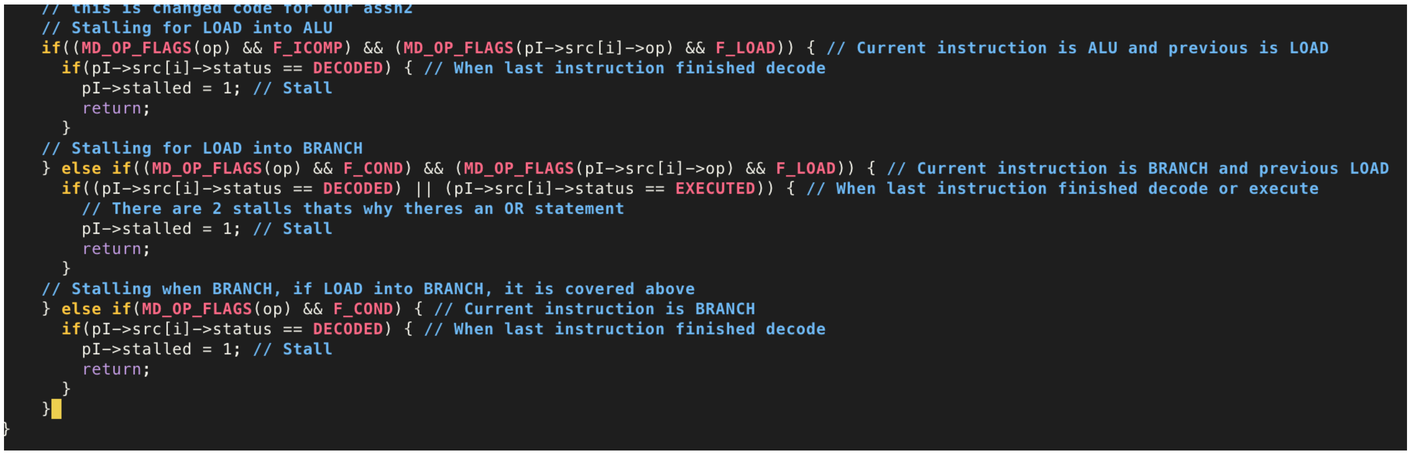
Jay Chang
39787114
Oct 20 2015

CPEN 411 Computer Architecture
W1 2015

Assignment #2: Performance Simulator of a Pipelined Processor

Report Analysis

For this assignment, we were given the task of simulating forwarding paths in our simulator to speed up the CPI when it is run. Please refer to Fig.1 below as I will explain my code below.



```
    // this is changed code for our assn2
    // Stalling for LOAD into ALU
    if((MD_OP_FLAGS(op) && F_ICOMP) && (MD_OP_FLAGS(pI->src[i]->op) && F_LOAD)) { // Current instruction is ALU and previous is LOAD
      if(pI->src[i]->status == DECODED) { // When last instruction finished decode
        pI->stalled = 1; // Stall
        return;
      }
    // Stalling for LOAD into BRANCH
    } else if((MD_OP_FLAGS(op) && F_COND) && (MD_OP_FLAGS(pI->src[i]->op) && F_LOAD)) { // Current instruction is BRANCH and previous LOAD
      if((pI->src[i]->status == DECODED) || (pI->src[i]->status == EXECUTED)) { // When last instruction finished decode or execute
        // There are 2 stalls thats why theres an OR statement
        pI->stalled = 1; // Stall
        return;
      }
    // Stalling when BRANCH, if LOAD into BRANCH, it is covered above
    } else if(MD_OP_FLAGS(op) && F_COND) { // Current instruction is BRANCH
      if(pI->src[i]->status == DECODED) { // When last instruction finished decode
        pI->stalled = 1; // Stall
        return;
      }
    }
}
```

Fig.1 The code I changed in sim-scalar-cpen411.c

For our case, there are 3 different types of stalls: ALU after LOAD, BRANCH after LOAD, and standalone BRANCH. The first IF statement covers the case when theres an ALU after LOAD, and we need to stall once after the previous code decoded. So we check if the current and previous instructions are ALU and LOAD respectively, and if it is we wait till the previous instruction is past the DECODE state, then we stall. The same methodology applies to the 2nd and 3rd IF statements, we check what are the current instructions and where the status is of the instruction, if it the right type, then we do the stalling.

For the second part, we can see the improvements and the speedup and mean on the next page (Fig.2). When analyzing the data we derived, we can notice that there are significant increase in CPI when we apply forwarding on the simulator. This is consistant with what we learn in class as forwarding reduces the cycles needed when there are stalls. Also, the harmonic mean that we calculate for the speedup is **1.729**.

Jay Chang
39787114
Oct 20 2015

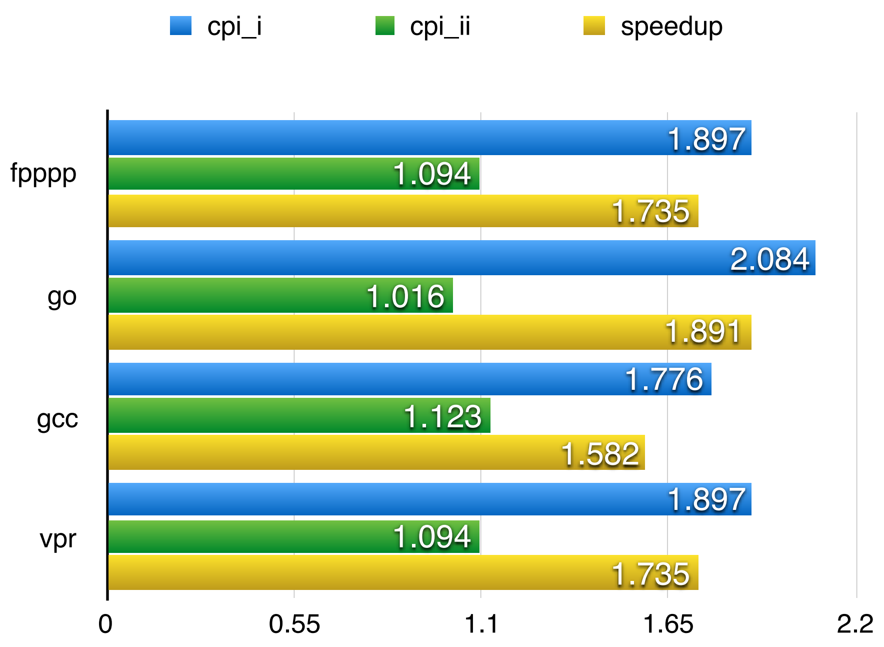


Fig.2 The bar graph for the change in CPI and the speedup change