**Tutorial 10:** WRES1201 – Computer System Architecture

1. Explain the function of the EU (execution unit) and the BIU (bus interface unit).

The execution unit executes instructions that fetched by the processor.

The bus interface unit access the memory and peripherals to fetch instruction

2. Why does pipelining improve performance?

The instruction are divided into stages and the overlapping of stages reduce the time required by the instruction execution.

3. What is pipeline latency?

The amount of time required by the execution of a single operation:

latency = 1/throughtput

4.

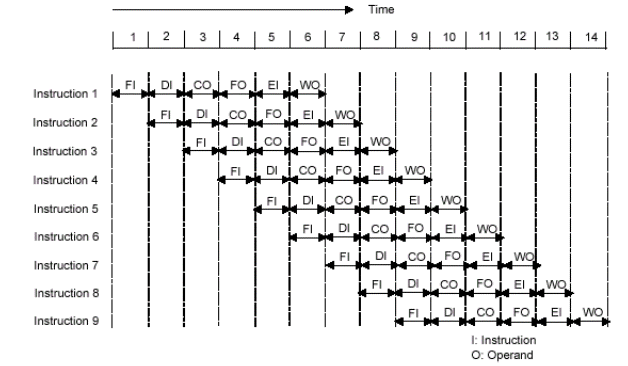


Figure 1

Consider the timing diagram of Figure 1. Assume that there is only a two stage pipeline (fetch, execute). Redraw the diagram to show how many time units are now needed for four instruction. What is *speedup* *2-stage pipeline* comparing with sequential pipelined.

Fetch

Execute

Instruction 1

Fetch

Execute

Instruction 2

Fetch

Execute

Instruction 3

Fetch

Execute

Instruction 4

1

2

3

4

5

Time

Total Time (sequential):

(Tk) = (k \* n ) г

k = 2 stages

n = 4 instructions

Tk = (2\*4) г

Total Time (pipelining):

(Tk) = [k + (n - 1)] г

k = 2 stages

n = 4 instructions

Tk= [2 + (4 - 1)] г

=5г

Speed up (Sk)

Sk = Sequential (Time)

Pipelining (Time)

= 8/5

5. How many time units are needed to execute 600 instructions with 6-stage pipeline. And how many time units if unpipelined. . Calculate the *speedup*.

k = 6 stages

n = 600 instructions

г = 1 (assuming)

(Tk) = [k + (n - 1)] г

= 600+(6-1)

= 605 hours

non pipeline

(Tk) = (k \* n ) г

= (600\*6)1  
 = 3600 hours

Speedup

Sk = Sequential (Time)

Pipelining (Time)

= 3600/605

= 5.95 (2 decimal point)

6. Given an unpipelined processor with 10ns cycle time and pipeline latches with 0.5ns latency, what are the cycle times of pipelined versions of the processor with 2, 4, 8, and 16 stages if the datapath logic is evenly divided among the pipeline stages? Also, what is the latency of each of the pipelined versions of processor?

Without pipeline the time is 10ns.

When pipeline is applied, assume the time is divided equally into each stages.

Therefore, CTpipe = (CTunpipe/no. of stages) + latch latency

CT for 2 = 10ns/ 2 + 0.5 =5.5ns

4 get 3, 8 get 1.75; 16 get 1.125ns

Latency for 2 get 11; 4 get 12, 8 get 14; 16 get 18ns

7. List and briefly explain various ways in which an instruction pipeline can deal with conditional branch instruction.

**Multiple streams**: A brute-force approach is to replicate the initial portions of the pipeline and allow the pipeline to fetch both instructions, making use of two streams.

**Prefetch branch target**: When a conditional branch is recognized, the target of the branch is prefetched, in addition to the instruction following the branch. This target is then saved until the branch instruction is executed. If the branch is taken, the target has already been prefetched.

**Loop buffer**: A loop buffer is a small, very-high-speed memory maintained by the instruction fetch stage of the pipeline and containing the n most recently fetched instructions, in sequence. If a branch is to be taken, the hardware first checks whether the branch target is within the buffer. If so, the next instruction is fetched from the buffer.

**Branch prediction**: A prediction is made whether a conditional branch will be taken when executed, and subsequent instructions are fetched accordingly.

**Delayed branch**: It is possible to improve pipeline performance by automatically rearranging instructions within a program, so that branch instructions occur later than actually desired.