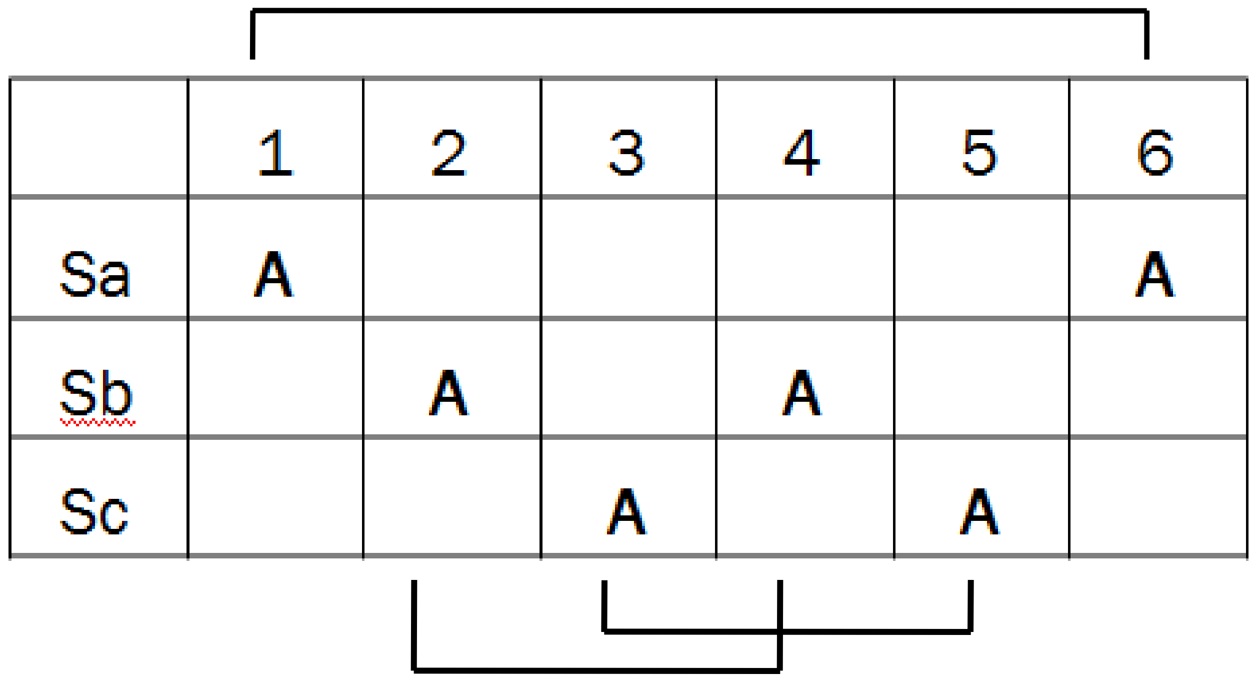
Chapter One
Introduction to Pipelined Processors

Principle of Designing Pipeline Processors

(Design Problems of Pipeline Processors)

State Diagram

Shortcut Method of finding Latency



	1	2	3	4	5	6
Sa	A					A
Sb		A		A		
Sc			A		A	

- Forbidden Latency Set,F = {5} U {2} U {2}

  = { 2,5}

State Diagram

- The initial collision vector (ICV) is a binary vector formed from F such that

  $C = (C_n \dots C_2\ C_1)$

  where $C_i = 1$ if $i \in F$ and $C_i = 0$ if otherwise
- Thus in our example

  $F = \{ 2,5 \}$

  $C = (1\ 0\ 0\ 1\ 0)$

State Diagram

- The procedure is as follows:

1. Start with the ICV
2. For each unprocessed state,

   For each bit i in the CV_i which is 0, do the following:

a. Shift CV_i right by i bits

b. Drop i rightmost bits

State Diagram

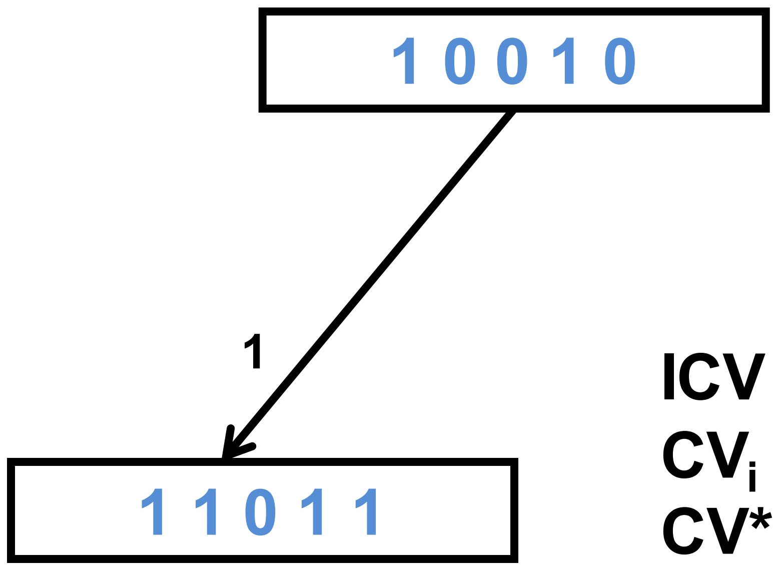
c. Append zeros to left
d. Logically OR with ICV
e. If step(d) results in a new state then form a new node for this state and join it with node of CV_i by an arc with a marking i.

- This shifting process needs to continue until no more new states can be generated.

State Diagram

$\boxed{1\ 0\ 0\ 1\ 0}$

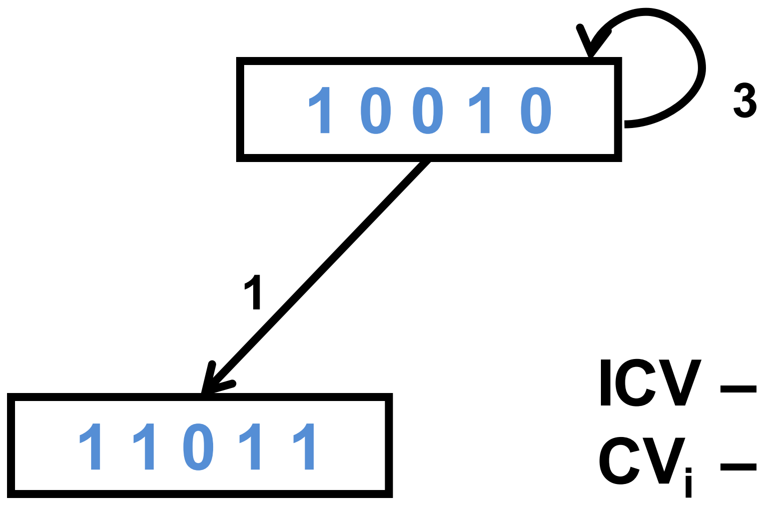
State Diagram



i = 1

$$\begin{array}{lll} \text{ICV} - & 10010 & \quad\text{OR} \\ \text{CV}_i - & \underline{01001} & \\ \text{CV*} & 11011 & \end{array}$$

State Diagram



i = 3

ICV – 10010 OR

CV_i – $\underline{00010}$

CV* 10010

State Diagram

$$\begin{tikzcd}
 & \boxed{1\,0\,0\,1\,0} \arrow[dl, "1"'] \arrow[dr, "4"] \arrow[loop right, "3"] & \\
\boxed{1\,1\,0\,1\,1} & & \boxed{1\,0\,0\,1\,1}
\end{tikzcd}$$

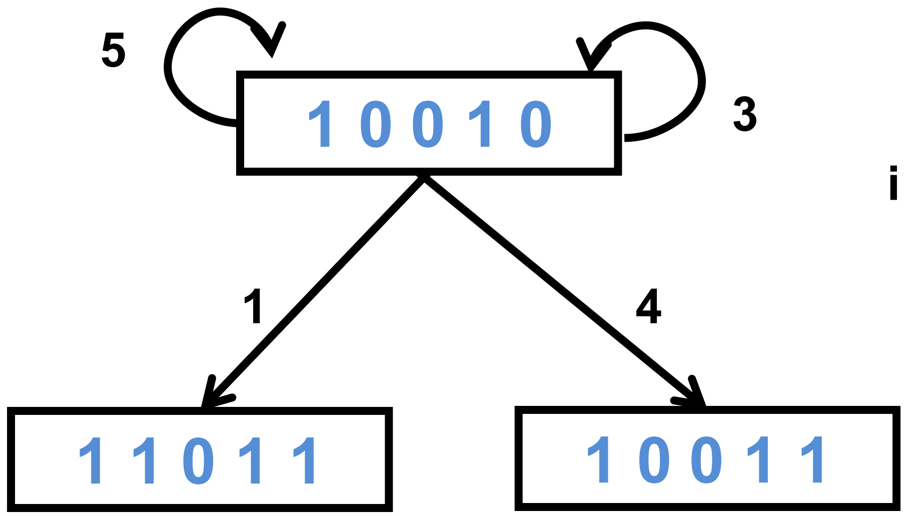
$i = 4$

ICV – 10010 OR

CV_i – $\underline{00001}$

CV* 10011

State Diagram



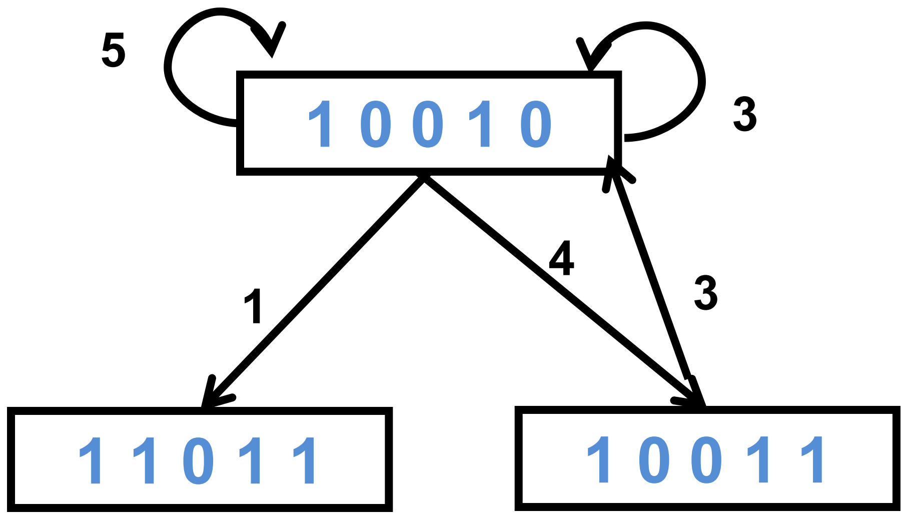
i = 5

ICV – 10010 OR

CV_i – <u>00000</u>

CV* 10010

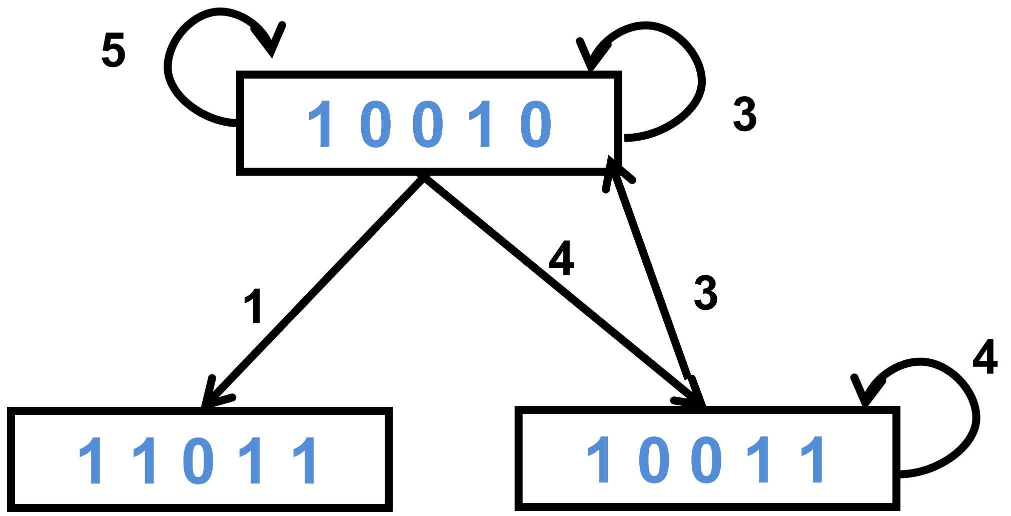
State Diagram



i =3

$$\begin{array}{lll} \text{ICV} - & 10010 & \text{OR} \\ \text{CV}_i - & \underline{00010} & \\ \text{CV*} & 10010 & \end{array}$$

State Diagram



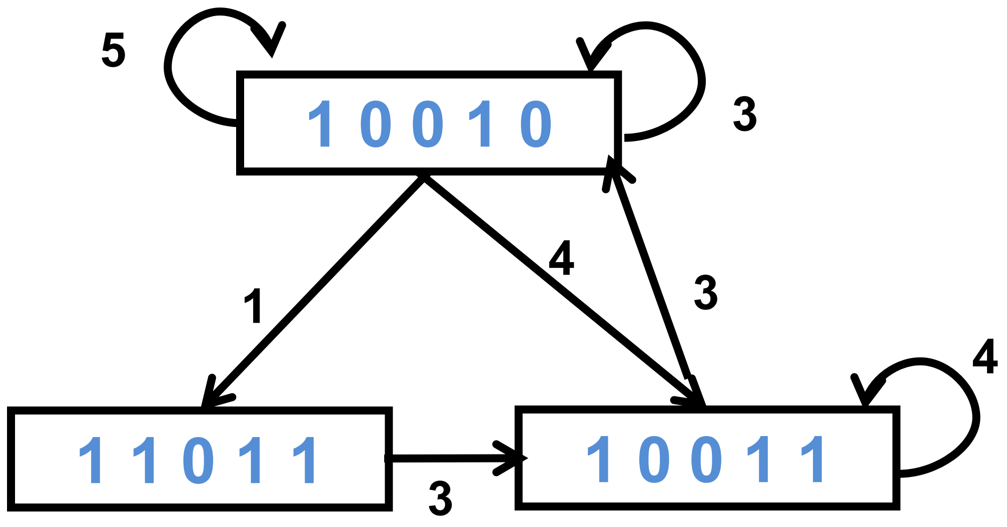
i =4

ICV – 10010 OR

CV_i – <u>00001</u>

CV* 10011

State Diagram



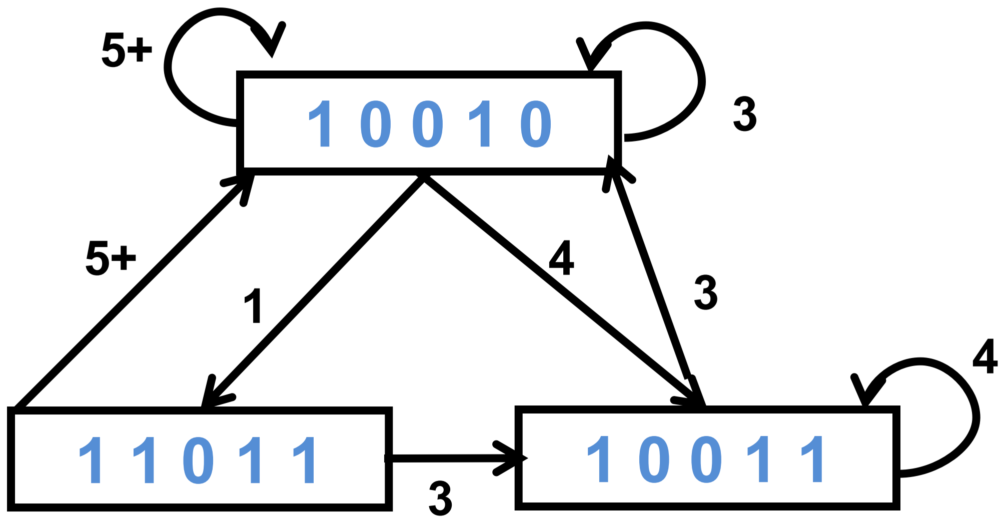
i =3

ICV – 10010 OR

CV_i – 00011

CV* 10011

State Diagram



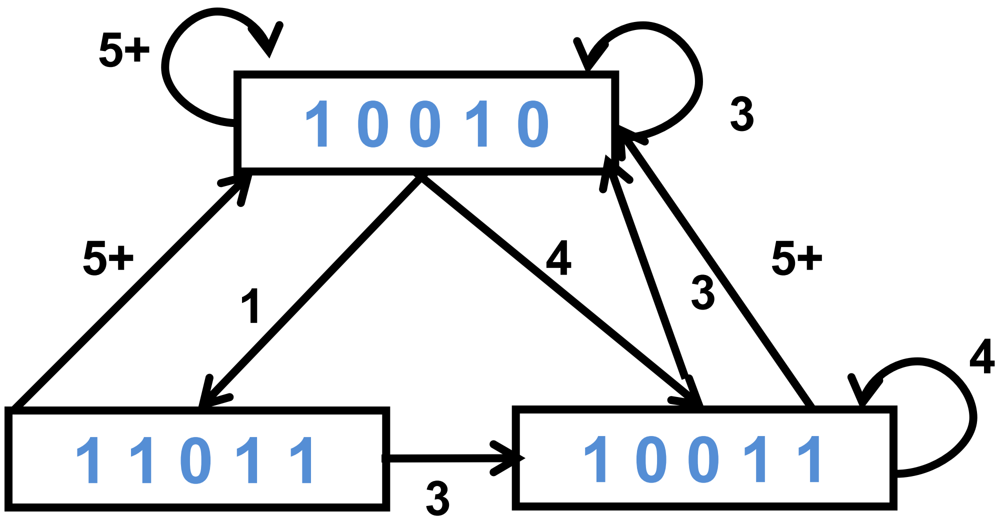
i = 5

ICV – 10010   OR

CV_i – <u>00000</u>

CV* 10010

State Diagram



i = 5

ICV – 10010 OR

CV_i – <u>00000</u>

CV* 10010

State Diagram

- The state with all zeros has a self-loop which corresponds to empty pipeline and it is possible to wait for indefinite number of latency cycles of the form (7),(8), (9),(10) etc.
- **Simple Cycle:** latency cycle in which each state is encountered only once.
- **Complex Cycle:** consists of more than one simple cycle in it.
- It is enough to look for simple cycles

State Diagram

- **Greedy Cycle:** A simple cycle is a greedy cycle if each latency contained in a cycle is the minimal latency(outgoing arc) from a state in the cycle.
- A good task initiation sequence should include the greedy cycle.

Simple cycles & Greedy cycles

- The Simple cycles are?
- The Greedy cycles are ?

Simple cycles & Greedy cycles

- The simple cycles are (3),(5) ,(1,3,3),(4,3) and (4)
- The Greedy cycle is (1,3,3)

State Diagram

- In the above example, the cycle that offers MAL is (1, 3, 3) (MAL = (1+3+3)/3 =2.333)

	1	2	3	4	5	6	7	8	9	10	11	12	13
Sa	A_1	A_2			A_5	A_1	A_2	A_8		A_5			A_8
Sb		A_1	A_2	A_1	A_2	A_5		A_5	A_8		A_8		
Sc			A_1	A_2	A_1	A_2	A_5		A_5	A_8		A_8	

UQ: Problem

- Consider the reservation table given below

	1	2	3	4	5	6	7	8	9
S_1	x								x
S_2		x	x					x	
S_3				x					
S_4					x	x			
S_5							x	x	

Problem

i. Find the forbidden set of latencies
ii. State the collision vector
iii. Draw the state transition diagram
iv. List simple cycles and greedy cycles
v. Calculate MAL (minimum average latency)