Archit Agrawal
2020512123
Archit Agrawal

CS208 : Computer Organisation and Architecture

Remote Mid Semester Exam

Question 1 (a)

The cycle time can be calculated as

$$\tau = \max\{150ns, 120ns, 160ns, 140ns\} + 5ns$$

$$\tau = 165ns$$

∴ since, there are four stages, $k = 4$.
items to be processed, $n = 1000$.

time taken to process 1000 items

$$= (k + n - 1)\tau$$
$$= (4 + 1000 - 1) \times 165ns$$
$$= 1003 \times 165ns$$
$$= 165495ns$$
$$= 165.495\ \mu s$$

Hence, time taken to process 1000 items is $165.495\ \mu s$.

Question 1(b)

i) RAW hazard between I_1 and I_2 as value of R_1 will be read before writing it.

ii) RAW hazard between I_3 and I_4 as value of R_1 will be read before writing it.

iii) RAW hazard between I_2 and I_4 as value of R_2 will be read before writing it.

Avdeit Agrawal
2020512213
Avdeit Agrawal (3)

Question 2(a)

i) SUB R_1, R_2, R_3

ADD R_4, R_1, R_8

(direct dependency) (I_1 & I_2)

ii) SUB R_1, R_2, R_3

OR R_6, R_1, R_7

(direct dependency) (I_1 & I_3)

iii) SUB R_1, R_2, R_3

AND R_8, R_4, R_8

(direct dependency) (I_1 & I_4)

iv) AND R_8, R_1, R_8

LW $R_{10}, 0(R_8)$

direct dependency (I_4 & I_5)

v) LW $R_{10}, 0(R_8)$

SW $10(R_5), R_{10}$

direct dependency (I_5 & I_6)

vi) AND R_8, R_1, R_8

LW $R_{10}, 0(R_9)$

SW $10(R_5), R_{10}$

(transitive dependency) (I_4, I_5 & I_6)

Question 2(b)

SUB R_1, R_2, R_3

~~(3 stalls)~~

3 stalls

ADD R_4, R_1, R_4

3 stalls

OR R_6, R_1, R_7

3 stalls

AND R_8, R_1, R_8

~~4 stalls~~ 0 stalls

LW $R_{10}, 0(R8)$

1 stall

SW $10(R5), R_{10}$

Note: There was no case of ALU and load in slides. If we consider it to be 1, stalls increases by 1.

Question 3(a)

In 4-bit CLA, expression for S_i is:

$$S_i = A_i \oplus B_i \oplus C_i$$

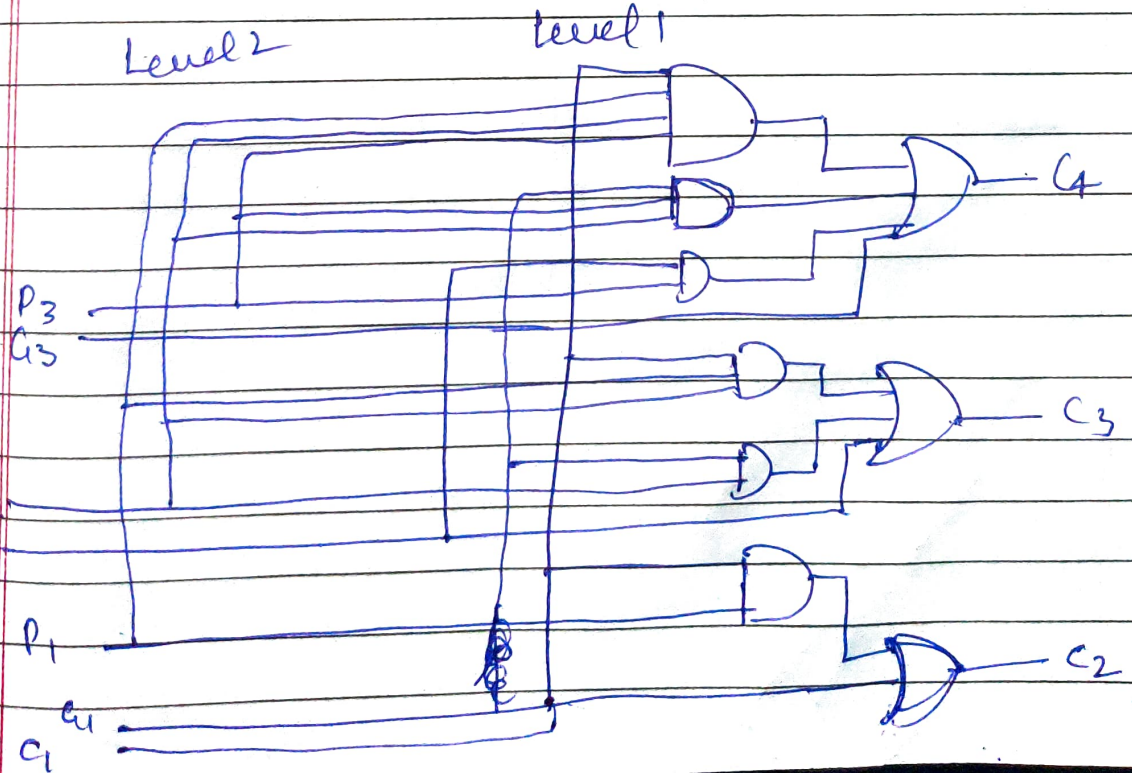
where C_i is carry-in.

So, two XOR gates are involved in computing the sum value. So, the time taken to calculate sum is 2 (delay of EX-OR) gate = 40 ns.

Now, for the carry,
C_{i+1} is the summation of P_i (carry propagator) * C_i and G_i (carry generator).

Since, in CLA,

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1$$
$$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in}$$



In level 1, all AND gates are considered.
In level 2, all OR gates are considered.

Since, in each level the respective gates work in parallel, so the total time is time taken at both levels that is 10 ns at level 1 and 10 ns at level 2.

total propagation delay = 40 ns + 20 ns
= 60 ns

Arshit Agrawal
2020512173
Arshit Agrawal

~~Roll~~ Question 3(b)

Roll No:- 2020512173

$(1)_{10} \rightarrow (0001)_2$

$(3)_{10} \rightarrow (0011)_2$

M → Multiplicand = 0001

Q → Multiplier = ~~(0011)~~ 0011

Q-1 → 0

AC → 0000

	AC	Q	Q-1	n	Operation
(1)	0000	0011	0	4	{10} → AC → AC-M → Arithm R.S.
	1111	0011	0	4	AC ← AC-M done
	1111	1001	1	3	ARS done
(2)	1111	1001	1	3	{11} → ARS
	1111	1100	1	2	ARS done
(3)	1111	1100	1	2	{01} → AC = AC+M → ARS
	0000	1100	1	2	AC ← AC+M done
	0000	0110	0	1	ARS done
(4)	0000	0110	0	1	{00} → ARS
	0000	0011	0	0	

R.W.

0001
0
1110
1111
1111
+ 0001
0000

$\therefore (0001)_2 \times (0011)_2 = (0011)_2$

Question 4 (a)

PC Relative	Direct
• Effective address is calculated by adding the contents of the CPU register with the address part of the instruction.	• The address for fetching the operand is already provided the address part of the instruction.
• It results in shorter address field	• It results in longer address field.
• Often with branch type instructions	• not used in branch type instructions
• used for writing relocatable code, position independent code etc.	• used to access static data.
• used to handle recursive procedures	• cannot handle recursive procedures

Archit Agrawal
2020512B

Question 4(b)

Stack Based	Accumulator Based
• Operands are implicitly on top of stack	• Operand is stored in an accumulator implicitly.
• Simple Mode of Expression Evaluation,	
• ~~short instructions~~.	