Pipelined Architecture

CSEN 3104

Lecture 10

Dr. Debranjan Sarkar

Minimum Latency

- Although the cycle with the minimum average latency maximizes the throughput of the pipeline, sometimes a less efficient cycle may be chosen to reduce the implementation complexity of the pipeline's control circuit (i.e., a trade-off between time and cost)
- For example, the cycle *C=(1,3,3), which has the MAL of 2.33, requires a circuit that counts one unit of time,* then three units, again three units, and so on
- However, if it is acceptable to initiate an input datum after every 3 units of time, the complexity of the circuit will be reduced. Therefore, sometimes it may be necessary to determine the smallest latency that can be used for initiating input data at all times
- Such a latency is called the *minimum latency*

Example 1: Minimum Latency

- *One way to determine the minimum latency is to choose a cycle of* length 1 with the smallest latency from the state diagram. In this example, it is 3/1 = 3
- Another way is to find the smallest integer whose product with any arbitrary integer is not a member of the forbidden list.
- For example, for the forbidden list (2, 5), the minimum latency can be determined as follows:

Example 1: Minimum Latency

Minimum Latency	Times an integer	Product	Result
1	*1	=1	OK
1	*2	=2	No good
2	*1	=2	No good
3	*1	=3	OK
3	*2	=6	OK
4	*1	=4	OK
4	*2	=8	OK

- Forbidden Latency Set is { 2,5}
- Therefore the minimum latency for this pipeline is 3.

Simple cycle and Greedy cycle

- **Simple Cycle:** latency cycle in which each state is encountered only once.
- In this example, the simple cycles are (3),(6), (1,6), (4,6), (4,3), (1,3,3), (1,3,6) and (4)
- The cycle (1,3,4,3) is not a simple cycle as the state 10011 is encountered twice
- **Greedy Cycle:** A simple cycle is a greedy cycle whose edges are all made with minimum latencies from their respective starting states
- At least one of the greedy cycles will lead to MAL.
- In this example, the Greedy cycle is (1,3,3)
- In the above example, the cycle that offers MAL is (1, 3, 3)

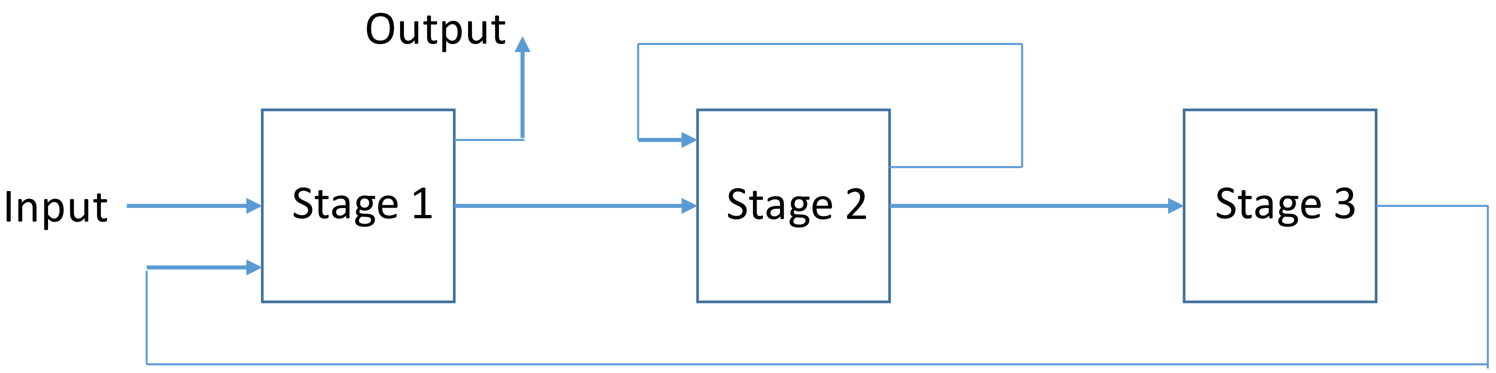
Upper and Lower Bounds of MAL

- Lower bound of MAL = the maximum number of checkmarks in any row of the Reservation Table
- MAL $\geq$ max no. of check marks in any row of the reservation table
- In this example, Lower bound of MAL = 2, since there are maximum 2 checkmarks in any row of the RT
- Upper bound of MAL= the number of 1's in the initial collision vector plus 1
- MAL $\leq$ avg latency of any greedy cycle

  $\leq$ no. of 1’s in initial collision vector + 1
- In this example, ICV is 10010, so number of 1’s = 2
- Upper bound of MAL = 2 + 1 = 3
- So $2 \leq$ MAL ≤ 3

Scheduling of Static Pipelines

Example 2



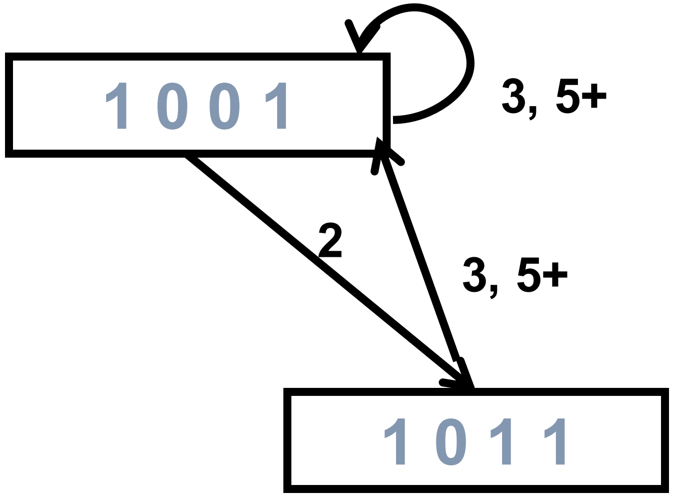
	1	2	3	4	5
Stage 1	X				X
Stage 2		X	X		
Stage3				X	

Forbidden Latencies: 1, 4

$$\begin{array}{lcccc} & C_4 & C_3 & C_2 & C_1 \\ \text{Initial Collision Vector:} & 1 & 0 & 0 & 1 \end{array}$$

Example 2

- State Diagram



- The following are the average latencies of the different cycles
- (2 + 3)/2 =2.5 (2 + 5)/2 =3.5 3/1= 3 5/1= 5
- The minimum average latency is 2.5

Example 2: Minimum Latency

- Although the cycle with the minimum average latency maximizes the throughput of the pipeline, sometimes a less efficient cycle may be chosen to reduce the implementation complexity of the pipeline's control circuit (i.e., a trade-off between time and cost)
- For example, the cycle *C=(2,3), which has the MAL of 2.5, requires a circuit that counts three units of time,* then two units, again three units, and so on.
- However, if it is acceptable to initiate an input datum after every three units of time, the complexity of the circuit will be reduced. Therefore, sometimes it may be necessary to determine the smallest latency that can be used for initiating input data at all times.
- Such a latency is called the *minimum latency*

Example 2: Minimum Latency

- One way to determine the minimum latency is to choose a cycle of length 1 with the smallest latency from the state diagram
- Another way is to find the smallest integer whose product with any arbitrary integer is not a member of the forbidden list.
- For example, for the forbidden list (1, 4), the minimum latency can be determined as follows:

Minimum Latency	Times an integer	Product	Result
1	* 1	= 1	No good
2	* 1	= 2	OK
2	* 2	= 4	No good
3	* 1	= 3	OK
3	* 2	= 6	OK
4	* 1	= 4	No good

- Therefore the minimum latency for this pipeline is 3

Example 2: Simple cycle and Greedy cycle

- **Simple Cycle:** latency cycle in which each state is encountered only once.
- In Example 2, the simple cycles are (3),(5), (2,3), (2,5)

- **Greedy Cycle:** A simple cycle is a greedy cycle whose edges are all made with minimum latencies from their respective starting states
- At least one of the greedy cycles will lead to MAL.
- In this example, the Greedy cycle is (2,3)
- In the above example, the cycle that offers MAL is (2, 3)

Example 2: Upper and Lower Bounds of MAL

- Lower bound of MAL = the maximum number of checkmarks in any row of the Reservation Table
- MAL $\geq$ max no. of check marks in any row of the reservation table
- In Example 2, Lower bound of MAL = 2, since there are maximum 2 checkmarks in any row of the RT
- Upper bound of MAL= the number of 1's in the initial collision vector plus 1
- MAL $\leq$ avg latency of any greedy cycle
  $\leq$ no. of 1’s in initial collision vector + 1
- In example 2, ICV is 1001, so number of 1’s = 2
- Upper bound of MAL = 2 + 1 = 3
- So 2 $\leq$ MAL $\leq$ 3

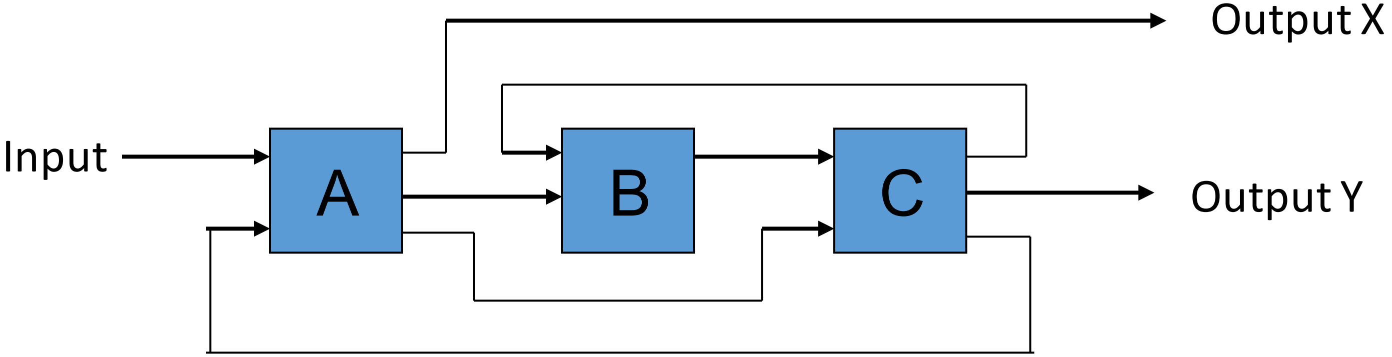
Example 3

Forbidden Latencies:
{5,7,2} ∪{2} ∪ {2,4} = {2,4,5,7}

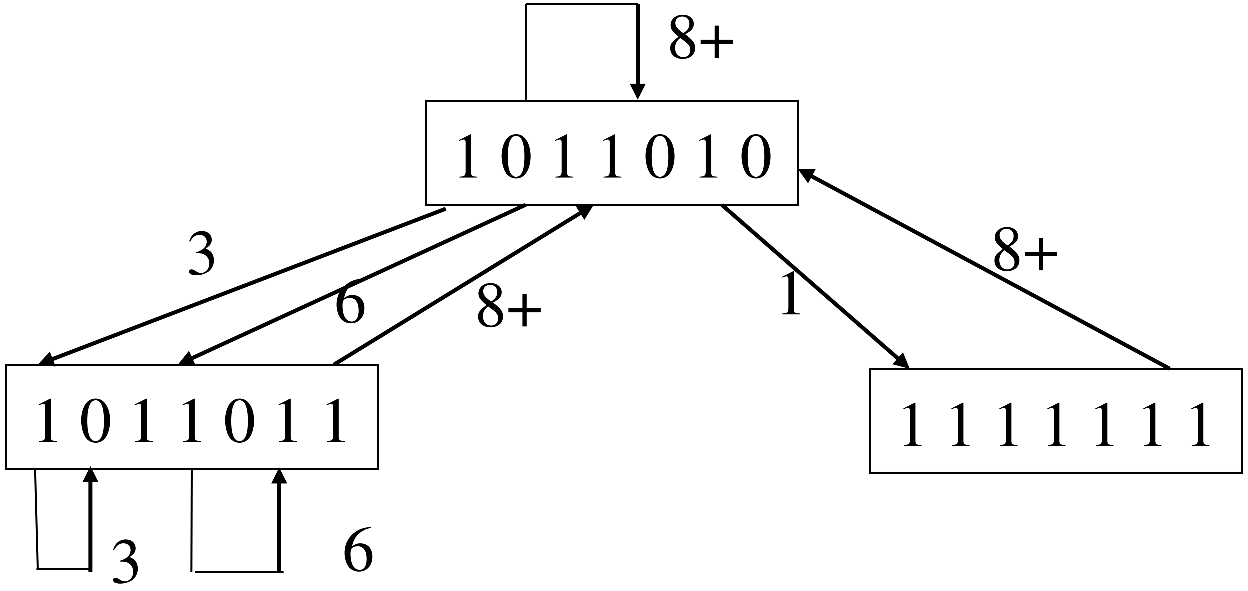
	1	2	3	4	5	6	7	8
A	X					X		X
B		X		X				
C			X		X		X	

Sequence for output X: A, B, C, B, C, A, C, A

Initial Collision Vector: $C_7C_6C_5\ C_4C_3C_2C_1$
1 0 1 1 0 1 0



Example 3: State Diagram for X



Example 3

- Simple Cycles: (3), (6), (8), (1, 8), (3, 8) and (6, 8) are simple cycles
- The cycle (6, 8, 1, 8) is not simple because the state (1011010) is encountered twice

- Greedy Cycles: The cycle (1, 8) and (3) are greedy cycles

- Minimum Average Latency (MAL): In greedy cycles (1, 8) and (3), the cycle (3) leads to MAL value 3

Example 3: Minimum Latency

Minimum Latency	Times an integer	Product	Result
1	*1	= 1	OK
1	*2	= 2	No good
2	*1	= 2	No good
3	*1	= 3	OK
3	*2	= 6	OK
3	*3	= 9	OK
4	*1	= 4	No good
5	*1	= 5	No good
6	*1	= 6	OK
6	*2	= 12	OK
7	*1	= 7	No good

- Forbidden Latency Set is { 2, 4, 5, 7}
- Therefore the minimum latency for this pipeline is 3.

Example 3: Upper and Lower Bounds of MAL

- Lower bound of MAL = the maximum number of checkmarks in any row of the Reservation Table
- In this example, Lower bound of MAL = 3, since there are maximum 3 checkmarks in any row of the RT
- Upper bound of MAL= the number of 1's in the initial collision vector plus 1
- MAL $\leq$ avg latency of any greedy cycle

  $\leq$ no. of 1’s in initial collision vector + 1
- IN this example, ICV is 1011010, so number of 1’s = 4
- Upper bound of MAL = 4 + 1 = 5
- So $3 \leq \text{MAL} \leq 5$

Assignment

Given the Reservation Table for output Y as follows:

	1	2	3	4	5	6
A	Y				Y	
B			Y			
C		Y		Y		Y

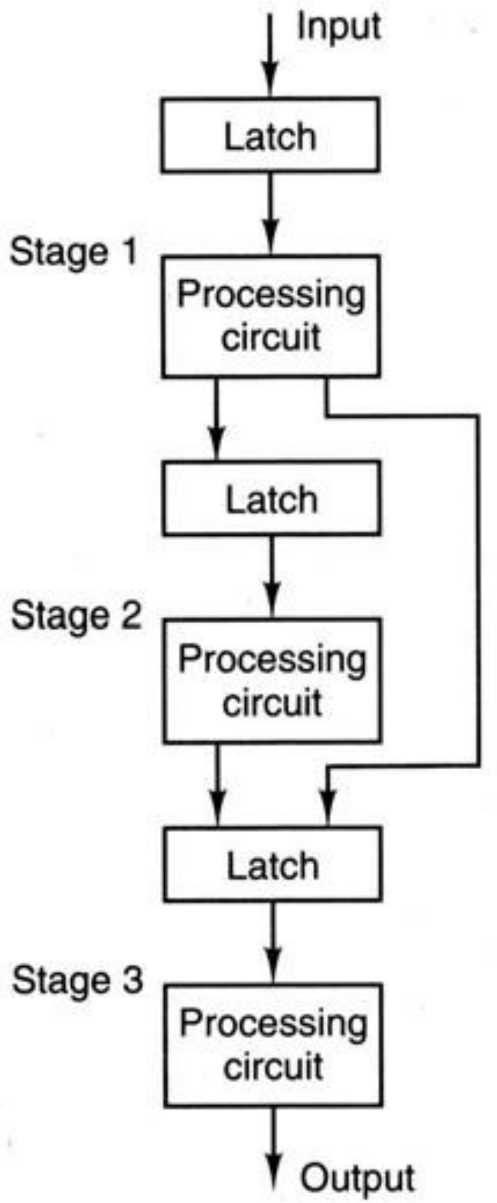
(a) Find out the (i) Forbidden latencies and (ii) Initial Collision Vector

(c) Draw the State Diagram for scheduling the pipeline

(d) Find out the simple cycle, greedy cycle and MAL

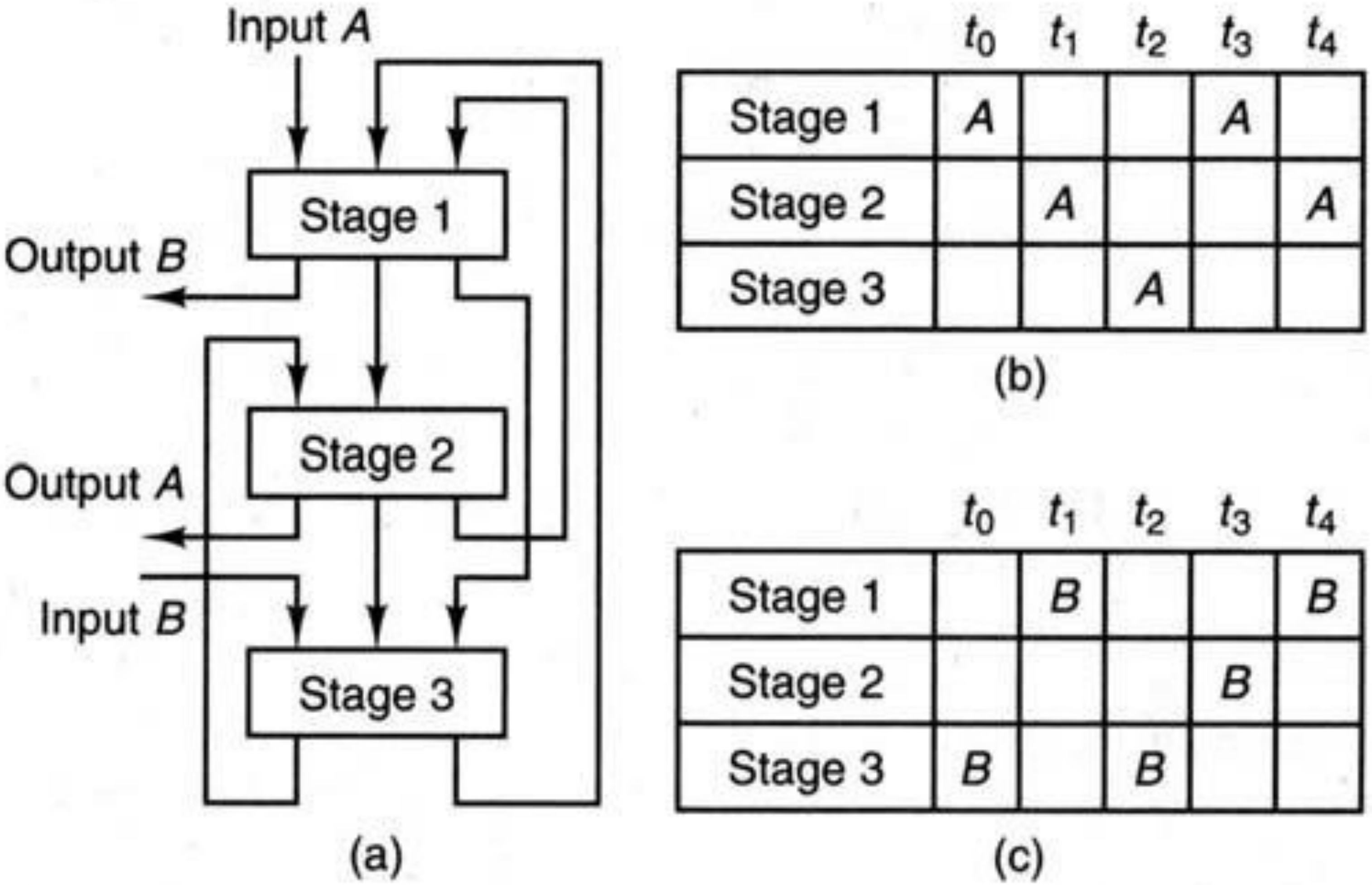
(e) What are bounds on MAL?

Dynamic Pipeline



- A dynamic pipeline can perform more than one operation at a time
- This 3-stage dynamic pipeline performs addition and multiplication on different data at the same time
- For multiplication, the input data must go through stages 1, 2, and 3
- For addition, the data only need to go through stages 1 and 3
- In dynamic pipeline, stage 1 can perform the first stage of the addition operation on an input data D1, and at the same time stage 3 can perform the last stage of the multiplication operation on another input data D2
- The time interval between the initiation of the inputs D1 and D2 to the pipeline should be such that they do not reach stage 3 at the same time; otherwise, there is a collision
- In general, in dynamic pipelines the mechanism that controls when data should be fed to the pipeline is much more complex than in static pipelines

A dynamic pipeline and its reservation tables



(a)

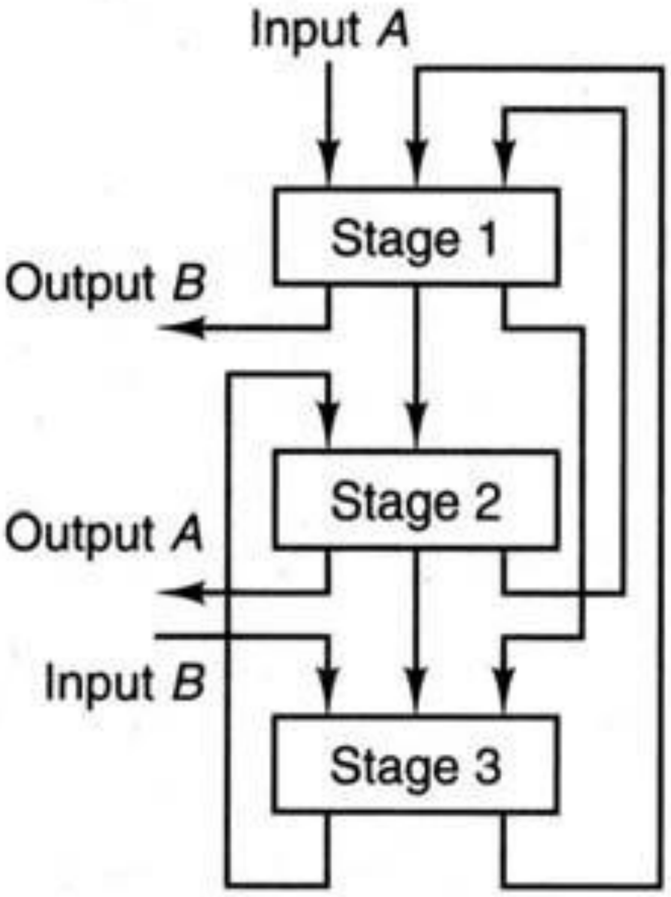
	t_0	t_1	t_2	t_3	t_4
Stage 1	*A*			*A*	
Stage 2		*A*			*A*
Stage 3			*A*		

(b)

	t_0	t_1	t_2	t_3	t_4
Stage 1		*B*			*B*
Stage 2				*B*	
Stage 3	*B*		*B*		

(c)

A dynamic pipeline and its reservation tables



(a)

	t_0	t_1	t_2	t_3	t_4
Stage 1	A			A	
Stage 2		A			A
Stage 3			A		

(b)

	t_0	t_1	t_2	t_3	t_4
Stage 1		B			B
Stage 2				B	
Stage 3	B		B		

(c)

Example of Overlaid Reservation Table

	t_0	t_1	t_2	t_3	t_4
Stage 1	A			A	
Stage 2		A			
Stage 3			A		A

For A

	t_0	t_1	t_2	t_3	t_4
Stage 1		B			B
Stage 2				B	
Stage 3	B		B		

For B

	t_0	t_1	t_2	t_3	t_4
Stage 1	A	B		A	B
Stage 2		A		B	
Stage 3	B		AB		A

Overlaid

Scheduling of Dynamic Pipeline

- When scheduling a static pipeline, only collisions between different input data for a particular function had to be avoided.
- With a dynamic pipeline, it is possible for different input data requiring different functions to be present in the pipeline at the same time.
- Therefore, collisions between these data must be considered as well.
- As with the static pipeline, however, dynamic pipeline scheduling begins with the compilation of a set of forbidden lists from function reservation tables.
- Next the collision vectors are obtained, and finally the state diagram is drawn.

Forbidden Lists of Dynamic Pipeline

- With a dynamic pipeline, in general there are p^2 forbidden lists and hence p^2 cross-collision vectors for a p-function pipeline
- In the earlier figure, there are 2 functions (A and B), so *the* number of forbidden lists is 4, denoted by *AA, AB, BA, and BB*
- Task A may collide with a previously initiated task B, if the latency between these two initiations is a member of the forbidden list
- AA = (2,3), AB = (1,2,4), BA = (2,4), and BB = (2,3),

Collision Vectors and Collision matrices

- The collision vectors are determined in the same manner as for a static pipeline
- 0 indicates a permissible latency and a 1 indicates a forbidden latency
- For the preceding example, the collision vectors are
- C_{AA} = (0 1 1 0), C_{BA} = (1 0 1 0), C_{AB} = (1 0 1 1), C_{BB} = (0 1 1 0)
- The collision vectors for the *A function form the collision matrix M_A,*

  that is, $M_A = [C_{AA}, C_{BA}]^T$
- The collision vectors for the *B function form the collision matrix M_B:*

  that is, $M_B = [C_{AB}, C_{BB}]^T$
- For the above collision vectors, the collision matrices are
- *$M_A = [0110, 1010]^T$* *$M_B = [1011, 0110]^T$*

State Diagram

- The state diagram for the dynamic pipeline is developed in the same way as for the static pipeline
- The resulting state diagram is much more complicated than a static pipeline state diagram due to the larger number of potential collisions

Problems of pipeline

- In practice, making the delays of pipeline stages equal is a complicated and time-consuming process
  - It is essential to maximize performance that the stages be close to balanced.
  - It is done for commercial processors, although it is not easy and cheap to do
- Another problem with pipelines is the overhead in term of handling exception or interrupts
  - A deep pipeline increases the interrupt handling overhead.

What is Microprocessor without Interlocked Pipelined Stages?

- One of the main disadvantages of pipelining is that there can be various dependencies that occur between stages:
  - Data hazards (RaW, WaW and WaR)
  - Control Hazards
  - Structural Hazards
- In an interlocked pipeline, hardware is used to check for hazards between stages. Sometimes this may lead to deadlock.
- In a non interlocked pipeline, no hardware is used to check for hazards. However, the burden of checking for dependencies is left to the programmer/compiler. If they fail to check for hazards, the program might use invalid data and even create a deadlock requiring a reboot.
- MIPS CPU was originally intended to simplify processor design by eliminating hardware interlocks between the five pipeline stages
- Later versions of the MIPS ISA did have interlocking however.

Thank you