Computer Design

Computer Science (2nd year B.Sc.)

MIPS datapath
(single-cycle)

Seminar #6

1. Pre-check

This section is designed as a check to help you determine whether you understand the concepts covered in class. Please answer "true/false" to the following questions and include an explanation.

1.1. The single-cycle datapath uses all hardware units for each instruction.
FALSE! All units are active in each cycle, but some outputs may be ignored by multiplexer control signals.

1.2. Single-cycle datapath stages can be executed in parallel to speed up the execution of a single instruction.
FALSE! Each stage depends on the values produced by the previous stage (for example, the instruction decoding step depends on the instruction fetched from memory).

1.3. Combinatorial circuits are only used in the instruction decoding stage.
FALSE! Other stages also use combinatorial circuits (multiplexers for path selection; ALU operations during the execution stage, ...).

2. Single-cycle CPU

2.1. In this tutorial, we'll work with the single-cycle datapath shown on the last page.

a) Fill each rounded box with the name of the datapath component and each square box with the name of the control signal.

b) Describe what happens at each step of the datapath.

Instruction **F**etch (**IF**)
Sends the address to the Instructions Memory (IMEM) and reads the content of IMEM at that address.

Instruction **D**ecode (**ID**)
Generates the control signals from the bits of the instruction. Generates the "Immediate" constant, and reads the contents of registers from the "Registers File".

EXecute (**EX**)
Executes ALU operations and performs comparisons for branching.

MEMory access (**MEM**)
(Reads from/Writes to) Data Memory (DMEM).

Write**b**ack (**WB**)
Transcribes the ALU output or the data returned from (DMEM) into a register.

2.2. Add the data wires and control signals for the circuit computing the next PC.

2.3. Implement the circuit logic that computes the next PC.

2.4. Fill in the following table with the control signals for the given instructions based on the datapath you completed. In the case where the nature of the control signal does not matter for the instruction under consideration, put an **X** in the corresponding box.

Instrs.	Control signals								
	Jump	RegDst	ImmSel	RegWEn	BSel	ALUSel	MemRW	WBSel	Branch
add	0	1	X	1	0	add	0	0	0
ori	0	0	0	1	1	or	0	0	0
lw	0	0	1	1	1	add	0	1	0
sw	0	X	1	0	1	add	1	X	0
beq	0	X	1	0	0	sub	0	X	1
j	1	X	X	0	X	XXXX	0	X	X

2.5. **A state circuit** is a component connected to the clock (designated by a triangle at the bottom of the circuit). In order for the component to function correctly, the input signal must be stable before each rising edge of the clock.
The critical path describes the minimum period required for clocking all state elements present in a circuit. The circuit cannot be clocked faster than this period because signal stability is no longer guaranteed for the individual components of the circuit.

For this exercise, assume the duration for each stage in the datapath is as follows (periods are given in picoseconds – 1 ps = $10^{-12} seconds$):

IF : 200 ps ; **ID** : 100 ps ; **EX** : 200 ps ; **MEM** : 200 ps ; **WB** : 100 ps

a) For each instruction listed in the table below, indicate with an X the datapath stages used and compute the minimum time required for its execution.

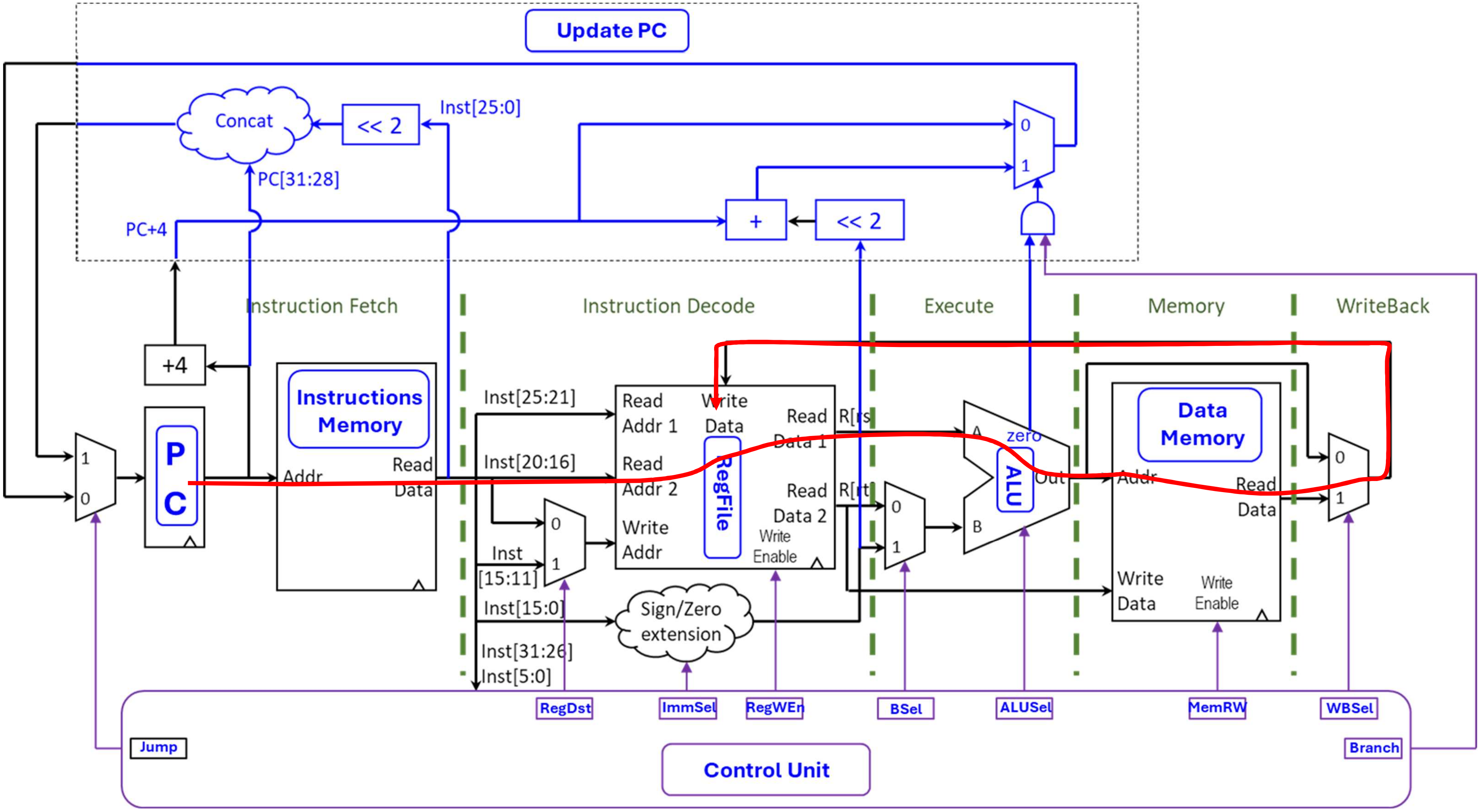
	IF	ID	EX	MEM	WB	Total duration
add	X	X	X		X	600 ps
ori	X	X	X		X	600 ps
lw	X	X	X	X	X	800 ps
sw	X	X	X	X		700 ps
beq	X	X	X			500 ps
j	X	X				300 ps

b) Which instruction(s) exercise the critical path?
The `lw` instruction! It uses all five stages of the datapath (see the drawn red line on the diagram).

c) Based on your previous answer, at what frequency we clock this single-cycle processor?
$\frac{1}{800\ ps} = \frac{1}{800 \times 10^{-12} sec} = 1{,}250{,}000{,}000\text{ Hz} = 1.25\text{ GHz}.$

d) Why single-cycle datapath is inefficient?
In a single-cycle data path, most stages of the circuit are often in an idle state (i.e., twiddling their thumbs doing nothing). Additionally, even though few instructions require the execution of all stages (i.e. the critical path), instructions that can be processed with fewer steps in the data path are clocked at the speed of the slowest instruction.

e) How can performance be improved? What is the purpose of pipelining?
Performance can be improved with pipelining. We will cover this in detail in the next tutorial.


Update PC
Concat
<< 2
Inst[25:0]
PC[31:28]
PC+4
+
<< 2
0
1
Instruction Fetch
Instruction Decode
Execute
Memory
WriteBack
+4
Instructions Memory
Addr
Read Data
P C
1
0
Inst[25:21]
Inst[20:16]
Inst [15:11]
Inst[15:0]
Inst[31:26]
Inst[5:0]
Read Addr 1
Read Addr 2
Write Addr
Write Data
RegFile
Read Data 1
Read Data 2
Write Enable
R[rs]
R[rt]
0
1
Sign/Zero extension
A
B
zero
ALU
Out
0
1
Data Memory
Addr
Read Data
Write Data
Write Enable
0
1
RegDst
ImmSel
RegWEn
BSel
ALUSel
MemRW
WBSel
Jump
Branch
Control Unit