

Sample questions and solutions

List the subtasks required to process any basic instruction.

The following subtasks can process any instruction supporting any addressing mode:

IF : Instruction fetch- read machine code of instruction and save to Instruction register

ID and Register Read: Instruction Decode and Register read- the machine code is decoded at the control unit. The control unit will generate control signals in a sequence. The generated control signals are used to select registers, read register, select ALU operation etc.

EXE: ALU operation

MEM: Memory operation- read data from memory and load to a register or save the content of a register to memory

WB: Write Back- save result to a register

List the sequence of subtasks required to process ALU instructions of any RISC processor.

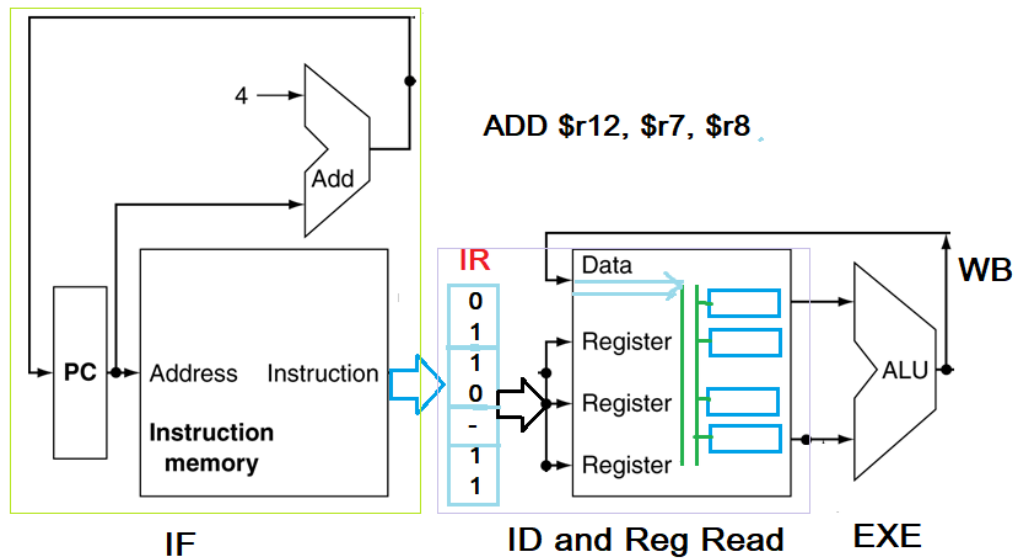
Since RISC processor supports only few addressing modes and ALU instructions are designed to support Register and Immediate mode, the list the sequence of subtasks required to process ALU instructions of any RISC processor is as follows:

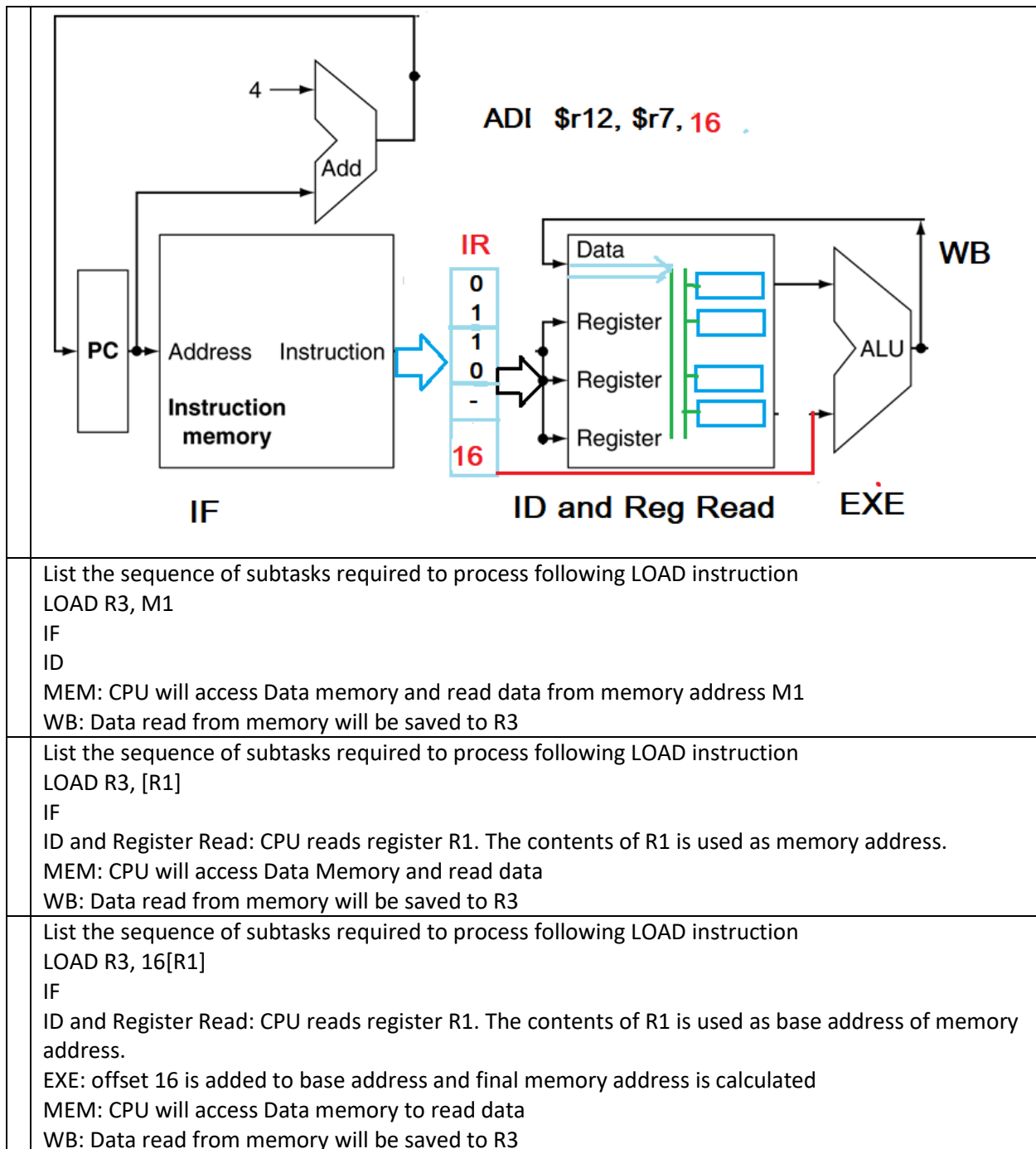
IF

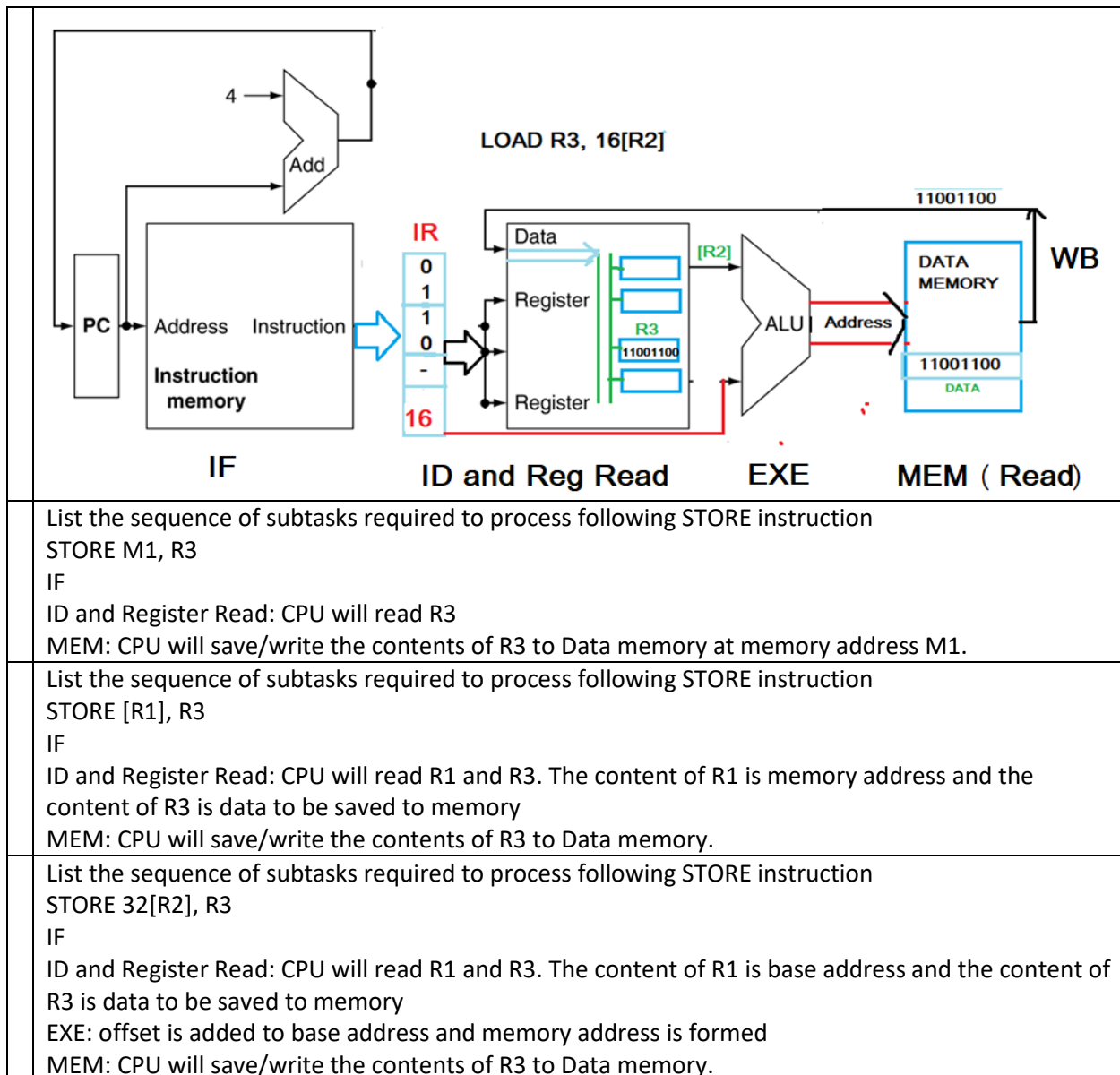
ID and Register Read

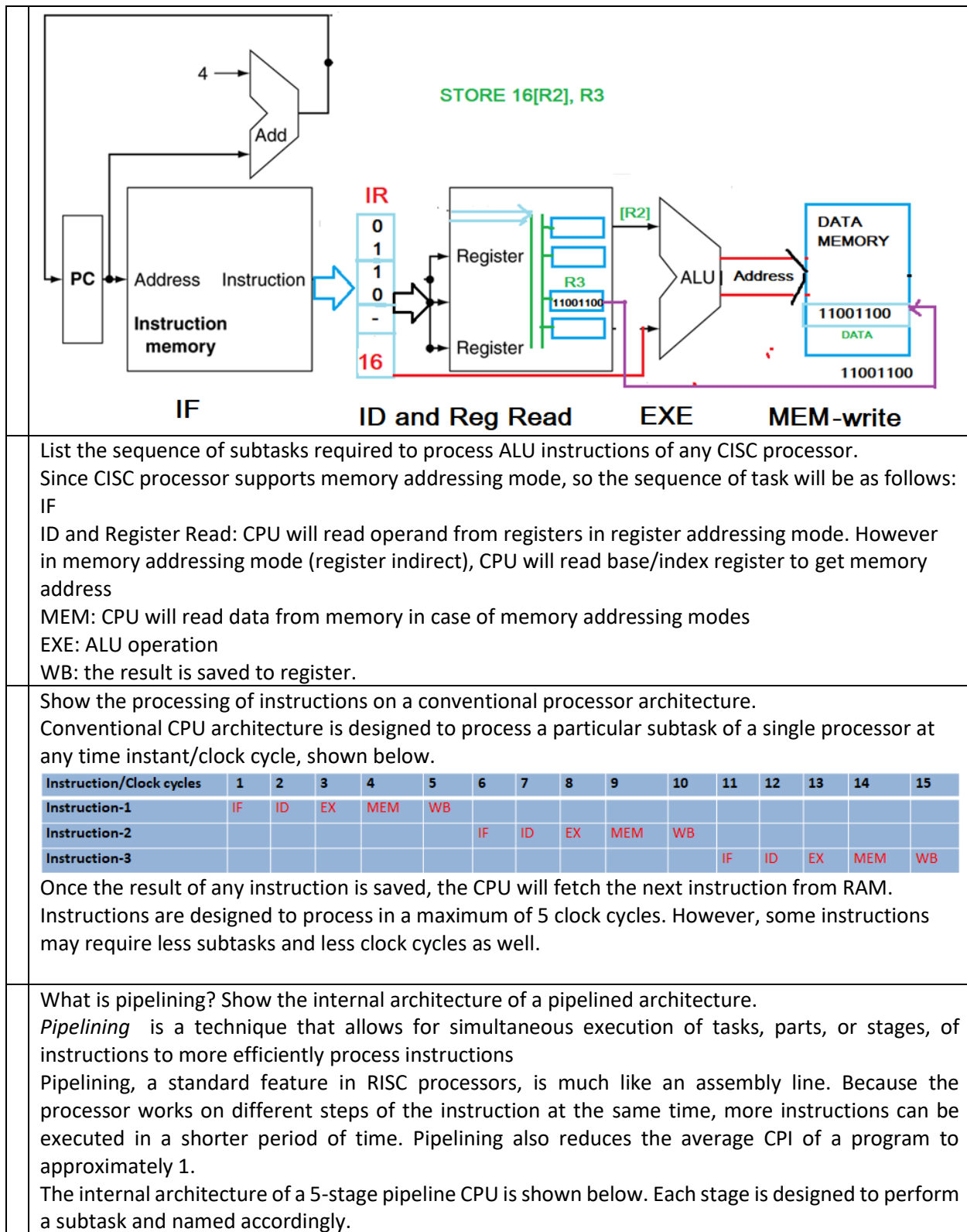
EXE

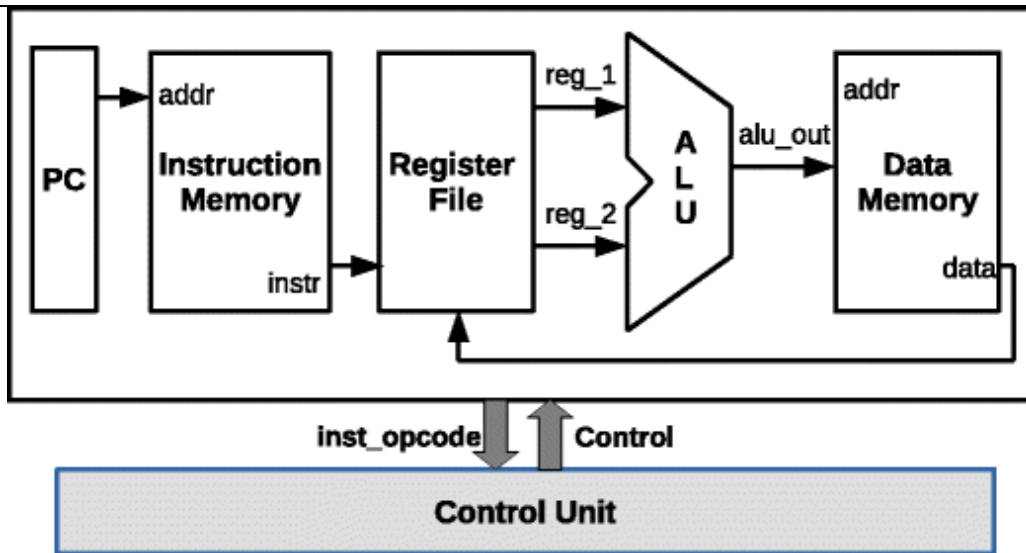
WB









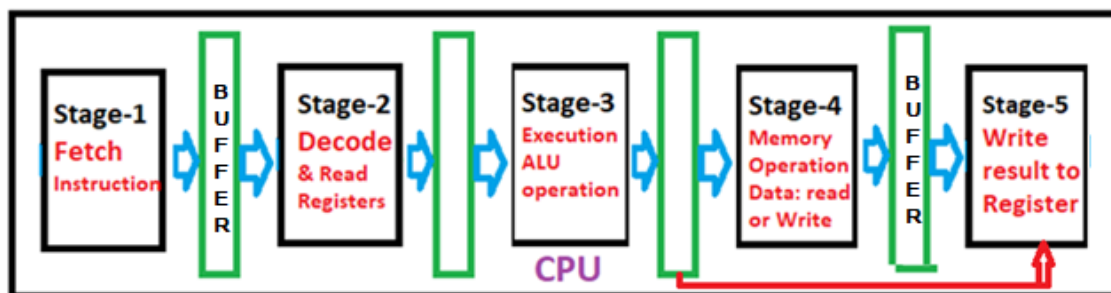


At the beginning of any instruction processing, it is required to read the machine code of instruction from memory. So the first task is called instruction fetch and first stage is called IF unit.

The machine code is decoded next at the control unit, it means different bits of machine codes are used to activate ALU and other functional units of CPU and other functional units of computer as required to process a particular instruction being processed. Depending on addressing modes of instructions, registers are read to get operand or memory address for data to be used in the operation. CPUs are designed to process some tasks at the same time if there is no resource conflicts or constraints. Subtasks: instruction decode and register read are designed to implement at the same clock cycle.

ALU operation will be executed next in case of RISC processor since ALU instructions support register and immediate addressing modes only.

Memory access will follow next, especially to implement LOAD & STORE instructions. CPU will read data in case of LOAD instruction and save data to data memory. It is to be noted that two separate memories; namely Instruction Memory and Data Memory are used in pipeline architecture to allow simultaneous execution of IF and MEM subtasks on different instructions at the same clock cycle.

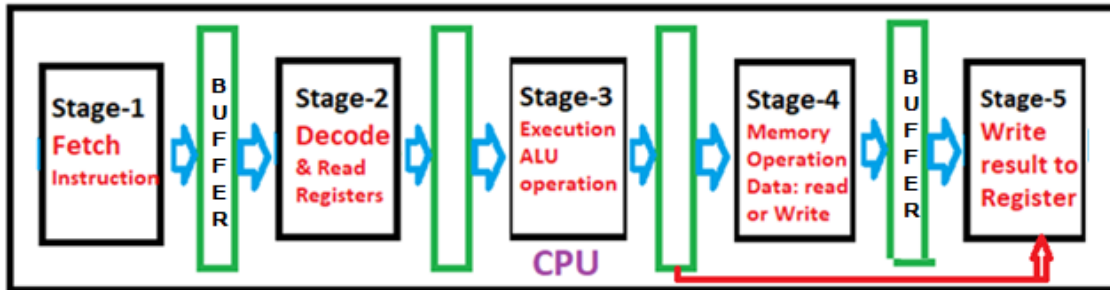


It is to be noted that buffers are used in between stages to hold the partially processed instructions, partial results, contents of registers etc to be sent to following sections as happens in an assembly line. Buffers also allow all stages work independently and simultaneously on different instructions at the same time.

What is a buffer? Why it is used in pipeline architecture?

Buffers are temporary electronic storage, used in between stages to hold the partially processed instructions, partial results, contents of registers etc to be sent to following sections as happens in an assembly line. Buffers also allow all stages work independently and simultaneously on different instructions at the same time.

How does the pipeline architecture process instructions?
Here a 5-stage pipeline architecture is considered, shown below



Processing of a number of instructions on above 5-stage pipeline architecture is shown below. Instructions are listed at the leftmost column. Numbers on first row indicate clock cycles.

Instruction \ clock cycles	1	2	3	4	5	6	7	8
Instruction-1	IF	ID	EX	MA	WB			
Instruction-2		IF	ID	EX	MA	WB		
Instruction-3			IF	ID	EX	MA	WB	
Instruction-4				IF	ID	EX	MA	
Instruction-5					IF	ID	EX	
Instruction-6						IF	ID	
Instruction-7							IF	

At clock cycle-1, fetch unit just fetches instruction-1. At clock cycle-2, fetch unit fetches new instruction (instruction-2) whereas instruction-1 will be decoded. So at clock cycle-2, two subtasks on two different instruction will be implemented simultaneously. The CPU will continue to process subtasks and at clock cycle-5, the pipeline will be full. It means that all 5-stages are simultaneously working on 5 instructions: fetch unit fetches instruction-5, instruction-4 is decoded, ALU operation of instruction-3 is executed, memory is accessed for instruction-2 and result of instruction-1 is saved. For the following clock cycles, the pipeline is expected to remain full and will five instructions simultaneously if the instructions are independent and there is no resource conflicts in pipeline architecture. It is to be noted that, each instruction still requires 5 clock cycles but due to overlapped in execution, number of clock cycles for a program (many instructions) is significantly reduced.

For example, to process 100 instruction, number of clock cycles required:

$5 + (100-1) \times 1 = 104$ clock cycle.

To process 100 instructions on a non-pipelined architecture, it takes: $100 \times 5 = 500$ clock cycles.

List the differences between a conventional and pipelined architecture.

Conventional CPU architecture is designed to process only one instruction at a time whereas a pipeline architecture can process multiple instructions at a time by overlapped in instruction execution.

Processing of instructions on a 5-stage pipeline structure

Instruction \ clock cycles	1	2	3	4	5	6	7	8
Instruction-1	IF	ID	EX	MA	WB			
Instruction-2		IF	ID	EX	MA	WB		
Instruction-3			IF	ID	EX	MA	WB	
Instruction-4				IF	ID	EX	MA	
Instruction-5					IF	ID	EX	
Instruction-6						IF	ID	
Instruction-7							IF	

Processing on instructions on a conventional architecture

Instruction/Clock cycles	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Instruction-1	IF	ID	EX	MEM	WB										
Instruction-2						IF	ID	EX	MEM	WB					
Instruction-3											IF	ID	EX	MEM	WB

Average CPI of a conventional architecture is approximately 4.5 whereas average CPI of a pipeline architecture is approximately 1.

More registers and buffers are required in designing pipeline architecture.

What does pipeline improve
 Processing time of a single instruction
 Processing time of a program
 Throughput of a CPU
 Explain

Calculate Average CPI of a pipelined architecture.
 Consider a program containing 1000 instructions.
 To process 1000 instruction on a 5-stage pipeline processor, total number of clock cycles required:
 $5 + (1000-1) \times 1 = 1004$ clock cycles.
 Average CPI = $1004/1000 = 1$ (approximately)

Compare the runtime of CISC and RISC processor for any program.
 Consider a program containing 1000 instructions. Both the processors use 1MHz clock.
 Clock period = 1 micro second
 To process 1000 instruction on a 5-stage pipeline processor, total number of clock cycles required:
 $5 + (1000-1) \times 1 = 1004$ clock cycles.
 Runtime (pipeline architecture) = 1004×1 microseconds = 1004 microseconds
 To process 1000 instruction on a conventional processor, total number of clock cycles required:
 $1000 \times 5 = 5000$ clock cycles.
 Runtime (conventional architecture) = 5000×1 microseconds = 5000 microseconds
 Speedup factor = run time of conventional architecture/ Runtime (pipeline architecture)

<div>= 5000/1004 = 5 (approximately) So a 5-stage pipeline architecture runs a program 5 times faster compared to a conventional architecture.</div>	
Compare addressing modes for ALU instructions of RISC and CISC processors.	
CISC	RISC
ALU instructions support Register addressing mode Immediate addressing mode Memory addressing mode	ALU instructions support Register addressing mode Immediate addressing mode
Compare the design of control unit of RISC and CISC processors.	
CISC	RISC
Microprogrammed control unit CISC instruction set includes a large number of instructions supporting many addressing modes. Length of machine codes is variable and it depends on addressing modes. So control unit is designed based on software, called microprogrammed control unit. Inside the control unit, there is a control memory and it holds micro programs correspond to instructions. Each microprogram contains a number of microinstructions. Each microinstruction contains information on control signals to be activated at a particular clock cycle.	Hardware control unit. RISC instruction set includes relatively fewer and simple instructions supporting a few addressing modes. All instructions are of same length. So the control unit is designed using logic gates and instruction decoding is very fast.
Compare the size of machine codes of RISC and CISC processors. Size of machine code is fixed in RISC processor. In many RISC processor, the length of machine code is 32 bits. It means any instruction when converted into binary format, 32 bits are used. For CISC, the length of instruction varies from 1 byte to 8/10 bytes since it support complex instructions and different types of memory addressing modes.	
Calculate the speed up factor of a pipeline architecture compared to a conventional architecture. Consider a program containing 1000 instructions. Both the processors use 1MHz clock. Clock period = 1 micro second To process 1000 instruction on a 5-stage pipeline processor, total number of clock cycles required: 5 + (1000-1)x1 = 1004 clock cycles. Runtime (pipeline architecture) = 1004 x 1 microseconds = 1004 microseconds To process 1000 instruction on a conventional processor, total number of clock cycles required: 1000 x 5 = 5000 clock cycles. Runtime (conventional architecture) = 5000 x 1 microseconds = 5000 microseconds Speedup factor = run time of conventional architecture/ Runtime (pipeline architecture) = 5000/1004 = 5 (approximately) So a 5-stage pipeline architecture runs a program 5 times faster compared to a conventional architecture. For a k-stage pipeline architecture, the ideal speedup will be k.	
List the factors that decide the speed up factor of a pipeline architecture.	

	<p>The following factors decide the speed up factor of a pipeline architecture:</p> <p>The number of pipeline stages.</p> <p>Whether the instructions are independent or dependent on other instructions in a program.</p> <p>Whether there is any resource conflicts in the pipeline architecture.</p> <p>Number/percentage of branch instructions.</p> <p>The design of compiler</p>
	<p>Which of the following design consideration is better</p> <p>Increasing the number of pipeline stages without reducing clock period</p> <p>Reducing the clock period without increasing pipeline stages</p>
	<p>How many instructions a pipeline processor can process at the first clock cycle?</p> <p>The CPU can process only one instruction at clock cycle-1.</p>
	<p>How many instructions a pipeline processor can process when it is full?</p> <p>For a 5-stage pipeline architecture, CPU can process 5 instructions when pipeline if full.</p>