PART - A

END SEM

Ans. 1.

For Non-pipelined circuit

Tpmax = 5ms

Tclk = 5ms.

Latency for non - pipelined circuit = 5ms

Throughput of Non-pipelined circuit = 1/(Tclk or Tpmax) = 200

For pipelined circuit

Tpmax = 1ms

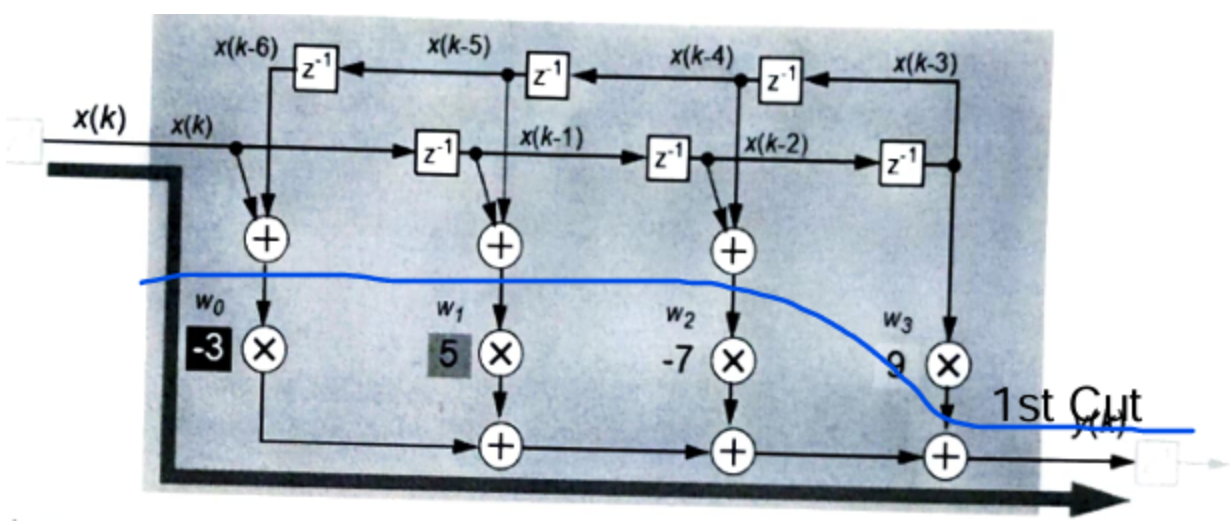
Tclk = 1ms

Latency for pipelined circuit = 5ms

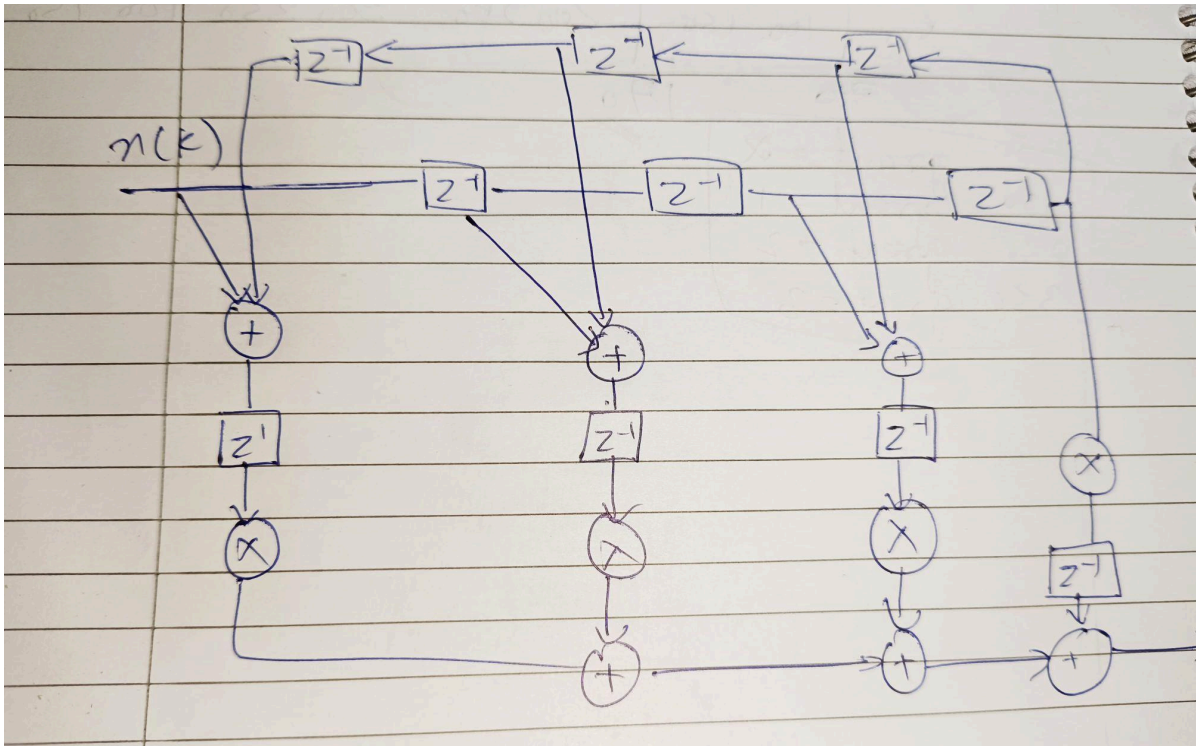
Throughput of pipelined circuit = 1/(Tclk or Tpmax) = 1000

Stages of Cut-set Retiming

1st Cut

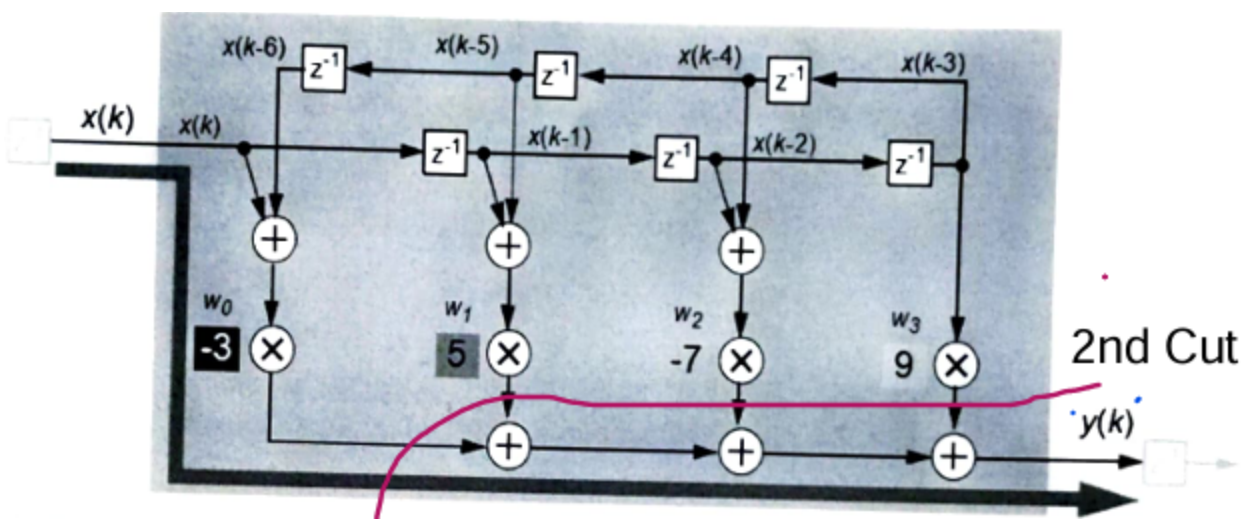


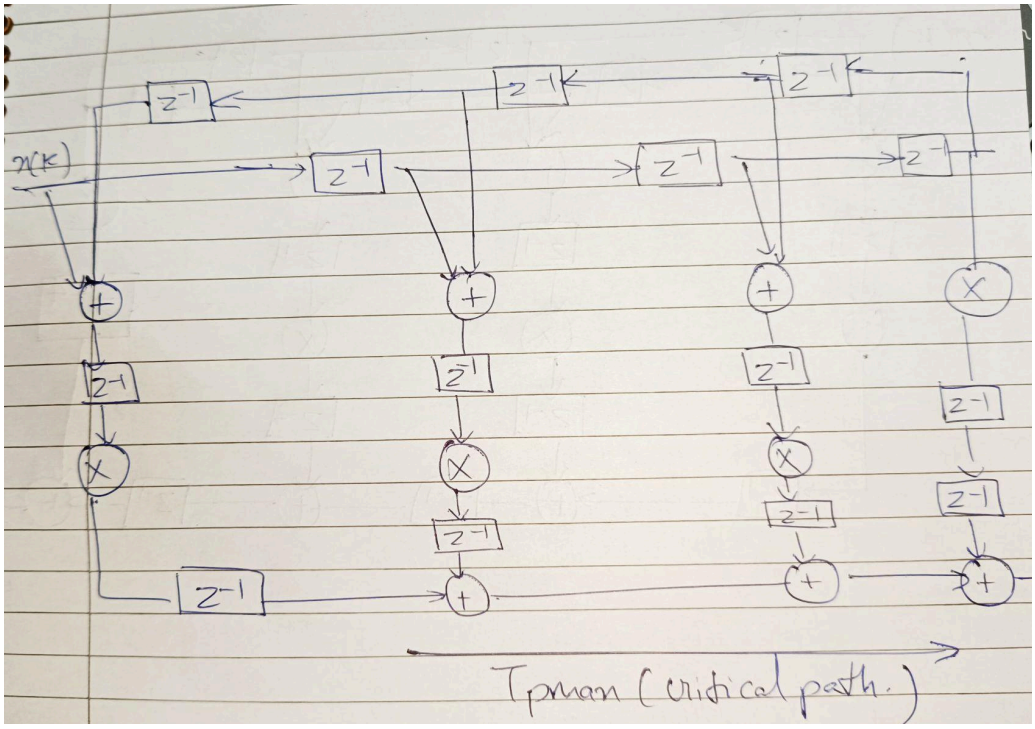
Consider outward as outbound and Weight it as Z-1



Now Tpmax = 4ms (critical Path)

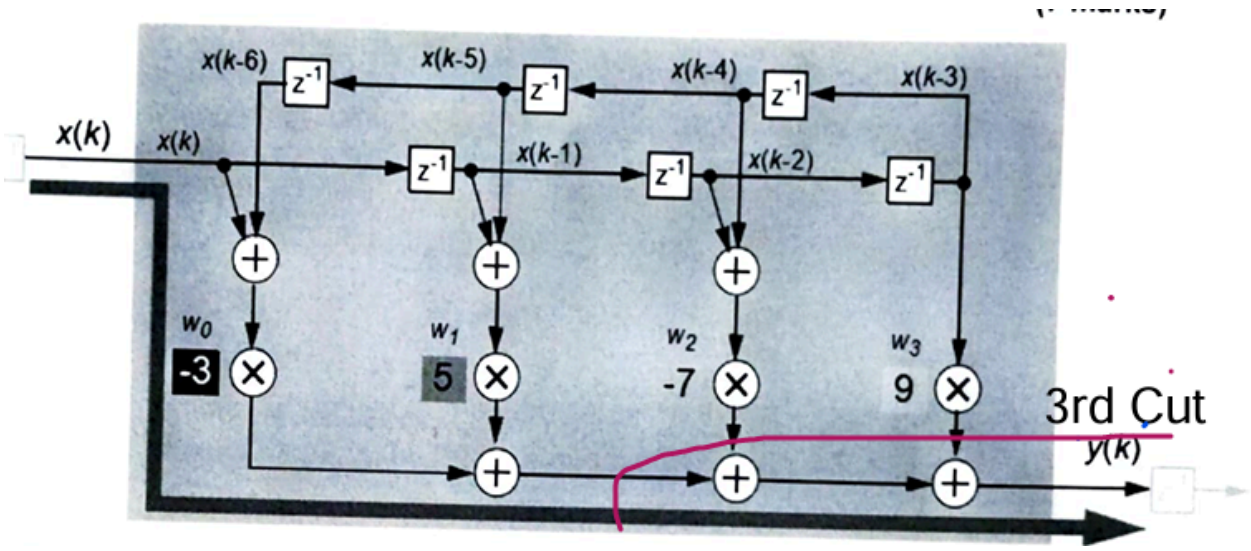
Now 2nd Cut

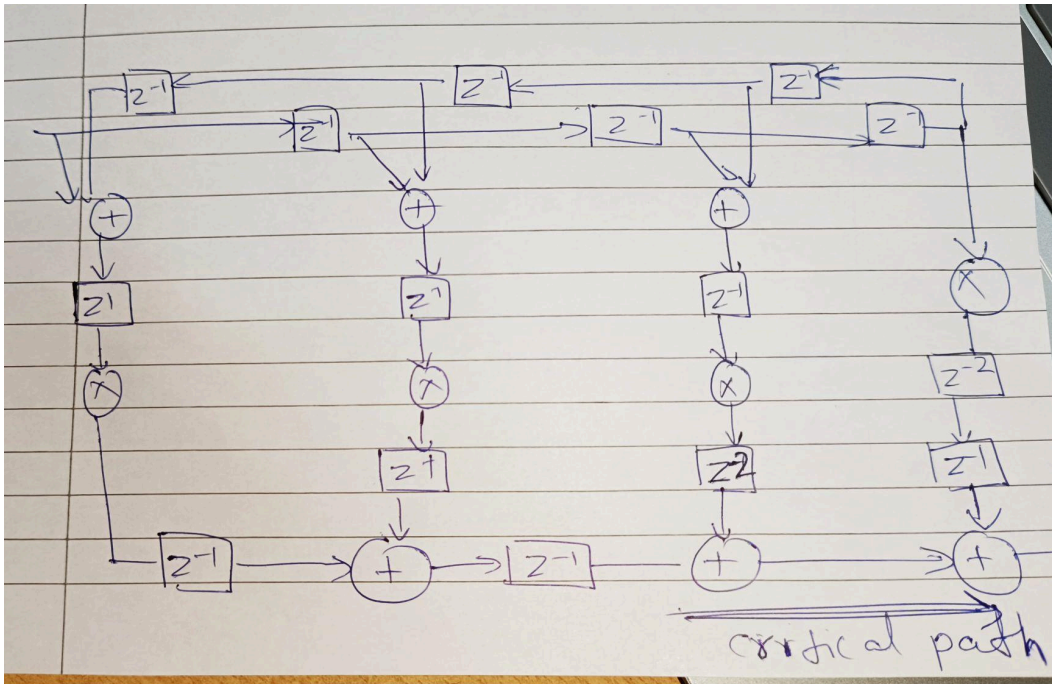




Now After 2nd cut Tpmax = 3ms(Critical Path)

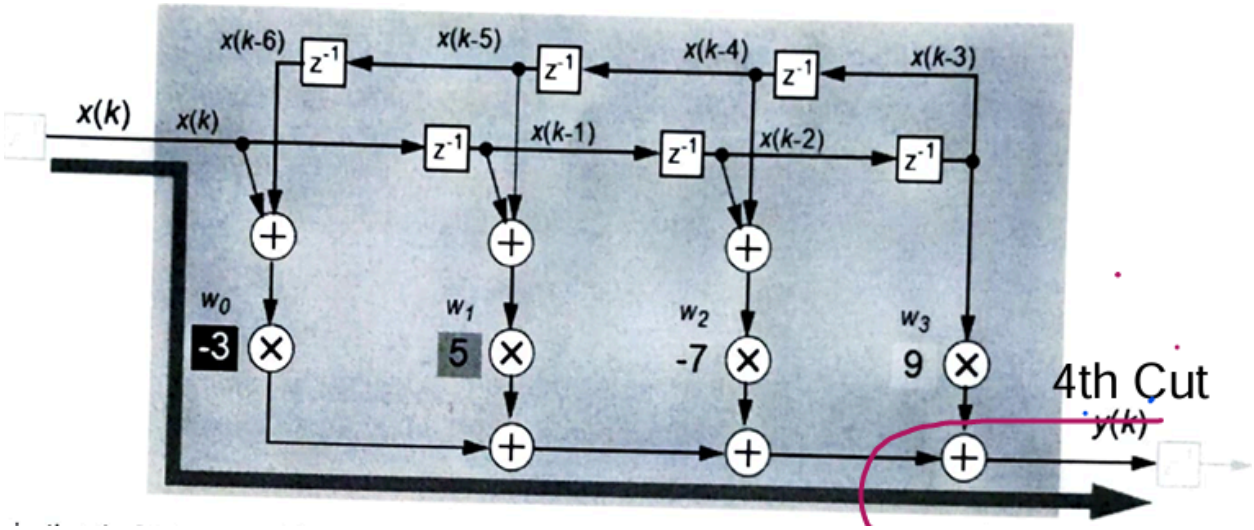
Now 3rd Cut

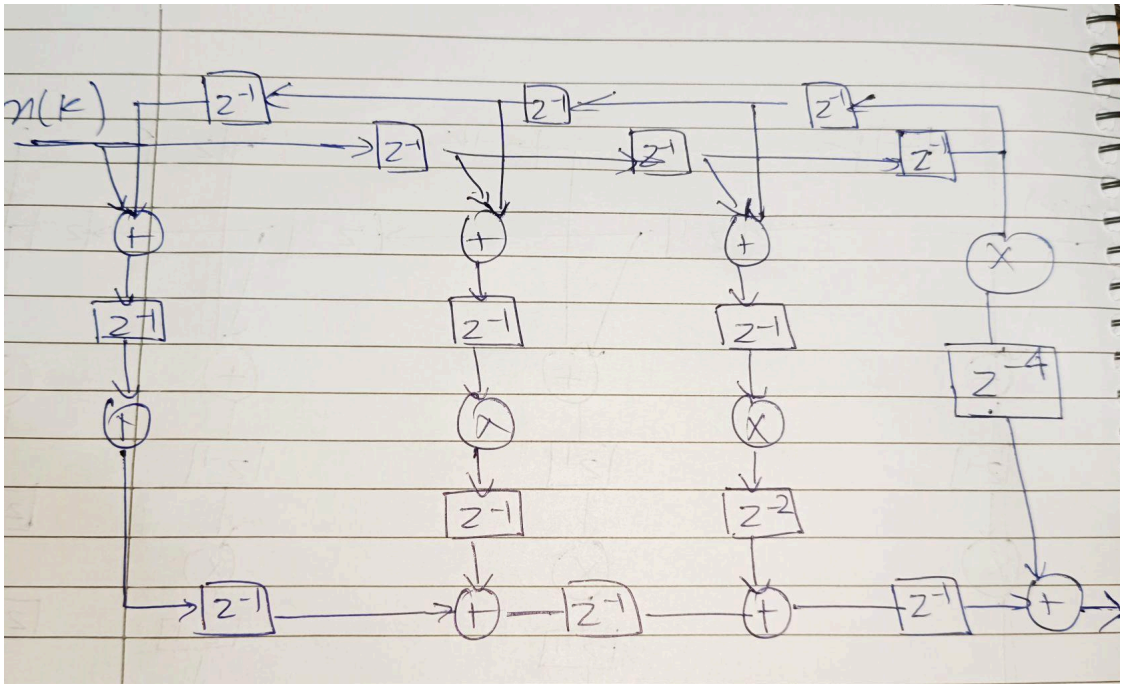




Now After 3nd cut Tpmax = 2ms(Critical Path)

Now 4th Cut

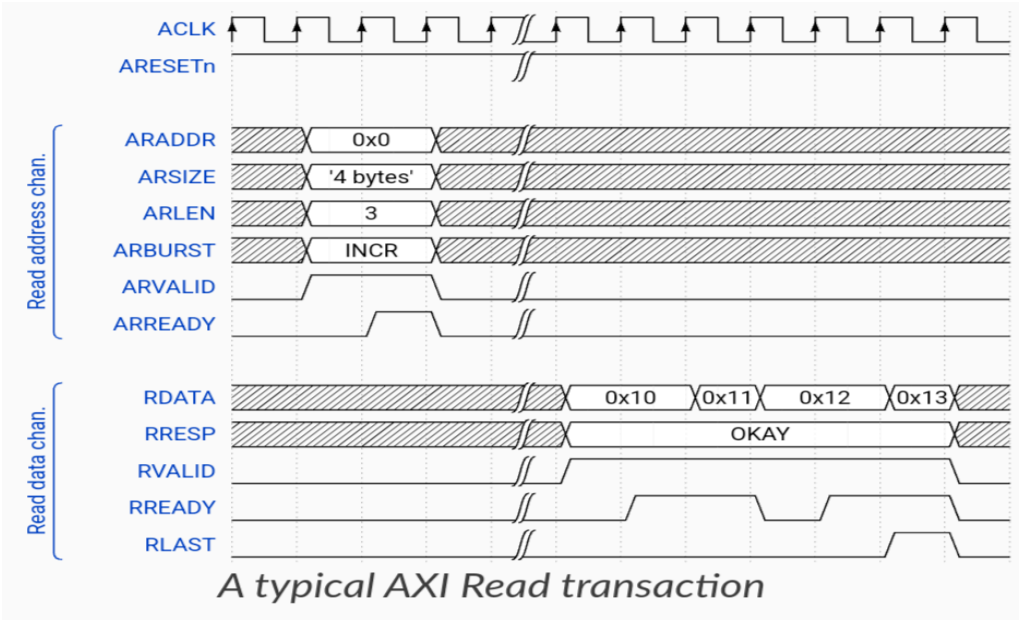




Now After 4th cut Tpmax = 1ms(Critical Path)

Ans. 2.

1. AxLEN describes the length of the transaction in the number of transfers. For AXI4, AxLEN[7:0] has 8 bits, which specifies a range of 1-256 data transfers in a transaction.
2. AxSize[2:0] describes the maximum number of bytes to transfer in each data transfer. Three bits of encoding indicate 1, 2, 4, 8, 16, 32, 64, or 128 bytes per transfer.
3. STRB: In the AXI protocol, the strobe signal indicates which bytes of the data signal are valid during a transaction. The strb signal is a bitmask where each bit corresponds to a byte in the data signal, indicating whether the byte is valid (1) or should be ignored (0). The number of bits in the strb signal depends on the data width of the bus. STRB signal is only available in write transaction (reference lecture 16 slide 28)



QUIZ 4

Ans.1. **Difference Between Pipelining and Parallel Processing**

Both **pipelining** and **parallel processing** are techniques to improve the performance of computing systems by increasing throughput. However, they differ in execution strategy, hardware requirements, and performance benefits.

Comparison Table **[2M]**

Aspect	Pipelining	Parallel Processing
Definition	Divides a task into sequential stages, overlapping execution.	Divides a task into independent sub-tasks, executing them simultaneously.
Execution	Stages run in sequence, but multiple tasks occupy different stages concurrently.	Independent tasks run at the same time on different hardwares.

Latency	No reduction (task still takes multiple cycles).	Reduced (tasks finish faster due to simultaneous execution).
Throughput (number of tasks per unit time)	Increases by overlapping stages.	Increases by executing multiple tasks concurrently.
Hardware Cost	Low (single unit reused for stages).	High (multiple hardware units needed).

Let's compare **pipelining** and **parallel processing** using two `ADD` instructions and analyze latency, throughput, and hardware requirements. We'll assume a **5-stage pipeline** with the following stages:

1. **Fetch (F)**
2. **Decode (D)**
3. **Execute (E)**
4. **Memory Access (M)**
5. **Writeback (W)**

Two `ADD` Instructions

- **Instruction 1**: `ADD R1, R2, R3` (R1 = R2 + R3)
- **Instruction 2**: `ADD R4, R5, R6` (R4 = R5 + R6)

---

1. Pipelining Example [1M]

In pipelining, the execution of instructions is divided into stages that overlap. Each stage takes **1 clock cycle (cc)**.

Clock Cycle	Instruction 1	Instruction 2
1	F	
2	D	F
3	E	D

4	M	E
5	W	M
6		W

Analysis

1. **Latency**:
   - Time to complete both the instructions = **6 cycles**
2. **Throughput**:
   - After the pipeline fills, a new instruction completes every **1 cycle**.
3. **Hardware Units**:
   - Single hardware unit reused for each stage (e.g., one ALU for the `Execute` stage).

2. Parallel Processing Example [1M]

In parallel processing, multiple instructions are executed simultaneously on separate hardware units (e.g., two ALUs). Each instruction goes through all 5 stages independently.

Clock Cycle	Instruction 1	Instruction 2
1	F	F
2	D	D
3	E	E
4	M	M
5	W	W

Analysis

1. **Latency**:
   - Time to complete both instructions = **5 cycles** (F → D → E → M → W)
2. **Throughput**:
   - Two instructions complete in **5 cycles** → Throughput = **1 instruction every 2.5 cycles**, improved from 5 cycles (which will take in unpipelined one)
3. **Hardware Units**:
   - **Separate hardware units** for each stage (e.g., two ALUs for the `Execute` stage).

Ans. 2.

QUIZ-4 (Q3)

i) $32MB = 32 \times 2^{20} = 2^5 \times 2^{20} = 2^{25}$

128 64 32 16 8 4 2 1

0x 0000 4002 – 0000 [Invalid SA]

0000 0000 0000 0000 0100 0000 0000 0010 0000 0000 0000 0000

25 zeroes should be there.

the possible Next Starting Addresses

0000 0000 0000 0000 0100 0010 0000 0000 0000 0000 0000 0000

0000 4200 0000 [Valid SA]

Invalid SA: 0x 0000 4002 0000

Mem. Wastage (1) 0x 0000 01FE 0000.

Valid SA 0x 0000 4200 0000

memory Wastage = Valid SA – given Invalid SA

= 0x 4200 0000 – 0x 4002 0000 = 0x- 01FE 0000

ii) $2GB = 2 \times 2^{30} = 2^{31}$ 31 0's required in LSB of SA

SA 0020 8000 0000 ✓

given SA itself a Valid SA

1000 0000 0000 0000

31 0's available

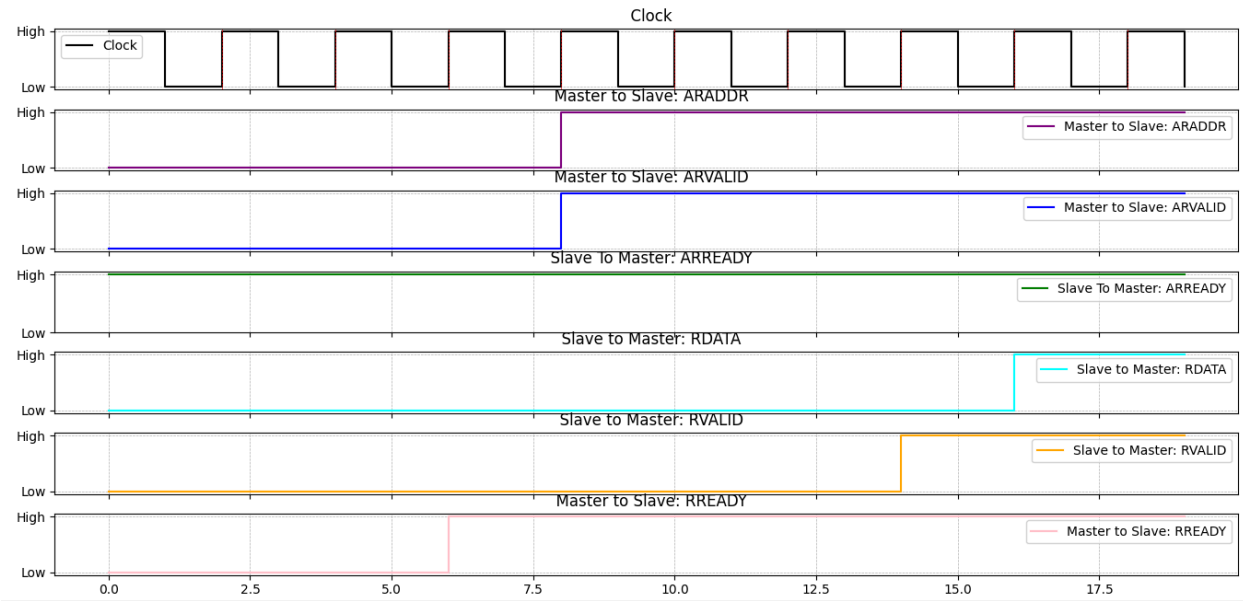
memory Wastage = Valid SA – given SA = 0

PART - B

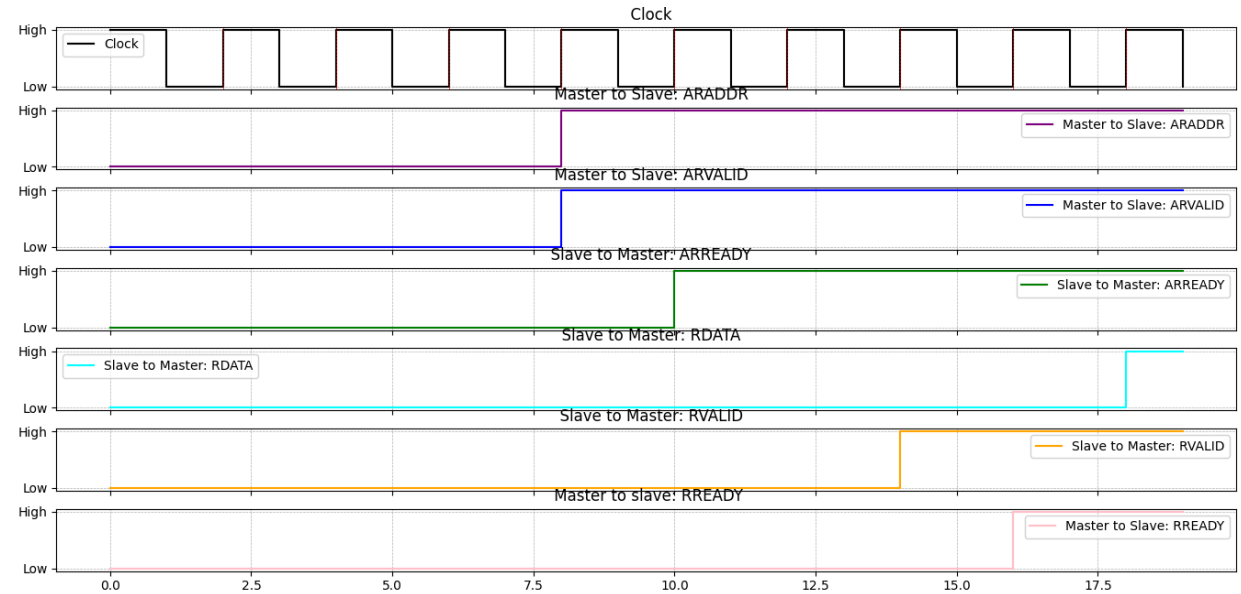
END SEM

Ans. 1.

Scenario 1 :



Scenario 2 :



Ans. 2.

B) Q2

			Size order
GPIO1 :	$4k = 2^2 \times 2^{10}$ B	$= 2^{12}$ B	(3)
BRAM :	$256k = 2^8 \times 2^{10}$ B	$= 2^{18}$ B	(1) Largest
CDMA :	$2k = 2^1 \times 2^{10}$ B	$= 2^{11}$ B	(4) Smallest
Timer :	$128k = 2^7 \times 2^{10}$ B	$= 2^{17}$ B	(2)

Constrain: Avoid Mem. Wastage

Case - 1 Large blocks First Approach.

To Fit BRAM, here we need 18 Zero in LSB

SA ≥ 0X 0000 07FF

0000 0000 0000 0000 0000 0111 1111 1111

∴ given Base Add is not Valid

First Starting Address 0004 0000 SA 1

Valid SA 0000 0002 0000 0000 0000 0111 1111 1111

Next SA2 128KB

0000 0000 0000 1000 0000 0000 0000 0000

0000 0000 0000 0001 1111 1111 1111 1111

EA2 0000 0000 0000 1001 1111 1111 1111 1111

EA2 : 0 0 0 9 F F F F

Next SA3 : 0 0 0 A 0 0 0 0

It has 12 0's ✓

4KB : 0 0 0 0 0 F F F

EA3 0 0 0 A 0 F F F

Next SA4 0 0 0 A 1 0 0 0

2KB 0 0 0 0 0 7 F F

EA4 0 0 0 A 1 7 F F

0X 0000 07FF BA

memory Wastage

0X 0004 0000 SA1

0X 0007 FFFF EA1

0X 0008 0000 SA2

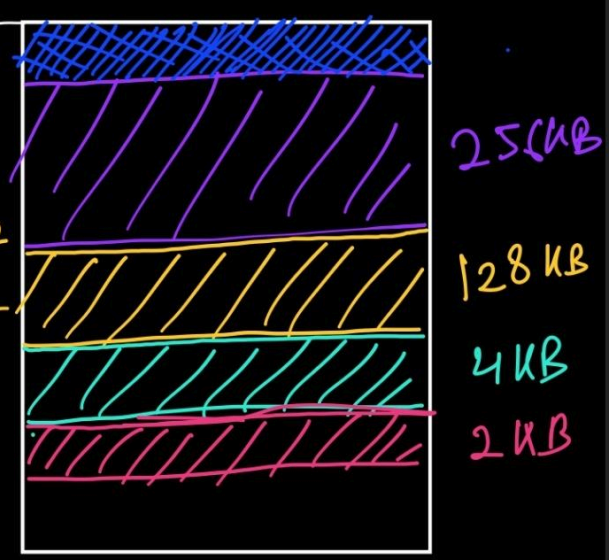
0X 0009 FFFF EA2

0X 000A 0000 SA3

0X 000A 0FFF EA3

0X 000A 1000 SA4

0X 000A 17FF EA4



SA1 − BA = 0X 0004 0000 − 0X 0000 07FF =

MW1 = 0X 0003 F801 = 260097

Case-2 Small blocks First

BA: 0000 07FF not Valid SA

SA1 [11 0s → Valid SA] 0000 0800
2KB (EDMA) 0000 07FF
0000 0FFF
EA1

SA2 [12 0s → Valid SA] 0000 1000
4KB (GPIO1) 0000 0FFF
0000 1FFF
EA2

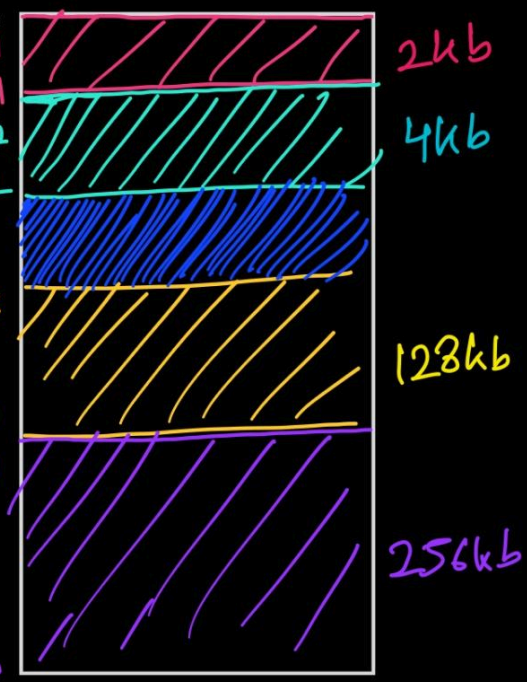
SA3 (Next) ✗ 0000 2000

Contains 13 0s only
But we need min 17 0s So Not Valid.

Next Valid Base Address 0002 0000 ✓
(0000 0000 0000 0010 0000 0000 0000 0000)
SA3 0001 FFFF
128KB 0003 FFFF
EA3

SA4 [has 18 0s → Valid] 0004 0000
256KB 0003 FFFF
0007 FFFF
EA4

	0000	07FF	BA
CDMA	0000	0800	SA1
	0000	0FFF	EA1
GPIO1	0000	1000	SA2
	0000	1FFF	EA2
	memory	wastage	
	0002	0000	SA3
Timer	0003	FFFF	EA3
	0004	0000	SA4
BRAM			
	0007	FFFF	EA4



Wastage = SA3 − EA2 = 0002 0000 − 0001 FFFF

$$\boxed{MW2 = 0x\ 1E001 = 1,22,881}$$

Conclussion: Compared to Case 1 & Case 2 case 2 has Lesser memory wastage

Marks Distribution

END SEM [B] Q2)		QUIZ-4 [A] Q3	
order of IP's	−1M	For IP1 & IP2	
BA of 4 IP	−1M x4	Identify the given SA Validity	−1M x2
Wastage & Mem	−1M	If not Find the Nearest Valid Address	−1M x2
Appropriate Explination	−1M	Find Memory Wastage	−1M x2
	7M		6M

QUIZ 4

Ans.1.

1. Unpipelined Combinational Delay [1.5M]

In the first diagram (without pipelining), let's say the combinational delay is: Tcomb=45 ns

The clock period Tclk must satisfy the timing constraint: **Tclk≥Tclk-Q+Tcomb+Tsetup+Tskew**
Assume all other delays to be 0.

Therefore, the original clock period: Tclk≥45 ns

2. After Pipelining (Dividing into Two Stages) [1.5M]

Now, we divide the combinational delay into two parts:

- **Tcomb1=20 ns**
- **Tcomb2=25 ns**

Considering other delays as same as unpipelined.

Timing Constraints for Each Stage

1. **First Stage (A → P)**:
   The clock period must account for the combinational delay of the first stage, clock-to-Q delay, setup time, and clock skew: Tclk≥Tclk-Q+Tcomb1+Tsetup+Tskew
   => Tclk≥20 ns

2. **Second Stage (P → B)**:
   Similarly, the second stage's clock period constraint is:
   Tclk≥Tclk-Q+Tcomb2+Tsetup+Tskew
   => Tclk≥25 ns

Resulting Clock Period

To satisfy both stages, the clock period must be: Tclk≥max(20 ns,25 ns)=25 ns

Why the Clock Period Doesn't Reduce by Half [3M]

1. **Pipeline Overhead**

- Introducing FFs between combinational logic stages adds overhead, such as setup time, clock skew, and clock-to-Q delay of the FFs.
- These delays prevent the clock period from being reduced directly by half, as the total clock period must account for both the combinational logic delay and the FF delays.

2. **Imbalance in Logic Delay**
    - The delays in the two new pipeline stages may not be perfectly balanced.
    - If one stage has more delay than the other, the clock period is determined by the slower stage, limiting the reduction in clock period.

Ans. 2.

The addresses 0x2300 0008 and 0x2300 0010 are 8 bytes apart. For the first address to be valid starting address we require 8 bytes of memory as it has 3 zeros in the last, for the second one we will require 16 bytes of memory as it has 4 zeros in last, and the first one will not be a valid address for it. Hence the memory size will be minimum of 8 bytes and 16 bytes which comes out to be 8 bytes

---

Ans. 3. **Connection Type**:

- **AXI DMA**: Operates in the programmable logic (PL) section and connects to the AXI interconnect. It is used for high-speed data transfers between PL peripherals (like custom hardware accelerators) and memory.
- **PS DMA**: Operates within the processing system (PS) and is tightly integrated with PS components like ARM Cortex-A processors. It is used for transferring data between PS peripherals (e.g., UART, SPI) and memory.

Performance:

- **AXI DMA**: Typically offers higher performance, supporting scatter-gather and high-throughput data transfers suitable for custom hardware accelerators in PL.
- **PS DMA**: Generally lower performance compared to AXI DMA and is better suited for lightweight, low-speed tasks or PS-only operations.

Configuration:

- **AXI DMA**: Requires configuration through the AXI interface in the programmable logic. It is highly customizable and designed to integrate with user-defined IP cores in the PL.
- **PS DMA**: Configured directly through the PS DMA controller registers in the processing system. It is less flexible compared to AXI DMA.

Integration:

- **AXI DMA**: Integrated as part of the PL fabric and requires interfacing with AXI-based peripherals or memory-mapped IPs.
- **PS DMA**: Part of the hardwired processing system, making it simpler to use for standard DMA tasks within the PS domain.

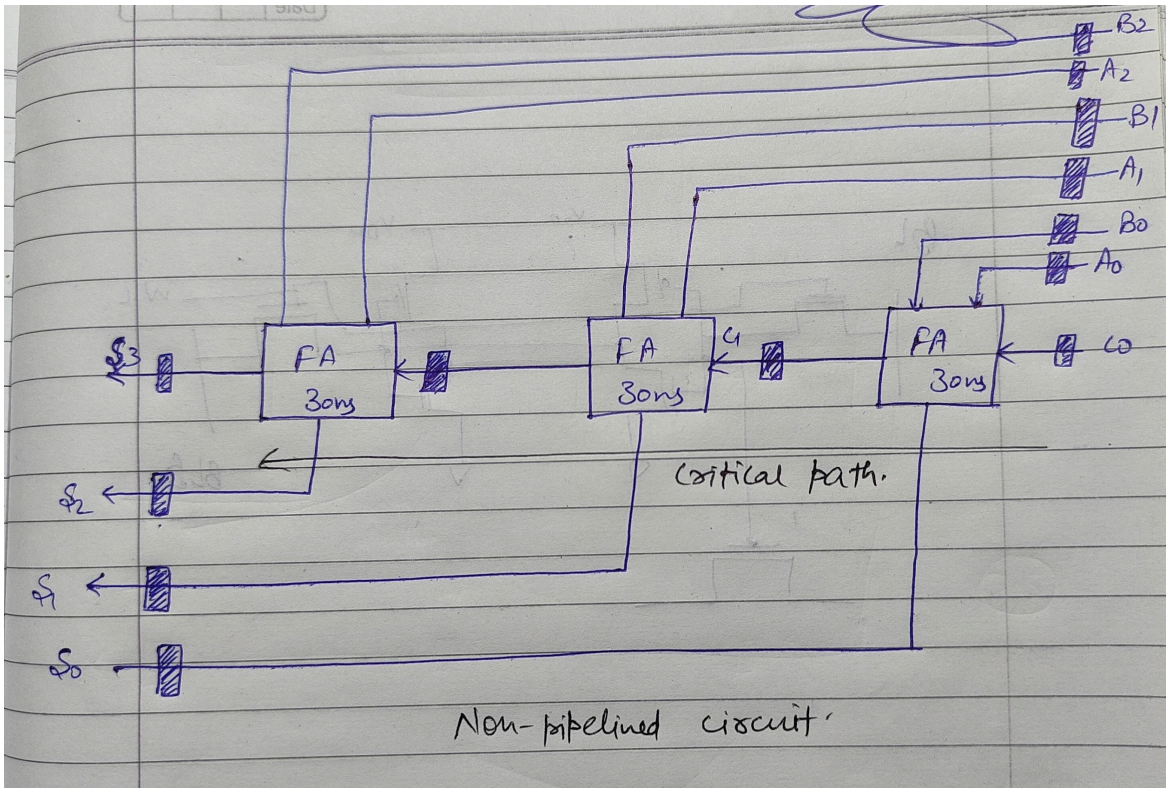
Throughput:

- **AXI DMA**: Capable of handling high-bandwidth data transfer, suitable for applications like video processing or data streaming from high-speed peripherals.
- **PS DMA**: Limited to moderate throughput, adequate for tasks like moving data from sensors or low-speed peripherals.

PART - C

END SEM

Ans. 1.



For non-pipelined circuit.

Clock period = Delay of ~~non~~ critical path = 30 + 30 + 30
= 90ns.

Latency = 30ns + 30ns + 30ns = 90ns. — 0.5

Throughput = $\frac{1}{90ns}$ — 0.5

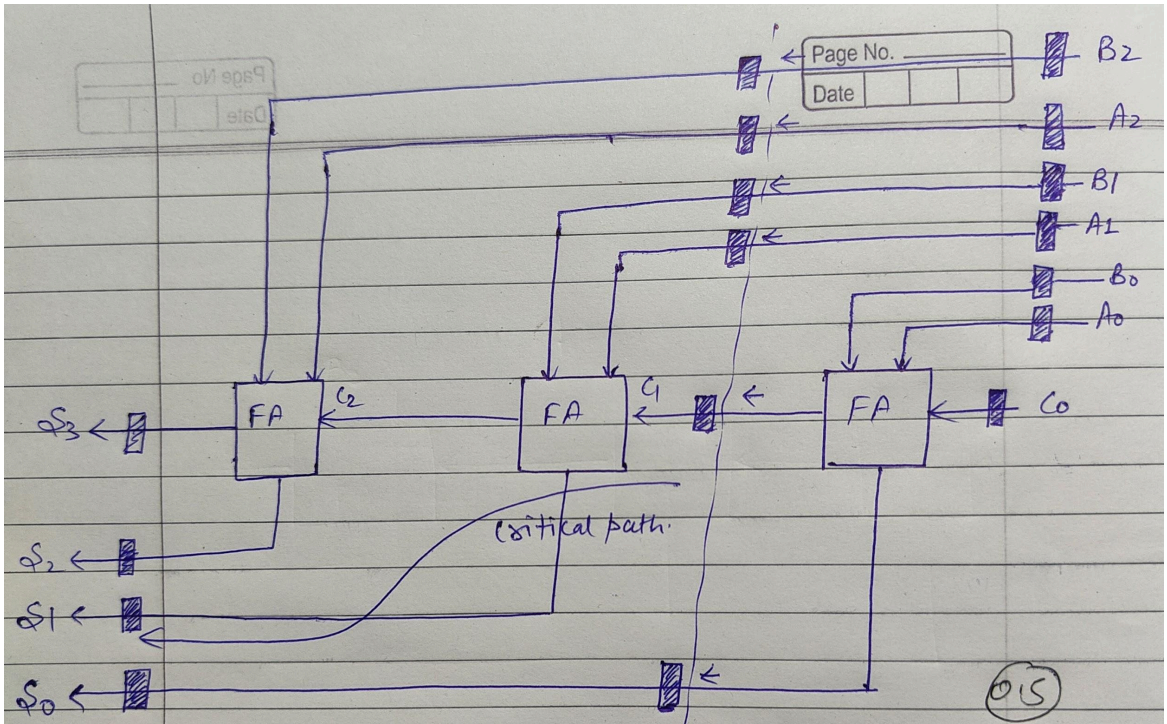


fig. 2 After first cut-set.

After first cut-set.

Clock period = Delay of critical path = 30 + 30 = 60ns.

— 0.5

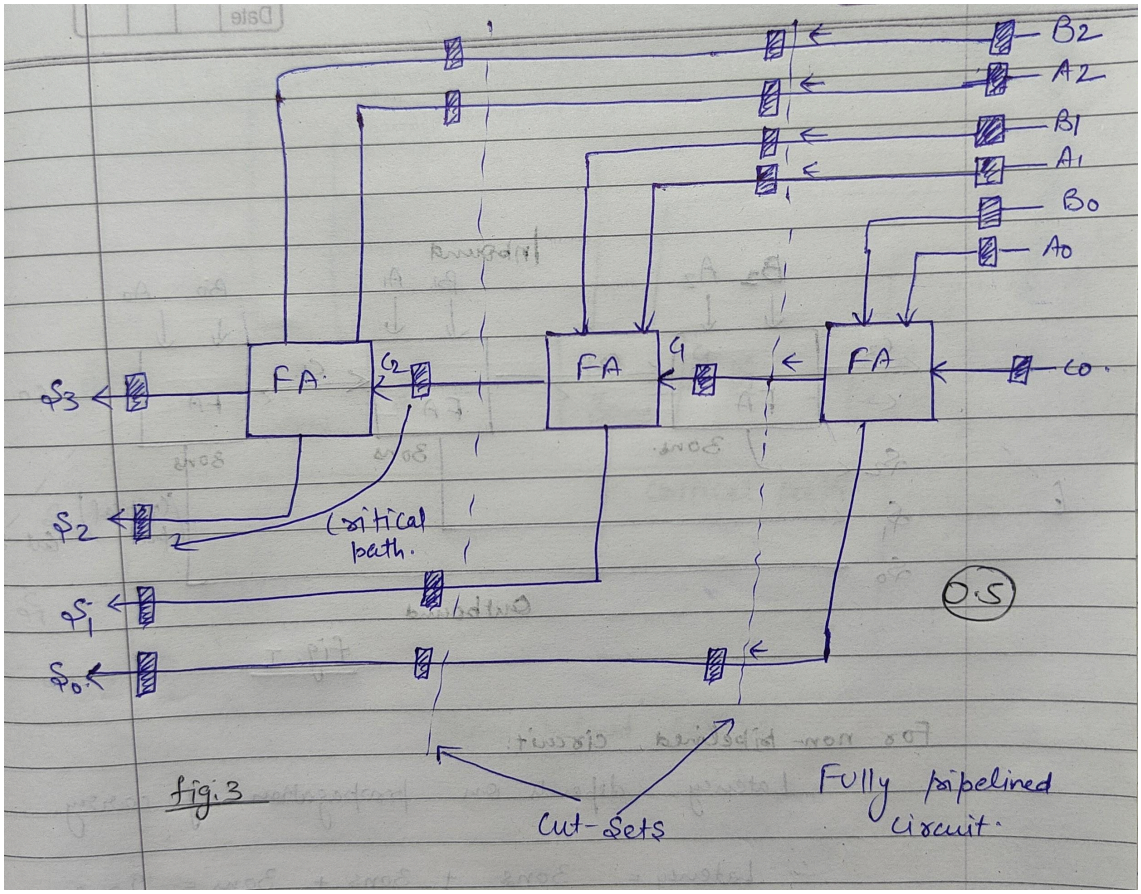


fig. 3

0.5

For pipelined circuit.

Insert a flip-flop after ~~a every~~ ~~of~~ adders ~~to~~ + adder to store the carry generated temporarily.

$\therefore$ $T_{clk} = \max(30, 30, 30) = 30ns$ — 0.5

$\therefore$ Latency $= 3 \times 30 = 90ns.$ — 0.5

Throughput $= \frac{1}{T_{clk}} = \frac{1}{30ns}$.

Critical path delay $= 30ns. = T_{clk}$ — 0.5

- If only one figure ~~has~~ of pipelined circuit with cut-sets has shown then direct 1 mark, otherwise as mentioned above.

- If any of the figure from fig. ① + fig. ② is wrong, then further answer will not be evaluated.

Ans. 2.

① Finding out the Base address →

Start address ≥ 0X0001_FFFF

The next valid address is 0X0002_0000

0X 0000 0000 0000 0010 — 0000 0000 0000 0000

17 0's at end

which means IP with N = 17 can fit first.

Given IP's →

AXI BRAM → 256K ⇒ N = 18

AXI Timer → 128K ⇒ N = 17

hence IP 2 will fit first

SA = 0X0002_0000

EA = 0X0003_FFFF

Next SA = 0X0004_0000

EA for IP 1 ⇒ ~~0X0007_0000~~

0X0007_FFFF ans.

The base addresses are assigned contiguously with no gaps(17 0's at the end of SA), starting from 0x0002_0000 for 128KB IP, and next SA as 0x0004_0000 for 256KB IP, Hence, both IPs are aligned to their respective memory boundaries (128 KB for Timer and 256 KB for BRAM)

QUIZ 4

Ans.1.

To ensure correct operation in a digital system, the clock period must be greater than the maximum latency of the combinational circuit. This requirement ensures that signals generated

at one clock edge have enough time to propagate through the combinational logic and stabilize before being sampled at the next clock edge, to meet the setup requirement of flip-flop(or any sequential circuit), clock-to-output delay of launching FF, and clock skew.
The equation is:

$$T_{clk} \geq T_{clk-out} + T_{pmax} + T_{setup} + T_{skew}$$

Consider an example of a synchronous circuit where the input data is processed through a combinational logic block (e.g., performing an addition A+B), and the result is stored in an output flip-flop. Let's assume the combinational circuit has a maximum propagation delay of 8 ns, which is the time it takes for the longest signal path through the logic to stabilize. Additionally, the output flip-flop has a setup time of 2 ns, time required by the FF to make the input available at the output is $T_{clk\text{-}out}$ = 0.5ns, and the skew between the two flops is T_{skew} = 1ns.

In this scenario, the total time required for the data to pass through the combinational logic and meet the flip-flop's setup requirement is 8 ns+2 ns+0.5 ns+1 ns=11.5 ns. Therefore, the clock period must be greater than 11.5 ns to ensure that the output data from the combinational logic has fully stabilized and meets all timing requirements. If the clock period is shorter, say 8 ns, the data might not stabilize in time, leading to a **timing violation** and causing the flip-flop to capture incorrect data.

Ans. 2. EMIO: EMIO stands for Extended Multiplexed Input Output. Certain connections can also be made via the Extended MIO (EMIO), which is not a direct path from the PS to external connections, but instead passes through and shares the I/O resources of the PL.

Why Useful?: EMIO enables access to additional I/O pins that are not directly available on the PS. The signals from the PS can be routed to the PL for custom processing, signal modification, or external access through FPGA pins.

Example: The PS has built-in GPIOs, but the design requires 20 additional GPIO pins for interfacing with external sensors and actuators. The PS can only provide a limited number of direct GPIOs. The additional GPIOs can be configured through the EMIO interface.

Ans. 3. The tasks of First Stage Boot Loader (FSBL) in the Zynq boot process are as follows:

1. Initialize the PS using the PS7 Init data that is generated by Vivado tools (MIO, DDR, e.t.c).
2. Program the PL using a bitstream (if provided).
3. Load the second stage boot loader or bare metal application code into DDR memory.
4. Hand off system control to the second stage bootloader or bare metal application.