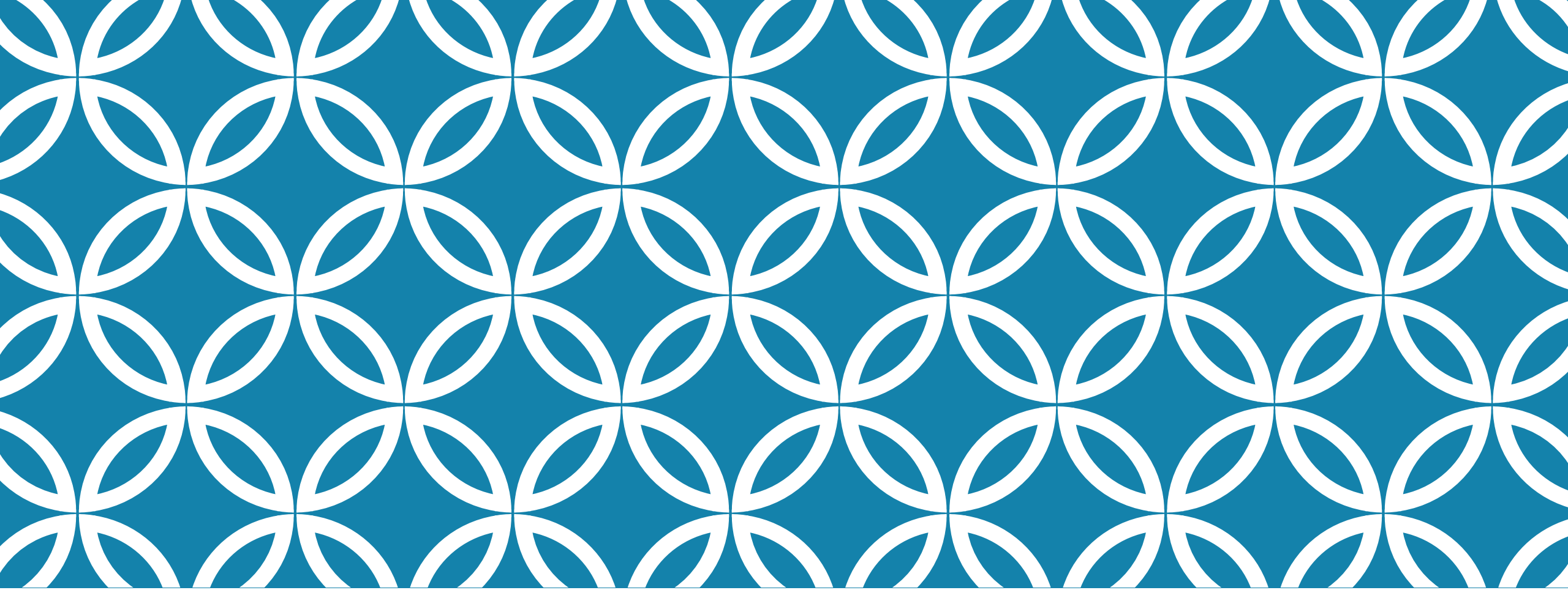


PIPELINING I

REAL WORLD ANALOGY::THROUGHPUT AND LATENCY

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

- **From slide 8,** throughput **with** pipelining is 3 Cars is 7.67 Hours ~0.4 Cars/hour
- **From slide 4,** throughput **without** pipelining is 3 Cars is 9Hours ~ 0.33 Cars/hour
- So we can see that the throughput with pipelining is higher.

Latency is time taken to complete one car

- **From slide 8,** latency **with** pipelining is 4.5 hours (4.5 hours consumed to create car 2 and 3)
- **From slide 4,** latency **without** pipelining is 3 hours (3 hours consumed to create each car)
- Thus we can see that latency is worse with pipelining

Observation: Pipeline does not improve latency, it only improves throughput (observe the bottlenecks)

PIPELINE IN COMPUTER:: STAGES

Let us consider the following decomposition of the instruction processing

- **Fetch instruction (FI):** Read the next expected instruction into a buffer.
- **Decode instruction (DI):** Determine the opcode and the operand specifiers.
- **Calculate operands (CO):** Calculate the effective address of each source operand. This may involve displacement, register indirect, indirect, or other forms of address calculation.
- **Fetch operands (FO):** Fetch each operand from memory. Operands in registers need not be fetched.
- **Execute instruction (EI):** Perform the indicated operation and store the result, if any, in the specified destination operand location.
- **Write operand (WO):** Store the result in memory.

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, next slide shows that a six-stage pipeline can reduce the execution time for 9 instructions from 54 time units to 14 time units.

PIPELINE IN COMPUTER:: STAGES

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

Note that the figure is for ideal case. In reality, all the instructions might not require to go through all these stages.

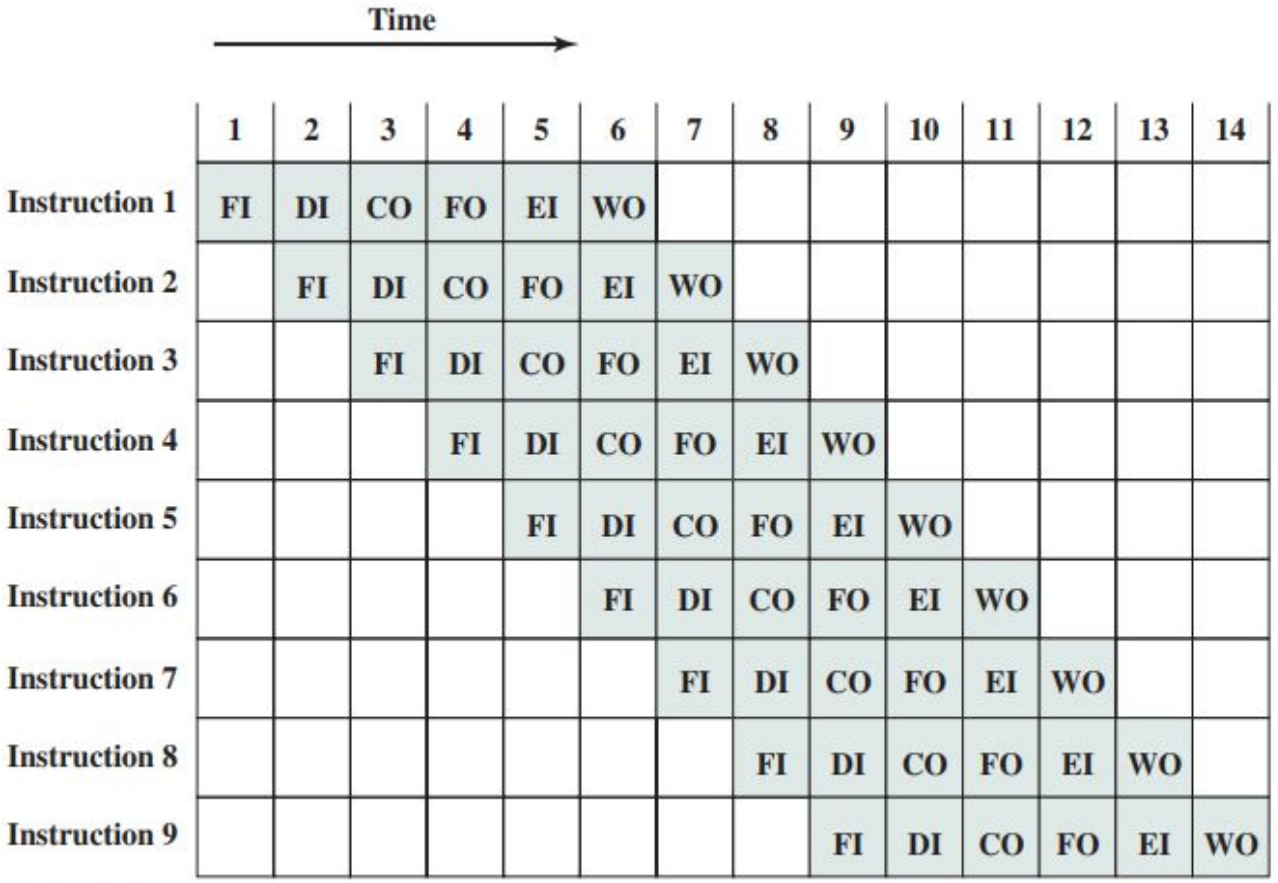


Figure 14.10 Timing Diagram for Instruction Pipeline Operation

PIPELINE IN COMPUTER:: 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.

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

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

One cell in figure 14.10 (previous slide) is 1 clock cycle.

PIPELINE IN COMPUTER:: CLOCK CYCLE

For example in if

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

For example:

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

For example, in processor with no pipelining

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

PIPELINE IN COMPUTER:: 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;

- **Frequency = 1/10ms= 1MHZ**
- **i.e. 1 000 000 cycles / second**

Frequency is AKA as clock speed

PIPELINE IN COMPUTER:: PERFORMANCE

In ideal scenario i.e., all the stages require same amount of time and ignoring time to transfer data from one stage to another

- Speedup= n*k/ (k+n-1)
- where n is number of instructions and k is number of stages

Time line	T	T	T	T	T
Car 1	A	P	**T**		
Car 2		A	**P**	T	
Car 3			**A**	P	T
..				…	..

PIPELINE IN COMPUTER:: 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 instruction?

What is the speedup from pipelining for 100 instructions?

PIPELINE IN COMPUTER:: PERFORMANCE

Another example

- It takes 5us to complete one instruction in non-pipeline processor
- We were able to convert the circuit into 5 equal duration sequential pipeline stages.
- Latch time is 0us
- 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?

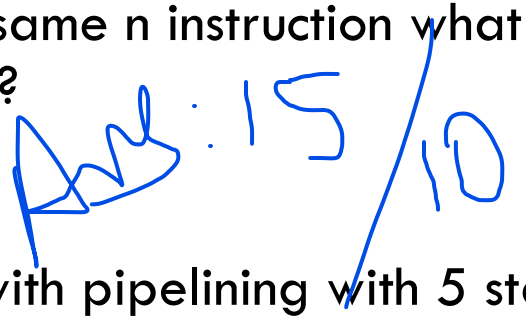
What is the throughput for 100 instructions with pipelining?

What is the speedup from pipelining for 1 instructions?

What is the speedup from pipelining for 100 instructions?

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?