Pipelining in Computer Architecture

Pipelining is a process of arrangement of hardware elements of the CPU such that its overall performance is increased. Simultaneous execution of more than one instruction takes place in a pipelined processor.

Real World Example 1:

In an automobile manufacturing line, there are 3 stages, each contributing something to the construction of the car.

Step 1: Assemble (A)

Step 2: Paint (P)

Step 3: Fix Tires (T)

Each step operates in parallel with the other steps, although on a different car.

Assuming **A + P + T** take 3 hours.

Time line	0-3 hours	3-6 Hours	6 -9Hours
Car 1	A+P+T		
Car 2		A+P+T	
Car 3			A+P+T

So, without pipelining it will take 9 hours to manufacture 3 cars.

In pipelining, we will use the advantage of parallelism that exists among the actions needed to manufacture a car.

Assume each stage takes 1 hour.

Time line	1st hour	2nd hour	3rd hour	4th hour	5th hour	6th hour	7th hour	8th hour	9th hour
Car 1	A	P	**T**						
Car 2		A	**P**	T					
Car 3			**A**	P	T				

So, with pipelining it will take 5 hours to manufacture 3 cars.

Speedup = Without Pipelining / With Pipelining = 9/5

Speed-up is a measure of how much faster a pipelined architecture is compared to a non-pipelined architecture when executing the same workload.

Here our assumptions are:

1. There is no time consumed to transition car from one working station to other.
2. All the stages consume equal time.

However these assumption are not valid

1. Each stage can take different amount of time. For example, painting a car takes 2 hours and assembly and installing tires take 0.5 hours each.
2. To transfer car from one stage to another will take time. For example, it takes 10 minutes to transfer a car from assembly room to paint room and 10minutes to transfer from paint room to tire fixing room.

NOTE: This will not be consume in un-pipeline factory as everything will be done in same room (this is the overhead of using pipelining).

Real World Example 2:

Automobile manufacturing line with pipelining, in 3 stages.
If Assemble (A), Paint (P), Fix tires (T) take different time.
If Assembly take 0.5 hour, Paint 2 hours and Tires 0.5 hours.

- Paint of Car 2 cannot start before 2.5th hour because paint unit is busy with Car 2. So after assembly Car 2 has to wait for 1.5 hours.

Total time taken to complete 3 instructions = 7 hours.

Speed Up = 9/7

Time line	0-0.5h	0.5-2.5h	2.5-4.5h	4.5-6.5 h	6.5-7 h
Car 1	A (0.5 h)	P (2h)	T (0.5h)		
Car 2		A (0.5 h) +wait for 1.5 h	P (2h)	T (0.5h)	
Car 3			A (0.5h)+ wait for 1.5 h	P (2h)	T (0.5h)

Real World Example 2:

Automobile manufacturing line with pipelining, in 3 stages.
If we add time for transition between stages as well.
Total time taken to complete 3 instructions = 7 h + 40 m = 7.67 h

Speed Up = 9/7.67

Time line	0-0.5h	Transition time 10 min	0.5 +10m-2.5h +10m	Transition time 10 min	2.5h+20m-4.5h+20m	Transition time 10 min	4.5h+30m-6.5 h+30m	Transition time 10 min	6.5h+40m -7 h+40m
Car 1	A (0.5 h)		P (2h)		T (0.5h)				
Car 2			A (0.5 h) +wait for 1.5 h		P (2h)		T (0.5h)		
Car 3					A (0.5h)+ wait for 1.5 h		P (2h)		T (0.5h)

Throughput and Latency:

In case of car manufacturing throughput is number of cars create per hour.

Latency is time taken to complete one car.

Case	Latency	Throughput	Speed up
Non pipeline	3	3/9=0.33	NA
Pipeline ideal case	3	3/5=0.6	9/5
Pipeline with all stages not of equal time	4.5	3/7=0.42	9/7
Non ideal case with latch time	4.833	3/7.67=0.39	9/7.67

Pipelining in Computer:

In a computer pipeline, each step in the pipeline completes a part of an instruction.

Like the assembly line, different steps are completing different parts of different instructions in parallel.

Each of these steps is called a pipe stage or a pipe segment.

The stages are connected one to the next to form a pipe — instructions enter at one end, progress through the stages, and exit at the other end, just as cars would in an assembly line.

The number of stages in which one instruction can be divided is different from processor to processor.

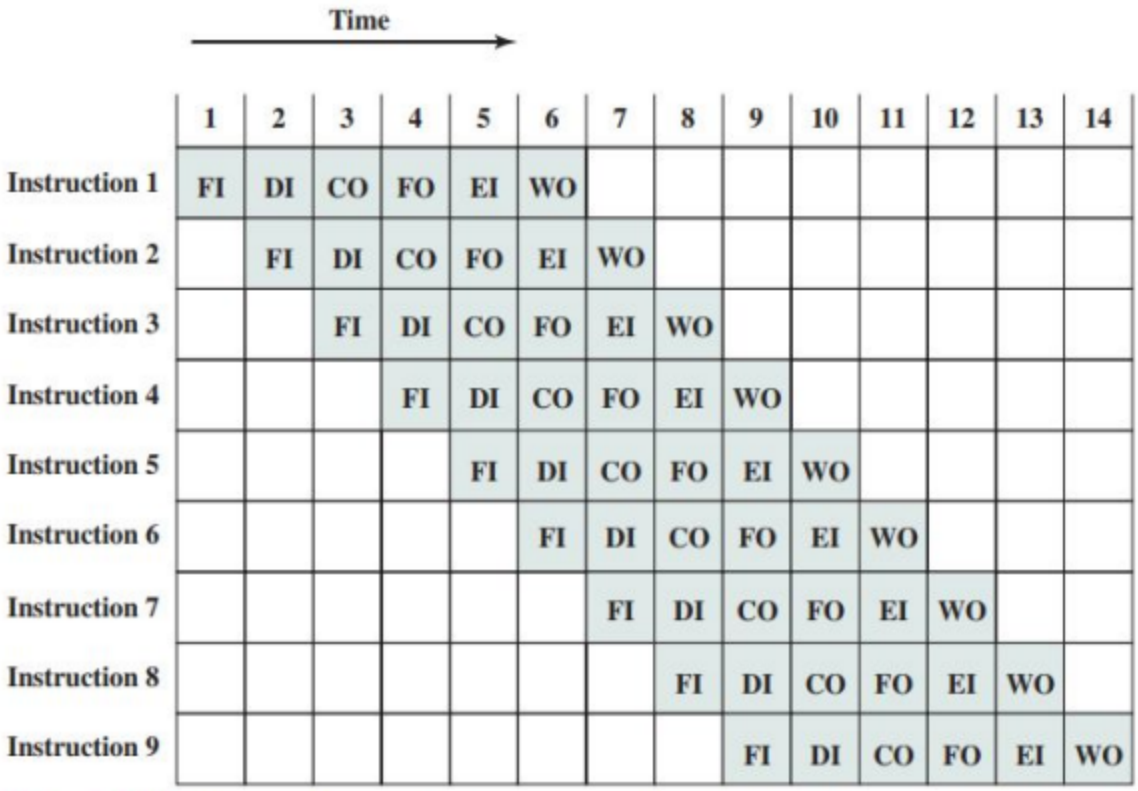
If we only create two stages Fetch Instruction (FI) and Execute Instruction (EI) – the bottleneck creator, EI will take a lot more time than FI.

To get a better throughput and thus the speedup pipeline must have more stages. The pipeline designer’s goal is to balance the length of each pipeline stage

Let us consider the following decomposition of the instruction processing:

1. Fetch instruction **(FI)**
2. Decode instruction **(DI)**
3. Calculate operands **(CO)**
4. Fetch operands **(FO)**
5. Execute instruction **(EI)**
6. Write operand **(WO)**

With this decomposition, the various stages will be of more nearly equal duration. For the sake of illustration, let us assume equal duration. Using this assumption, a six-stage pipeline can reduce the execution time for 9 instructions from 54 time units to 14 time units.



Throughput with pipeline= 9/14
Throughput without pipeline=
9/54
Speedup=
54/14

Clock Cycle:

The time required between moving an instruction one step down the pipeline is a processor cycle.

Because all stages proceed at the same time, the length of a processor cycle is determined by the time required for the slowest pipe stage, just as in an auto assembly line.

The longest step would determine the time between advancing the line.

In a computer, this processor cycle is usually 1 clock cycle.

For example:

If:

- FI DI CO FO EI WO each takes 1us.
- Clock cycle should be of 1us.
- Although this is the ideal case.

If:

- FI takes 5 us, DI takes 1us, CO takes 1us, FO takes 10us, EI takes 1us, and WO also takes 10us.
- Clock cycle should be of 10us

In processor with no pipelining:

- If one instruction takes 10us then clock cycle should 10us.

Frequency:

Computer processors can execute one or more instructions per clock cycle, depending on the type of processor.

Frequency of computer is 1/clock cycle

For example: if 1 clock cycle = 10 ms;

f = 1/10ms = 1MHZ

Frequency is also known as clock speed.

Performance:

Ideal Case (Derivations)

If

- All stages take equal amount of time T
- Latch time = 0
- Stages = k
- Number of instructions = n

Then

- Clock cycle of pipeline = T , clock cycle of non pipeline= k*T
- Frequency of pipeline=1/T , Frequency of non pipeline=1/k*T
- Time taken to complete n instructions without pipeline= n*k*T= n*clock cycle of non pipeline
- Through put for n instructions without pipeline= n/ n*k*T
- Time taken to complete n instructions with pipeline= (k+n-1)*T= (k+n-1)*(Clock Cycle of pipeline)
- Through put for n instructions with pipeline= n/ (k+n-1)*T
- Speedup for n instructions = n*k*T/ (k+n-1)*T = n*k/ (k+n-1)
- Latency without pipelining = k*T
- Latency with pipelining= k*T

PIPELINE IN COMPUTER:: PERFORMANCE IDEAL CASE

Numerical:

- It takes 6us to complete one instruction in non-pipeline processor
- We were able to convert the circuit into 6 equal sequential pipeline stages.
- Assume latch time is 0
- Answer the following, assuming that there are no stalls in the pipeline.

What are the clock cycle in the two processors?

What are the clock speeds(frequency) in two processors?

How long does it take to finish one instruction in pipeline and no pipeline (latency)?

What is the throughput for 100 instructions without pipelining?

What is the throughput for 100 instructions with pipelining?

What is the speedup from pipelining for 1 instructions?

What is the speedup from pipelining for 100 instructions?

Given:

Time to complete one instruction in non-pipeline = 6us = 6×10^{-6} s

No. of stages in pipeline = k = 6

Latch time = 0

Time taken by each step = 1 us = 1×10^{-6} s

① Clock cycle of pipeline = T = 1 us

Clock cycle of non-pipeline = k x T

= 6 x 1 us

= 6 us

② Frequency of pipeline = $\frac{1}{T} = \frac{1}{1\,us} = \frac{1}{1 \times 10^{-6}}$

= 1 MHZ

Frequency of non-pipeline = $\frac{1}{k \times T} = \frac{1}{6 \times 10^{-6}}$

= 166.67 KH

③ Time taken to complete one instruction with pipeline = (k+n-1)xT

= k x T = 6 x 1us = 6us

In non-pipeline = n x k x T

= k x T = 6 x 1us = 6us

④ $n = 100$

$$\text{Throughput} = \frac{n}{n \times k \times T} = \frac{100}{100 \times 6 \times 1\mu s}$$

$$= \frac{100}{600\mu s}$$

$$= 166.67 \text{ instructions per ms}$$

⑤ $n = 100$

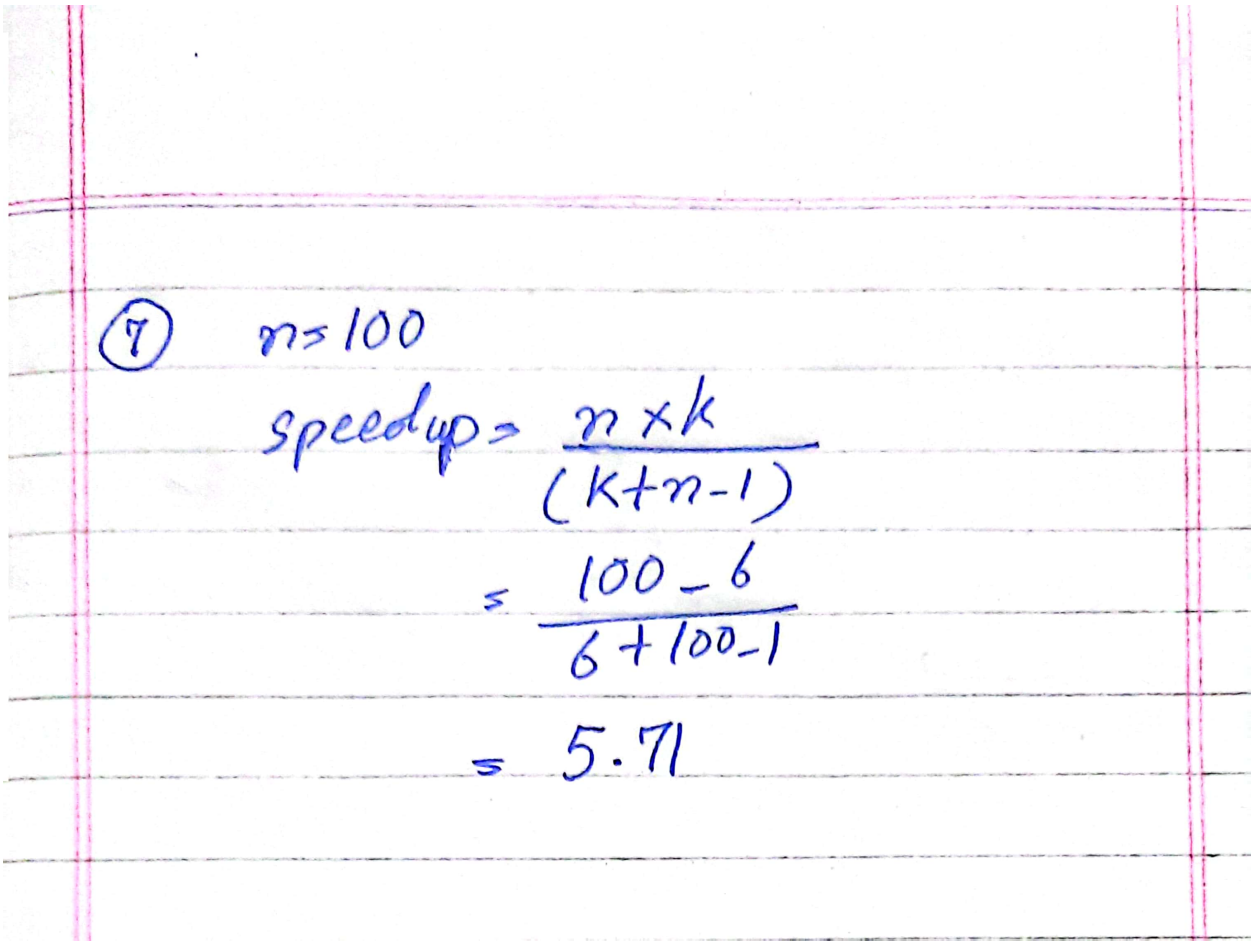
$$\text{Throughput} = \frac{n}{(k+n-1) \times T} = \frac{100}{(6+100-1) \times 1\mu s}$$

$$= 952.38 \text{ instructions per ms}$$

⑥ $n = 1$

$$\text{speedup} = \frac{n \times k \times T}{(k+n-1) \times T}$$

$$= \frac{n \times k}{(k+n-1)} = \frac{1 \times 6}{(6+1-1)} = 1$$



Non-ideal Case (Derivations)

(DERIVATIONS)

If all stages do not take same time

- Non pipeline processor takes T1 time to complete one instruction
- In pipeline processor max time take by any stage is T2
- Latch time=0
- Stages =k
- Number of instructions = n

Then

- Clock cycle of pipeline processor = T2, Clock Cycle of processor without pipelining = T1
- Frequency of pipeline processor =1/T2, Frequency of processor without pipelining = 1/T1
- Time taken to complete n instructions without pipeline= n*T1= n*clock cycle of non pipeline
- Through put for n instructions without pipeline= n/ n*T1
- Time taken to complete n instructions with pipeline= (k+n-1)*T2 =(k+n-1)*(Clock Cycle of pipeline)
- Through put for n instructions with pipeline= n/ (k+n-1)*T2
- Speedup for n instructions = n*T1/ (k+n-1)*T2
- Latency without pipelining = T1
- Latency with pipelining= K*T2

PIPELINE IN COMPUTER:: PERFORMANCE

Example NON-IDEAL

- It takes 6us to complete one instruction in non-pipeline processor
- We were able to convert the circuit into 6 sequential pipeline stages.
- Stage 1 and 2 take 2us each and stage 3-6 take 0.5us each
- Assume latch time is 0
- Answer the following, assuming that there are no stalls in the pipeline.

What are the clock cycle in the two processors?

What are the clock speeds(frequency) in two processors?

How long does it take to finish one instruction in pipeline and no pipeline (latency)?

What is the throughput for 100 instructions without pipelining?

What is the throughput for 100 instructions with pipelining?

What is the speedup from pipelining for 1 instructions?

What is the speedup from pipelining for 100 instructions?

Given:

Time taken to complete one instruction in non-pipeline = $6us = 6 \times 10^{-6} s$

Stages in pipeline = $k = 6$

Stage 1 and 2 take 2us each

Stage 3 to 6 take 0.5 us each

Latch time = 0

Here $T1 = 6us$ and $T2 = 2us$

(1)

Clock cycle of pipeline = 2 us which is equal to the time required for longest pipeline stage

Clock cycle of none-pipeline = 6 us

(2) Frequency of pipeline $= \frac{1}{T2} = \frac{1}{2us}$

$= 500 KHZ$

Frequency of non-pipeline $= \frac{1}{T1} = \frac{1}{6us}$

$= 166.67 KHZ$

③ $n = 1$

Latency of pipeline $= k \times T2$
$= 6 \times 2us$
$= 12us$

Latency with non-pipeline $= T1$
$= 6us$

④ $n = 100$

$$\text{Throughput} = \frac{n}{n \times T1} = \frac{100}{100 \times 6us}$$

$= 166.67$ instructions per ms

⑤ $n = 100$

$$\text{Throughput} = \frac{n}{(k + n - 1) \times T2}$$

$$= \frac{100}{(6 + 100 - 1) \times 2us}$$

$= 476.19$ instructions per ms

(6) $n = 1$

$$\text{Speedup} = \frac{n \times T1}{(k+n-1) \times T2}$$

$$= \frac{1 \times 6us}{(6+1-1) \times 2us}$$

$$= \frac{1}{2} = 0.5$$

(7) $n = 100$

$$\text{Speedup} = \frac{n \times T1}{(k+n-1) \times T2}$$

$$= \frac{100 \times 6us}{(6+100-1) \times 2us}$$

$$= \frac{20}{7} = 2.85$$

Including Latch Time

TIME (DERIVATIONS)

If all stages do not take same time

- Non pipeline processor takes T1 time to complete one instruction
- In pipeline processor max time take by any stage is T2
- Latch time=T3
- Stages =k
- Number of instructions = n

Then

- Clock cycle of pipeline processor = T2+T3, Clock Cycle of processor without pipelining = T1
- Frequency of pipeline processor =1/(T2+T3), Frequency of processor without pipelining = 1/(T1)
- Time taken to complete n instructions without pipeline= n*T1= n*clock cycle of non pipeline
- Through put for n instructions without pipeline= n/ n*T1
- Time taken to complete n instructions with pipeline= (k+n-1)*(T2+T3)= (k+n-1)*(Clock Cycle of pipeline)
- Through put for n instructions with pipeline= n/ (k+n-1)*(T2+T3)
- Speedup for n instructions = n*T1/ (k+n-1)*(T2+T3)
- Latency without pipelining = T1
- Latency with pipelining= K*(T2+T3)

PIPELINE IN COMPUTER:: PERFORMANCE

Example NON-IDEAL CASE with latch time

- It takes 6us to complete one instruction in non-pipeline processor
- We were able to convert the circuit into 6 sequential pipeline stages.
- Stage 1 and 2 take 2us each and stage 3-6 take 0.5us each
- Latch time is 2us
- Answer the following, assuming that there are no stalls in the pipeline.

What are the clock cycle in the two processors?

What are the clock speeds(frequency) in two processors?

How long does it take to finish one instr in pipeline and no pipeline (latency)?

What is the throughput for 100 instructions without pipelining?

What is the throughput for 100 instructions with pipelining?

What is the speedup from pipelining for 1 instructions?

What is the speedup from pipelining for 100 instructions?

Given:

Time Taken to complete one instruction in non-pipeline = 6 us = 6×10^{-6} s
Stages = k = 6
Stage 1 and 2 take 2us each
Stages 3 to 6 take 0.5 us each
Latch time is 2us

Here, T1 = 6us , T2 = 2us , T3 = 2us

(1)

Clock time in pipeline = T2 + T3
= 2us + 2us
= 4us

Clock time in non-pipeline = T1
= 6us

(2)

$$P.F = \frac{1}{T2+T3} = \frac{1}{4us} = 250\ KHZ$$

$$N.P.F = \frac{1}{6us} = 166.67\ KHZ$$

(3)

Latency with pipeline = $K \times (T2 + T3)$

$= 6 \times (2us + 2us)$

$= 6 \times 4us$

$= 24us$

Latency without pipeline = 6us

(4) $n = 100$

$$\text{Throughput} = \frac{n}{n \times T1} = \frac{100}{100 \times 6us} = \frac{1}{1 \times 6 \times 10^{-6}}$$

= 166.67 instructions per ms

(5) $n = 100$

$$\text{Throughput} = \frac{n}{(k + n - 1) \times (T2 + T3)}$$

$$= \frac{100}{(6 + 100 - 1)(4us)}$$

= 238.09 instructions per ms

(6) $n = 1$

$$\text{speedup} = \frac{n \times T1}{(k + n - 1) \times (T2 + T3)}$$

$$= \frac{1 \times 6us}{(6 + 1 - 1) \times (4us)}$$

$$= \frac{1}{4} = 0.25$$

EXERCISE

Question 1: If throughput of one processor1 for n instruction is 10us and through put of processor 2 is 15us for same n instruction what speed up is achieved by processor2 are compare to processor 1?

Question 2: If Latency with pipelining with 5 stages is 6us and latch time is 0 what is the clock cycle time?

Question: 1

$$\text{Speed up} = \frac{\text{Throughput of none-pipeline}}{\text{Throughput of pipeline}}$$

$$= \frac{10}{15}$$

$$= \frac{2}{3}$$

Question: 2

As, $k = 5$ L.P = 6us $T3 = 0$ $T2 = ?$

Latency = k x Clock cycle

6us = 5 x Clock cycle

clock cycle = $\frac{6us}{5} = 1.2 \times 10^{-6}$ s

$= 1.2$ us