INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Mid Sem Solutions

Date: 24-02-23 **Timing:** **2:00 pm - 4:00 pm** **Place:** **NR223** **Total Marks =** 30
Subject No : **CS60003** **HIGH PERFORMANCE COMPUTER ARCHITECTURE**
Department/Centre/School : **Computer Science and Engineering**
Answer all questions. In case of reasonable doubt, make assumptions and state them upfront. Marks will be deducted for claims without proper reasoning. Write your answers (with all analysis, justification, calculation etc) within the allotted time.

---

1. Solution 1 **[5]**

$$\begin{aligned} CPI_{original} &= 1*0.4 + 4*0.2 + 2*0.3 + 3*0.1 \\ &= 0.4 + 0.8 + 0.6 + 0.3 \\ &= 2.1 \end{aligned}$$

$$\begin{aligned} CPI_{enhanced} &= 1*0.4 + 2*0.2 + 2*0.3 + 3*0.1 \\ &= 0.4 + 0.4 + 0.6 + 0.3 \\ &= 1.7 \end{aligned}$$

(a) Assuming IC to be same in both cases and $cycle\ time = 1/4GHz$, we have

$$\begin{aligned} Speedup &= \frac{IC * CPI_{original} * cycle\ time}{IC * CPI_{enhanced} * cycle\ time} \\ &= \frac{2.1}{1.7} \\ &= 1.24 \end{aligned}$$

(b) Choice 1: As execution time of arithmetic instructions decreases by 20%,

$$\begin{aligned} CPI_{new}^{arithmetic} &= CPI_{old}^{arithmetic} - 0.2 * CPI_{old}^{arithmetic} \\ &= CPI_{old}^{arithmetic} * (1 - 0.2) \\ &= 0.8 * 1 = 0.8 \end{aligned}$$

On the other hand, as execution time of load instructions increases by 50%,

$$\begin{aligned} CPI_{new}^{load} &= CPI_{old}^{load} + 0.5 * CPI_{old}^{load} \\ &= CPI_{old}^{load} * (1 + 0.5) \\ &= 1.5 * 2 = 3 \end{aligned}$$

$$\begin{aligned} CPI_{choice\_1} &= 0.8*0.4 + 4*0.2 + 3*0.3 + 3*0.1 \\ &= 0.32 + 0.8 + 0.9 + 0.3 \\ &= 2.32 \end{aligned}$$

Choice 2: As execution time of load instructions decreases by 20%,

$$\begin{aligned} CPI_{new}^{load} &= CPI_{old}^{load} - 0.2 * CPI_{old}^{load} \\ &= CPI_{old}^{load} * (1 - 0.2) \\ &= 0.8 * 2 = 1.6 \end{aligned}$$

Similarly, as execution time of store instructions decreases by 20%,

$$\begin{aligned} CPI_{new}^{store} &= CPI_{old}^{store} - 0.2 * CPI_{old}^{store} \\ &= CPI_{old}^{store} * (1 - 0.2) \\ &= 0.8 * 3 = 2.4 \end{aligned}$$

On the other hand, as execution time of branch instructions increases by 10%,

$$\begin{aligned} CPI_{new}^{branch} &= CPI_{old}^{branch} + 0.1 * CPI_{old}^{branch} \\ &= CPI_{old}^{branch} * (1 + 0.1) \\ &= 1.1 * 4 = 4.4 \end{aligned}$$

$$\begin{aligned} CPI_{choice\_2} &= 1 * 0.4 + 4.4 * 0.2 + 1.6 * 0.3 + 2.4 * 0.1 \\ &= 0.4 + 0.88 + 0.48 + 0.24 \\ &= 2 \end{aligned}$$

Since the CPI for the original enhancement is the least among all the enhancements, we prefer to keep the original enhancement.

2. Solution 2 **[10]**

(a)
$$\begin{aligned} CPI_{P1} &= 10 * 0.2 + 8 * 0.2 + 4 * 0.5 + 4 * 0.1 \\ &= 2 + 1.6 + 2 + 0.4 \\ &= 6 \end{aligned}$$

(b)
$$\begin{aligned} CPI_{P2} &= 12 * 0.2 + 10 * 0.2 + 6 * 0.5 + 5 * 0.1 \\ &= 2.4 + 2 + 3 + 0.5 \\ &= 7.9 \end{aligned}$$

(c) Given $f_{P2} = 2 * f_{P1}$. Now $C_{P1} = 1/f_{P1}$ and $C_{P2} = 1/f_{P2} = 1/(2 * f_{P1}) = C_{P1}/2$, where C_{P1} and C_{P2} are clock cycle time of processors $P1$ and $P2$. Also $IC_{P1} = IC_{P2}$

$$\begin{aligned} Speedup &= \frac{Ex_{P1}}{Ex_{P2}} \\ &= \frac{IC_{P1} * CPI_{P1} * C_{P1}}{IC_{P2} * CPI_{P2} * C_{P2}} \\ &= \frac{CPI_{P1} * C_{P1}}{CPI_{P2} * (C_{P1}/2)} \\ &= \frac{CPI_{P1} * 2}{CPI_{P2}} \\ &= \frac{6 * 2}{7.9} \\ &= \frac{12}{7.9} = 1.52 > 1 \end{aligned}$$

Since speedup is greater than 1, $P2$ is faster than $P1$.

(d) Since faster branch execution unit reduces the latency of branch instructions by a factor of 4, $CPI_{new}^{branch} = CPI_{old}^{branch}/4 = 4/4 = 1$.

$$\begin{aligned} CPI_{P1}^{branch} &= 10 * 0.2 + 8 * 0.2 + 4 * 0.5 + 1 * 0.1 \\ &= 2 + 1.6 + 2 + 0.1 \\ &= 5.7 \end{aligned}$$

Since memory device reduces the latency of the memory operations (both load and store) by a factor of 2, $CPI_{new}^{load} = CPI_{old}^{load}/2 = 10/2 = 5$, and $CPI_{new}^{store} = CPI_{old}^{store}/2 = 8/2 = 4$.

$$\begin{aligned} CPI_{P1}^{memory} &= 5 * 0.2 + 4 * 0.2 + 4 * 0.5 + 4 * 0.1 \\ &= 1 + 0.8 + 2 + 0.4 \\ &= 4.2 \end{aligned}$$

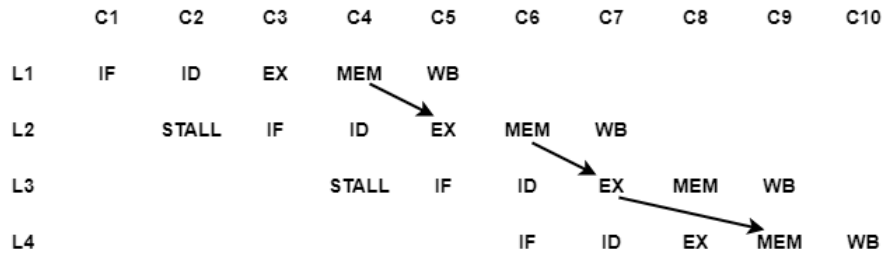
Since $CPI_{P1}^{memory} < CPI_{P1}^{branch}$, we choose the second design.

3. Solution 3 **[5]**

(a) We have the following dependencies:

1. WAR on r1 from L_1 to L_2
2. RAW on r2 from L_1 to L_2
3. WAR on r1 from L_1 to L_3
4. RAW on r2 from L_1 to L_3
5. WAW on r1 from L_2 to L_3
6. RAW on r2 from L_1 to L_4
7. RAW on r1 from L_2 to L_4
8. RAW on r1 from L_3 to L_4

(b) The optimal pipeline schedule using forwarding is as follows:



$$CPI = \frac{number\ of\ cycles\ consumed}{number\ of\ instructions} = \frac{10}{4} = 2.5$$

4. Solution 4 **[10]**

The source of stalls are load-ALU (load followed by ALU operation), ALU-branch, load-branch, and the branch delay slots.
Memory indirect stalls = 0.10 * (27% + 10%) * 1 = 0.037
Load-ALU = 0.40 * 0.27 * 1 = 0.108
ALU-branch = 0.60 * 0.15 * 1 = 0.09
Load-branch = 0.10 * 0.15 * 2 = 0.03
Branch delay = (0.15 + 0.03) * 1 = 0.18

$$CPI_{unoptimized} = 1 + stalls = 1 + 0.037 + 0.108 + 0.09 + 0.03 + 0.18 = 1.445$$

Calculating stall for the optimized version
Memory indirect stalls: 0.037 * 0.5 = 0.0185 (we remove 50% of these)
Load-ALU: 0.108 * 0.4 = 0.0432 (we remove 60% of these)

ALU-branch: 0.09 * 0.55 = 0.0495
Load-branch: 0.03 * 0.5 = 0.015 (considering 50% load as 2 cycle stall and 50% no stall)
Branch delay: (0.15 + 0.03) * 1 * 0.35 = 0.063

$$CPI_{optimized} = 1 + stalls = 1 + 0.0185 + 0.0432 + 0.0495 + 0.015 + 0.063 = 1.1892$$

Optimized vs Unoptimized versions:

$$\begin{aligned} Speedup &= \frac{CPI_{unoptimzed}}{CPI_{optimzed}} \\ &= \frac{1.445}{1.1892} = 1.215 \end{aligned}$$

The optimized version is 21.5% faster than the unoptimized.

Optimized vs Idea:

$$\begin{aligned} Speedup &= \frac{CPI_{optimized}}{CPI_{ideal}} \\ &= \frac{1.1892}{1} = 1.1892 \end{aligned}$$

The ideal machine is 18.92% faster than the optimized version.

5. Solution 5 **[5]**

We can use the formula $CPU\ time = IC * CPI * Clock\ cycle\ time$

For RISC:

$CPI = 1.3$ and $Clock\ cycle\ time = \frac{1}{1.75GHz}$. Thus

$$\begin{aligned} CPU\ time &= IC_{RISC} * 1.3 * 1/1.75 \\ CPU\ time &= 0.743 * IC_{RISC} \end{aligned}$$

For CISC:

$Clock\ cycle\ time = \frac{1}{3.5GHz}$ and $CPI = 4 * 0.38 + 4 * 0.10 + 3 * 0.35 + 10 * 0.03 + 10 * 0.03 + 3 * 0.11 = 1.52 + 0.4 + 1.05 + 0.3 + 0.3 + 0.33 = 3.9$.

Also $IC_{CISC} = 0.7 * IC_{RISC}$, since IC for CISC machine is 30% smaller than IC for RISC machine. Thus

$$\begin{aligned} CPU\ time &= IC_{CISC} * 3.9 * 1/3.5 \\ CPU\ time &= IC_{RISC} * 0.7 * 3.9/1.75 \\ CPU\ time &= 0.78 * IC_{RISC} \end{aligned}$$

Thus, the RISC machine is faster on this benchmark by $0.78/0.743 = 1.0498$ or about 4.98%.

6. Solution 6 **[10]**

(a) Here since the repair is made speculatively at the execute time, all the entries between the one that got mispredicted and the tail of the queue are removed. To facilitate such operation, each branch must

also carry a tag indicating its entry in the FIFO queue of global registers. Therefore, we need another counter in the pipeline such that it will carry the tag with the branch statement indicating its entry in the FIFO queue.

If there are maximum of m branches in flight, the FIFO queue must contain atleast m entries. In case of misprediction, there is no problem if there are less than m entries as they will anyway be squashed. But in case of correct prediction, some entries will be lost.

(b) Consider the case of a `for` or `while` loop. If the tagging for branches is done using PC values instead of FIFO queue entries, there will be a chance of aliasing which will lead to erroneous updation/deletions from the queue.

7. Solution 7 **[5]**

(a) Number of entries in PHT = $2^3 = 8$

(b)

GR	**Prediction**	**Updated GR**	**Actual**	**Correct?**	**PHT**
000	NT	000	T	No	00, 00, 00, 00, 00, 00, 00, 00
001	NT	010	NT	Yes	01, 00, 00, 00, 00, 00, 00, 00
010	NT	100	NT	Yes	01, 00, 00, 00, 00, 00, 00, 00
100	NT	000	NT	Yes	01, 00, 00, 00, 00, 00, 00, 00
000	NT	000	T	No	10, 00, 00, 00, 00, 00, 00, 00
001	NT	010	NT	Yes	10, 00, 00, 00, 00, 00, 00, 00
010	NT	100	NT	Yes	10, 00, 00, 00, 00, 00, 00, 00
100	NT	000	NT	Yes	10, 00, 00, 00, 00, 00, 00, 00
000	T	001	T	Yes	11, 00, 00, 00, 00, 00, 00, 00

8. Solution 8 **[10]**

(a) B0 = 4, B1 = 4, B2 = 3, Total = 11

(b)

B0	B1	B2	B0	B1	B2	B0	B1	B2	B0	B1
11	10	01	10	01	00	01	00	00	01	00
X	X	X	X	✓	X	✓	✓	X	✓	X

(i) B0=2, B1=2, B2=0, Total=4

(ii) Branch prediction hit rate = (4/11)*100 =36.36%

(c) B0:

NT	NT	NT	NT
00	00	00	00
✓	✓	✓	✓

B1:

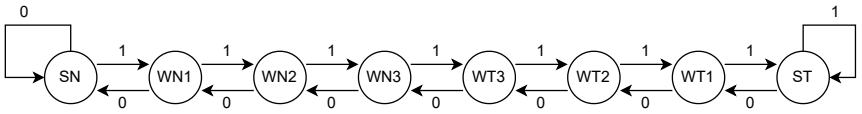
NT	NT	NT	NT
00	00	00	00
✓	✓	✓	X

B2:

T	T	T
00	01	10
X	X	✓

(i) B0:4, B1:3, B2:1, Total=8
(ii) Branch prediction hit rate = (8/11)*100 = 72.72%

9. Solution 9 **[10]**
(a)



(b) From the table we see that in the initial warm up period there are 8 mispredictions, after that there will be 100% correct predictions.

GR, 00, 01, 10, 11	**Prediction**	**Actual**	**Correct**
00, SN, SN, SN, SN	N	N	✓
00, SN, SN, SN, SN	N	T	X
01, WN1, SN, SN, SN	N	T	X
11, WN1, WN1, SN, SN	N	N	✓
10, WN1, WN1, SN, SN	N	N	✓
00, WN1, WN1, SN, SN	N	T	X
01, WN2, WN1, SN, SN	N	T	X
11, WN2, WN2, SN, SN	N	N	✓
10, WN2, WN2, SN, SN	N	N	✓
00, WN2, WN2, SN, SN	N	T	X
01, WN3, WN2, SN, SN	N	T	X
11, WN3, WN3, SN, SN	N	N	✓
10, WN3, WN3, SN, SN	N	N	✓
00, WN3, WN3, SN, SN	N	T	X
01, WT3, WN3, SN, SN	N	T	X
11, WT3, WT3, SN, SN	N	N	✓
10, WT3, WT3, SN, SN	N	N	✓
00, WT3, WT3, SN, SN	T	T	✓
01, WT2, WT3, SN, SN	T	T	✓
11, WT2, WT2, SN, SN	N	N	✓
10, WT2, WT2, SN, SN	N	N	✓
00, WT2, WT2, SN, SN	T	T	✓
01, WT1, WT2, SN, SN	T	T	✓
11, WT1, WT1, SN, SN	N	N	✓
10, WT1, WT1, SN, SN	N	N	✓
00, WT1, WT1, SN, SN	T	T	✓
01, ST, WT1, SN, SN	T	T	✓
11, ST, ST, SN, SN	N	N	✓

(c) Cost per entry = 2 + 4 * 3 = 14 bits