Pipelined Architecture

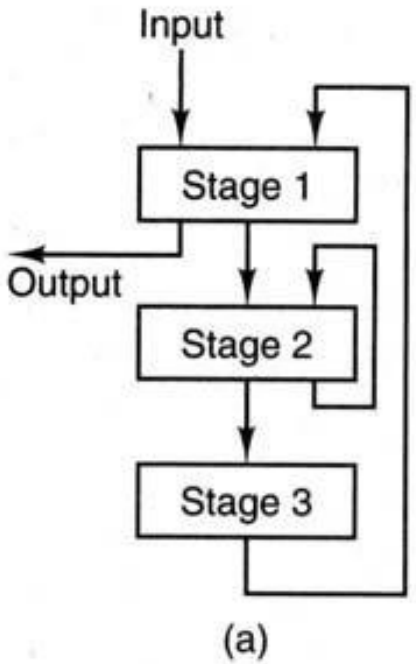
CSEN 3104

Lecture 9

Dr. Debranjan Sarkar

Reservation Table

- *3 stages, 5 clock cycles*
- The input data goes through the stages 1, 2, 2, 3 and 1 progressively



(a)

	Time t_0	t_1	t_2	t_3	t_4
Stage 1	X				X
Stage 2		X	X		
Stage 3				X	

(b)

A static pipeline and its corresponding reservation table.

Scheduling of Static Pipelines

Pipeline Control: Scheduling

- Controlling the sequence of tasks presented to a pipeline for execution is extremely important for maximizing its utilization.
- If two tasks are initiated requiring the same stage of the pipeline at the same time, a collision occurs, which temporarily disrupts execution.
- The reservation table can be used for determining the time difference between input data initiations so that collisions won't occur.
- Initiation of an input data refers to the time that the data enter the first stage of the pipeline.

Some Definitions

- Initiation: Launching of an operation into the pipeline. i.e. the time that the data enter the first stage of the pipeline

- Latency: the delay (or, the number of cycles) that elapse between two initiations of a pipeline
- Latency values must be non-negative integers
- A collision will occur if two pieces of input data are initiated with a latency equal to the distance between two X's in a reservation table
- For example, the reservation table, shown earlier, has two X's with a distance of 1 in the second row
- So, if a second piece of data is passed to the pipeline one time unit after the first, a collision will occur in stage 2
- A latency value *k* means that two initiations are separated by *k* clock cycles

Some definitions

- Collision
  - Collision occurs if a stage in the pipeline is required to perform more than one task at any time
  - Any attempt by two or more initiations to use the same pipeline stage at the same time will cause a *collision*.
  - A collision will occur if two pieces of input data are initiated with a latency equal to the distance between two X's in a reservation table
  - For example, the reservation table, shown earlier, has two X's with a distance of 1 in the second row
  - Therefore, if a second piece of data is passed to the pipeline one time unit after the first, a collision will occur in stage 2
  - A collision implies resource conflicts between two initiations in the pipeline, so it should be avoided

Forbidden Latency

- Latencies that cause collision are called forbidden latencies
- Forbidden latencies should be prohibited
- Otherwise, the two data would arrive at the same stage of the pipeline at the same time and lead to collision
- Forbidden latencies for the previous RT are 1 and 4
- Let the Maximum forbidden latency be m (here 4)
- n = no. of columns ($m \leq n-1$)
- With static pipelines, zero is always considered a forbidden latency, since it is impossible to initiate two jobs to the same pipeline at the same time.
- However, such initiations are possible with dynamic pipelines

Permissible Latency

- Latencies that do not cause any collision are called permissible latencies
- Permissible latencies for the previous RT are 2 and 3, as they do not cause collision
- Let the Maximum forbidden latency be m
- All the latencies greater than m do not cause collisions
- Permissible Latency p, lies in the range $1 \leq p \leq m-1$
- Value of p should be as small as possible
- Permissible latency p=1 corresponds to an ideal case, can be achieved by a static pipeline

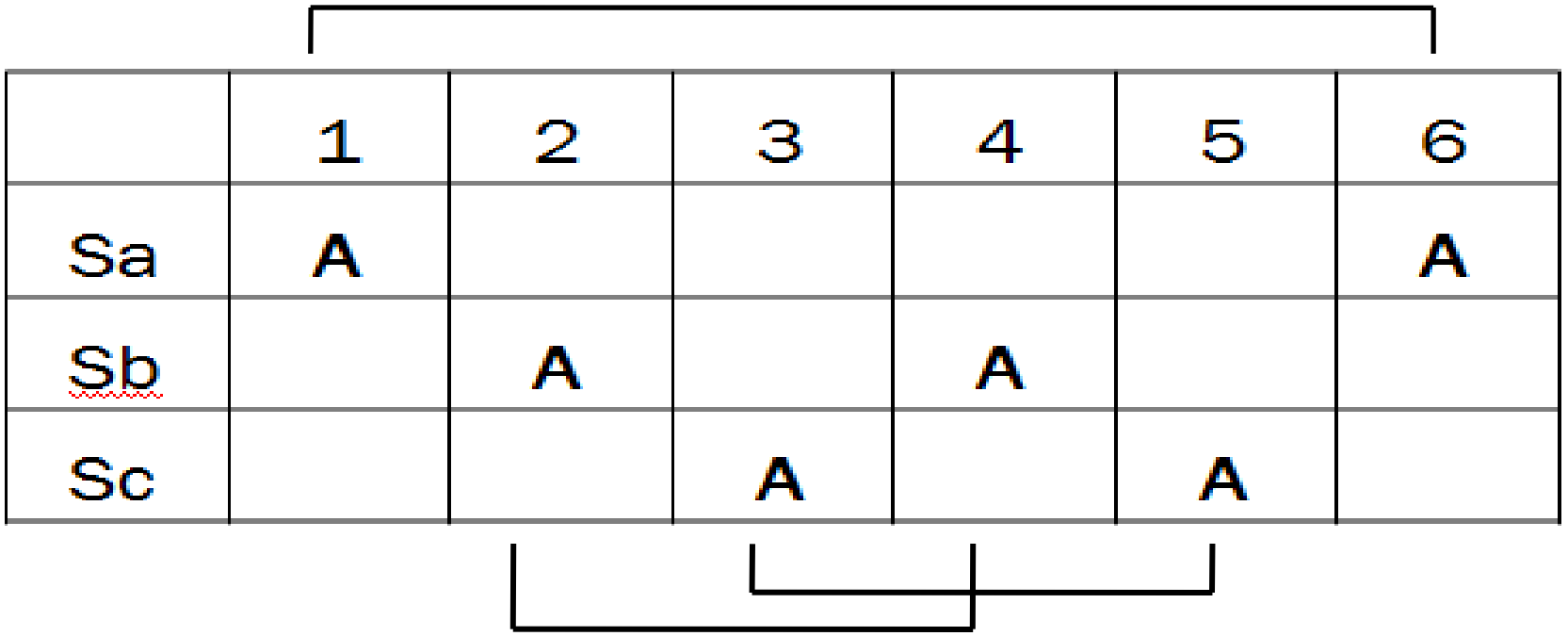
Forbidden Latency set or Forbidden List

- Reservation table, having more than one X's in any given row, has one or more forbidden latencies, which, if not prohibited, would allow two data to collide or arrive at the same stage of the pipeline at the same time
- Forbidden Latency set: the set of all possible column distances between two entries in a particular row of Reservation Table
- The forbidden list F is simply a list of integers corresponding to these prohibited Latencies
- As 0 is always considered a forbidden latency for static pipelines, 0 may be included in the Forbidden List

Collision Vectors

- A collision vector is a combined set of permissible and forbidden latencies.
- Let the Maximum forbidden latency be m (here 4) and n = no. of columns ($m \leq n-1$)
- Then the collision vector is an m-bit binary number $C = (C_m\ C_{m-1}....C_2\ C_1\ )$
- The initial collision vector, C, is created from the forbidden list in the following way:
  - each component C_i of C, for i=1 to m, is 1 if i is an element of the forbidden list.
  - Otherwise, C_i is zero

Example 1



	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}
- Permissible Latency List = { 1,3,4}
- Initial Collision Vector = 10010

State Diagram

- State Diagram is a graph of all the possible operation sequences through the pipeline
- State diagrams can be constructed to specify the permissible transitions among successive initiations
- State diagrams can be used to show the different states of a pipeline for a given time slice
- Once a state diagram is created, it is easier to derive schedules of input data for the pipeline that have no collisions

Procedure for construction of State Diagram

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

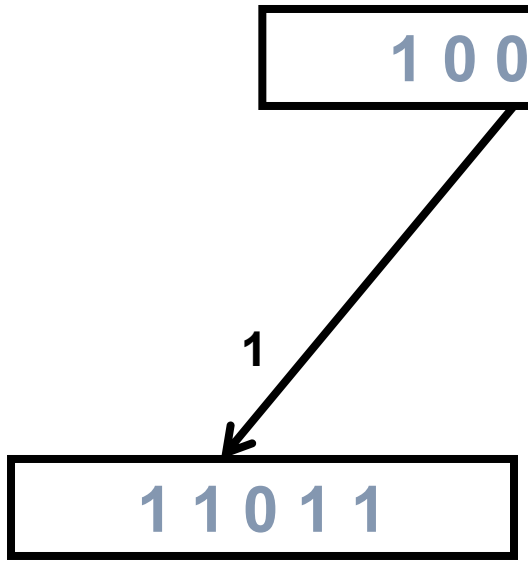
   For each bit i in the CV_i which is 0, do the following:

a. Shift CV_i right by i bits
b. Drop i rightmost bits
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.

Example 1: State Diagram

Example 1: State Diagram



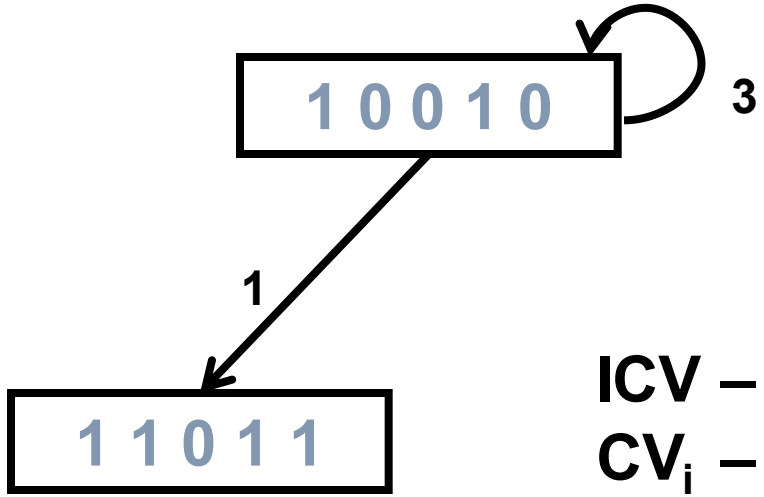
i = 1

ICV – 10010 OR

CV_i – 01001

CV* 11011

Example 1: State Diagram



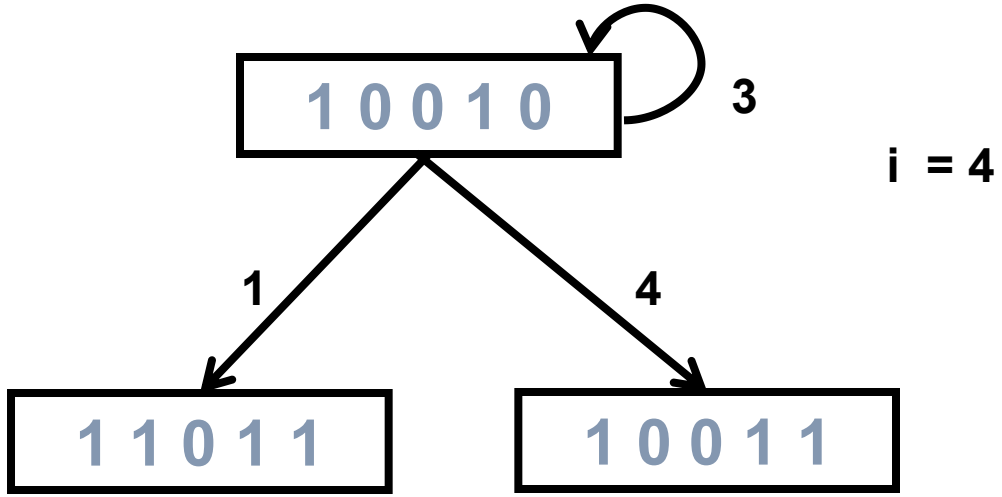
i = 3

ICV – 10010 OR

CV_i – 00010

CV* 10010

Example 1: State Diagram

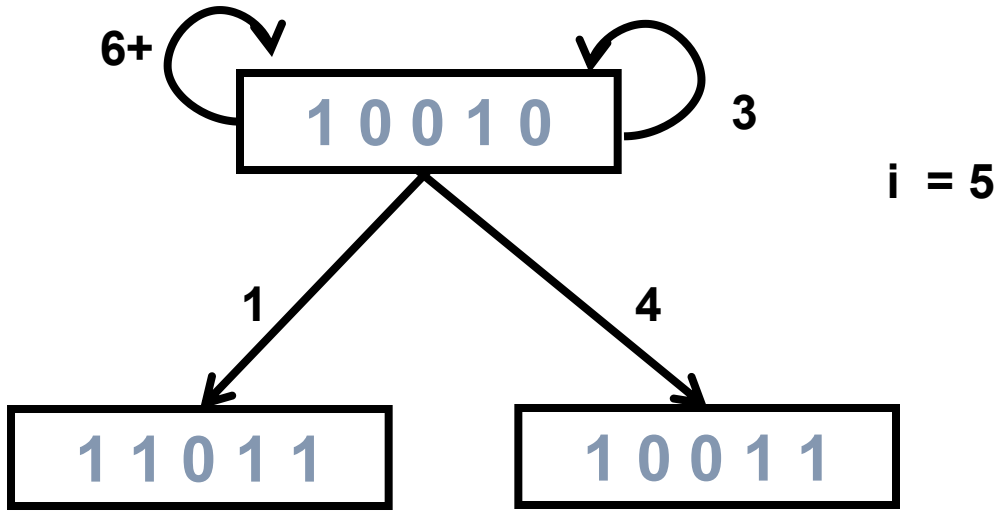


ICV – 10010 OR

CV_i – <u>00001</u>

CV* 10011

Example 1: State Diagram

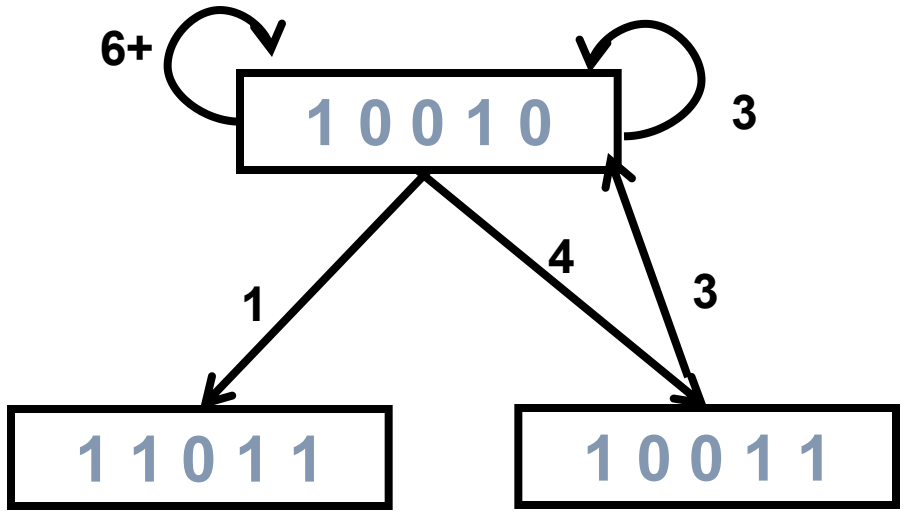


ICV – 10010 OR

CV_i – 00000

CV* 10010

Example 1: State Diagram



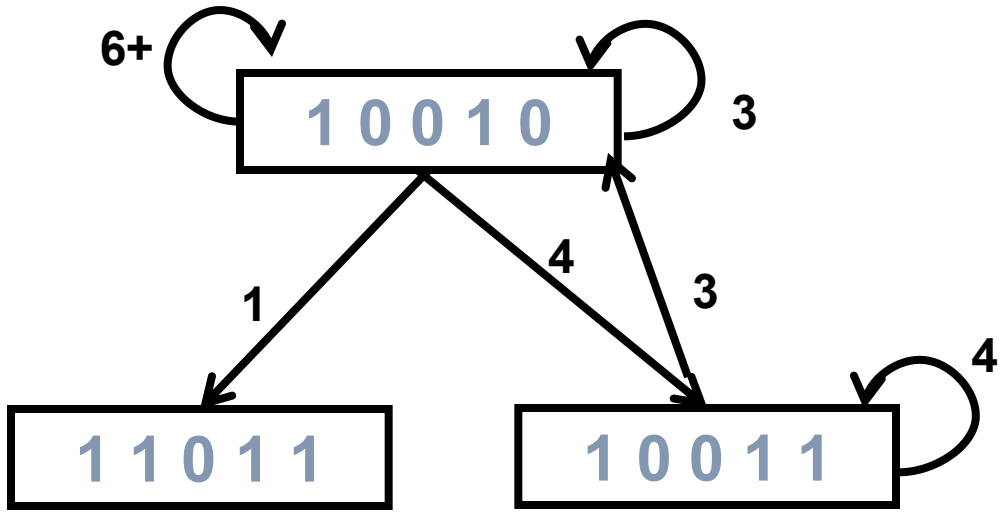
i =3

ICV – 10010 OR

CV_i – <u>00010</u>

CV* 10010

Example 1: State Diagram



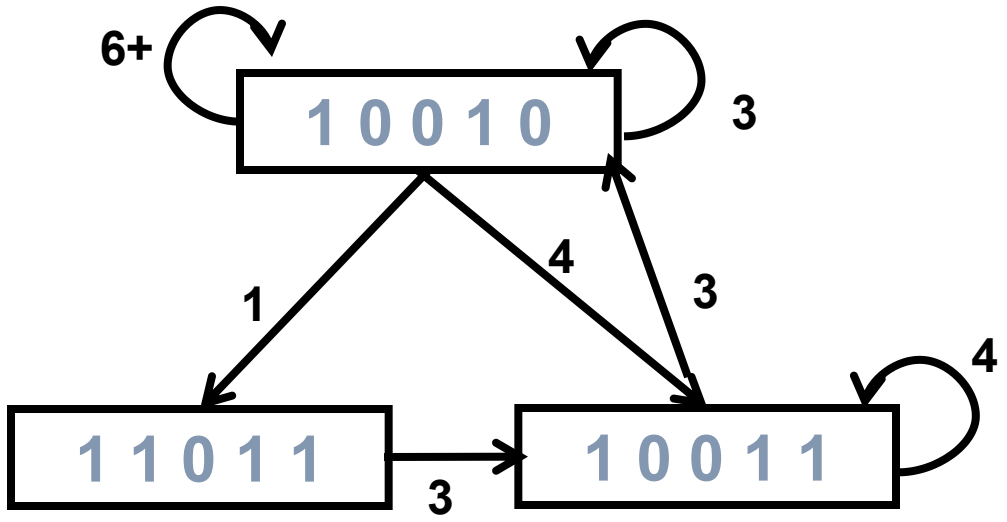
i =4

ICV – 10010 OR

CV_i – <u>00001</u>

CV* 10011

Example 1: State Diagram



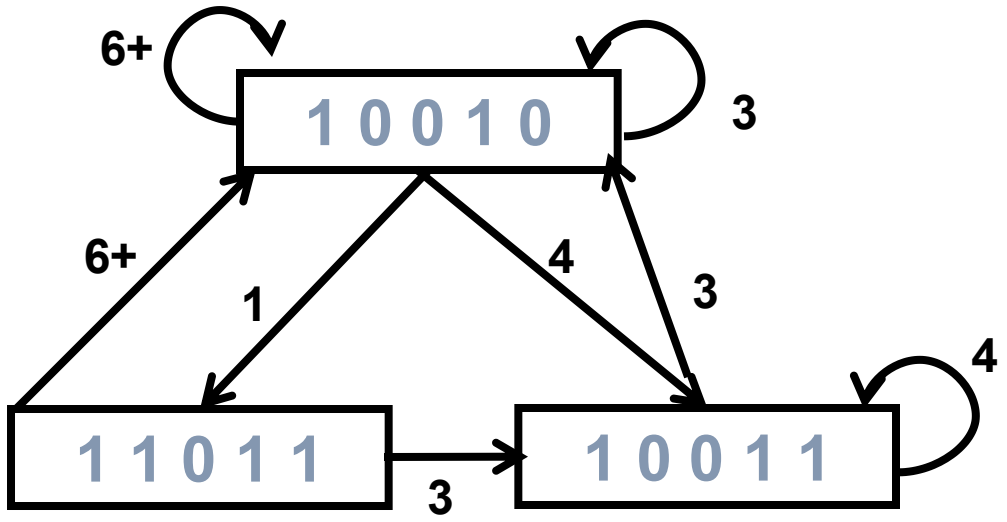
i =3

ICV – 10010 OR

CV_i – <u>00011</u>

CV* 10011

Example 1: State Diagram



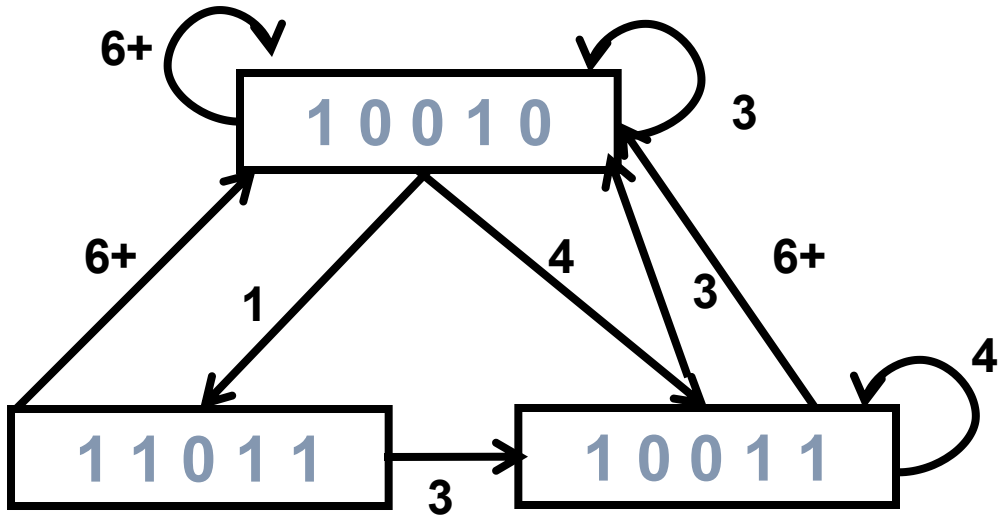
i = 5

ICV – 10010 OR

CV_i – <u>00000</u>

CV* 10010

Example 1: State Diagram



i = 5

ICV – 10010 OR

CV_i – <u>00000</u>

CV* 10010

Cycle, Average Latency and MAL

- A cycle in a state diagram is an alternating sequence of collision vectors and arcs: C0, a1, C1, ..., an, Cn
- Each arc ai connects collision vector Ci-1 to Ci
- All the collision vectors are distinct except the first and last
- A cycle is simply represented by a sequence of latencies of its arcs
- A Constant Cycle is a latency cycle which contains only one latency value
- The average latency for a cycle is determined by adding the latencies of the arcs of the cycle and then dividing it by the total number of arcs in the cycle
- Minimum Average Latency (MAL) is the minimum of all the average latencies of a state diagram

Example 1: Average Latency

- Let C0 is 10010, C1 is 11011 and C2 is 10011
- The cycle *C0, a1, C1, a2, C2, a3, C0*
- *a1 is an arc (latency = 1) from C0 to C1*
- *a2 is an arc (latency = 3) from C1* to *C2*
- *a3 is an arc (latency = 6) from C2* to *C0*
- *The cycle is represented as C=(1, 3, 6)*
- The average latency for a cycle is determined by adding the latencies of the arcs of the cycle and then dividing it by the total number of arcs in the cycle
- The cycle *C=(1, 3, 6) has the average latency:*

$$(1 + 3 + 6)/3 = 3.33$$

Example 1: Minimum Average Latency (MAL)

- The following are the average latencies of the different cycles in example 1
- 3/1= 3
- 6/1= 6
- (1 + 6)/2 =3.5
- (1 + 3 + 3)/3 =2.33
- (1 + 3 + 6)/3 = 3.33
- (4 + 3)/2 = 3.5
- (4 + 6) /2 = 5
- 4/1 = 4
- The minimum average latency is simply the minimum of these values
- So the MAL is 2.33

Thank you