University of KwaZulu-Natal

School of Engineering

Examinations: May 2026

ENEL4DP : Digital Processes

Duration: **2 Hours** Marks: 100

Examiners : Prof H. Xu (Internal)
Dr S Rezenom (Moderator)
Dr. Y. A. Gaffar (External)

Instructions : 1. Answer all questions.
2. This is not an open book exam and no notes may be used, either electronic or handwritten.
3. Questions 3 and 4 must be answered on the attached answer sheet, which must be detached and handed in with your answer book. Ensure that you fill in your student number.

Question 1 VHDL [40 marks]

The simple instruction set of practical is shown below:

Instruction Set		Instruction Format			
add	rd ← rs + rt	00	rd	rs	Rt
addi	rd ← rs + Imm	01	rd	rs	Imm
ld	rd ← Mem[rs + Imm]	10	rd	rs	Imm
st	Mem[rs + Imm] ← rd	11	rd	rs	Imm

Each field is 2 bits wide

The datapath of the above instruction set is shown in Figure 1.1

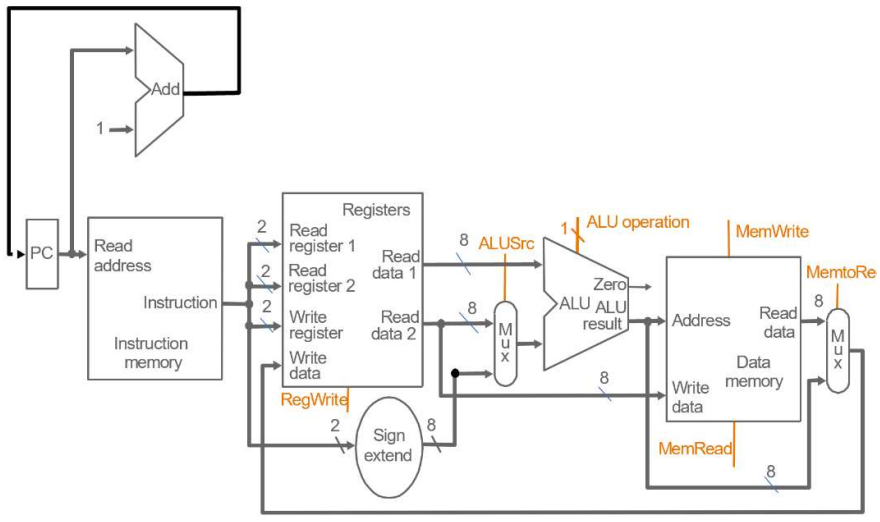

Figure 1.1 the datapath of the simple instruction set

All VHDL questions are based on Figure 1.1

(1.1) [**5 marks**] In Figure 1.1 there are two MUXs, multiplexer 2-to-1. Write the VHDL code to implement the architecture of the entity MUX2_8, **multiplexer 2-to-1**.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
entity MUX2_8 is
        port( A, B: in STD_LOGIC_VECTOR(7 downto 0);
              Z: out STD_LOGIC_VECTOR(7 downto 0);
              Sel: in STD_LOGIC);
end MUX2_8;
```

(1.2) [**5 marks**] In Figure 1.1 there is one sign extend component which signed extends two bits to eight bits. Write the VHDL code to implement the architecture of the entity **SExtend2_8**.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_SIGNED.ALL;
entity SExtend2_8 is
        port( InData: in STD_LOGIC_VECTOR(1 downto 0);
             OutData: out STD_LOGIC_VECTOR(7 downto 0));
end SExtend2_8;
```

(1.3) [**5 marks**] In Figure 1.1 the ALU always performs addition for the above simple instruction set. Determine control signals for Figure 1.1 in Table 1.1.

Table 1.1 Control signals for Figure 1.1

	RegWrite	MemWrite	MemRead	ALUSrc	MemToRe
add: 00					
addi: 01					
Ld: 10					
St: 11					

(1.4) [**5 marks**] Based on Table 1.1, write the VHDL code to implement the architecture of the entity **control**.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_SIGNED.ALL;
entity Control is
        port( OPcode :in STD_LOGIC_VECTOR (1 downto 0);
            RegWrite, MemRead, MemWrite: out STD_LOGIC;
            ALUSrc, MemToRe : out STD_LOGIC);
end Control ;
```

(1.5) [**5 marks**] Write the programme VHDL mif file for the code below.

```
Add $1, $2, $2;
Lw $2, 1($0);
Sw $1, 1($3);
Lw $0, 1($3);
```

(1.6) [**15 marks**] Suppose the mif file is "prac.mif". Write the VHDL code to implement the architecture of the entity **fetch_exam,** fetch instruction.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
LIBRARY lpm;
USE lpm.lpm_components.ALL;

ENTITY  fetch_exam IS
  PORT(SIGNAL  Instruction : OUT        STD_LOGIC_VECTOR( 7 DOWNTO 0 );
       SIGNAL  PC_plus_1_out : OUT   STD_LOGIC_VECTOR( 7 DOWNTO 0 );
       SIGNAL  clock, reset      : IN      STD_LOGIC);
END  fetch_exam;
```

Question 2 Single Cycle Datapath Design (20 marks)

We have implemented 8-bit instruction set in our Prac. This question is a 16-bit instruction set which is an extension of 8-bit instruction set. The 16-bit instruction set is shown in Table 2.1.

Table. 2.1

Instruction Set		Instruction Format			
add	rd ← rs + rt	0	rd	rs	rt
sub	rd ← rs – rt	1	rd	rs	rt
and	rd ← rs AND rt	2	rd	rs	rt
or	rd ← rs OR rt	3	rd	rs	rt
ld	rd ← Mem[rs + Imm]	4	rd	rs	Imm
st	Mem[rs + Imm] ← rd	5	rd	rs	Imm
jn	If rs < rt then PC ← rs else PC←PC+1 end	6	rd	rs	rt
jz	If rs = rt then PC ← rs else PC←PC+1 end	7	rd	rs	rt

Each field is 4 bits wide

Assume that the memory address is 16 bits, each memory unit has 16 bits and two memories are used for instruction and data, respectively.

(2.1) [**5 marks**] Briefly write the steps to design a single-cycle datapath.

(2.2) [**12 marks**] Clearly draw single cycle datapath for the above instruction set and design control signals for the above instruction set. Use an 'X' to indicate a don't care.

(2.3) [**3 marks**] Given the functional unit latencies as shown in Table 2.2, calculate the minimum time in Table 2.3 to perform ***st*** instruction in the datapath you designed in (2.2).

Table 2.2 Unit lantency

Function Unit	Latency
Memory read or write	2 ns
ALU	2 ns
Register File read or write	1 ns

Table 2.3 Lantency of st instruction

Instruction	Minimum Time	Explain
st		

Question 3 Multicycle Datapath [20 marks]

The multicycle datapath of MIPS shown in Figure 3.1 will be used to answer the following question.

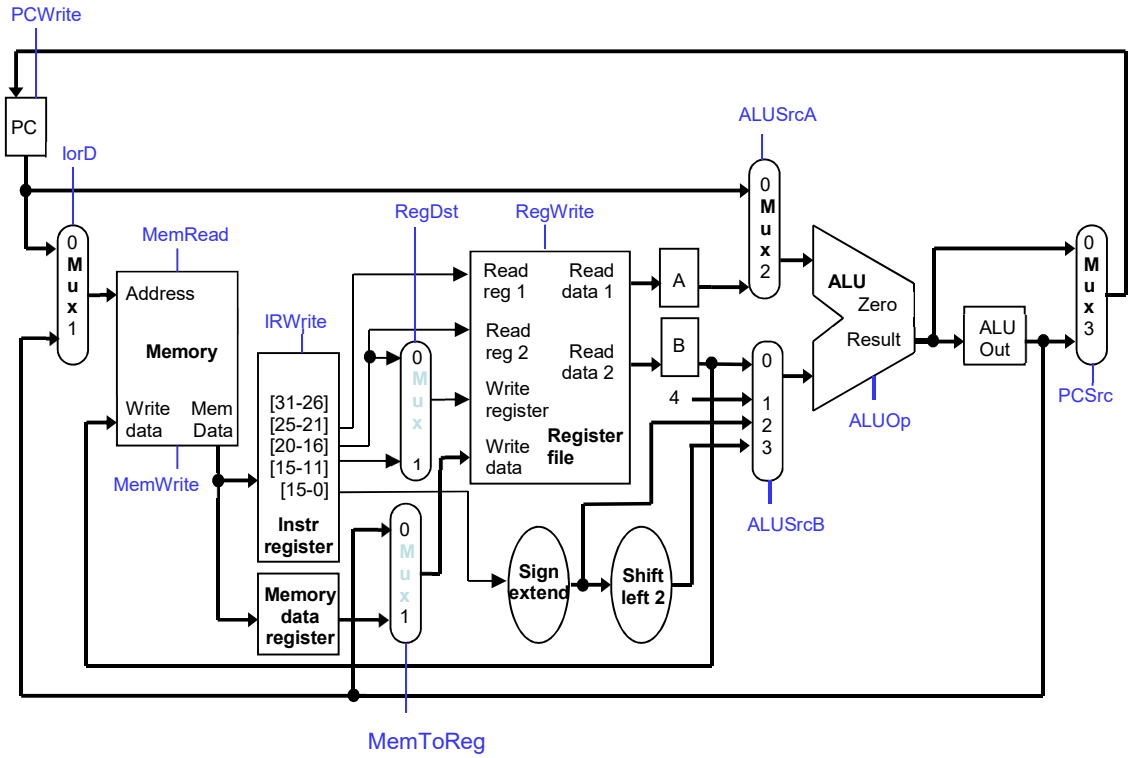

Figure 3.1 The Multicycle datapath of MIPS

Suppose we run add, addi, lw, sw, and beq in the datapath shown in Figure 3.1.

The instruction formats of add, addi, lw, sw, and beq are shown below.

Add format: add $rd, $rs, $rt R[$rd]<=R[$rs]+R[$rt];

31-26:opcode	25:21:rs	20:16:rt	11-15:rd	6-10:shamt	0-5:func

Addi format: addi $rs, $rt, imm R[$rt]<=R[$rs] +imm;

31-26:opcode	25:21:rs	20:16:rt	15:0:immediate

lw format: lw $rt, offset($rt), R[$rt]<=mem(R[$rs] +offset);

31-26:opcode	25:21:rs	20:16:rt	15:0:offset

sw format: sw $rt, offset($rt), mem(R[$rs] +offset)<=R[$rt];

31-26:opcode	25:21:rs	20:16:rt	15:0:offset

beq format: beq $rt, $rt, 25 , if R[$rs]==R[$rt] then PC<=PC+4+100;

31-26:opcode	25:21:rs	20:16:rt	15:0:25

Please placing an X in tables for Question (3.1) to (3.4).

(3.1) (**2 marks**) Indicate what clock cycle and which instruction(s) select(s) 1 for IorD control signal.

	Clock cycle 1	Clock cycle 2	Clock cycle 3	Clock cycle 4	Clock cycle 5
add					
addi					
lw					
sw					
beq					

(3.2) (**2 marks**) Indicate what clock cycle and which instruction(s) select(s) 1 for MemToReg control signal.

	Clock cycle 1	Clock cycle 2	Clock cycle 3	Clock cycle 4	Clock cycle 5
add					
addi					
lw					
sw					
beq					

(3.3) (**2 marks**) Indicate what clock cycle and which instruction(s) select(s) 1 for ALUSrcB control signal.

	Clock cycle 1	Clock cycle 2	Clock cycle 3	Clock cycle 4	Clock cycle 5
add					
addi					
lw					
sw					
beq					

(3.4) (**2 marks**) Indicate what clock cycle and which instruction(s) select(s) 1 for RegDst control signal.

	Clock cycle 1	Clock cycle 2	Clock cycle 3	Clock cycle 4	Clock cycle 5
add					
addi					
lw					
sw					
beq					

(3.5) (**12 marks**) Assume that the ALU can perform the **max2** operation (i.e., return the greater of two inputs):

alu_result = A_input if A_input >B_input; otherwise alu_result = B_input

ALU operation for this instruction is **max2**.

Consider extending the multicycle datapath of MIPS with the new instruction ***max3*** by using **max2**. The **max3** instruction has the same format as R-format, shown below.

Max3 $rs, $rt, $rd $rd=max($rs, $rt, $rd)

31-26:opcode	25:21:rs	20:16:rt	11-15:rd	6-10:shamt	0-5:func

(a) (**4 marks**) Show what changes of the datapath shown in Figure 3.1 are needed to support ***max3*** instruction and explain how ***max3*** instruction is executed in the modified datapath. You should not need to modify the main functional units (the memory, register file and ALU), but you can make any other changes or additions necessary. Try to keep your diagram neat!

(b) (**8 Marks**) Complete the finite state machine diagram shown in Figure 3.2 for the **max3** instruction. Control values not shown in each stage are assumed to be 0. Remember to account

for any control signals that you added or modified in the previous of the question! **Note that if (a) is wrong no mark is awarded for (b)**

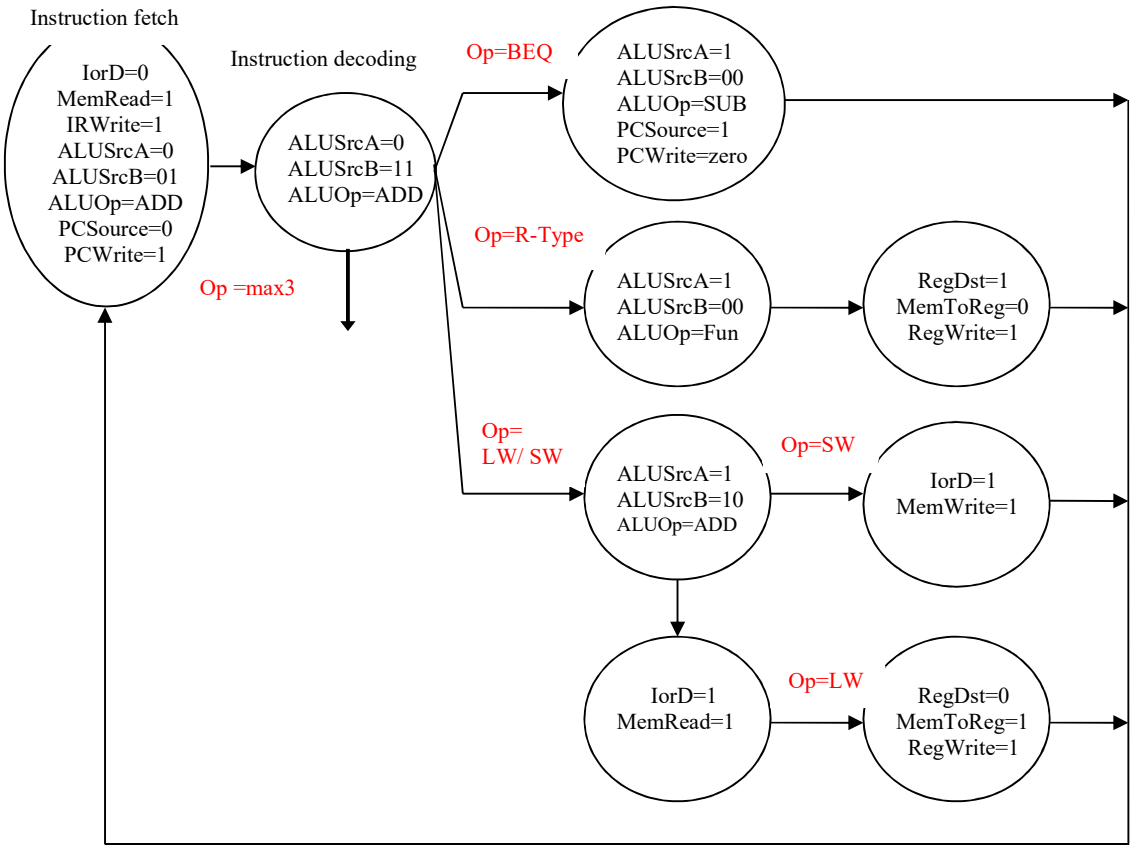

Figure. 3.2 The finite state machine of improved MIPS

Question 4 Pipelined Datapath [20 marks]

This question is based on the pipelined datapath shown in Figure 4.1. Consider the following piece of code.

```
Add $15, $18, $19
Add $15, $17, $15
Lw  $13, 4($15)
Add $12. $14. $16
```

(4.1) [**3 marks**] if we run the code in the pipelined datapath shown in Figure 4.1 what stage Add $15, $17, $15, Lw $13, 4($15), and Add $12, $14, $16 are at when Add $15, $18, $19 is at WB stage ?

(4.2) [**17 marks**] Fill in the correct datapath values for twenty-one question marks in Figure 4.1 when Add $15, $18, $19 is at WB stage.

Again, please :

- Write your answers directly on the diagram on the attached answer sheet.
- Show decimal values.
- Assume that registers initially contain their number plus 100_{10}: \$15 contains 115, \$18 contains 118, etc., and Memory[xx] = xx. All values are decimal.
- Add format: add \$Rd, \$Rs, \$Rt; Lw format: Lw \$Rt, offset(\$Rs).
- Write 'X' for any numbers that can not be determined.

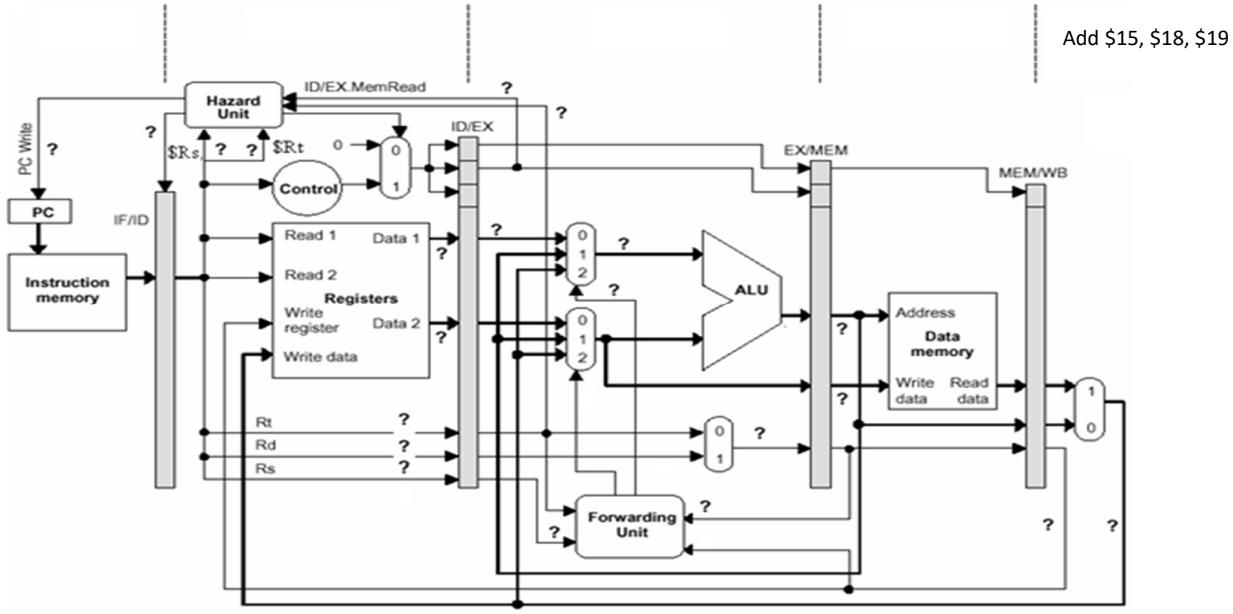

Figure 4.1 The pipelined datapath