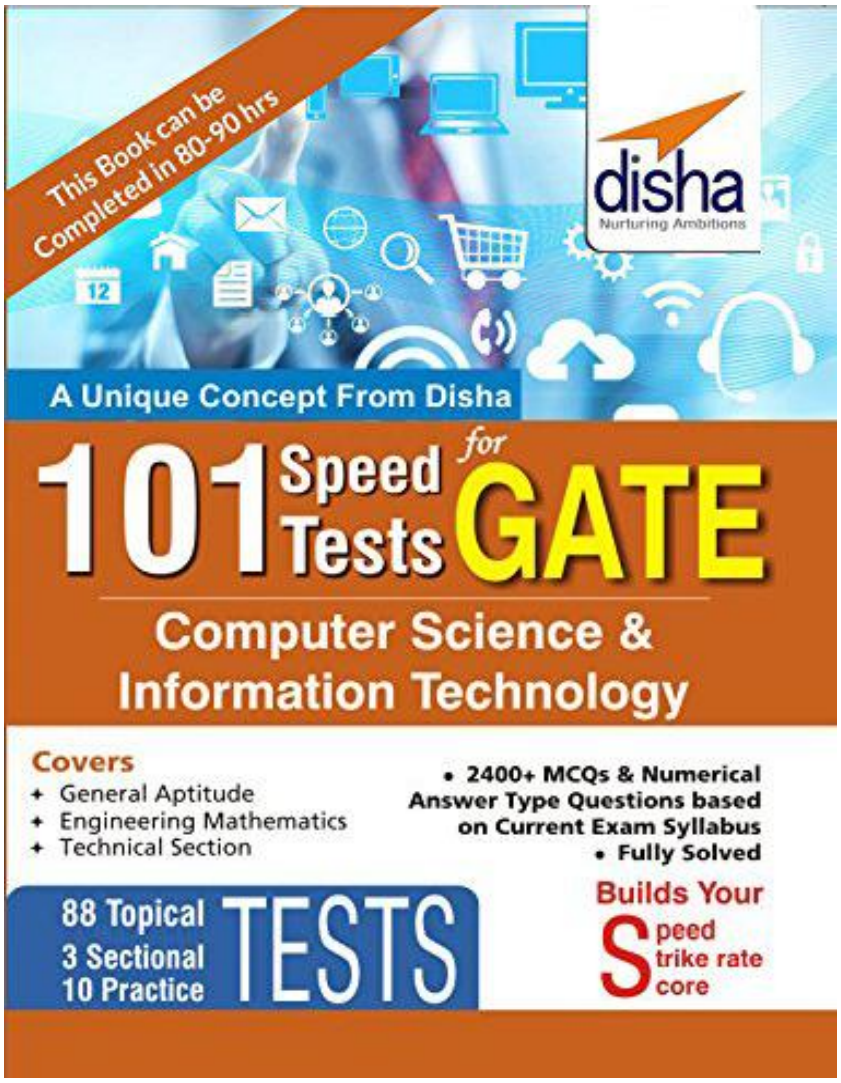
INSTRUCTION PIPELINING

This Chapter is taken from our Book:



ISBN : 9788193288979

101 SPEED TEST

35

Instruction Pipelining

Max. Marks : 20 | No. of Qs. 20 | Time : 30 min. | Date : ...../...../.....

1. Consider a computer with 4 floating point processors suppose that each processor uses a cycle time of 40nsec. How long will it take to perform 400 floating point operations? What is the time if the operations are carried out on a single processor with cycle time of 10 nsec?
(a) 400n sec, 4000n sec (b) 4000n sec, 4000n sec
(c) 4000n sec, 400n sec (d) None of these

2. An instruction pipeline is having 5 stages
(1) fetch (2) decode (3) execute (4) memory (5) write back. All the instructions can be over lapped except the branch instructions.
If there exists 30% branch instruction clock cycle is 10ns and penalty for branch is 4 cycle, what is the throughput with pipeline?

3. A non pipeline system takes 50nsec to process a task. The same task can be processed in a 6 segment pipeline with a clock cycle of 10n sec. Determine the speedup ratio of the pipeline for 100 tasks?

4. While Implementing a DLX processor, Alyssa observes that because of increasing clock speeds, the D-cache no longer can be accessed in 1 cycle; instead it takes two cycles. (The smaller I-cache can still be accessed in 1 cycle.) Consequently she suggests replacing the MEM stage of the 5-stage DLX pipeline by two stages MEM1 and MEM2 to perform the 2-cycle access, yielding a 6-stage pipeline. Moreover she designs a two-stage pipelined D-cache so that a new instruction can continue to be issued per cycle.
A program having 0.5 million load/store instructions with an ideal (100% hit rate D-cache takes 2.5 million cycles to execute in a 5-stage pipeline (with a 1 cycle cache hit) and 2.6 million cycles in a 6-stage pipeline (with a 2 cycle cache hit). The increase is because of additional stalls from hazards. Now consider that the ideal cache is replaced by a real cache with a 90% hit rate. Assume that after a cache miss, a DRAM with a 20 cycle latency is accessed. How many cycles does the program take to execute in the above 5-stage and 6-stage pipelines?
(a) 3.6 million cycle 3.5 million cycle
(b) 3.5 million cycle 3.6 million cycle
(c) 3.6 million cycle 3.6 million cycle
(d) None of these

5. Consider a LW instruction followed by a dependent XOR instruction in a program. With the best bypass path possible, how many stall cycles result ?

DIRECTIONS for (6-7) A VLIW processor issues groups of three instructions per cycle. Three pipelines are present, each very much like a 5-stage DLX pipeline. A program executes 120 million instructions in 50 million groups, of which 25 million groups contain exactly one branch. (A VLIW does not allow groups with more than one branch sinch the result is undefined.) Branches can only target the start of instruction groups. Branches are predicted "never taken". If they are found to be taken instead, subsequent fall-through instructions are squashed and execution restarts at the target. Branch prediction accuracy is 30%.

6. How many NOPs are present in the program code?
(a) 3 million instruction (b) 30 million instruction
(c) 120 million instruction (d) None of these

7. What is the run-time of the program assuming perfect bypassing, and ideal cache, and the real predictor mentioned?

8. A 5-stage pipelined processor has Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Perform Operation (PO) and Write Operand (WO) stages. The IF, ID, OF and WO stages take 1 clock cycle each for any instruction. The PO stage takes 1 clock cycle for ADD and SUB instructions, 3 clock cycles for MUL instruction, and 6 clock cycles for DIV instruction respectively. Operand forwarding is used in the pipeline. **How many number of clock cycles needed to execute the following sequence of the instructions?**

Instruction	**Meaning of instruction**
I_0: MUL R_2, R_0, R_1	$R_2 \rightarrow R_0 * R_1$
I_1: DIV R_5, R_3, R_4	$R_5 \rightarrow R_3 / R_4$
I_2: ADD R_2, R_5, R_2	$R_2 \rightarrow R_5 + R_2$
I_3: SUB R_5, R_2, R_6	$R_5 \rightarrow R_2 - R_6$

9. Consider an instruction pipeline with five stages without any branch prediction: Fetch Instruction (FI), Decode Instruction (DI), Fetch Operand (FO), Execute Instruction (EI) and Write Operand (WO). The stage delays for FI, DI, FO, EI and WO are 5 ns, 7 ns, 10 ns, 8 ns and 6 ns, respectively. There are intermediate storage buffers after each stage and the delay of each buffer is 1 ns. A program consisting of 12 instructions I_1, I_2, I_3, ..., I_{12} is executed in this pipelined processor. Instruction I_4 is the only branch instruction and its branch target is I_9. If the branch is taken during the execution of this program, the time (in ns) needed to complete the program is

10. Consider a pipelined processor with the following four stages
IF : Instruction Fetch
ID : Instruction Decode and Operand Fetch
EX : Execute
WB : Write Black
The IF, ID and WB stages take 1 clock cycle each to complete the operation.
- The number of clock cycle for the EX stage depends on the instruction. the ADD and SUB instructions need 1 clock cycle and the MUL instruction needs 3 clock cycles in the EX stage. Operand forwarding is used in the pipelined processor. What is the number of clock cycles taken to complete the following sequence of instructions:

ADD R_2, R_1, R_0 $R_2 \leftarrow R_1 + R_0$
MUL R_4, R_3, R_2 $R_4 \leftarrow R_3 * R_2$
SUB R_6, R_5, R_4 $R_6 \leftarrow R_5 - R_4$

11. A 5 stage pipelined CPU has the following sequence of stages
IF : Instruction fetch from instruction memory
RD : Instruction decode and register read
EX : Execute: ALU operations for data and address computation

MA : Data memory access : for write access, the register read at RD stage is used
WB : Register write back
Consider the following sequence of instructions:
I_1 : L R_0 loc 1; R_0 <= M [loc_1]
I_2 : A R_0 ; R_0; R_0 <= R_0 + R_0
I_3 : S R_2 ; R_0; R_2 <= R_2 – R_0
Let each state takes on clock cycle.
What is the number of clock cycles taken to complete the above sequence of instructions starting from the fetch of I_1?

12. Which of the following is/are true of the auto-increment addressing mode?
1. It is used in creating self-relocating code.
2. If it is included in an Instruction Set Architecture, then an additional ALU is required for effective address calculation.
3. The amount of increment depends on the size of the data item accessed.

(a) 1 only (b) 2 only
(c) 3 only (d) 2 and 3

13. Consider a 4 stage pipeline processor. The number of cycles needed by the four instructions I_1, I_2, I_3, I_4 in stages S_1, S_2, S_3, S_4 is shown below.

	S_1	S_2	S_3	S_4
I_1	2	1	1	1
I_2	1	3	2	2
I_3	2	1	1	3
I_4	1	2	2	2

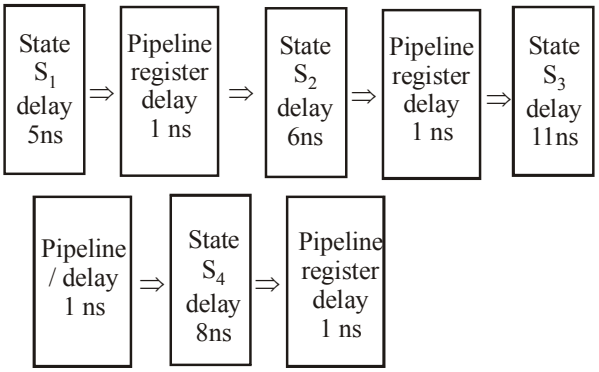
What is the number of cycles needed to execute the following loop?
For (i = 1 to 2) {I_1; I_2; I_3; I_4;}

14. An instruction pipeline has five stages, namely, instruction fetch (IF), instruction decode and register fetch (ID/RF), instruction execution (EX), memory access (MEM), and register writeback (WB) with stage latencies 1 ns, 2.2 ns, 2 ns, 1 ns, and 0.75 ns, respectively (ns stands for nanoseconds). To gain in terms of frequency, the designers have decided to split the ID/RF stage into three stages (ID, RF1, RF2) each of latency 2.2/3 ns. Also, the EX stage is split into two stages (EX1, EX2) each of latency 1 ns. The new design has a total of eight pipeline stages. A program has 20% branch instructions which execute in the EX stage and produce the next instruction pointer at the end of the EX stage in the old design and at the end of the EX2 stage in the new design. The IF stage stalls after fetching a branch instruction until the next instruction pointer is computed. All instructions other than the branch instruction have an average CPI of one in both the designs. The execution times of this program on the old and the new design are P and Q nanoseconds, respectively. The value of P/Q is _____.

15. The memory access time is 1 nanosecond for a read operation with a hit in cache, 5 nanoseconds for a read operation with a miss in cache, 2 nanoseconds for a write operation with a hit in cache and 10 nanoseconds for a write operation with a miss in cache. Execution of a sequence of instructions involves 100 instruction fetch operations, 60 memory operand read operations and 40 memory operand write operations. The cache hit-ratio is 0.9. The average memory access time (in nanoseconds) in executing the sequence of instructions is _____.

16. Consider a 6-stage instruction pipeline, where all stages are perfectly balanced. Assume that there is no cycle-time overhead of pipelining. When an application is executing on this 6-stage pipeline, the speedup achieved with respect to non-pipelined execution if 25% of the instructions incur 2 pipeline stall cycles is _____.

17. Consider an instruction pipeline with four stages (S_1, S_2, S_3 and S_4) each with combinational circuit only. The pipeline registers are required between each stage and at the end of the last stage. Delays for the stages and for the pipeline registers are as given in the figure



What is the approximate speed up of the pipeline in steady state under ideal conditions when compared to the corresponding non-pipeline implementation?

18. A non-pipeline sysem takes 40 ns to process a task. The same task can be processed in a 6-segment pipeline with a clock cycle of 10s. What is the speed up ratio of the pipeline for 100 tasks.....?

19. In a non-pipelined single-cycle-per-instruction processor with an instruction cache, the average instruction cache miss rate is 5%. It takes 8 clock cycles to fetch a cache line from the main memory. Disregarding data cache misses, what is the approximate average CPI (cycles per instruction).....?

20. Suppose that an unpipelined processor has a cycle time of 25 ns, and that its data path is made up of modules with latencies of 2, 3, 4, 7, 3, 2 and 4 ns (in that order). In pipelining this processor, it is not possible to rearrange the order of the modules (for examples, putting the register read stage before the instruction decode stage) or to divide a module into multiple pipeline stages (for complexity reasons). Given pipeline latches with 1 ns latency: If the processor is divided into the rewest number of stages that allow is to achieve the minimum latency from part 1, what is the latency of the pipeline?

Hints & Solutions

1. **(b)** Processing time = $\frac{400}{4}*40$ = 4000n sec, sinch each processor gets 100 instructions they can finish them in parallel.
Single pipeline processing time = 400 × 10 = 4000n sec

2. (0.48) t avg = (1 + 30% × 4) × 10ns = 22ns

throughput = $\frac{1}{22}=0.48$

3. (4.76) t_n = 50n sec $\Rightarrow 5=\frac{50\times100}{(6+99)\times10}=4.76,$ time required for the first task is 60ns and for the remaining 99 tasks time required is 990

4. **(b)** On a 5-stage pipeline :
Extra memory latency = (1-h)* extra miss latency*number of memory instruction = (1-0.9) * 20 * 500,000 = 1,000,000 cycles.
On a 6-stage pipeline, the extra memory latency is the same as the above.
Total run-time on 5-stage pipeline = 2.5 + 1 = 3.5 million cycles.
Total run-time on 6-stage pipeline = 2.6 + 1 = 3.6 million cycles.

5. **(2)** A bypass is needed from the start of WB to the start of EX. Two stall cycles are needed. One more than in regular DLX pipeline.

6. **(b)** 50 million groups can hold 50 million × 3 = 150 million instructions.
However only 120 million instructions are present.
Remaining 150 - 120 = 30 million instructions are NOPs.

7. **(17.5)** Run-time for ignoring control hazards = 50 million cycles.
1 cycle per group. Number of mis-predicted branches = Number of branches * mis-prediction rate = 25 million × 0.7 = 17.5 million.
Stalls due to control hozards = Number of mis predicted branches branch penalty = 17.5 million ★$_2$ cycles = 35 million cycles.
= Total run time = 50 + 35 = 85 million cycles.

8. **(15)**

9. **(165)** Instruction pipeline with five stages without any branch predicition: Delays for FI, DI, FO.EI and WO are 5,7,10,8,6 ns respectively.
The maximum time taken by any stage is 10 ns and additional 1ns is required for delay of buffer.
∴ The total time for an instruction to pass from one stage to another in 11 ns.
The instructions are executed in the following order
I_1, I_2, I_3, I_4, I_9, I_{10}, I_{11}, I_{12}
Execution with Time
Now when I_4 is in its execution stage we detect the branch and when I_4 is in WO stage we fetch I_9 so time for execution of instructions from I_9 to I_{12} is = 11* 5 +(4 – 1) * 11= 88 ns.
But we save 11 ns when fetching I_9 i.e., I_9 requires only 44 ns additional instead of 55ns because time for fetching I_9 can be overlap with WO of I_4.
∴ Total Time is = 88 + 88 – 11 = 165 ns

10. **(8)**

11 **(10)**

Clock cycle	$R_0 = M$ [loc 1]	$R_0 = R_0 + R_0$	$R_2 = R_2 - R_0$
1	IF		
2	RD		
3	EX	IF	
4	MA	RD	
5	WB	EX	
6		MA	IF
7		WB	RD
8			EX
9			MA
10			WB

Thus, total number of clock cycles required = 10

12. **(c)** For incrementing the data, the auto-increment addressing mode is used which purely depends on the size of the data. For example:
Regs [R_1] ← Regs [R_1] + Mem [Regs [R_2]
Regs [R_2] ← Regs [R_2] + d

13. **(30)** Number of cycle for executing loop i=1 is 15
So, total no. of cycle is 2*15(loop runs 2 time), if we calculate no. of pipeline continuous no option will match, so here we have to take loop independently.

14. **1.54**

	No. of stages	Stall cycle	Stall frequency	Clock period	Avg. access time
Old design	5	5	20%	2.2ns	P
New Design	8	5	20%	1 ns	Q

$$P=\left[80\%(1\text{clock})\quad 20\%\left(\begin{matrix}1\\ \text{completion}\end{matrix}\quad\begin{matrix}2\\ \text{stall clock}\end{matrix}\right)\right]\times T_{c-p}$$

P = (.8 + .6) × 2.2ns = 3.08ns

$$Q=\left[80\%(1\text{clock})\quad 20\%\left(\begin{matrix}1\\ \text{completion}\end{matrix}\quad\begin{matrix}5\\ \text{stall clock}\end{matrix}\right)\right]\times T_{c-p}$$

P = (.8 + .12) × 1ns = 2ns

So the value of $\frac{P}{Q}$ $\frac{3.08ns}{2ns}$ = 1.54

15. **1.68**
Total instructions = 100 instruction fetch operation + 60 memory operant read operation + 40 operant write operation
Therefore, total of 200 instructions
Time taken for fetching 100 instructions (equivalent to read) = 90 × 1ns + 10 × 5ns = 140ns
Memory operand Read operations = 90%(60) × 1ns + 10% (60) × 5ns
= 54ns + 30ns = 84ms
Memory operands write operation time = 90%(40) × 2ns + 10%(40) × 10ns
= 72ns + 40ns = 112ns
Total time taken for executing 200 instructions = 140 + 84 + 112 = 336ns

∴ Average memory access time = $\frac{336}{200}$ns = 1.68ns

16. **4 to 4**

17. **(b)** Speed up = $\frac{(5+6+11+8)\times1}{(11+1)}=\frac{30}{12}=2.5$

18. **4.76**

$$S=\frac{nt_p}{(k+n-1)t_p}=\frac{100\times50}{(6+100-1)\times10}=\frac{500}{105}$$

19. **1.4**
CPI = (1 inst-per-cycle) + (0.05)(8 cycles/miss) = 1.4

20. **(40)** Minimum cycle time =latency of longest module in data path plus pipeline latch time =7ns + 1 ns=8 ns.
assuming that thereis no limit on the number of pipeline stages. We need to know how many pipeline stages the processor requires to operate at a cycle time of 8ns. We can group any set of adajacent modules with total latencies of 7 ns or less into a single stage. Doing this gives 5 pipeline stages.
5 stages * 8 ns cycle time = 40 ns latency