Project Report

of

Design of 6-stage pipelined RISC processor

by

Sanket P Gedam (213070078)

Abstract

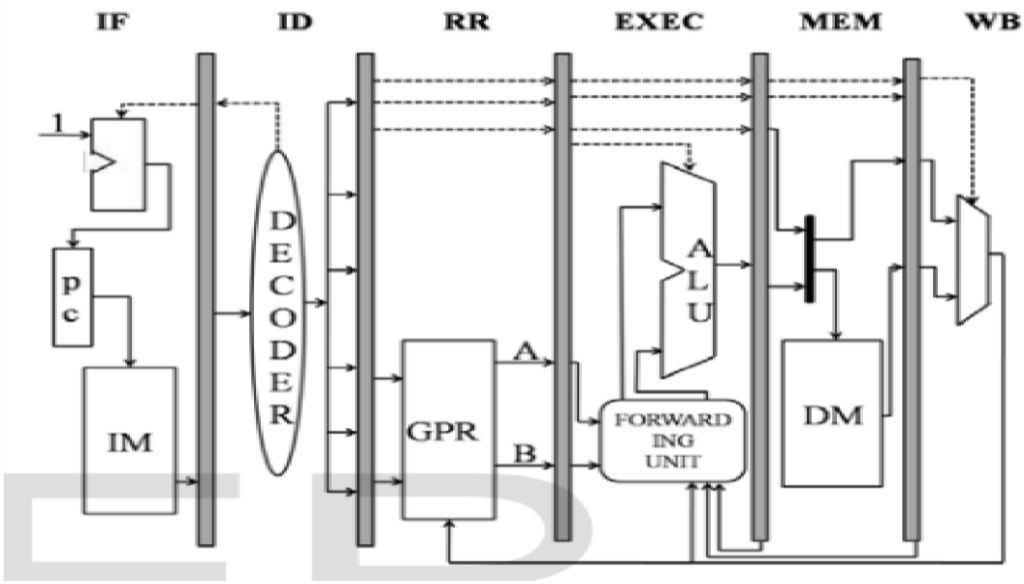
The aim of this project is to design a 6-stage pipelined RISC processor, using VHDL. The 6 stages being used are Instruction Fetch (IF), Instruction Decode (ID), Register Read (RR), Execute (EX), Memory (MEM) and Write Back (WB). The instruction set being used is of 16-bits. The various modules being used are Instruction Memory, Data Memory, ALU, Registers etc. I also have implemented a Forwarding Unit and a hazard detection unit for the detection of Data hazards.

MIPS is basically a Microprocessor without Interlocked Pipelined Stages. By designing MIPS, we are basically designing a RISC based CPU (Central Processing Unit). A central processing unit (CPU), also referred to as a central processor unit, is the hardware within a computer that carries out the instructions of a computer program by performing the basic arithmetical, logical, and input/output operations of the system. RISC (Reduced Instruction Set Computer) CPU, made use of simpler instructions, larger no. of registers, simpler pipelined processor, a small transistor count. Because of this, it becomes cheaper and easy to design at high clk rate. Single-cycle based instructions became easier because of RISC.

Pipelined DATAPATH and Control Path

The division of an instruction into six stages means a six-stage pipeline, which in turn means that up to 6 instructions will be in execution during any single clock cycle. The 6-stages of instruction execution are :

1. IF: Instruction fetch
2. ID: Instruction decode and register file read
3. RR: Register read.
4. EX: Execution or address calculation
5. MEM: Data memory access
6. WB: Write back



single-cycle pipelined datapath

Pipelined Stage Description

Before knowing what MIPS is, we need to focus upon the concept of Pipelining. Pipelining is nothing but doing more than one operation, in a single data path. A multi-cycle CPU consists of many processes. For example load might take up to 5 clock cycles, but beq takes only 3 clock cycles. So if one process is taking place, instead of waiting for the process to complete, we can simultaneously start a new process in the same data path, without disturbing the previous process.

For this to happen, the each part of the process is divided into various pipelined stages. So after every clock, the process is stored into next pipelined stage, enabling another operation to start in that stage without disturbing the previous process. Hence all the stages in the path can be used simultaneously. This in turn can increase the throughput of your design.

The Six stages are the following:

1. Instruction fetch UNIT : In the IF stage instructions are fetched one by one from the instruction memory according to the PC value. Program counter (PC) keeps the track of instruction that is being fetched. Instructions are fetched at every clock cycle from instruction memory.

2. INSTRUCTION DECODE This unit read instruction from instruction register and decodes the operands according to operation. The 16 bit instruction will be divided into several parts.

3. REGISTER READ UNIT In this unit we calculate the address of read register and this address is send to the GPR. Data from GPR (general purpose register) is read

and forwarded to the next unit. GPR is dual ports RAM in which read and write operations can be done simultaneously at different address.

4. EXECUTION UNIT The main function of this stage is arithmetic calculation. This unit contain ALU unit. The inputs to the ALU are selected by forwarding unit. It decides that from where data is forwarded whether from EXEC-to-EXEC unit forwarding or MEM-toEXEC forwarding or WB-to-EXEC forwarding.

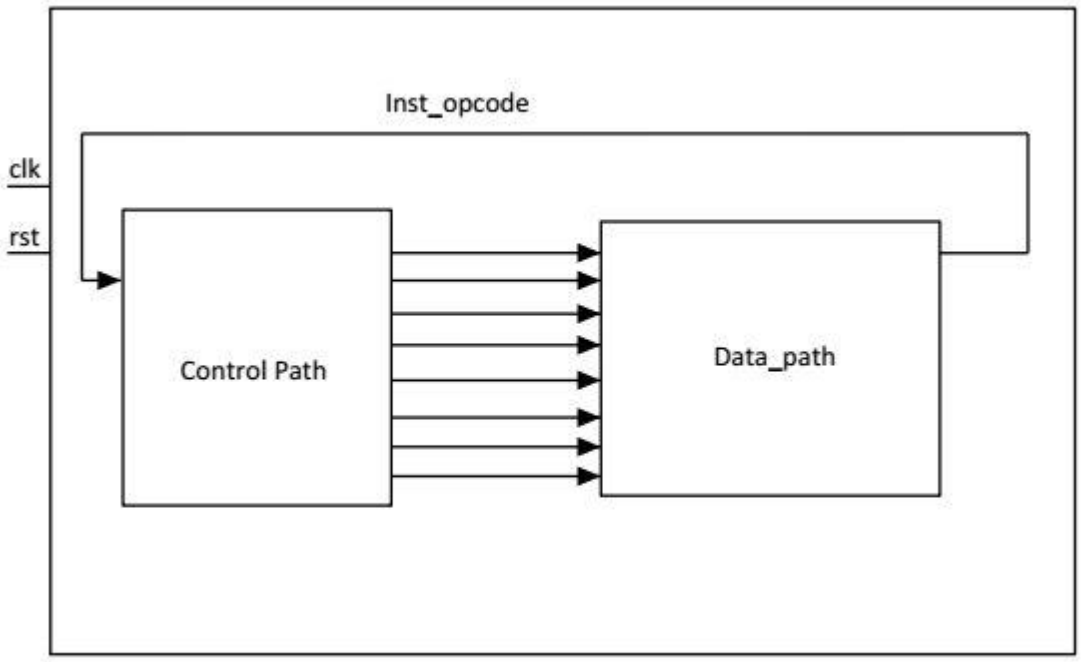
5. DATA MEMORY If there is any data to be written or read from the data memory then this unit is used. So this stage is only used for loading and storing instruction, which read and writes the data memory respectively. For other Instruction this unit is not used. The result of the ALU can be directly stored in the data memory. This unit interface with the data memory.

6. WRITE BACK This stage is used when we need to write back to the GPR. It is used for writing any data from instruction or storing result of the ALU to the GPR

Control:

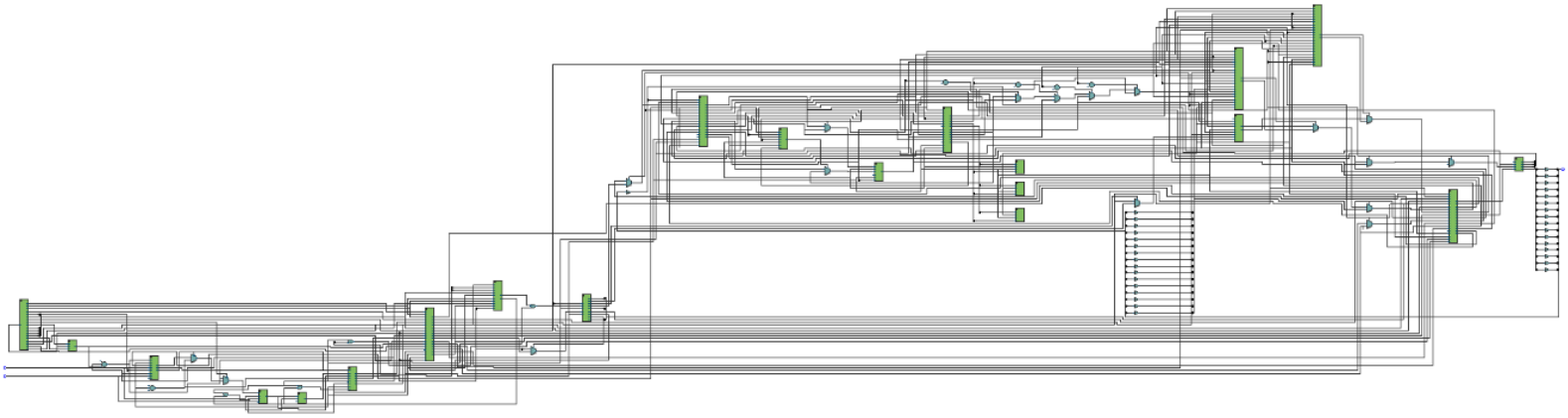
Control is one of the most important modules for the design. It generates different control signals for various modules in the design. It mainly implements control signals which are passed from pipelined registers.

The top module was divided into 2 modules, control module cntl_path and data module data_path respectively. The cntl_path gives all the control signals, required for datapath. It is as below:


Inst_opcode
clk
rst
Control Path
Data_path

Simulation

The simulation of the 6 stage pipelined RISC Processor was carried out on Intel Quartus Prime and Modelsim tool. The RTL view of the implemented circuit is



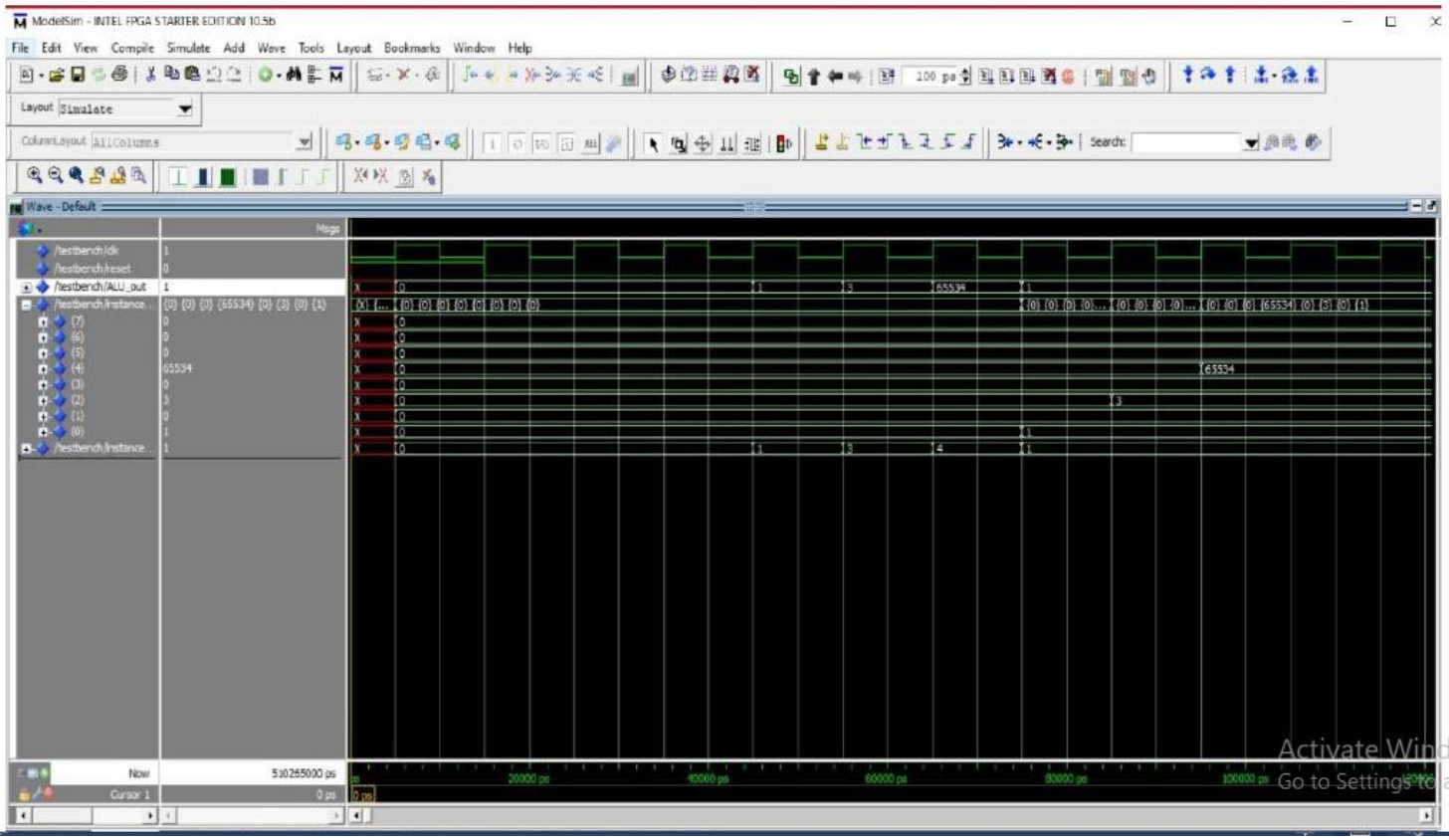
Results : Waveform of Add and ADI instruction

adi $3 $0 1

adi $1 $2 3

add $0 $2 $4

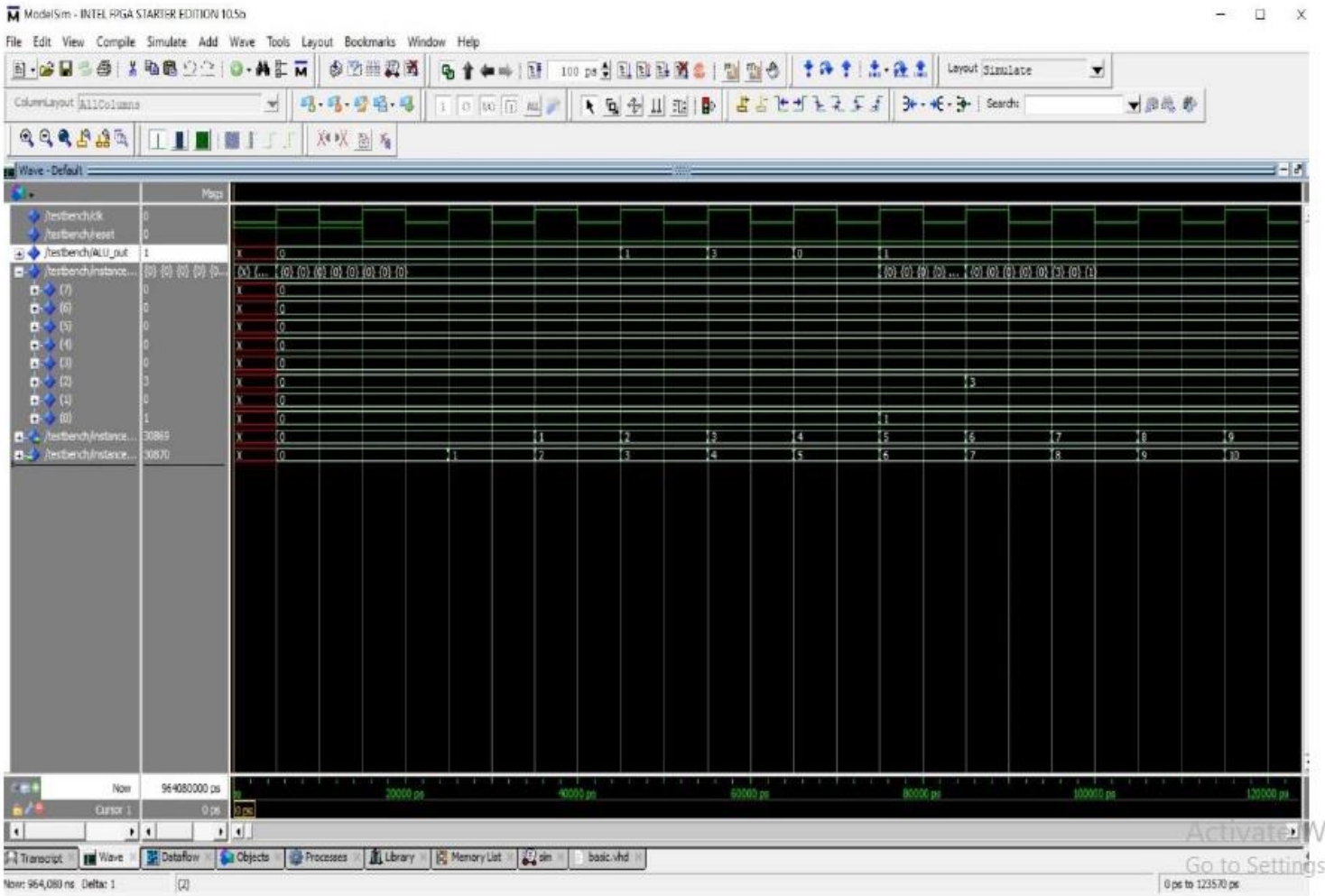
register R0 and R2 gets loaded by 1,3 respectively as R3 and R1 contains 0 value. Their addition is done in alu which can be seen at last row in simulation. But, during simulation latch is forming at output; because of that output of ALU changes to 65534 which gets stored in R4.



Waveform for LM instruction

LM $1 0 11111111

As we can see R1 contains 0 value which is the address of memory and it will load the content of the memory of location 0 to 7 to register R0 to R7; as all are given high in the immediate field.



Conclusion

6-stage pipelined RISC is a 16-bit very simple computer developed for the teaching that is based on the Little Computer Architecture. The 6-stage pipelined -RISC is an 8-register, 16-bit computer system. It has 8 general-purpose registers (R0 to R7). Register R7 is always stores Program Counter. All addresses are short word addresses (i.e. ad- dress 0 corresponds to the first two bytes of main memory, address 1 corresponds to the second two bytes of main memory, etc.). This architecture uses condition code register which has two flags Carry flag (C) and Zero flag (Z).

The 6-stage pipelined -RISC is very simple, but it is general enough to solve complex problems. The architecture allows predicated instruction execution and multiple load and store execution. There are three machine-code instruction formats (R, I, and J type) and a total of 13 instructions.