DESIGNING A 16 BIT PROCESSOR

Muhammad Asfandyar(180387) Abdullah(180374)

Abstract—This paper presents the design of a 16 bit Reduced Instruction Set Computing (RISC) processor Designing on logisim.Various functional blocks of the processor such as the Control Unit, Instruction Decoder, Instruction Register unit and Arithmetic and Logical Unit (ALU) are designed using the logisim tool.

Index Terms—— Processor,Designing,mips

1 INTRODUCTION

THE microprocessor is a deep example of the Very Large Scale Integration industry, which takes input data as 0 and 1 and processes it according to instructions. It is expected to yield output according to specific instructions at maximum speed. A physical set of hardware modules serves the purpose. Main components are Arithmetic Logic Unit (ALU), Control Unit, Memory unit etc. The emergence of Low Instruction Set Computing (RISC) processors was an evolution in computing research platform during recent years. It has a much simpler and designed architecture with fewer fixed length instructions than Complex Instruction Set Computing which, on the other hand, had a complex architecture with a high number of instruction sets.

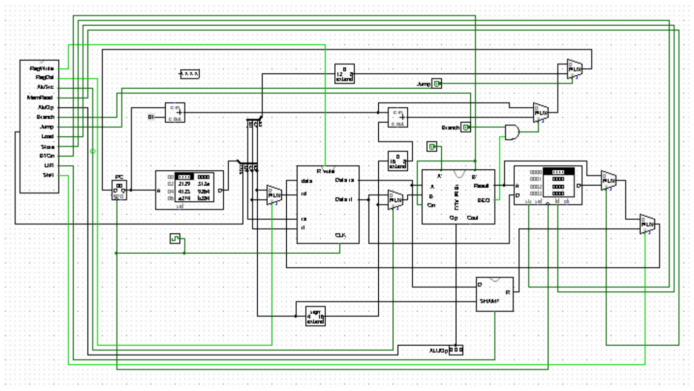


Fig. 1. 16-bit processor desinge on logisim

2 ARCHITECTURE

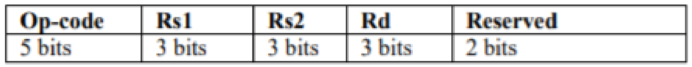
The objective of the project is to design a 16- bit MIPS processor based on Harvard architecture which utilizes minimum functional units. The architecture of proposed 16-bit Processor is shown in Fig.1. The processor incorporates 16-bit ALU capable of performing 11 arithmetical and logical operations, 16- bit program counter, 24-bit Instruction register, Sixteen 16-bit general purpose registers, 3-bit flag register to indicate carry, zero and parity. The processor has four states idle, fetch, decode and execute. The control unit provides necessary signal interaction to perform expected function in all the states. The 16-bit program counter indicates the address of memory location from which the instruction is to be fetched. After the execution of current instruction, the program counter is incremented by one unless JUMP instruction is encountered. When the JUMP instruction is executed the program counter is incremented or decremented by the amount indicated by the offset.

3 MIPS 16 INSTRUCTION SET DESCRIPTION

MIPS instructions have fixed width. The original MIPS 32 ISA has 32 bits wide instructions. Each instruction in MIPS16 is 16 bits wide. Further, MIPS16 has 8 internal registers as opposed to the 32 registers of MIPS32. As the name suggests, data bus is 16 bits wide and address bus is preferably 16 bits wide too. I/O support is memory mapped. Memory is accessed by LOAD and STORE instructions. The instructions follow an ¡operand register, register, register¿ format.The Instructions can be divided into 4 groups:

- Arithmetic: Basic computational instructions add and subtract.
- Logical: Operations like AND, OR, XOR.
- Data Transfer: Load and Store .
- Branch and control: Jump, Call, Return, etc.

A MIPS16 instruction is 16 bits wide. Since MIPS uses a Register-Register type of instruction a general instruction specifies two source registers and a destination registers. The format of such an instruction will be ADD R1,R2,R3 R1 = First source operand register R2 = Second Source operand register R3 = Destination register The instruction word has a 5 bit op-code specifying the operation to be performed. Number of operands may be variable e.g. ADD requires three operands while NOT requires only two. Format of a three operand instruction word is shown in Fig:1



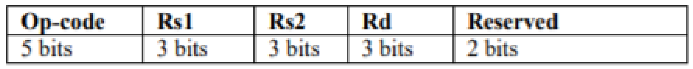
Op-code	Rs1	Rs2	Rd	Reserved
5 bits	3 bits	3 bits	3 bits	2 bits

Fig. 2.

- *E-mail: 180374@students.au.edu.pk*
- *E-mail: 180387@students.au.edu.pk*

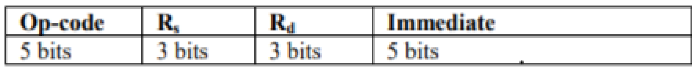
3.1 Instruction Decoder

The instruction decoder decodes the 16 bit instruction and it enables the respective functional units for execution. There are three types of instructions, namely, R type, I type and J type instructions. As a typical Mips processor,the design employs the instructions of 32 bit word size, b ut we have to make it 16 bit.The three types of mips instructions are depicted in Figure 3and Figure 4 and Figure 5 .



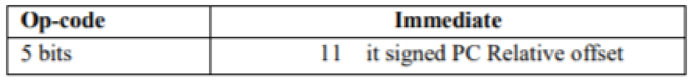
Op-code	Rs1	Rs2	Rd	Reserved
5 bits	3 bits	3 bits	3 bits	2 bits

Fig. 3. R-type instruction



Op-code	R_s	R_d	Immediate
5 bits	3 bits	3 bits	5 bits

Fig. 4. I-type instruction



Op-code	Immediate
5 bits	11 it signed PC Relative offset

Fig. 5. J type instruction

3.2 ALU

MIPS has instructions to perform addition and subtraction. These instructions source their operations from the two the operation using registers (RS and RT), and write the result to the third register (RD). Alternatively, an operand can also be 16-bit instantiated (which is sign-extended to 32 bits).

MIPS contains bitwise logical and instructions for OR, XOR, and NOR. These instructions source their operations from the two registers and write the result to the third register. The AND, OR, and XOR instructions can optionally source one of the operands from a 16-bit instant (which extends from zero to 32 bits).In the case of ALU instructions, the 2 reserved bits act as function bits. For example, ADD and ADDi instructions have the same opcode but varies in Function Bits. This results in simple control logic because the reserved bits are directly decoded by the ALU control. The design of the ALU on logisim is shown in the fig.6

The ALU performs the 16 bit arithmetic operations, such as the Addition, Subtraction, Multiplication and Division, and 16 bit logical operations, such as AND, OR and exclusive OR. The ALU consists of the following blocks.

- 16 bit AND operation unit
- 16 bit OR operation unit
- 16 bit XOR operation unit
- 16 bit Adder unit
- 16 bit Multiplier unit
- 16 bit Subtracter unit
- 16 bit Shifter

3.3 Control Unit

The instruction register brings in the instruction, which is decoded by the instruction decoder. The output of the instruction decoder is fed to the control unit. The control part of the processor asserts the necessary control signals to the ALU and the appropriate registers and ports. It also initiates the program counter to fetch the next instruction. The control unit of our processor is shown in the fig.7

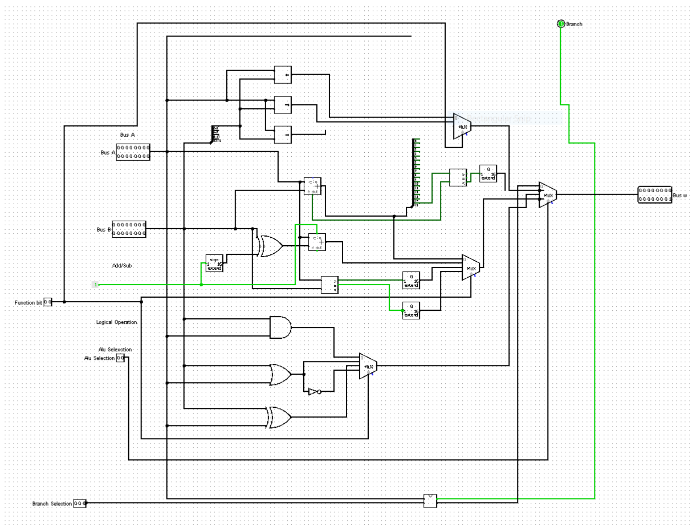


Fig. 6. The ALU Design on logisim

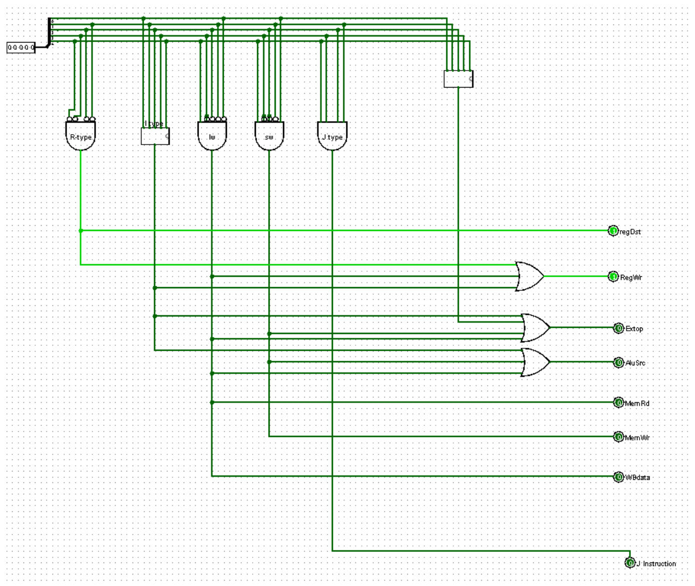
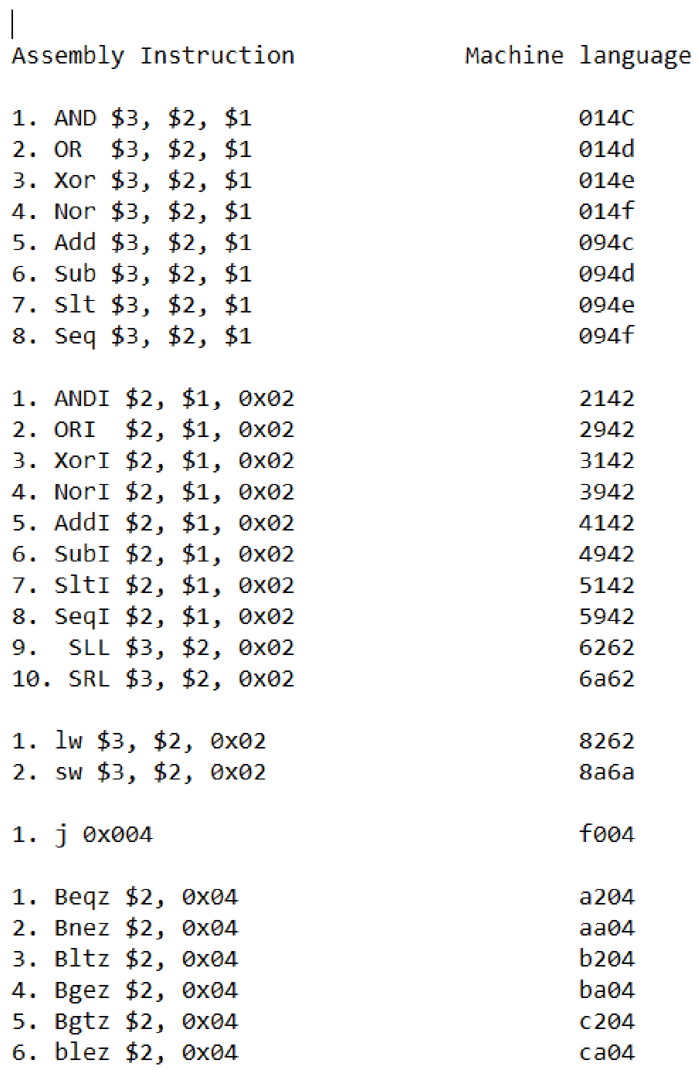


Fig. 7. The control unit of 16-bit processor

4 List OF Instruction

As the op-code has a 5 bit length there are 32 possible distinct instructions. If the reserved bits at the end of the instruction are utilized for grouping 2 or more similar instructions more op-codes can be incorporated in the instruction set e.g. ADD and ADDi can be grouped together as they perform similar function with the difference being inclusion of carry into the sum.We are not going to implement all of the 32 instructions, we are only going to implement major part of it.List of instructions that our processor is going to execute are shown in the fig.8



```
|
Assembly Instruction              Machine language

1. AND $3, $2, $1                        014C
2. OR  $3, $2, $1                        014d
3. Xor $3, $2, $1                        014e
4. Nor $3, $2, $1                        014f
5. Add $3, $2, $1                        094c
6. Sub $3, $2, $1                        094d
7. Slt $3, $2, $1                        094e
8. Seq $3, $2, $1                        094f

1. ANDI $2, $1, 0x02                     2142
2. ORI  $2, $1, 0x02                     2942
3. XorI $2, $1, 0x02                     3142
4. NorI $2, $1, 0x02                     3942
5. AddI $2, $1, 0x02                     4142
6. SubI $2, $1, 0x02                     4942
7. SltI $2, $1, 0x02                     5142
8. SeqI $2, $1, 0x02                     5942
9.  SLL $3, $2, 0x02                     6262
10. SRL $3, $2, 0x02                     6a62

1. lw $3, $2, 0x02                       8262
2. sw $3, $2, 0x02                       8a6a

1. j 0x004                               f004

1. Beqz $2, 0x04                         a204
2. Bnez $2, 0x04                         aa04
3. Bltz $2, 0x04                         b204
4. Bgez $2, 0x04                         ba04
5. Bgtz $2, 0x04                         c204
6. blez $2, 0x04                         ca04
```

Fig. 8. Main instruction that can be executed by our processor with respective hexadecimal value.

The above mentioned processor components are single cycle, means that in a single cycle processor it carries out one instruction in a single Clock cycle.

5 MULTI CYCLE 16-BIT PROCESSOR DESIGN

Multi-cycle datapath: instruction execution. Breaking instruction execution into multiple clock cycles: Balance amount of work done in each cycle (minimizes the cycle time).Multi cycle processors also allow computers that have a single memory unit instead of the two separate instructions and data memory units of a traditional cutting machine. The reason for this is that instructions are loaded on one cycle, and data memory is interfered on another cycle. CPI can be ¡1 on machines that execute more than 1 instruction per cycle .The CPU takes minimal time to do any work. The clock cycle time or clock duration is only one cycle length.

5.1 Implementing Pipelining in Processor

Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing. Pipelining is a technique where multiple instructions are overlapped during execution.

A multi cycle implementation divides the execution of an instruction into well-defined time states. The execution happens in a timely order and might require different number of time states for different instruction. The main advantage is the hardware can be shared for similar elementary functions in different time states. Multi cycle should be preferred for applications with smaller chip area requirements.Pipelining provides performance enhancement by concurrent execution of more than one pair able instructions. The design involves use of multiple datapaths and a logic to check for pairability and hazard removal that occurs due to concurrent execution. This significantly complicates the design and takes a larger chip area. But the performance improvement would be tremendous.

6 PIPELINE HAZARDS

Pipeline hazards are situations that prevent the next instruction in the instruction stream from executing during its designated clock cycles. Any condition that causes a stall in the pipeline operations can be called a hazard.Pipeline hazards prevent next instruction from executing during designated clock cycle.Common solution is to stall the pipeline until the hazard is resolved.

6.1 Structural Hazards

- Arise from resource conflicts.Avoid structural hazards by duplicating resources
- Using same resource by two instructions during the same cycle.If not all possible combinations of instructions can be executed, structural hazards occur

6.2 Data hazards Hazards

- Occur when given instruction depends on data from an instruction ahead of it in pipeline.
- Hardware Can detect dependencies between instructions.

6.3 Control Hazards

- Result from branch, other instructions that change flow of program (i.e. change PC).
- Delays in changing the flow of control.

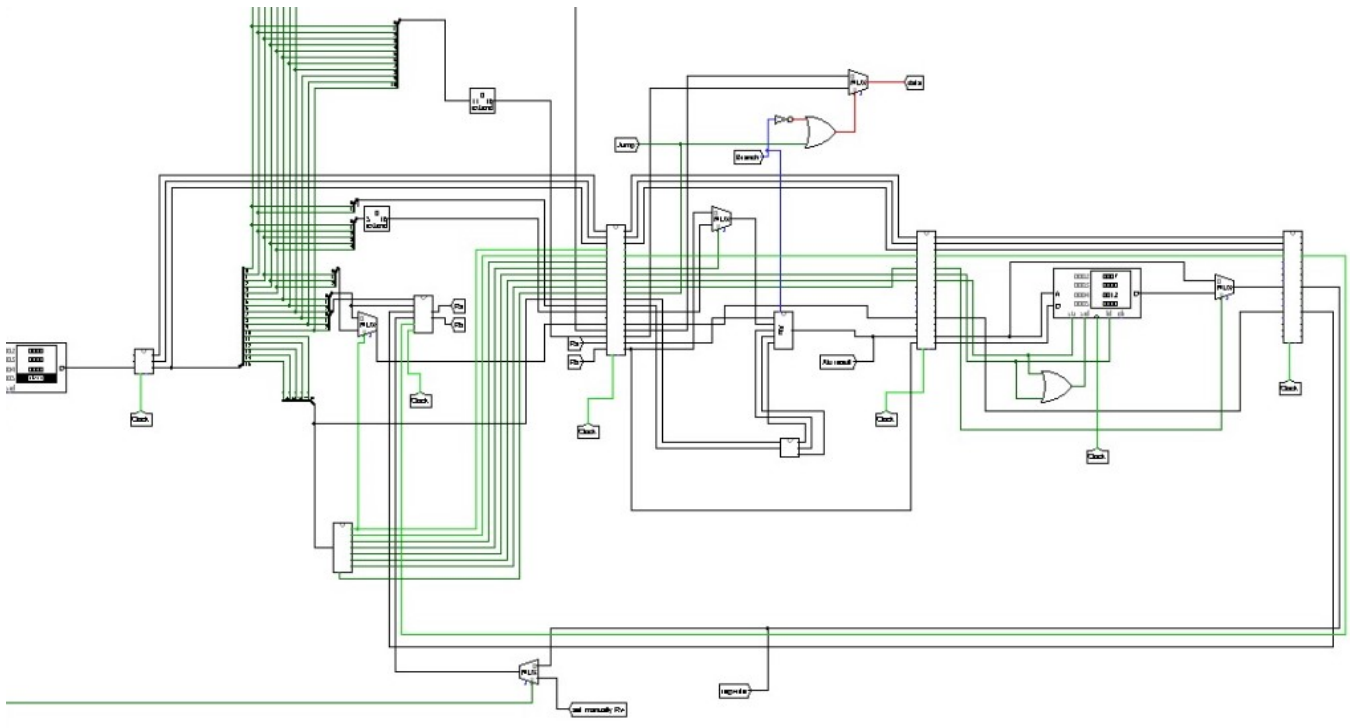


Fig. 9. Processor design with pipeline registers

7 CONCLUSION

A 16-bit MIPS processor has been designed that utilizes minimum functional units. Simulation is done using logisim simulator. The simulation output is compared with the expected results and the functionality is found correct.The design can be improved in number of ways. To achieve a more sophisticate design more features can be added to the current design. The number of instructions that the processor supports can be increased. Pipelining can be added to improve the performance of the proposed design.

REFERENCES

[1] https://www.101computing.net/enigma-machine-emulator/: McGraw-Hill.
[2] https://piotte13.github.io/enigma-cipher
[3] Ellsbury,Graham (1998),The Turing Bombe:what it was and how it worked,retrieved 1 May 2010.
[4] Davies,Donald (*April 1999),"The Bombe-a Remarkable Logic Machine ", Cryptologia,23.
[5] https://piotte13.github.io/enigma-cipher/
[6] :https://www.youtube.com/watch?v=4L6KtS0t75w
[7] :https://www.youtube.com/watch?v=2D2bJWHvqJo