

| Cycle                          | 1  | 2  | 3  | 4   | 5   | 6   | 7  | 8   | 9   | 10 | 11  | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|--------------------------------|----|----|----|-----|-----|-----|----|-----|-----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| <b>lw \$t2, 12(\$s0)</b>       | IF | ID | EX | MEM | WB  |     |    |     |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>la \$t0, player_x</b>       |    | IF | ID | EX  | MEM | WB  |    |     |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>lw \$t1, 0(\$t0)*</b>       |    |    | IF | ID  | EX  | MEM | WB |     |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>addi \$t1, \$t1, 16**</b>   |    |    |    | IF  | ID  | -   | EX | MEM | WB  |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>blt \$t1, \$t2, loop***</b> |    |    |    |     | IF  | -   | ID | EX  | MEM | WB |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>lw \$t2, 12(\$s0)****</b>   |    |    |    |     |     |     | -  | IF  | ID  | EX | MEM | WB |    |    |    |    |    |    |    |    |    |    |    |    |    |

- \* t0 is computed at the end of EX stage of la. lw gets t0 forwarded at beginning of EX stage through a forwarding wire at the beginning of MEM stage of la.
- \*\* t1 is loaded at the end of MEM stage of lw. addi gets t1 forwarded at beginning of EX stage through a forwarding wire at the beginning of WB stage of la.
- \*\*\* blt ID stage can only happen at cycle 7 because at cycle 6 it was occupied by addi. Blt gets t1 forwarded at beginning of EX stage from MEM stage of addi.
- \*\*\*\* Thanks to branch predictor, by cycle 6, the processor already knows direction of branch. But since there is no BTB, it is only at cycle 7 that the branch target is decoded and the branch target can be fetched at cycle 8.