karthi Balasubramanium, "Low Power pipelined MIPS Processor design", in the proceedings of the 2009,12th international symposium, 2009 Research in Computer Science (JGRCS), September 2011.

[3] Kaur Harpreet, Gulati Nitika, "Pipelined MIPS processor architecture with Improved Datapath", International Journal of Research in Engineering and Application (IJREA), January-February 2013.

[4] David A. Patterson and John L. Hennessy, Computer Organization and Design, 3rd ed.

[5] MIPS Official Website.
Available:http://www.mips.com/products/architectures/mips32/