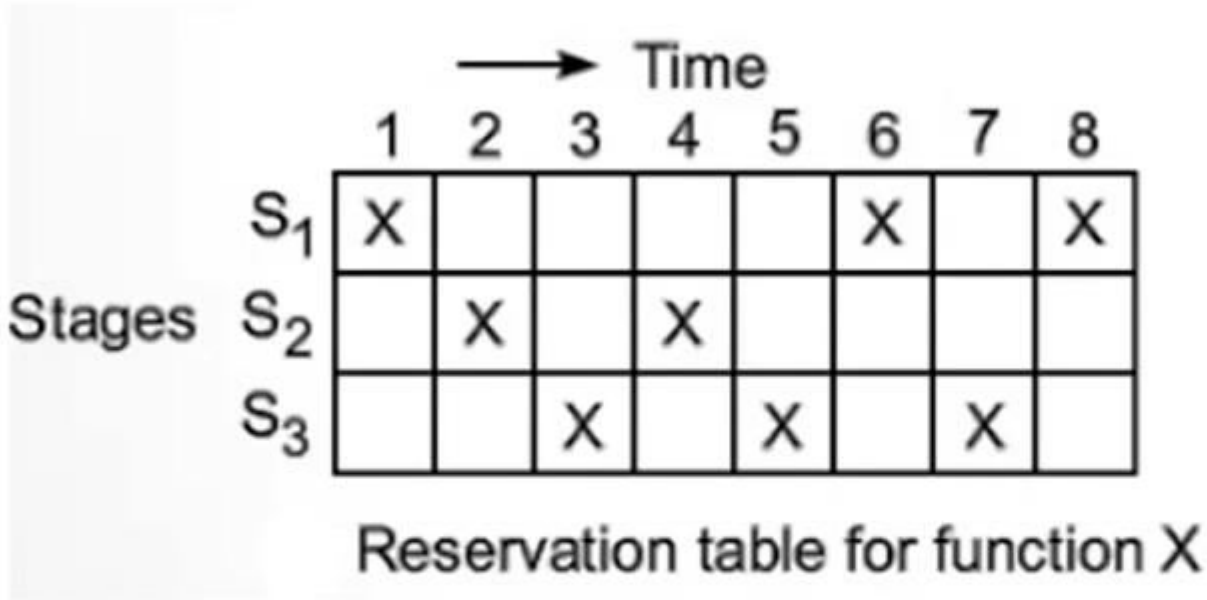
NUMERICAL-1

on

RESERVATION TABLE

CALCULATE COLLISION VECTOR, STATE DIAGRAM, GREEDY CYCLES, MAL

Question 1: Consider the following Reservation Table:



Reservation table for function X

1. What are the forbidden latencies?
2. Draw the state transition diagram.
3. List all the simple cycles and greedy cycles.
4. Determine the minimal average latency (MAL).

1. What are the forbidden latencies

S1: {(6-1), (8-6), (8-1)} i.e. **{5, 2, 7}**

S2: {(4-2)} i.e. **{2}**

S3: {(5-3), (7-5), (7-3)} i.e. **{2, 2, 4}**

Forbidden latencies = {2, 4, 5, 7}

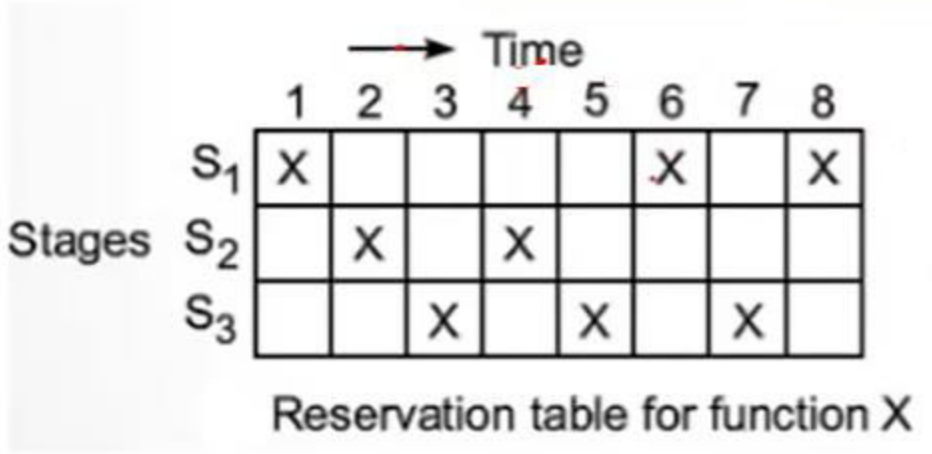
- **It causes collision**

Permissible Latency = {1, 3, 6, 8^+}

- **It does not causes collision**

Maximum forbidden latency (m) = 7,

- So the **collision vector** is of **7 bits**



Reservation table for function X

Collision Vector = $\{C_7\ C_6\ C_5\ C_4\ C_3\ C_2\ C_1\}$

0 1 0 1 1 0 1 0

1 = forbidden latency **0 = Permissible laten**

Collision Vector = (1011010)

Note: The ***last permissible latency of 8*** can be ***ignored*** because we need only 7 bits to represent Collision Vector

Initial Collision Vector (ICV) = 1011010

2. Draw State Transition diagram

A **state diagram** is constructed from **Collision Vector.** It specifies the **permissible state transitions** among successive initiations based on the **collision vector.**

How to draw state transition diagram?

- The **Permissible Latency = {1, 3, 6, 8^+ },** so we have to **find** the **next states of collision vector** for the **transitions 1, 3, 6, 8^+**
- ***Method:***

  To **find the collision vector** of the **next state** the **collision vector** of the **present state** is **shifted right** for each **transitions (1, 3, 6, 8^+).** The **shifted collision vector of present state** is **bitwise-ORed** with Initial Collision Vector (**ICV**).

Initial Collision Vector (ICV) = 1011010

Step 1: Calculating next state with **Present State (PS) = 1011010**

Permissible latencies (or transitions) = 1, 3, 6, 8^+

Initial Collision Vector (ICV) = 1011010

With latency 1:

1011010 **ICV**

\+ 0101101 **After Right Shift PS by 1**

1111111 **Next state**

- So, the **new state 1111111** with **latency 1**

With latency 3:

1011010 **ICV**

\+ 0001011 **After Right Shift PS by 3**

1011011 **Next state**

With latency 6:

1011010 **ICV**

\+ 0000001 **After Right Shift PS by 6**

1011011 **Next state**

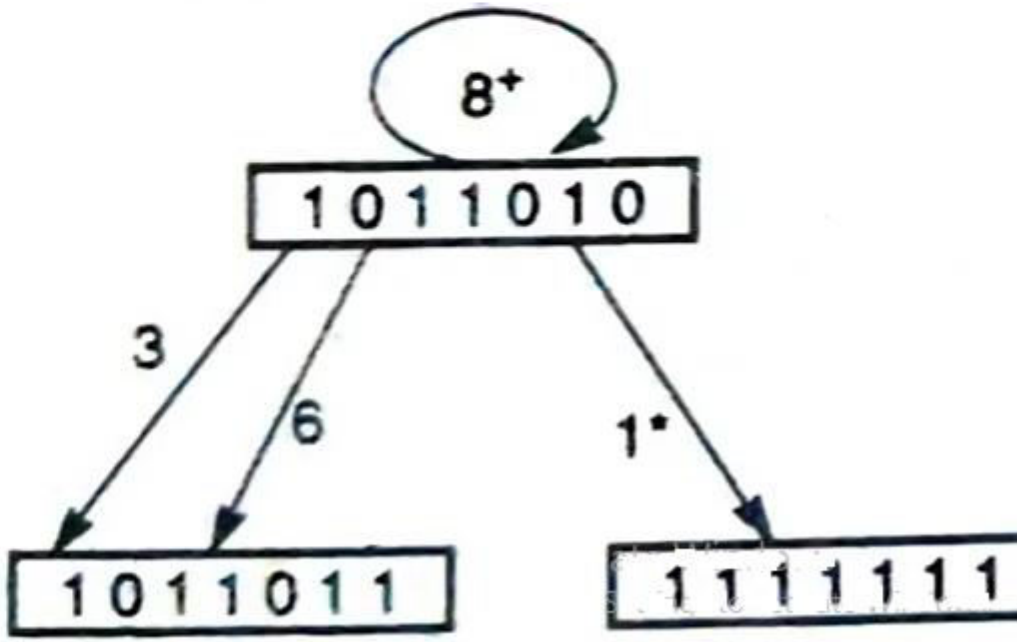
- **With latency 8^+:**

1011010 **ICV**

\+ 0000000 **After Right Shift PS by 8**

1011010 **Next state**

- So the **latency 8** returns to **original ICV (1011010)**



Initial State Diagram

Step 2: Calculating Next state with Present State (PS) = 1011011

Permissible latencies (or transitions) = 3, 6, 8⁺.

Initial Collision Vector (ICV) = 1011010

- **With latency 3:**

  1011010 **ICV**

  \+ 0001011 **After right shifting PS by 3**

  1011011 Next State

- **With latency 6:**

  1011010 **ICV**

  \+ 0000001 **After right shifting PS by 3**

  1011011 Next State

  - So the **latency 3 & 6** returns to **same state.**

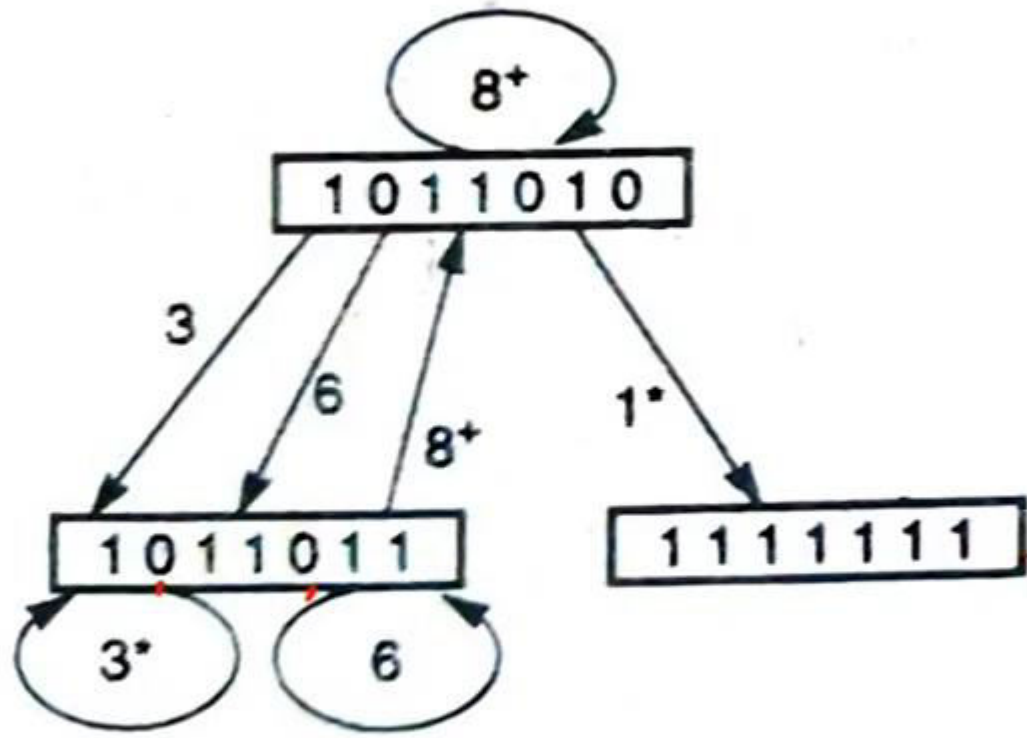
- **With latency 8⁺ :**

  1011010 **ICV**

  \+ 0000000 **After right shifting PS by 8⁺**

  1011010 Next state

  - So the **latency 8** returns to **original ICV (1011010)**



Intermediate State Diagram

Step 3: Calculating Next state with Present State (PS) = 1111111

Permissible latencies (or transitions) = 8^+

Initial Collision Vector (ICV) =(101101(

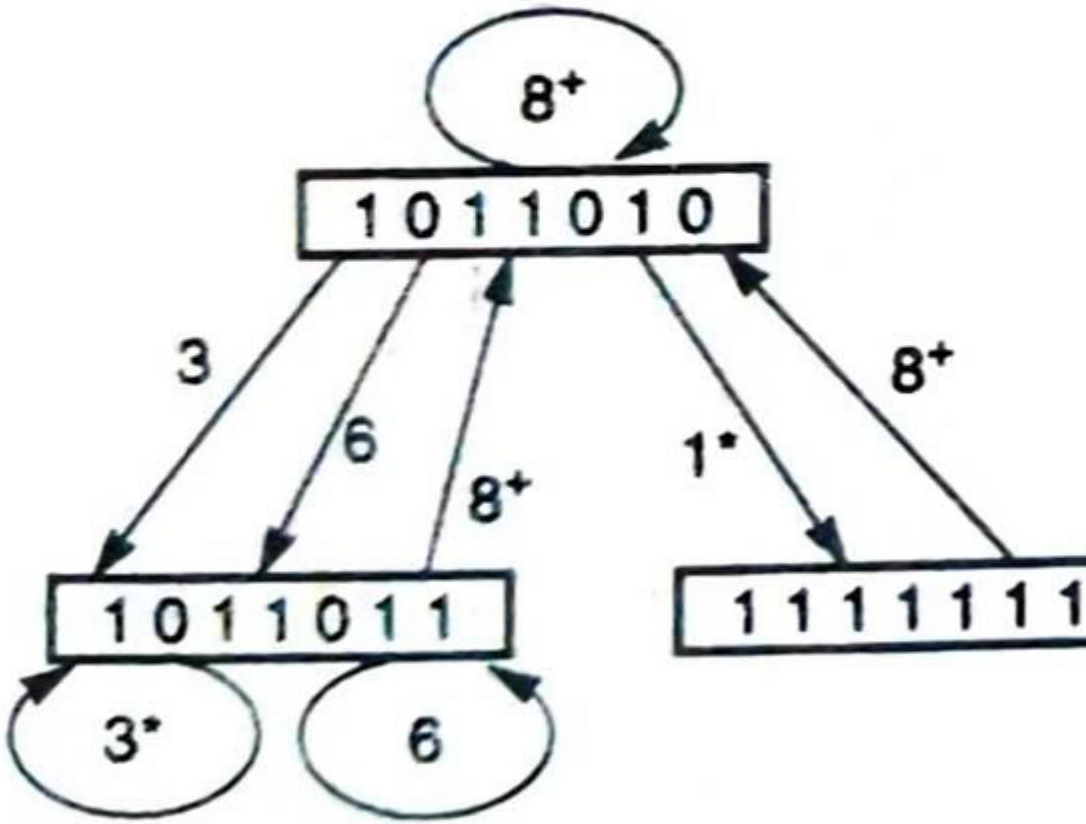
With latency 8^+ :

1011010 **ICV**

\+ 0000000 **After right shifting PS by 8^+**

1011010 Next State

- So the **latency 8** returns to **original ICV (1011010)**



Final State Diagram

Note:

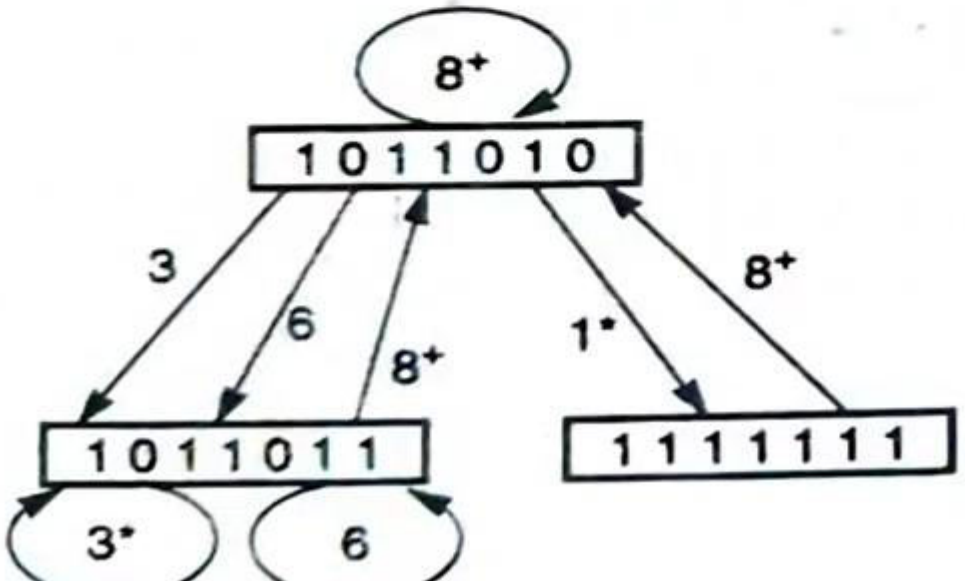
1. The **bitwise ORing** of the **shifted version of the present state** with the **initial collision vector (ICV)** is meant *to prevent collisions from future initiations.*
2. When the number of shifts is **m + 1** (7+1 =**8**) **or greater** (**8^+**), *all transitions are redirected back to the **initial state**.*
   - For example, after **eight or more shifts** (denoted as), the next 8+ state must be the initial state, regardless of which state the transition starts from.

3. Simple & Greedy Cycles…

Simple Cycle: A **Simple Cycle** is a **latency cycle** in which **each state appears only once.**

Greedy Cycle: A **Greedy Cycle** is a simple cycle whose edges are all made with **minimum latencies** from their respective starting states.

- Their **average latencies** must be **lower**

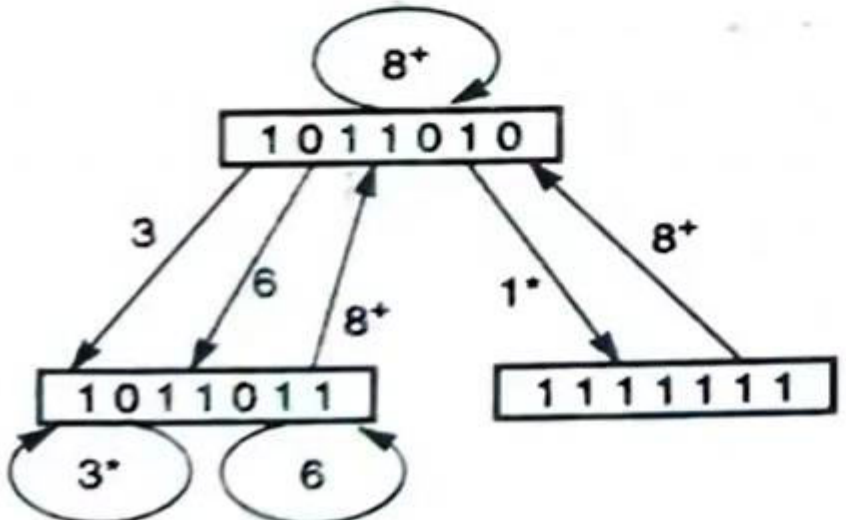


Latency Cycles	Types of cycles	Average Latency
(3)	Simple cycle (Greedy cycle)	3
(6)	Simple cycle	6
(8)	Simple cycle	8
(1,8)	Simple cycle (Greedy cycle)	4.5
(3,8)	Simple cycle	5
(6,8)	Simple cycle	7
(3,6,3)	-----	-----
(3,3,6)	-----	-----
(1,8,3,8)	-----	-----
(1,8,6,8)…	-----	-----

4. MAL (Minimum Average Latency)

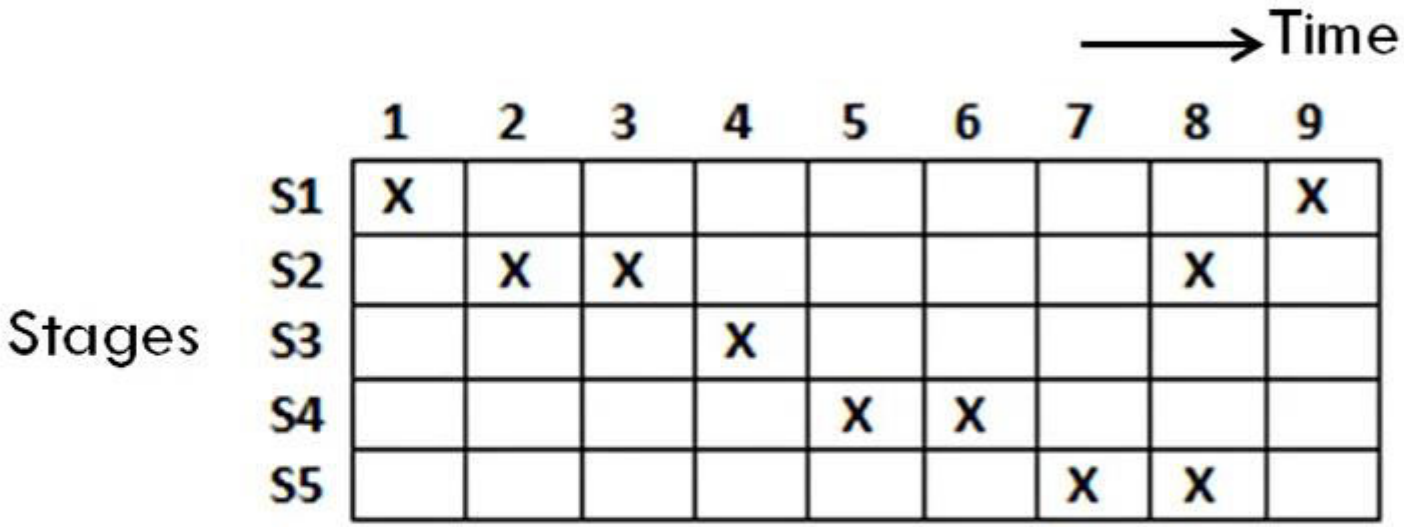
MAL (Minimum Average Latency) is the **minimum average latency** obtained from the **greedy cycle.**

- The minimum-latency edges in the state diagrams are marked with asterisks.
- Atleast one of the greedy cycles will lead to the MAL



Latency Cycles	Types of cycles	Average Latency
(3)	Simple cycle (Greedy cycle)	3 (MAL
(6)	Simple cycle	6
(8)	Simple cycle	8
(1,8)	**Simple cycle (Greedy cycle)**	**4.5**
(3,8)	Simple cycle	5
(6,8)	Simple cycle	7
(3,6,3)	-----	-----
(3,3,6)	-----	-----
(1,8,3,8)	-----	-----
(1,8,6,8)...	-----	-----

Question 2: Consider the following Reservation Table:

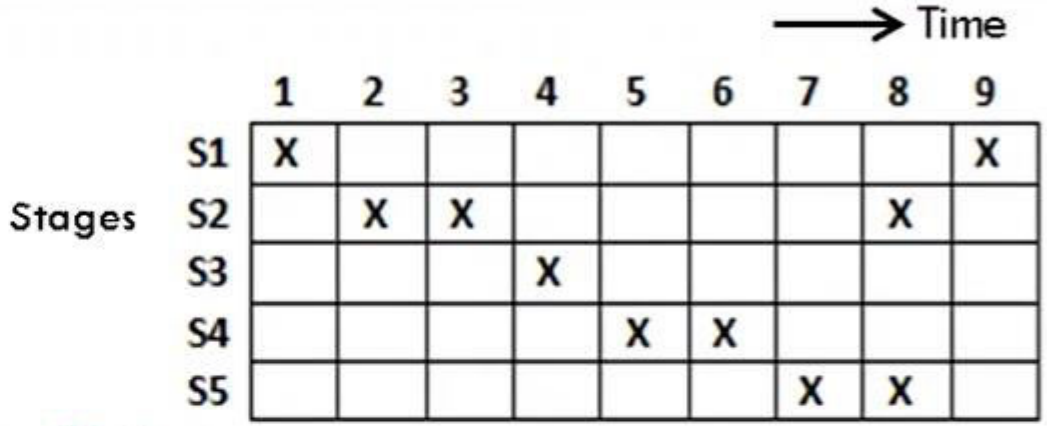


1. What are the forbidden latencies and initial collision vector?
2. Draw the state transition diagram for scheduling the pipeline.
3. List all the simple cycles and greedy cycles.
4. Determine the minimal average latency (MAL).

1. What are the forbidden latencies & initial collision vector?

Forbidden latencies

S1: {(9-1)}	i.e.	**8**
S2: {(3-2), (8-3), (8-2)}	i.e.	**1, 5, 6**
S3: 0	i.e.	**0**
S4: {(6-5)}	i.e.	**1**
S5: {(8-7)}	i.e.	**1**



Forbidden latencies = {1, 5, 6, 8} — **It causes collision**

Permissible Latency = {2, 3, 4, 7, 9^+ } — **It does not causes collision**

Maximum forbidden latency (m) = 8,

So the **collision vector** is of **8 bits**

Collision Vector : {C_8 C_7 C_6 C_5 C_4 C_3 C_2 C_1 } = {10110001}

1 0 1 1 0 0 0 1 — **1= forbidden latency** **0= Permissible latency**

Initial Collision Vector (ICV) = 10110001

2. Draw State Transition diagram

- ***How to draw state transition diagram?***
  - The **Permissible Latency = {2, 3, 4, 7, 9+ }**, so we have to **find** the **next states of collision vector** for the **transitions 2, 3, 4, 7, 9+**
  - ***Method:***

    To **find** the **collision vector** of the **next state** the **collision vector** of the **present state** is **shifted right** for each **transitions (2, 3, 4, 7, 9+)**. The **shifted collision vector of present state** is **bitwise-ORed** with Initial Collision Vector (**ICV**).

Initial Collision Vector (ICV) = 10110001

Step 1: Calculating next state with Present State (PS) = 10110001

Permissible latencies (or transitions) = 2, 3, 4, 7, 9+

Initial Collision Vector (ICV) = 10110001

With latency 2:

10110001 ICV

\+ 00101100 After Right Shift PS by 2

10111101 Next state

- So, the new state is 10111101 with latency 2

With latency 3:

10110001 ICV

\+ 00010110 After Right Shift PS by 3

10110111 Next state

- So, the new state is 10110111 with latency 3

With latency 4:

10110001 ICV

\+ 00001011 After Right Shift PS by 4

10111011 Next state

- So the new state is 10111011 with latency 4

With latency 7:

10110001 ICV

\+ 00000001 After Right Shift PS by 7

10110001 Next state

- So the latency 7 returns to original ICV (1011010)

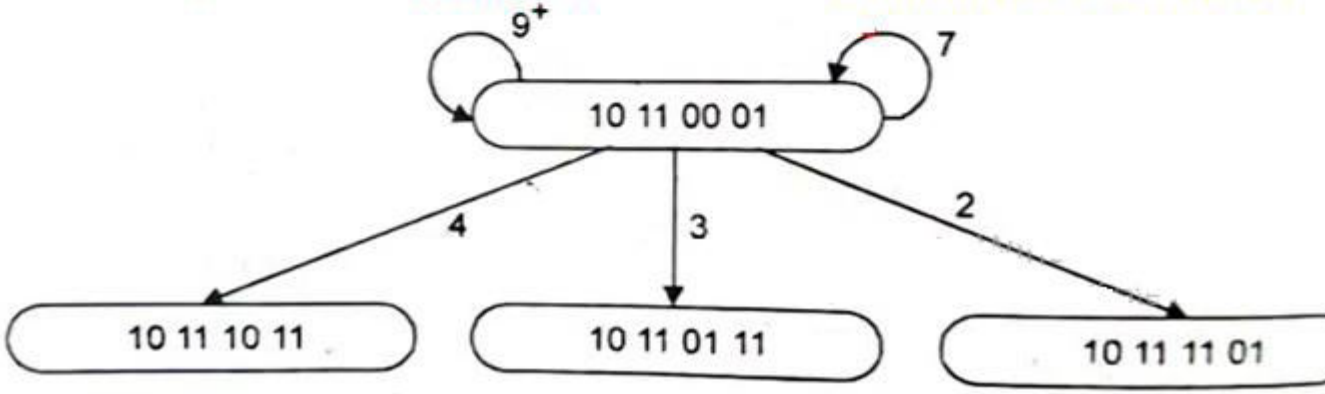
With latency 9+:

10110001 ICV

\+ 00000000 After Right Shift PS by 9

10110001 Next state

- So the latency 9+ returns to oriainal ICV (1011010)



Step 2: Calculating Next state with Present State (PS) = 10111101

Permissible latencies (or transitions) = 2, 7, 9+.

Initial Collision Vector (ICV) = 10110001

- **With latency 2:**

```
  10110001   ICV
+ 00101111   After right shifting PS by 2
------------
  10111111   Next State
```

  - So, the new state is 10111111 with latency 2

- **With latency 7:**

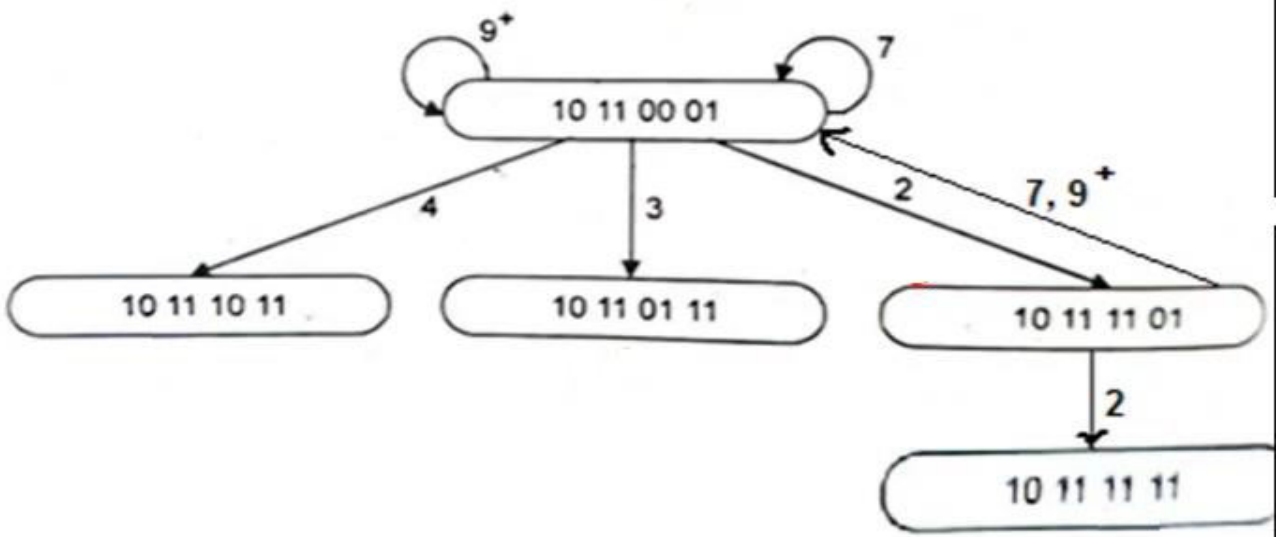
```
  10110001   ICV
+ 00000001   After right shifting PS by 7
------------
  10110001   Next State
```

  - So the latency 7 returns to original ICV (10110001)

- **With latency 9+:**

```
  10110001   ICV
+ 00000000   After right shifting PS by 9+
------------
  10110001   Next state
```

  - So the latency 9+ returns to original ICV (10110001)



Intermediate State Transition Diagram

Note: The process of shifting continues by taking the collision vector of new state, shifting and ORing with the initial collision vector (ICV).

Step 5: Present State (PS) = 10111111

Permissible latencies (or transitions) = 7, 9⁺.

- **With latency 7:**

  10110001 ICV

  \+ 00000001 After right shifting PS by 7

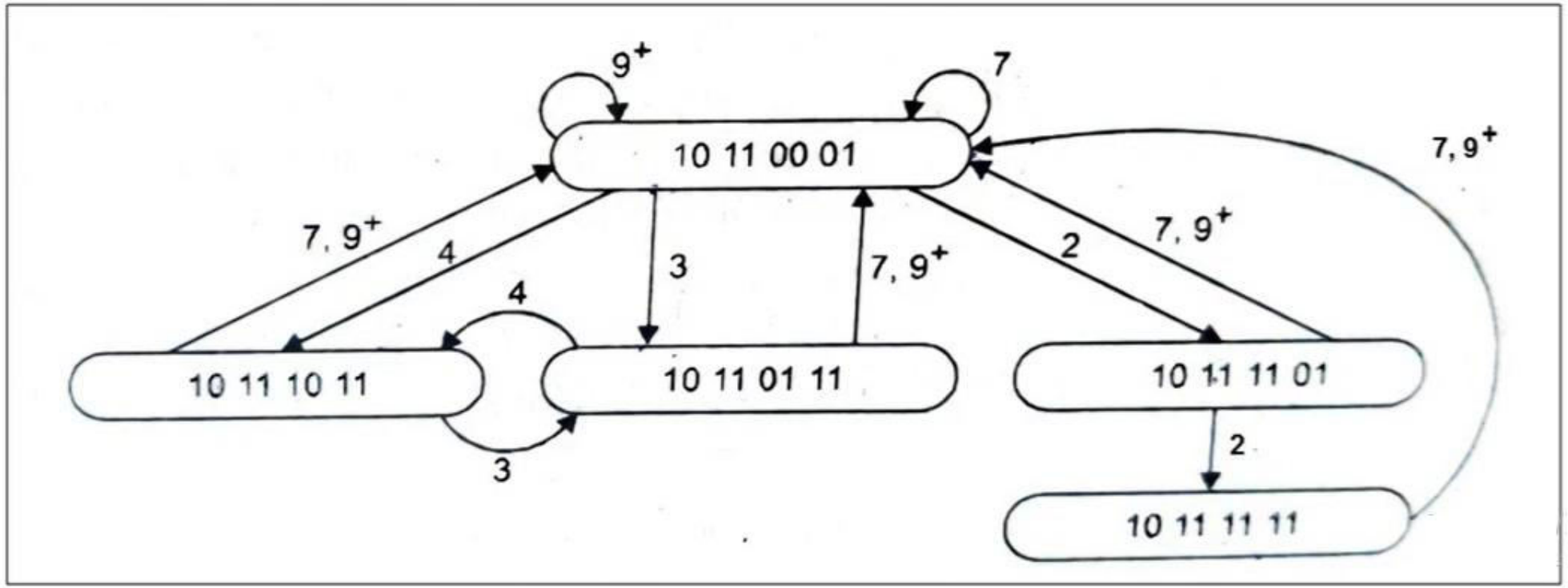
  10110001 Next State

- **With latency 9⁺:**

  10110001 ICV

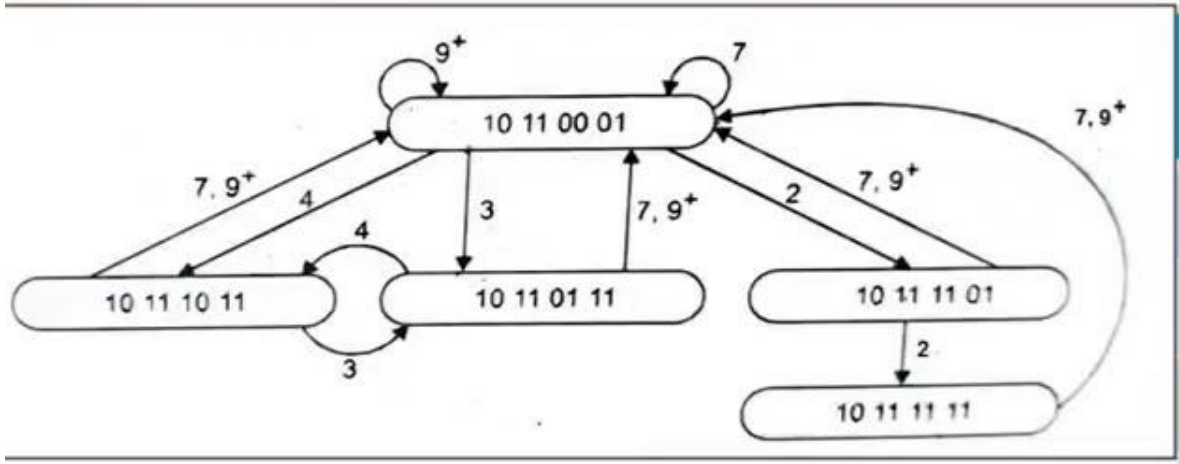
  \+ 00000000 After right shifting PS by 9⁺

  10110001 Next state



Final State Transition Diagram

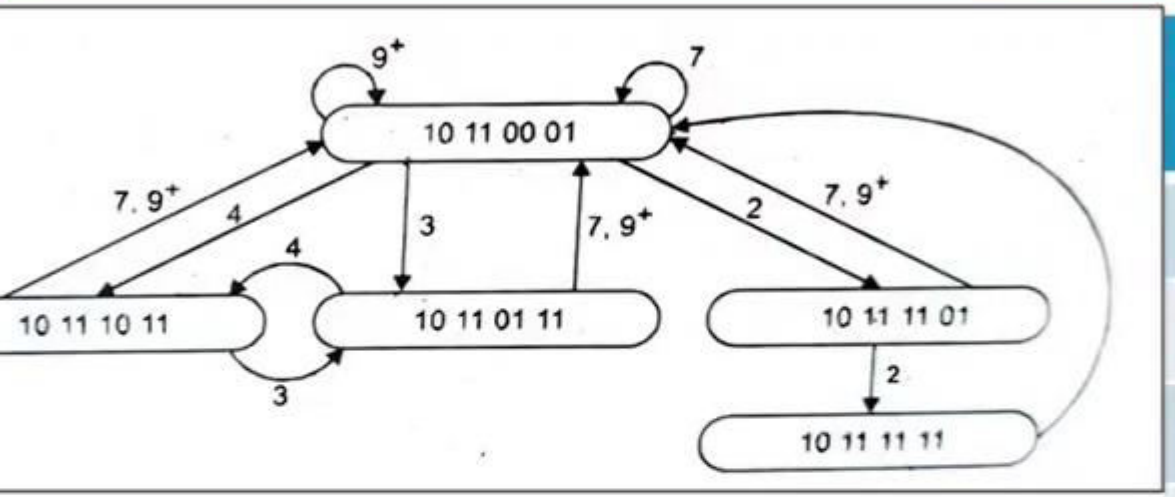
3. Simple & Greedy Cycles



Latency Cycles	Types of cycles	Average Latency
(7)	Simple cycle (Constant Cycle)	7
(2, 7)	Simple cycle	4.5
(2, 2, 7)	**Simple cycle (Greedy Cycle)**	**3.6**
(3, 7)	Simple cycle	5
(3, 4)	**Simple cycle (Greedy Cycle)**	**3.5**
(4, 7)	Simple cycle	5.5
(4, 3, 7)	Simple cycle	4.6
(3, 4, 7)	Simple cycle	4.6

- A **Simple Cycle** is a **latency cycle** in which **each state appears only once.**
- A **Greedy Cycle** is a simple cycle whose edges are all made with **minimum latencies** from their respective starting states.
  - Their **average latencies** must be **lower**

4. MAL Minimum Average Latency



Latency Cycles	Types of cycles	Average Latency
(7)	Simple cycle (Constant Cycle)	7
(2, 7)	Simple cycle	4.5
(2, 2, 7)	**Simple cycle (Greedy Cycle)**	**3.6**
(3, 7)	Simple cycle	5
(3, 4)	**Simple cycle (Greedy Cycle)**	**3.5 (MA**
(4, 7)	Simple cycle	5.5
(4, 3, 7)	Simple cycle	4.6
(3, 4, 7)	Simple cycle	4.6

MAL (Minimum Average Latency) is the **minimum average latency** obtained from the **greedy cycle.**

The **greedy cycle (3, 4)** leads to **MAL = 3.5**