|  |
| --- |
| **DIGITAL SYSTEM DESIGN LABORATORY** |
| **LAB 5** |

**SINGLE CYCLE MICROPROCESSOR DESIGN**

**Instruction Datapath**

**Part 1**

### I. LAB OBJECTIVES

### This Lab experiments are intended to design and test a Single Cycle Microprocessor

### II. DESCRIPTION

### Single Cycle Microprocessor datapath to be implemented is in figure 2.1.

### 

### Figure 2.1: Single Cycle Microprocessor DataPath

### III. LAB PROCEDURE

### III.1 EXPERIMENT NO. 1

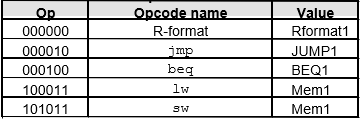
##### III.1.1 AIM: To understand and write the assembly code using MIPS Instruction set

##### register number of MIPS compiler conventions

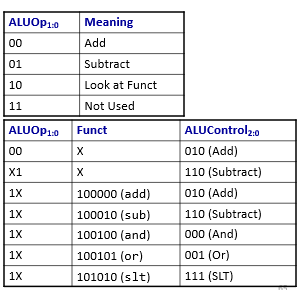
##### 

##### MIPS Assembly Language Sumarize:

##### Operation code (Op code) sumarize:

****

**ALU opcode and Function**

****

##### Instruction Formats

**Timeline

Description automatically generated**

**III.1.2 CODE**

**a) Assembly Code sample 1:**

Instruction Meaning

addi $s0, $zero, 33 load immediate value 33 to register $s0

addi $s1, $zero, 66 load immediate value 66 to register $s1

add $s2, $s0, $s1 $s2 = $s0 + $s1  
sub $s3, $s1, $s0 $s1 = $s1 – $s0  
sw $s3, 10($s2) Memory[$s2+10] = $s3

lw $s1, 10($s2) $s1 = Memory[$s2+10]

**b) Assembly Code sample 2:**

Assume the code start from address PC=0x00000000, one instruction is store in one memory location.

Instruction Meaning

addi $s2, $zero, 55 load immediate value 55 to register $S2

addi $s3, $zero, 22 load immediate value 22 to register $S3

addi $s5, $zero, 33 load immediate value 55 to register $S3

add $s4,$s2,$s3 $s4 = $s2 + $s3  
sub $s1,$s2,$s3 $s1 = $s2 – $s3  
sw $s1,100($s2) Memory[$s2+100] = $s1

lw $s1,100($s2) $s1 = Memory[$s2+100]   
bne $s1,$s5,End Next instr. is at End if $s4 !=$s5

addi $s6, $zero, 10 load immediate value 10 to register $s6  
beq $s4,$s5, End Next instr. is at End if $s4 = $s5

addi $s6, $zero, 20 load immediate value 20 to register $s6

End: j End jump Here

**III.1.3 LAB ASSIGNMENT**a) Compile the Assembly **Assembly Code sample 1** into machine code (decimal code and binary code)

b) Explain briefly the meaning of **Assembly Code sample 1**

c) Compile the Assembly **Assembly Code sample 2** into machine code (decimal code and binary code)

d) Explain briefly the meaning of **Assembly Code sample 2**

### III.2 EXPERIMENT NO. 2

##### III.2.1 AIM: To implement R-Type Datapath

##### 

##### Given the assembly code

addi $s1, $zero, 33 load immediate value 33 to register $s0

addi $s2, $zero, 66 load immediate value 66 to register $s1

##### add $s0, $s1, $s2 $s0 = $s1 + $s2

##### Translate into Macchine code:

##### 

##### 

##### Modify the code register file:

##### Assign initial value of the register 17=33 ( $s1=33)

##### Assign initial value of the register 18=33 ( $s2=66)

##### 

##### 

##### 

**III.2.2 CODE**

module Datapath R\_Type\_Add(rs, rt, rd,ALUop,Zero,ALU\_out);

// Your code here

endmodule

module Datapath R\_Type\_Sub(rs, rt, rd,ALUop,Zero,ALU\_out);

// Your code here

endmodule

**III.2.3 LAB ASSIGNMENT**1) Write Verilog code to implement R\_Type\_Add module

2) Write testbenches to verify R\_Type\_Add module, simulate and check the simulation output data.

3) Write Top level Verilog code to implement R\_Type\_Add module in FPGA Kit

4) Write testbenches to verify R\_Type\_Sub , simulate and check the simulation output data.

5) Write Top level Verilog code to implement R\_Type\_Sub module in FPGA Kit

### III.3 EXPERIMENT NO. 3

##### III.3.1 AIM: To implement SW I-Type Instruction Datapath

##### 

**III.3.2 CODE**

module SW\_datapath (rs, rt, offset,ALUOp,MemWrite,MemRead,Mem\_out);

//Your Verilog code here

endmodule

**III.3.3 LAB ASSIGNMENT**1) Write Verilog code to implement SW\_datapath module

2) Write testbenches to verify SW\_datapath module, simulate and verify the output data.

3) Write Top level Verilog code to implement SW\_datapath module in FPGA Kit

### III.4 EXPERIMENT NO. 4

##### III.4.1 AIM: To implement LW I-Type Instruction Datapath

##### 

**III.4.2 CODE**

module Datapath\_LW\_Type(clk,reset,RegWrite,MEM\_read,MEM\_write,ALU\_op, rs, rt,A\_ckeck,offset,Zero,LW\_out);

//Your Verilog code here

endmodule

**III.4.3 LAB ASSIGNMENT**1) Write Verilog code to implement LW\_datapath module

2) Write testbenches to verify LW\_datapath module, simulate and verify the output data.

3) Write Top level Verilog code to implement LW\_datapath module in FPGA Kit

### III.5 EXPERIMENT NO. 5

##### III.5.1 AIM: To implement I-Type Instruction Beq datapath

##### 

**III.5.2 CODE**

module beq\_Datapath (rs,rt,offset,ALUOp,PC\_plus\_4, Zero,beq\_add)

endmodule

**III.5.3 LAB ASSIGNMENT**1) Write Verilog code to implement beq\_Datapath module

2) Write testbenches to verify beq\_Datapath module, simulate and verify the output data.

3) Write Top level Verilog code to implement beq\_Datapath module in FPGA Kit

**III. LAB REPORT GUIDELINES**

Students write up a report on the Verilog HDL implementation experiment projects created in this lab. The lab report should include Verilog code for the module under test, Verilog test bench code and a truth table results, and example data input and output to validate the experiment. Simulation Result in form of Simulation Capture Screen. The implementation results in FPGA Kit, compare the simulation results and implementation results.