

12 BONUS: Branch Prediction [40 points]

Assume a processor that implements an ISA with eight registers (R0-R7). In this ISA, the main memory is byte-addressable and each word contains 4 bytes. The processor employs a branch predictor. The ISA implements the instructions given in the following table:

| Instructions | Description |
|---------------------------|--|
| la R_i , Address | load the <i>Address</i> into R_i |
| move R_i , R_j | $R_i \leftarrow R_j$ |
| move R_i , (R_j) | $R_i \leftarrow \text{Memory}[R_j]$ |
| move (R_i), R_j | $\text{Memory}[R_i] \leftarrow R_j$ |
| li R_i , Imm | $R_i \leftarrow \text{Imm}$ |
| add R_i , R_j , R_k | $R_i \leftarrow R_j + R_k$ |
| addi R_i , R_j , Imm | $R_i \leftarrow R_j + \text{Imm}$ |
| cmp R_i , R_j | Compare: Set sign flag, if $R_i < R_j$; set zero flag, if $R_i = R_j$ |
| cmp R_i , (R_j) | Compare: Set sign flag, if $R_i < \text{Memory}[R_j]$; set zero flag, if $R_i = \text{Memory}[R_j]$ |
| cmpi R_i , Imm | Compare: Set sign flag, if $R_i < \text{Imm}$; set zero flag, if $R_i = \text{Imm}$. |
| jg label | Jump to the target address if both of sign and zero flags are zero. |
| jnz label | Jump to the target address if zero flag is zero. |
| halt | Stop executing instructions. |

The processor executes the following program. Answer the questions below related to the accuracy of the branch predictors that the processor can potentially implement.

```

1      la R0, Array
2      move R6, R0
3      li R1, 4
4      move R5, R1
5      move R7, R1
6      move R2, R0
7      addi R2, R2, 4
8  Loop:
9      move R3, (R2)
10     cmp R3, (R0)
11     jg Next_Iteration
12     move R4, (R0)
13     move (R0), R3
14     move (R2), R4
15  Next_Iteration:
16     addi R0, R0, 4
17     addi R2, R2, 4
18     addi R1, R1, -1
19     cmpi R1, 0
20     jnz Loop
21     move R1, R7
22     addi R5, R5, -1
23     move R0, R6
24     move R2, R0
25     addi R2, R2, 4
26     cmpi R5, 0
27     jnz Loop
28     halt
29  .data
30  Array: word 5, 20, 1, -5, 34

```

- (a) [15 points] What would be the prediction accuracy using a global one-bit-history (last-time) branch predictor shared between *all* the branches? The initial state of the predictor is "taken".

Answer: 19/36.

Note that initial values of both R_1 and R_5 are 4; and they change only before the branches in lines 20 and 27 respectively. Both branches follow the pattern of T-T-T-NT, which creates a nested loop.

At each iteration of the internal loop, adjacent elements (pointed by R_0 and R_2) are swapped, if $Memory[R_0] \leq Memory[R_2]$. Then, both R_0 and R_4 are incremented by 4. So they point to the next element in the next iteration.

Therefore, the code sorts the elements in *Array* in increasing order.

Table below shows the behavior of each branch through the code. Here T means that the corresponding branch is taken at specified turn, whereas N indicates that it is not taken.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Line11 | T | | N | | N | | T | | | N | | N | | T | | T | | |
| Line20 | | T | | T | | T | | N | | | T | | T | | T | | N | |
| Line27 | | | | | | | | | T | | | | | | | | | T |
| | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| Line11 | N | | T | | T | | T | | | T | | T | | T | | T | | |
| Line20 | | T | | T | | T | | N | | | T | | T | | T | | N | |
| Line27 | | | | | | | | | T | | | | | | | | | N |

One-bit-history branch predictor suggests that the next branch's behavior will be the same with the last one. Table below shows the predictor states, hits, and misses through the execution.

| | | | | | | | | | | | | | | | |
|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Predictor State | T | T | T | N | T | N | T | T | N | T | N | T | N | T | T |
| Branch Behavior | T | T | N | T | N | T | T | N | T | N | T | N | T | T | T |
| Hit/Miss | H | H | M | M | M | M | H | M | M | M | M | M | M | H | H |
| | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| Predictor State | T | T | N | T | N | T | T | T | T | T | T | N | T | T | T |
| Branch Behavior | T | N | T | N | T | T | T | T | T | T | N | T | T | T | T |
| Hit/Miss | H | M | M | M | M | H | H | H | H | H | M | M | H | H | H |
| | 31 | 32 | 33 | 34 | 35 | 36 | | | | | | | | | |
| Predictor State | T | T | T | T | T | N | | | | | | | | | |
| Branch Behavior | T | T | T | T | N | N | | | | | | | | | |
| Hit/Miss | H | H | H | H | M | H | | | | | | | | | |

- (b) [15 points] What would be the prediction accuracy using a global two-bit-history (two-bit counter) branch predictor shared between *all* the branches? Assume that the initial state of the two-bit counter is "weakly taken". The "weakly taken" state transitions to the "weakly not-taken" state on misprediction. Similarly, the "weakly not-taken" state transitions to the "weakly taken" state on misprediction. A correct prediction in one of the "weak" states transitions the state to the corresponding "strong" state.

Answer: 26/36.

Explanation:

Table below shows the predictor states, hits, and misses through the code. Used abbreviations are as follows: ST: Strongly Taken, WT: Weakly Taken, WN: Weakly Not-taken, SN: Strongly Not-taken.

Branch behavior is the same with question (a), since both of them are shared predictors.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Predictor State | WT | ST | ST | WT | ST | WT | ST | ST | WT | ST | WT | ST | WT | ST |
| Branch Behavior | T | T | N | T | N | T | T | N | T | N | T | N | T | T |
| Hit/Miss | H | H | M | H | M | H | H | M | H | M | H | M | H | H |
| | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| Predictor State | ST | ST | ST | WT | ST | WT | ST | ST | ST | ST | ST | ST | WT | ST |
| Branch Behavior | T | T | N | T | N | T | T | T | T | T | T | N | T | T |
| Hit/Miss | H | H | M | H | M | H | H | H | H | H | H | M | H | H |
| | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | | | | | | |
| Predictor State | ST | ST | ST | ST | ST | ST | ST | WT | | | | | | |
| Branch Behavior | T | T | T | T | T | T | N | N | | | | | | |
| Hit/Miss | H | H | H | H | H | H | M | M | | | | | | |

- (c) [10 points] What would be the prediction accuracy using a local two-bit-history (two-bit counter) branch predictor that is separate for *each* branch? The initial state is "weakly taken" and the state transitions are the same as in part (b).

Answer:

- L11: 8/16
- L20: 12/16
- L27: 3/4
- All Branches: 23/36

Explanation: Private predictors update their states only based on the behaviors of corresponding branches.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---------------------|----|----|----|----|----|----|----|----|----|----|----|----|
| L11 Predictor State | WT | | ST | | WT | | WN | | | WT | | WN |
| L11 Branch Behavior | T | | N | | N | | T | | | N | | N |
| L11 Hit/Miss | H | | M | | M | | M | | | M | | H |
| L20 Predictor State | | WT | | ST | | ST | | ST | | | WT | |
| L20 Branch Behavior | | T | | T | | T | | N | | | T | |
| L20 Hit/Miss | | H | | H | | H | | M | | | H | |
| L27 Predictor State | | | | | | | | | WT | | | |
| L27 Branch Behavior | | | | | | | | | T | | | |
| L27 Hit/Miss | | | | | | | | | H | | | |
| | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| L11 Predictor State | | SN | | WN | | | WT | | WN | | WT | |
| L11 Branch Behavior | | T | | T | | | N | | T | | T | |
| L11 Hit/Miss | | M | | M | | | M | | M | | H | |
| L20 Predictor State | ST | | ST | | ST | | | WT | | ST | | ST |
| L20 Branch Behavior | T | | T | | N | | | T | | T | | T |
| L20 Hit/Miss | H | | H | | M | | | H | | H | | H |
| L27 Predictor State | | | | | | ST | | | | | | |
| L27 Branch Behavior | | | | | | T | | | | | | |
| L27 Hit/Miss | | | | | | H | | | | | | |
| | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| L11 Predictor State | ST | | | ST | | ST | | ST | | ST | | |
| L11 Branch Behavior | T | | | T | | T | | T | | T | | |
| L11 Hit/Miss | H | | | H | | H | | H | | H | | |
| L20 Predictor State | | ST | | | WT | | ST | | ST | | ST | |
| L20 Branch Behavior | | N | | | T | | T | | T | | N | |
| L20 Hit/Miss | | M | | | H | | H | | H | | M | |
| L27 Predictor State | | | ST | | | | | | | | | ST |
| L27 Branch Behavior | | | T | | | | | | | | | N |
| L27 Hit/Miss | | | H | | | | | | | | | M |