EC6020 – ARSITEKTUR KOMPUTER LANJUT

TUGAS – 5

CHAPTER 6

PIPELINING AND SUPERSCALAR TECHNIQUES

SOAL-SOAL TENTANG *SUPERSCALAR AND SUPERPIPELINE DESIGN*

ARWIN
NIM. 232 06 008

MAGISTER TEKNIK ELEKTRO
SEKOLAH TINGGI ELEKTRONIKA DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2006

Problem 6.1 – Consider the execution of a program of **15,000 instructions** by a **linear pipeline** processor with a **clock rate of 25 Mhz**. Assume that the instruction pipeline has **five stages** and that **one instruction is issued per clock cycle**. The penalties due to branch instructions and out-of-sequence executions are ignored.

a. Calculate the speedup factor using this pipeline to execute the program as compared with the use of an equivalent nonpipelined processor with an equal amout of flow-through delay.

b. What are the efficiency and throughput of this pipelined processor ?

Answer :

Information we get are :

☞ $\boldsymbol{n = 15,000}$ instructions or tasks.

☞ $\boldsymbol{f = 25\ MHz}$.

☞ $\boldsymbol{k = 5}$ stages.

☞ $\boldsymbol{1}-$issued processor.

The Speedup $(\boldsymbol{S_k})$, Efficiency, $(\boldsymbol{E_k})$, and Throughput $(\boldsymbol{H_k})$ factors are :

$$\begin{aligned} \boldsymbol{S_k} &= \frac{\boldsymbol{T_1}}{\boldsymbol{T_k}} = \frac{\boldsymbol{nk\tau}}{\boldsymbol{k\tau + (n-1)\tau}} \\ &= \frac{\boldsymbol{nk}}{\boldsymbol{k + (n-1)}} \\ &= \frac{\boldsymbol{(15,000)(5)}}{\boldsymbol{5 + (15,000-1)}} \\ &= \frac{\boldsymbol{75,000}}{\boldsymbol{15,004}} \\ &= \boldsymbol{4,999} \end{aligned}$$

$$\begin{aligned} \boldsymbol{H_k} &= \frac{\boldsymbol{nf}}{\boldsymbol{k + (n-1)}} \\ &= \frac{\boldsymbol{(15,000)(25)}}{\boldsymbol{5 + (15,000-1)}} \\ &= \frac{\boldsymbol{375,000}}{\boldsymbol{15,004}} \\ &= \boldsymbol{24,99\ MIPS} \end{aligned}$$

$$\begin{aligned} \boldsymbol{E_k} &= \frac{\boldsymbol{S_k}}{\boldsymbol{k}} \\ &= \frac{\boldsymbol{4,999}}{\boldsymbol{5}} \\ &= \boldsymbol{0,999} \end{aligned}$$

Problem 6.2 – Study the **DEC Alpha architecture** in Example 6.13, find more information in the DEC Alpha handbook, and then answer the following questions with reasoning :

a. Analyze the scalability of the Alpha processor implementation in terms of superscalar degree and superpipeline degree.

b. Analyze the scalability of an Alpha-based multiprocessor system in terms of address space and multiprocessor support.

Answer :

The superpipelined superscalar DEC-21064-A architecture consists of :

☞ 32 64-bit integer registers (EBOX) with $\boldsymbol{k} = \mathbf{7}$ stages.

☞ 32 64-bit floating-point registers (FBOX) with $\boldsymbol{k} = \mathbf{10}$ stages.

☞ Clock rate, $\boldsymbol{f} = \mathbf{150}\ \boldsymbol{MHz}$.

☞ $\boldsymbol{m} = \mathbf{2}$.

☞ $\boldsymbol{H}_{\mathbf{7(int)}} = \mathbf{300}\ \boldsymbol{MIPS}$ and $\boldsymbol{H}_{\mathbf{10}(\boldsymbol{flop})} = \mathbf{150}\ \boldsymbol{Mflops}$.

☞ 34-bit address bus.

☞ 128-bit data bus.

It also features :

o Multiprocessor support (fast interlocking and interrupts).

o Multiple operating system.

o Possibility to handle more number of issues in future implementation.

a. From the superscalar and superpipeline perspectives, we must compare it with a scalar processor which has a clock rate $\mathbf{25}\ \boldsymbol{MHz}$. So that, we will have a superpipelining degree of $\boldsymbol{n} = \mathbf{150/25} = \mathbf{6}$. Combining it with $\boldsymbol{m} = \mathbf{2}$, we obtain a superpipelined superscalar machine with degree of $(\boldsymbol{m}, \boldsymbol{n}) = (\mathbf{2}, \mathbf{6})$. By using these values, the performance of DEC-21064 Alpha machine is :

- Execution time, $\boldsymbol{T}(\mathbf{2,6})$ is : (assume $\boldsymbol{N} = \mathbf{10,000}$ instructions)

$$\boldsymbol{T}(\boldsymbol{m,n}) = \boldsymbol{k} + \frac{\boldsymbol{N} - \boldsymbol{m}}{\boldsymbol{mn}}$$

$$\begin{aligned} \boldsymbol{T}(\mathbf{2,6}) &= \mathbf{7} + \frac{\mathbf{10,000} - \mathbf{2}}{(\mathbf{2})(\mathbf{6})} \\ &= \mathbf{7} + \frac{\mathbf{9,998}}{\mathbf{12}} = \mathbf{7} + \mathbf{833,2} \qquad \text{for EBOX} \\ &= \mathbf{840,2} \\ &\approx \mathbf{840}\ \boldsymbol{cycles} \end{aligned}$$

$$\begin{aligned} \boldsymbol{T}(\mathbf{2,6}) &= \mathbf{10} + \frac{\mathbf{10,000} - \mathbf{2}}{(\mathbf{2})(\mathbf{6})} \\ &= \mathbf{10} + \frac{\mathbf{9,998}}{\mathbf{12}} = \mathbf{10} + \mathbf{833,2} \qquad \text{for FBOX} \\ &= \mathbf{843,2} \\ &\approx \mathbf{843}\ \boldsymbol{cycles} \end{aligned}$$

- Speedup, $\boldsymbol{S}(\mathbf{2,6})$

$$\boldsymbol{S}(\boldsymbol{m,n}) = \frac{\boldsymbol{mn}(\boldsymbol{k} + \boldsymbol{N} - \mathbf{1})}{\boldsymbol{mnk} + \boldsymbol{N} - \mathbf{1}}$$

$$\begin{aligned} \boldsymbol{S}(\mathbf{2,6}) &= \frac{(\mathbf{2})(\mathbf{6})(\mathbf{7} + \mathbf{10,000} - \mathbf{1})}{(\mathbf{2})(\mathbf{6})(\mathbf{7}) + \mathbf{10,000} - \mathbf{1}} \\ &= \frac{(\mathbf{12})(\mathbf{10,006})}{\mathbf{84} + \mathbf{9,999}} \qquad \text{for EBOX} \\ &= \frac{\mathbf{120,072}}{\mathbf{10,083}} \\ &= \mathbf{11.91} \\ &\approx \mathbf{12}\ \boldsymbol{times} \end{aligned}$$

$$\begin{aligned} S(2,6) &= \frac{(2)(6)(10+10{,}000-1)}{(2)(6)(10)+10{,}000-1} \\ &= \frac{(12)(10{,}009)}{120+9{,}999} \\ &= \frac{120{,}108}{10{,}119} \qquad \text{for FBOX} \\ &= 11.87 \\ &\approx 12 \; times \end{aligned}$$

The implementation of a superscalarity requires more transistors whereas the implementation of a superpipelinarity requires faster transistors and more careful circuit design to minimize the effects of clock skews. Those requirements have been coped with the fast-growing of IC fabrication technology by the time to time. It means that as long as the technology grows, there always opportunities to design a better machine in the future in terms of scalability.

b. We also have noted the scalability possibilities of the DEC-21064-A from multiprocessor support and address space perspectives.

- For multiprocessor system implementation, all processors need a shared memory to share data between them. With 128-bit data bus, it was designed to handle $\mathbf{2^{128}}$ data ($\mathbf{2^{64}}$ integer data and $\mathbf{2^{64}}$ floating-point data) which was very advanced in its time. From this point, we can see that the machine has a large memory capacity which means DEC-21064-A was prepared for multiprocessor support purpose.

- The DEC-21064-A was also equipped with 34-bit address bus that could cover $\mathbf{2^{34}}$ machine addressing. A shared-memory multiprocessor system will share their memory locations to other processors. This feature enables processors to exchange data via a large shared-memory.

Problem 6.4 – Find the **optimal number of pipeline stages** $\boldsymbol{k_0}$ given in Eq. 6.7 using the performance/cost ratio (PCR) given in Eq. 6.6.

Answer :

Eq. 6.6 defines that :

$\boldsymbol{PCR = \frac{f}{c + kh}}$ where $\boldsymbol{f}$ is clock rate, $\boldsymbol{c}$ is the cost of all logic stages and $\boldsymbol{h}$ represents the cost of each latch on a $\boldsymbol{k}-$stages pipeline. The optimal pipeline stages – in term of PCR – can be formulated :

$$\boldsymbol{PCR = \frac{f}{c + kh} \quad \leftrightarrow \quad PCR\left(c + kh\right) = f}$$

$$\boldsymbol{cPCR + khPCR = f}$$

$$\boldsymbol{khPCR = f - cPCR}$$

$$\boldsymbol{k = \frac{f - cPCR}{hPCR} \quad \rightarrow \quad k_0}$$

Problem 6.5 – Prove the **lower bound** and **upper bound** on the **minimal average latency (MAL)** as specified in page 277.

Answer :

[Shar72] states the MAL as followed :

1. The **MAL is lower-bounded** by the **maximum number of checkmarks in any row** of the reservation table.
2. The average latency of any greedy cycle is upper-bounded by **the number of 1's in the initial collision vector plus 1**. This is also **an upper bound on the MAL**.

Proof :

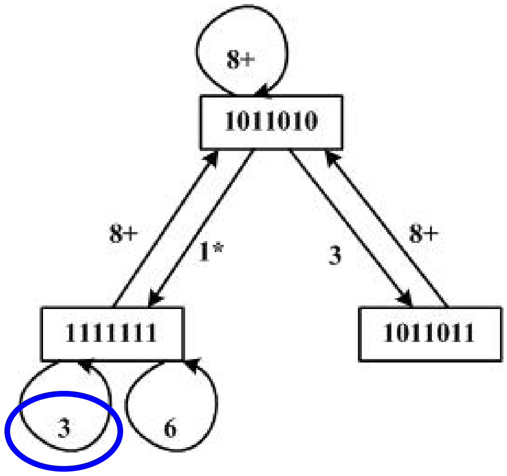
Take the case on the function X's reservation table on page 271 as followed :

	1	2	3	4	5	6	7	8
S1	X	🟥	🟥	🟥	🟥	X	🟥	X
S2		X	🟥	X				
S3			X	🟥	X	🟥	X	

From the table we see that S1 has 3 checkmarks, S2 has 2 checkmarks and S3 has 3 checkmarks which means that the minimum MAL for function X is 3. The forbidden latencies are 2, 4, 5 and 7 while the permissible latencies are 1, 3 and 6. We, then, can obtain the collision vector $C_X = \mathbf{1011010}$. Let's shift this vector bit-by-bit to the right to find the state transition diagram.

1st shift		3rd shift		6th shift	
shifted bit	**0101101**	***shifted bit***	**0001011**	***shifted bit***	**0000001**
$\boldsymbol{C_x}$	**1011010**	$\boldsymbol{C_x}$	**1011010**	$\boldsymbol{C_x}$	**1011010**
new value	**1111111**	***new value***	**1011011**	***new value***	**1011011**

3rd shift from **1011011**		6th shift from **1011011**	
shifted bit	**0001011**	***shifted bit***	**0000001**
$\boldsymbol{C_x}$	**1011010**	$\boldsymbol{C_x}$	**1011010**
new value	**1011011**	***new value***	**1011011**



a. The **simple cycles** are (3), (6), (1,8) and (3,8), such that the **greedy cycles** are (3) and (1,8). The lowest greedy cycle is the **MAL** and in this case is (3), so the **MAL is 3**…………… (*1st statement proved*).

b. The **collision vector** $\boldsymbol{C_X = 1011010}$ has 4 1-bits and the maximum MAL is the number of this 1-bit plus 1 or equal to 5. The largest average latency of any greedy cycle on function X's state diagram is **3.5** which is taken from greedy cycle (1,8) ……………….. (*2nd statement proved*).

Problem 6.6 – Consider the following reservation table for a **four-stage pipeline** with a **clock cycle** $\tau = \mathbf{20}$ ns.

	1	2	3	4	5	6
S1	X	🟥	🟥	🟥	🟥	X
S2		X	🟥	X		
S3			X			
S4				X	X	

a. What are the forbidden latency and the intial collision vector ?

b. Draw the state transition diagram for the scheduling the pipeline.

c. Determine the MAL associated with the shortest greedy cycle.

d. Determine the pipeline throughput corresponding to the MAL and given τ .

e. Determine the lower bound on the MAL for this pipeline.

Answer :

a. The **forbidden latencies** are 1, 2, and 5 (S1 : 5; S2 : 2; S3 : 0; S4: 1) and the c**ollision vector** is, $C_x = C_5C_4C_3C_2C_1 = \mathbf{10011}$. The **permissible latencies** are 3 and 4.

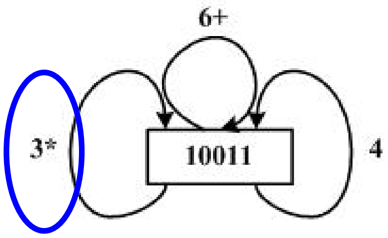
b. State diagram can be obtained by tracing each C_x shift as followed :

3rd shift

shifted bit	**00010**
C_x	**10011**
new value	**10011**

4th shift

shifted bit	**00001**
C_x	**10011**
new value	**10011**



c. The **greedy cycle** is 3 and so is the **MAL**.

d. The pipeline **throughput** according to :

1) The MAL is $\frac{\mathbf{1}}{\mathbf{3}} = \mathbf{0.33}$

2) The τ is $\frac{\mathbf{1}}{\mathbf{2}} = \mathbf{0.5}$ (at clock cycle 4 there are two task initiated)

e. The **MAL's** lower bound of this function according to [Shar72] is **2** (the maximum number of checkmarks in any row). The optimal latency have not been found – needs a modification on the reservation table.

Problem 6.7 – You are allowed to insert one noncompute delay stage into the pipeline in Problem 6.6 to make latency of 1 permissible in the shortest greedy cycle. The purpose is to yield a new reservation table leading to an optimal latency equal to the lower bound.

a. Show the modified reservation table with five rows and seven columns.

b. Draw the new state transition diagram for obtaining the optimal cycle.

c. List all the simple cycles and greedy cycles from the state diagram.

d. Prove that the new MAL equals to the lower bound.

e. What is the optimal throughput of this pipeline ? Indicate the percentage of throughput improvement compared with that obtained in part (d) of Problem 6.6.

Answer :

a. The modified reservation table is :

	1	2	3	4	5	6	7
S1	X	🟥	🟥	🟥	🟥	🟥	X
S2		X	🟥	X			
S3			X				
S4				X	🟥 →	X_1	
D					D_1		

The **forbidden latencies** are 2 and 6, then the **collision vector** is $\boldsymbol{C_x} = \mathbf{100010}$. The **permissible latencies** are 1, 3, 4, and 5.

b. State diagram can be obtained by tracing each $\boldsymbol{C_x}$ shift as followed :

1st shift

shifted bit **010001**
$\boldsymbol{C_x}$ **100010**
new value **110011**

3rd shift

shifted bit **000100**
$\boldsymbol{C_x}$ **100010**
new value **100110**

4th shift

shifted bit **000010**
$\boldsymbol{C_x}$ **100010**
new value **100010**

5th shift

shifted bit **000001**
$\boldsymbol{C_x}$ **100010**
new value **100011**

1-shift from 3rd shift		1-shift from 4th shift		1-shift from 5th shift	
shifted bit	**010011**	***shifted bit***	**010001**	***shifted bit***	**010001**
$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**
new value	**110011**	***new value***	**110011**	***new value***	**110011**

3-shift from 1st shift		4-shift from 1st shift		5-shift from 1st shift	
shifted bit	**000110**	***shifted bit***	**000011**	***shifted bit***	**000001**
$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**
new value	**100110**	***new value***	**100011**	***new value***	**100011**

3-shift from 3rd shift		4-shift from 3rd shift		5-shift from 3rd shift	
shifted bit	**000100**	***shifted bit***	**000010**	***shifted bit***	**000001**
$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**
new value	**100110**	***new value***	**100010**	***new value***	**100011**

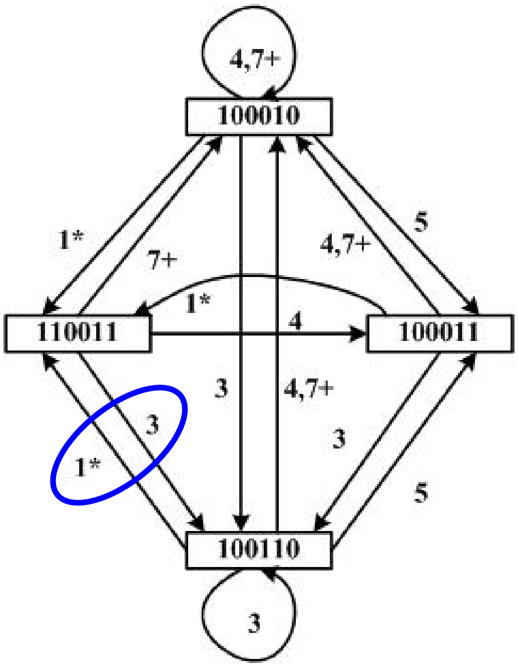
3-shift from 4th shift		4-shift from 4th shift		5-shift from 4th shift	
shifted bit	**000100**	***shifted bit***	**000010**	***shifted bit***	**000001**
$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**
new value	**100110**	***new value***	**100010**	***new value***	**100011**

3-shift from 5th shift		4-shift from 5th shift		5-shift from 5th shift	
shifted bit	**000100**	***shifted bit***	**000010**	***shifted bit***	**000001**
$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**	$\boldsymbol{C_x}$	**100010**
new value	**100110**	***new value***	**100010**	***new value***	**100011**

c. The **simple cycles** are (3), (4), (5), (1,3), (1,4), (1,7), (3,4), (3,5), (3,7), (4,5), (5,7), (1,3,4), (1,3,7), (1,4,4), (1,4,7), (3,5,4), (3,5,7), and (5,1,7). The **greedy cycles** are (3) and (1,3).

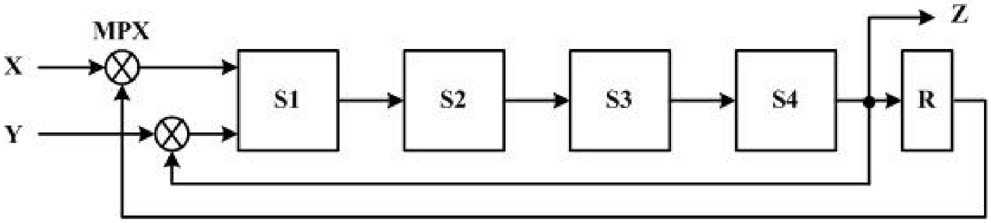
d. The greedy cycle **(1,3)** has the lowest average latency which is equal to **2**. This greedy cycle leads to the MAL of this pipeline machine. It can be seen on the reservation table that **this MAL is equal to the maximum number of checkmarks in any row in the reservation table** …. (proved).

e. The **throughput** is $\frac{\mathbf{1}}{\boldsymbol{MAL}} = \frac{\mathbf{1}}{\mathbf{2}} = \mathbf{0.5}$ and it has improvement percentage of $\frac{\mathbf{0.5}}{\mathbf{0.33}} = \mathbf{1{,}52}\ \boldsymbol{x}\ \mathbf{100\%} = \mathbf{152\%}$.



The state diagram from the modified reservation table.

Problem 6.8 – Consider an **adder pipeline** with **four stages** which consists of input lines X and Y and output line Z. The pipeline has a register R as its **output** where the **temporary result** can be stored and fec back to S1 at a later point in time. The inputs X and Y are multiplexed with the **outputs** R and Z.



a. Assume the elements of the vector A are fed into the pipeline through **input X**, one element per cycle. What is the **minimum number of clock cycles** required to compute the sum of an N-element vector $\boldsymbol{A : s = \sum_{I=1}^{N} A(I)}$? In the absence of an operand, a value 0 is input into the pipeline by default. Neglect the pipeline's setup time.

b. Let $\boldsymbol{\tau}$ be the clock period of the pipelined cycle. Consider an equivalent nonpipelined adder with a flow through delay of $\boldsymbol{4\tau}$. Find the actual **speedup** $\boldsymbol{S_4(64)}$ and **efficiency** $\boldsymbol{\eta_4(64)}$ of using the above pipeline adder for $\boldsymbol{N = 64}$.

c. Find the **maximum speedup** $\boldsymbol{S_4(\infty)}$ and the **efficiency** $\boldsymbol{\eta_4(\infty)}$ when N tends to infinity.

d. Find $\boldsymbol{N_{\frac{1}{2}}}$, the **minimun vector length** required to achieve half of the maximum speedup.

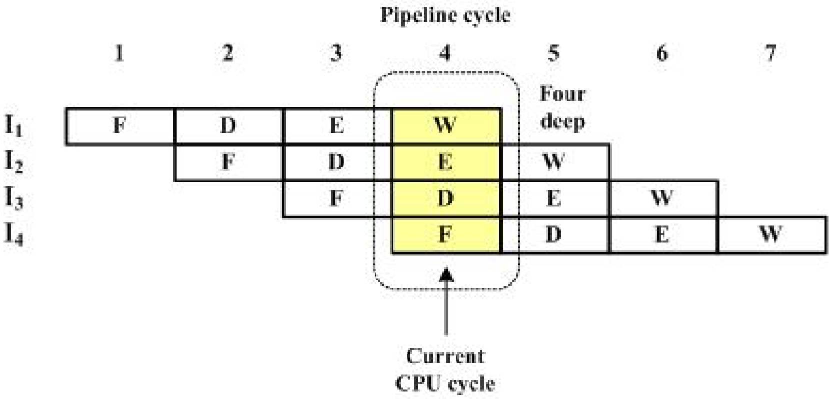
Answer :

a. The minimum number of clock cycle can be obtained by writing its assembly code, creating its reservation table and trying to imagine its instruction scheduling. The process in obtaining the outputs Z and R from input line X or Y via the same manner, such that the codes are not much different except there is a line code to execute store command when using input line X. The code is as follow :

```
I1 :     Load ACC, R          / ACC ← (R)/
I2 :     Inc, R               / R ← (R) + 1 /
I3 :     Add ACC, R           / ACC ← (ACC) + (R)/
I4 :     Store R, ACC         / R ← (ACC)/
```

The reservation table is :

	1	2	3	4
S1	X			
S2		X		
S3			X	
S4				X



	1	2	3	4	5	6	7
I_1	F	D	E	W			
I_2		F	D	E	W		
I_3			F	D	E	W	
I_4				F	D	E	W

So that, the **minimum clock cycles** required to complete one process is **7 clock cycles**.

b. The actual speedup and efficiency are :

For a nonpipelined processor, the CPU time is $\boldsymbol{T_1 = nk\tau}$ where $\boldsymbol{n = 64}$ and $\boldsymbol{k\tau = 4\tau}$, then $\boldsymbol{T_1 = 64.4\tau = 256\tau}$. For a pipeline processor, the CPU time is $\boldsymbol{T_k = \left[k + (n-1)\right]\tau}$, so that

$$\boldsymbol{T_4 = \left[4 + (64-1)\right]\tau}$$
$$\boldsymbol{= 67\tau}$$

The **actual speedup**, $\boldsymbol{S_4}$ is

$$S_k = \frac{T_1}{T_k} \quad \leftrightarrow \quad S_4 = \frac{256\tau}{67\tau} = 3.82$$

and the **actual efficiency**, $\boldsymbol{E_4}$ is

$$E_k = \frac{S_k}{k} \quad \leftrightarrow \quad E_4 = \frac{3.82}{4} = 0.96 \text{ or } 96\%$$

c. The **maximum speedup and efficiency** if $N \to \infty$ are

$$S_\infty = \frac{(\infty)(4)}{4+(\infty-1)} = \frac{4\infty}{\infty} \quad \text{(this is an ideal condition)}$$
$$= 4$$

$$E_\infty = \frac{S_\infty}{4} = 1 \text{ (this is an ideal condition)}$$

d. The **minimum vector length** to achieve half of the maximum speedup $(S_4 = 2)$ is

$$S_4 = \frac{Nk}{k+(N-1)} \quad \leftrightarrow \quad kS_4 + NS_4 - S_4 = Nk$$
$$S_4(k-1) = N(k-S_4)$$

$$N = \frac{S_4(k-1)}{k-S_4}$$

$$N = \frac{2(4-1)}{4-2} = \frac{6}{2} = 3$$

The **minimum vector length** required is **3**.

Problem 6.9 – Consider the following **pipeline** reservation table :

	1	2	3	4
S1	X	🟥	🟥	X
S2		X		
S3			X	

a. What are the forbidden latency ?

b. Draw the state transition diagram.

c. List all the simple cycles and greedy cycles.

d. Determine the optimal contant latency cycle and the minimal average latency (MAL).

e. Let the pipeline clock period be $\tau = 20$ ns. Determine the throughput of this pipeline.

Answer :

a. The **forbidden latency** is 3. From the table we can obtain the **collision vector**, $C_x = C_3C_2C_1 = 100$. The **permissible latencies** are 1 and 2.

b. State diagram can be obtained by tracing each C_X shift as followed :

1st shift

$$\begin{array}{ll} \textit{shifted bit} & \mathbf{010} \\ C_x & \mathbf{100} \\ \hline \textit{new value} & \mathbf{110} \end{array}$$

2nd shift

$$\begin{array}{ll} \textit{shifted bit} & \mathbf{001} \\ C_x & \mathbf{100} \\ \hline \textit{new value} & \mathbf{101} \end{array}$$

3rd shift

$$\begin{array}{ll} \textit{shifted bit} & \mathbf{000} \\ C_x & \mathbf{100} \\ \hline \textit{new value} & \mathbf{100} \end{array}$$

4th shift

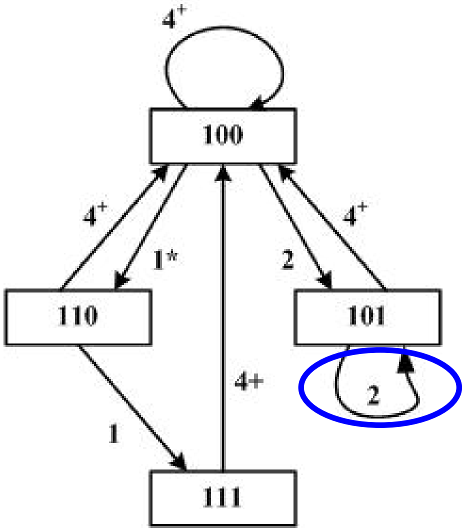
$$\begin{array}{ll} \textit{shifted bit} & \mathbf{000} \\ C_x & \mathbf{100} \\ \hline \textit{new value} & \mathbf{100} \end{array}$$

1st shift from **110**

$$\begin{array}{ll} \textit{shifted bit} & \mathbf{011} \\ C_x & \mathbf{100} \\ \hline \textit{new value} & \mathbf{111} \end{array}$$

2nd shift from **110**

$$\begin{array}{ll} \textit{shifted bit} & \mathbf{001} \\ C_x & \mathbf{100} \\ \hline \textit{new value} & \mathbf{101} \end{array}$$



c. The **simple cycles** are (2), (4), (1,4), (2,4), and (1,1,4). The **greedy cycles** are (2) and (1,4)

d. The **optimal constant latency** is (2), which is equal to MAL.

e. The maximum **throughput** is for this is liner pipeline is :

$$H_k = \frac{n}{\left[k + \left(n-1\right)\right]\tau}$$

$$H_3 = \frac{2}{\left[3 + \left(2-1\right)\right]20.10^{-9}}$$

$$= \frac{1}{40.10^{-9}}$$

$$= 25\ MIPS$$

Problem 6.10 – Consider the **five-staged pipelined** processor specified by the following reservation table:

	1	2	3	4	5	6
S1	X	🟥	🟥	🟥	🟥	X
S2		X	🟥	🟥	X	
S3			X			
S4				X		
S5		X	🟥	🟥	🟥	X

a. List the set of forbidden latencies and the collision vector.

b. Draw the state transition diagram showing all possible initial sequence (cycles) without causing a collision in the pipeline.

c. List all the simple cycles from the state diagram.

d. Identify the greedy cycles among the simple cycles.

e. What is the MAL of this pipeline ?

f. What is the minimum allowed constant cycle in using this pipeline ?

g. What will be the maximum throughput of this pipeline ?

h. What will be the throughput if the minimum constant cycle is used ?

Answer :

a. The **forbidden latencies** are 3, 4, and 5 (S1 ; 5; S2 : 3; S3 : 0; S4 : 0 and S5 : 4), so that the **collision vector** is $\boldsymbol{C_X = 11100}$ where the **permissible latencies** are 1 and 2.

b. State diagram can be obtained by tracing each $\boldsymbol{C_X}$ shift as followed :

1st shift

$$\begin{array}{ll} \textbf{\textit{shifted bit}} & \textbf{01110} \\ \boldsymbol{C_x} & \textbf{11100} \\ \hline \textbf{\textit{new value}} & \textbf{11110} \end{array}$$

2nd shift

$$\begin{array}{ll} \textbf{\textit{shifted bit}} & \textbf{00111} \\ \boldsymbol{C_x} & \textbf{11100} \\ \hline \textbf{\textit{new value}} & \textbf{11111} \end{array}$$

1-shift from 2nd shift

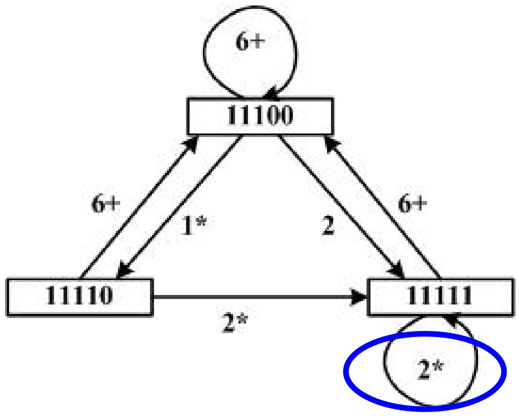
$$\begin{array}{ll} \textbf{\textit{shifted bit}} & \mathbf{01111} \\ \mathbf{\mathit{C_x}} & \mathbf{11100} \\ \hline \textbf{\textit{new value}} & \mathbf{11111} \end{array}$$

2-shift from 2nd shift

$$\begin{array}{ll} \textbf{\textit{shifted bit}} & \mathbf{00111} \\ \mathbf{\mathit{C_x}} & \mathbf{11100} \\ \hline \textbf{\textit{new value}} & \mathbf{11111} \end{array}$$

2-shift from 1st shift

$$\begin{array}{ll} \textbf{\textit{shifted bit}} & \mathbf{00111} \\ \mathbf{\mathit{C_x}} & \mathbf{11100} \\ \hline \textbf{\textit{new value}} & \mathbf{11111} \end{array}$$



c. The **simple cycles** are (2), (6), (1,6), and (2,6).

d. **The greedy cycles** are (2) and (1,6).

e. According to the lowest greedy cycle's average latency, the **MAL** is 2.

f. The **greedy cycle** is also the constant cycle which is equal to the MAL.

g. The maximum throughput is $\frac{\mathbf{1}}{\boldsymbol{MAL}} = \frac{\mathbf{1}}{\mathbf{2}} = \mathbf{0.5}$ or only $\mathbf{50\%}$.

h. The **minimum constant cycle** is **2**, so that the he maximum throughput does not change, only $\mathbf{50\%}$

Problem 6.11 – The following assembly code is to be executed in a three-stage pipelined processor with **hazard detection** and resolution in each stage. The stage are instruction fetch, operand fetch (one or more as required), and execution (including write-back operation). Explain **all possible hazards** in the execution of the code.

```
I1 :  Inc R0                      / R0 ← (R0)+ 1/
I2 :  Mul ACC, R0                 / ACC ← (ACC) x (R0)/
I3 :  Store R1, ACC               / R1 ← (ACC)/
I4 :  Add ACC, R0                 / ACC ← (ACC) + (R0)/
I5 :  Store M, ACC                / M ← (ACC)/
```

Answer :

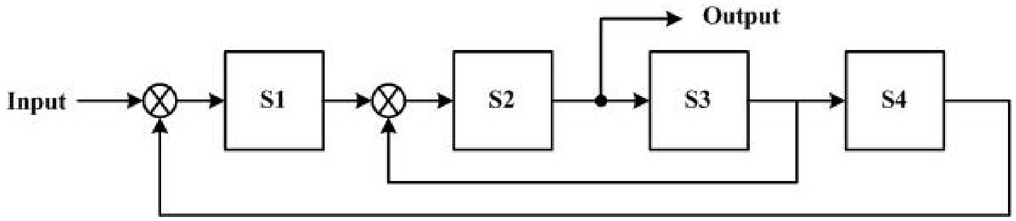
Analyzing ……………….

a. **RAW hazards** are I1- I2, I1 - I4, I2 - I3, I2 - I4, I2 - I5, and I4 - I5.

b. **WAW hazard** is I2 – I4.

c. **WAR hazards** are I3 – I4 and I2 – I4.

Problem 6.13 – Consider the following pipelined processor with **four stages**. This pipeline has a total **evaluation time of six clock cycles**. All successor stages must be used after each clock cycle.



a. Specify the reservation table for this pipeline with six columns and four fows.

b. List the set of forbidden latencies between initiations.

c. Draw the state diagram which shows all possible latency cycles.

d. List all greedy cycles from the state diagram.

e. What is the value of the minimal average latency (MAL) ?

f. What is the maximal throughput of this pipeline ?

Answer :

a. Reservation table

	1	2	3	4	5	6
S1	X					
S2		X	🟥	X		
S3			X	🟥	X	
S4				X	🟥	X

b. The **forbidden latencies** are 2 and 4 with a **collision vector**, $C_X = \mathbf{01010}$. The **permissible latencies** are 1, 3 and 5.

c. State diagram can be obtained by tracing each C_x shift as followed :

1st shift		3rd shift		5th shift	
shifted bit	**00101**	***shifted bit***	**00001**	***shifted bit***	**00000**
C_x	**01010**	C_x	**01010**	C_x	**01010**
new value	**01111**	***new value***	**01011**	***new value***	**01010**

1-shift from 3rd shift

shifted bit	**00101**
C_x	**01010**
new value	**01111**

1-shift from 5th shift

shifted bit	**00101**
C_x	**01010**
new value	**01111**

3-shift from 3rd shift

shifted bit	**00001**
C_x	**01010**
new value	**01011**

5-shift from 3rd shift

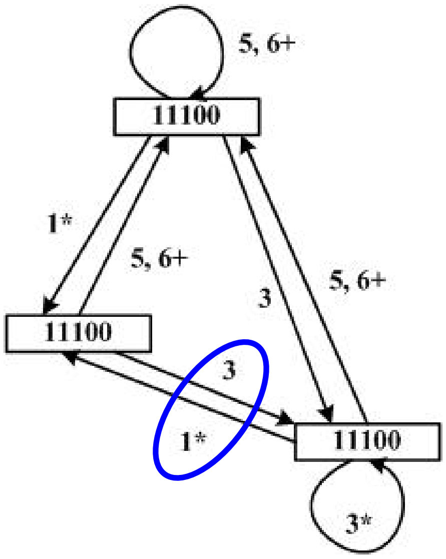
shifted bit	**00000**
C_x	**01010**
new value	**01010**

3-shift from 5th shift

shifted bit	**00001**
C_x	**01010**
new value	**01011**

5-shift from 5th shift

shifted bit	**00000**
C_x	**01010**
new value	**01010**

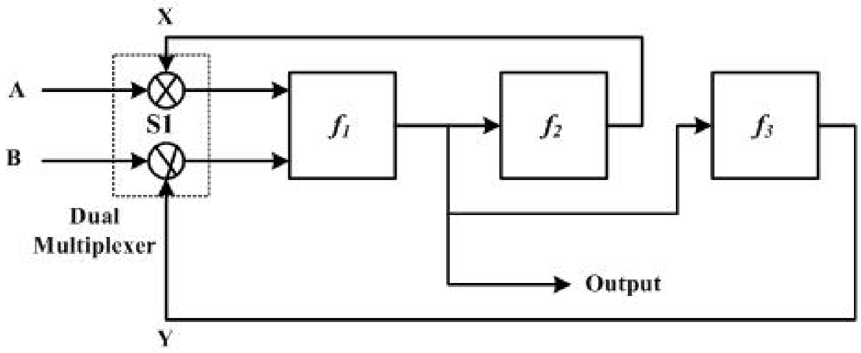


d. The **greedy cycles** are (3), (5), (6), (1,3), (1,5), (1,6), (3,5), (3,6), (1,3,5) and (1,3,6).

e. The value of **MAL is 2** according to the **greedy cycle (1,3)** obtained from the state diagram.

f. The maximum throughput is the inverse of MAL or $\frac{\mathbf{1}}{\mathbf{2}} = \mathbf{0.5}$ or $\mathbf{50\%}$.

Problem 6.14 – Three functional pipelines $f_1, f_2,$ and f_3 are characterized by the following reservation tables. Using these three pipelines, a composite pipeline network is formed below :



Each task going through this composite pipeline uses the pipeline in the following **order**: f_1 first, f_2 and f_3 next, f_1 again, and then the output is obtained. The dual multiplexer selects a pair of inputs, (A,B) or (X,Y), and feeds them into the input of f_1. The use of composite pipeline is described by the combined reservation table.

	1	2	3	4
S1	X			
S2		X		
S3			X	X

	1	2	3	4
T1	X			X
T2		X		
T3			X	

	1	2	3	4
U1	X		X	
U2				X
U3		X		

a. Complete the reservation table for this composite pipeline.

b. Write the forbidden list and the intial collison vector.

c. Draw a state diagram clearly showing all latency cycles.

d. List all simple cycles and greedy cycles.

e. Calculate the MAL and the maximal throughput of this composite pipeline.

Answer :

a. The reservation table is :

	1	2	3	4	5	6	7	8	9	10	11	12
S1	X	🟥	🟥	🟥	🟥	🟥	🟥	🟥	X			
S2		X	🟥	🟥	🟥	🟥	🟥	🟥	🟥	X		
S3			X	X	🟥	🟥	🟥	🟥	🟥	🟥	X	X
T1					X	🟥	🟥	X				
T2						X						
T3							X					
U1					X	🟥	X					
U2								X				
U3						X						

b. The **forbidden latencies** are 1, 2, 3, 7, 8, and 9 which create a **collision vector**, $\boldsymbol{C_X} = \mathbf{00111000111}$. The **permissible latencies** are 4, 5, 6, 10, and 11.

	Latencies
S1	8
S2	8
S3	1, 7, 8, 9

	Latencies
T1	3
T2	0
T3	0

	Latencies
U1	2
U2	0
U3	0

c. State diagram can be obtained by tracing each $\boldsymbol{C_X}$ shift as followed :

4th shift

shifted bit	**00000011100**
$\boldsymbol{C_x}$	**00111000111**
new value	**00111011111**

5th shift

shifted bit	**00000001110**
$\boldsymbol{C_x}$	**00111000111**
new value	**00111001111**

6th shift

shifted bit	**00000000111**
$\boldsymbol{C_x}$	**00111000111**
new value	**00111000111**

return to initial state

10th shift

shifted bit	**00000000000**
$\boldsymbol{C_x}$	**00111000111**
new value	**00111000111**

return to initial state

11th shift

shifted bit	**00000000000**
$\boldsymbol{C_x}$	**00111000111**
new value	**00111000111**

return to initial state

4-shift from 4th shift

shifted bit	**00000011101**
C_x	**00111000111**
new value	**00111011111**

return to its state

5-shift from 4th shift

shifted bit	**00000001111**
C_x	**00111000111**
new value	**00111001111**

go to 5th shift state

6-shift 4th shift

shifted bit	**00000000111**
C_x	**00111000111**
new value	**00111000111**

return to initial state

10-shift from 4th shift

shifted bit	**00000000000**
C_x	**00111000111**
new value	**00111011111**

go to initial state

11-shift from 4th shift

shifted bit	**00000000000**
C_x	**00111000111**
new value	**00111001111**

return to initial state

4-shift from 5th shift

shifted bit	**00000011100**
C_x	**00111000111**
new value	**00111011111**

go to 4th shift state

5-shift from 5th shift

shifted bit	**00000001110**
C_x	**00111000111**
new value	**00111001111**

return to its state

6-shift 5th shift

shifted bit	**00000000111**
C_x	**00111000111**
new value	**00111000111**

return to initial state

10-shift from 5th shift

shifted bit	**00000000000**
C_x	**00111000111**
new value	**00111011111**

return to initial state

11-shift from 5th shift

shifted bit	**00000000000**
C_x	**00111000111**
new value	**00111001111**

return to initial state

4-shift from 6th shift

shifted bit	**00000011100**
C_x	**00111000111**
new value	**00111011111**

go to 4th shift state

5-shift from 6th shift

shifted bit	**00000001110**
C_x	**00111000111**
new value	**00111001111**

go to 5th shift state

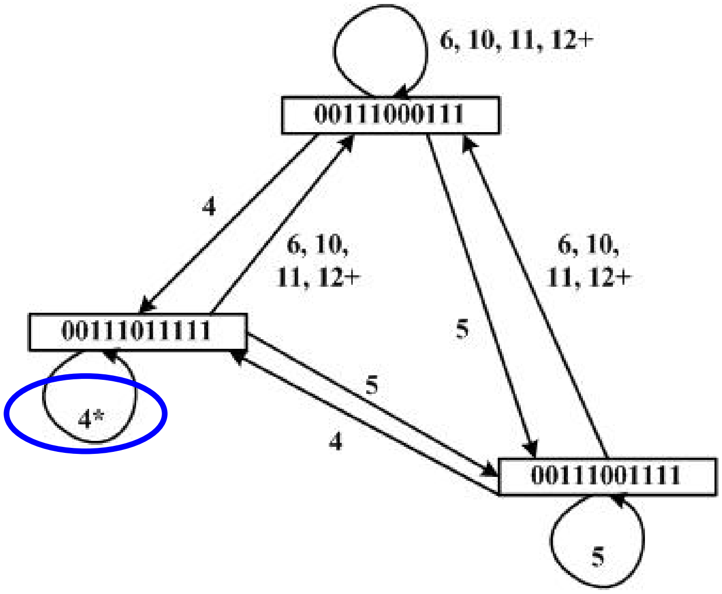
6-shift 6th shift

shifted bit	**00000000111**
C_x	**00111000111**
new value	**00111000111**

return to initial state

10-shift from 5th shift		11-shift from 5th shift	
shifted bit	**00000000000**	***shifted bit***	**00000000000**
$\boldsymbol{C_x}$	**00111000111**	$\boldsymbol{C_x}$	**00111000111**
new value	**00111011111**	***new value***	**00111001111**
return to initial state		return to initial state	

1-shift for 10th and 11th shifts will lead to 4th state and the rest will follow as the 1st round shifts.



d. The **simple cycles** are (4), (5), (6), (10), (11), (12), (4,5), (4,6), (4,10), (4,11), (4,12), (5,6), (5,10), (5,11), (5,12), (4, 5, 6), (4, 5, 10), (4, 5, 11) and (4, 5, 12). The **greedy cycles** are (4) and (4,5).

e. The **MAL** of this composite pipeline machine is **4**, so that the throughput is $\dfrac{\mathbf{1}}{\boldsymbol{MAL}} = \dfrac{\mathbf{1}}{\mathbf{4}} = \mathbf{0.25}$ or $\mathbf{25\%}$.

Problem 6.15 – A **nonpipelined processor X** has a **clock rate of 25 MHz** and an average **CPI of 4**. Processor Y, an improved successor of X, is designed with a **five-stage linear instruction pipeline**. However, due to the latch delay and clock skew effects, the **clock rate** of Y is only **20 MHz**.

a. If **a program containing 100 instructions** is executed on both processor, what is the **speedup** of processor Y compared with that of processor X.

b. Calculate the **MIPS rate** of each processor during the execution of this particular program.

Answer :

a. $\boldsymbol{CPI_X = 4}$, $\boldsymbol{f_X = 25\ MHz}$ and $\boldsymbol{I_C = 100}$.

Find the CPU $(\boldsymbol{T})$ time for each processor as followed :

$$\begin{aligned} T_X &= CPI_X . I_c . \tau_x \\ &= \frac{CPI_X . I_c}{f_x} \\ &= \frac{(4)(100)}{25.10^6} \\ &= 16\ \mu s \end{aligned}$$

and

$$\begin{aligned} T_Y &= \left[k + (n-1)\right]\tau_Y \\ &= \frac{k + (n-1)}{f_Y} \\ &= \frac{5 + (100-1)}{20.10^6} \\ &= 5.2\ \mu s \end{aligned}$$

so that the **speedup** is

$$\begin{aligned} S_k &= \frac{T_X}{T_Y} \\ &= \frac{16}{5.2} \\ &= 3.078 \quad \approx \quad 3.08 \end{aligned}$$

b. The MIPS rate for each processor is

$$\begin{aligned} \boldsymbol{MIPS_X} &= \frac{\boldsymbol{f}}{\boldsymbol{CPI}.\mathbf{10^6}} \\ &= \frac{\mathbf{25.10^6}}{\mathbf{4.10^6}} \\ &= \mathbf{6.25}\ \boldsymbol{MIPS} \end{aligned}$$

Find the **CPI** for Processor Y as followed :

$$\boldsymbol{T_Y} = \boldsymbol{CPI_Y}.\boldsymbol{I_c}.\boldsymbol{\tau_Y} \quad \leftrightarrow \quad \boldsymbol{CPI_Y} = \frac{\boldsymbol{T_Y}}{\boldsymbol{I_c}.\boldsymbol{\tau_Y}}$$

$$\begin{aligned} \boldsymbol{CPI_Y} &= \frac{\boldsymbol{T_Y} \cdot \boldsymbol{f_Y}}{\boldsymbol{I_c}} \\ &= \frac{\left(\mathbf{5.2.10^{-6}}\right)\left(\mathbf{20.10^6}\right)}{\mathbf{100}} \\ &= \mathbf{1.04} \end{aligned}$$

Then, the **MIPS rate** is

$$\begin{aligned} \boldsymbol{MIPS_Y} &= \frac{\boldsymbol{f_Y}}{\boldsymbol{CPI}.\mathbf{10^6}} \\ &= \frac{\mathbf{20.10^6}}{\left(\mathbf{1.04}\right).\mathbf{10^6}} \\ &= \mathbf{19.23}\ \boldsymbol{MIPS} \end{aligned}$$