

Midterm Exam

الشعبة:

رقم التسجيل:

KEY

الاسم:

Exam Instructions:**Duration:** 60 minutes**Materials Allowed:** Open book and notes. No electronic devices permitted.**Instructions:** Please answer all six problems in the spaces provided. Limit your responses to the space allocated for each problem. Each problem is worth 5 marks. Ensure clarity and conciseness in your answers.

P1. Below are five **True/False questions** designed to assess your understanding of recent technological trends in computer architecture. Carefully read each statement and indicate whether it is True or False based on your knowledge of the current trends in computer technology.

No.	Statement	True/False
1	The number of transistors per chip has generally doubled approximately every two years, aligning with Moore's Law.	True
2	The performance of a single core has continued to significantly improve year over year due to increases in processor clock rate alone.	False
3	Processor clock rates have consistently increased over the past decade, leading to faster overall computer speeds.	False
4	Typical processor power consumption has decreased as the number of transistors on chips has increased, due to enhancements in transistor technology.	False
5	The number of cores per chip has remained relatively constant over the past five years as manufacturers focus more on improving the efficiency of single-core performance.	False

P2. You are given a single-cycle implementation of a processor that operates at a clock rate of 100 MHz. This processor is being considered for conversion to an 8-stage pipelined implementation, which can operate at a higher clock rate of 500 MHz. However, the pipelined implementation is not perfect and suffers from an average of one stall cycle every 4 cycles due to data and control hazards. **Calculate the peak speedup** that could be achieved with this conversion. Assume that the execution time for a single instruction in the single-cycle processor equals the cycle time, and that the pipelined processor's effective cycle time includes the impact of stall cycles.

Hint: Start by computing the cycle times for the single-cycle and pipelined implementations, then determining the effective cycle time per instruction for the pipelined implementation by accounting for stall cycles, and calculating the execution time for a given large number of instructions in both implementations.

Solution:

Step 1: Compute the Cycle Time for Both Implementations

- **Single-Cycle Implementation:**
 - Clock rate = 100 MHz
 - Cycle time = $1 / 100 \text{ MHz} = 10 \text{ nanoseconds}$
- **Pipelined Implementation:**
 - Clock rate = 500 MHz
 - Cycle time = $1 / 500 \text{ MHz} = 2 \text{ nanoseconds}$

Step 2: Determine the Effective Cycle Time with Stall Cycles

- Given that there is one stall cycle every 4 cycles:
 - Effective cycles per instruction = 1 normal cycle + 0.25 stall cycles per cycle (because 1 stall every 4 cycles) = 1.25 cycles per instruction
- Effective cycle time = $1.25 \times 2 \text{ nanoseconds} = 2.5 \text{ nanoseconds}$

Step 3: Calculate the Execution Time for N Instructions

- **Single-Cycle Implementation:**
 - Time = $N \times 10 \text{ nanoseconds}$
- **Pipelined Implementation:**
 - Time = $7 + N \times 2.5 \text{ nanoseconds} \approx N \times 2.5 \text{ nanoseconds}$ (for large N)

Step 4: Calculate the Speedup

- The speedup S can be calculated using the formula:

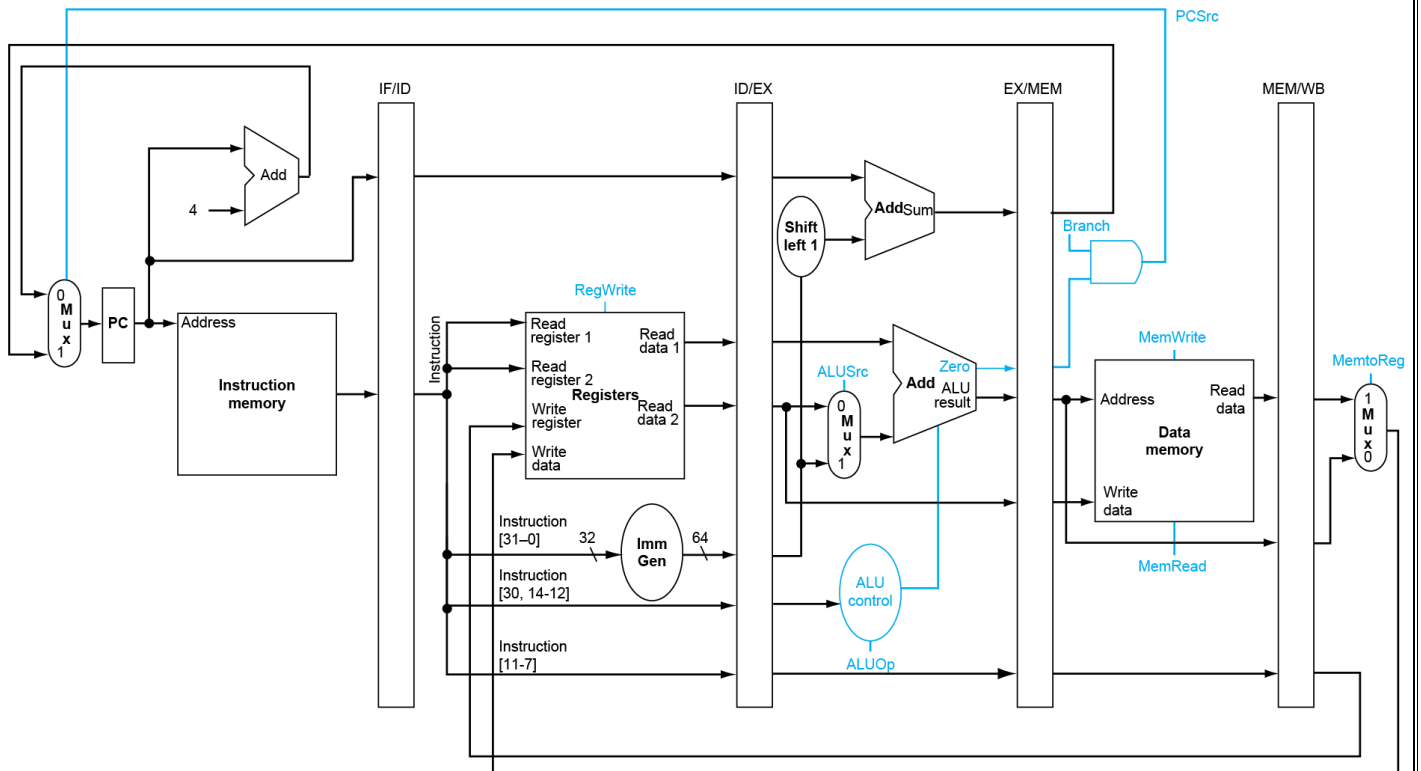
$$S = T_{\text{single}} / T_{\text{pipeline}}$$

$$S = N \times 10 \text{ ns} / N \times 2.5 \text{ ns}$$

$$S = 4$$

P3. Assume that the following five instructions are executed by the RISC-V pipeline shown below. Assume that this pipeline has the needed forwarding paths to solve data hazards. In the table below, **specify** the values of the shown five **fields/signals** when the first instruction has reached the end of the write-back stage. Note: All numbers shown are in decimal.

Address	Instruction
100020	add x10, x0, x0
100024	sd x10, 8(x13)
100028	sub x14, x3, x3
100032	subi x11, x11, 20
100036	ld x15, 0(x11)



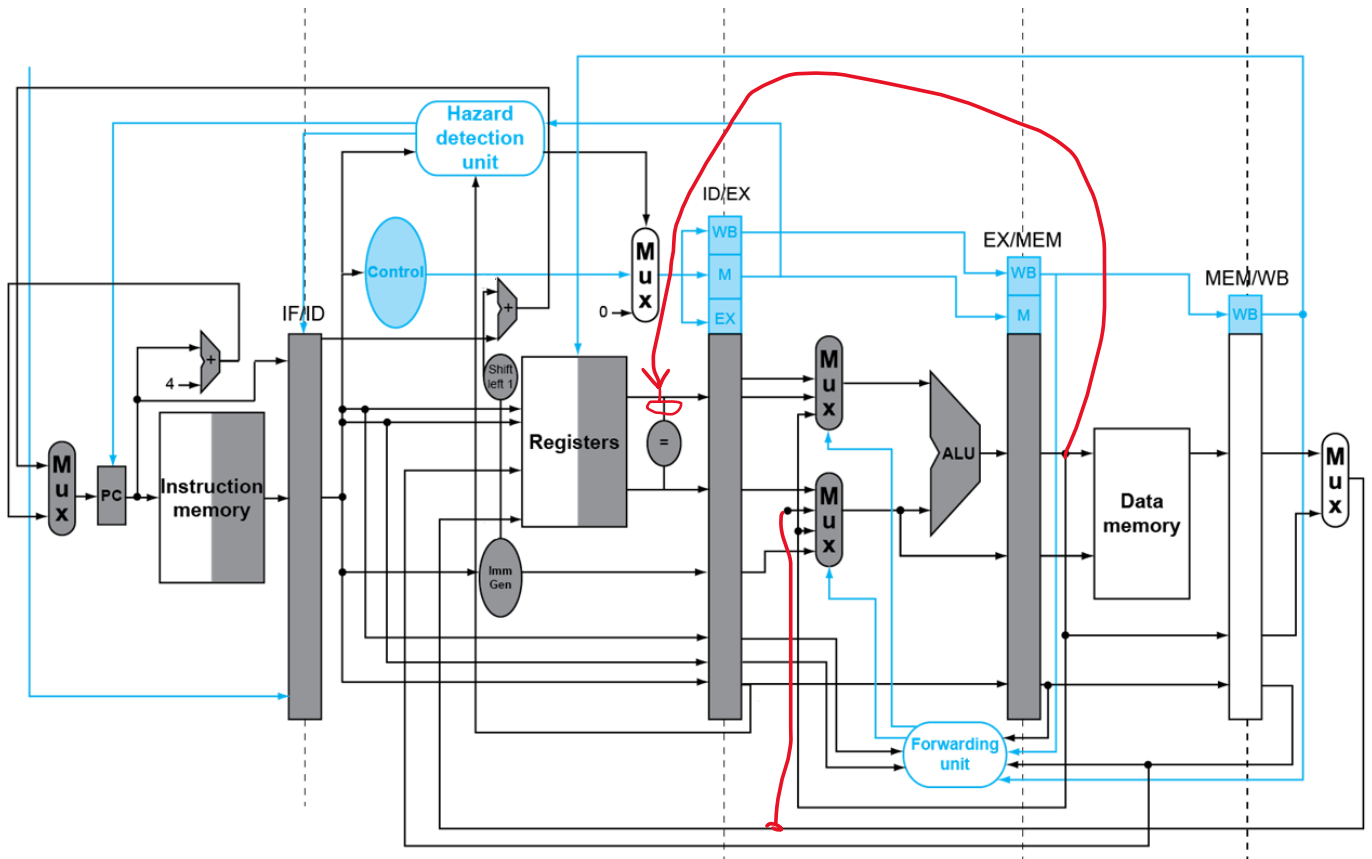
Field/Signal	Value (in decimal)
The input of the program counter	100040
The output of the immediate generator	20
ID/EX.RegisterRd	14
The value of the zero status in the execute stage	1
EX/MEM.MemWrite	1
The output of the multiplexer of the write-back stage	0

P4. In this problem, we consider a 5-stage pipelined processor that has been discussed in class. This processor handles data hazards using stalls and forwarding mechanisms, equipped with all necessary forwarding paths to resolve data hazards that can be addressed through forwarding. It is also noted that the branch instruction resolution occurs in the decode stage. Using the **multi-cycle pipeline diagram** provided below, demonstrate how this processor executes the given sequence of instructions. Clearly indicate any occurrences of forwarding by drawing **arrows** between the relevant pipeline stages where forwarding takes place.

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
add x4, x2, x3	F	D	E	M	W										
ld x5, 16(x4)		F	D	E	M	W									
sub x7, x5, x6			F	D	D	E	M	W							
bne x7, x8, L22				F	F	D	D	E	M	W					

P5. Assume that the following pipeline processor resolves branch instructions in the decode stage. On the diagram below draw lines to indicate any needed **additional forwarding paths** to execute the following instruction sequence without stalls. Only draw the forwarding busses; no need to draw the forwarding control circuits.

```
ld    x10, 40(x1)
subi  x11, x11, 16
add   x14, x11, x10
bne   x11, x10, 20
```



The solution is in red.

P6. Assume that the following code sequence is executed by a dynamic **triple-issue** pipelined processor. This processor uses reservation stations, reorder buffer, and three common data buses to execute the instructions out of order. The fetch, issue, execute, write, and commit stages each take one cycle. However, the memory latency is 2 cycles (1 cycle for address calculation and 1 cycle for data memory access). The processor has one address calculation unit, one memory access unit, and two integer ALU units. Using the **multi-cycle pipeline diagram** below, specify the execution of these instructions in this processor pipeline.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ld x1, 0(x2)	F	I	A	M	W	C								
ld x3, 8(x2)	F	I		A	M	W	C							
add x4, x1, x3	F	I					E	W	C					
sd x4, 16(x5)		F	I		A				C					
add x8, x6, x7		F	I	E	W				C					
sub x10, x8, x9		F	I			E	W			C				

<Good Luck>