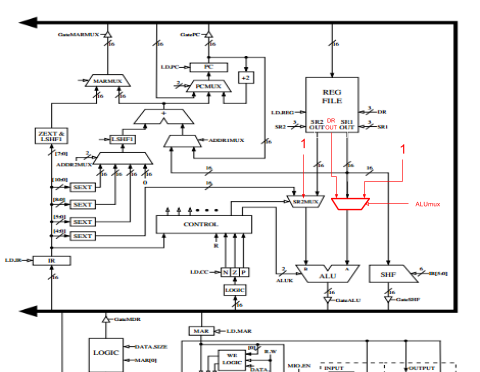
Cody Martin

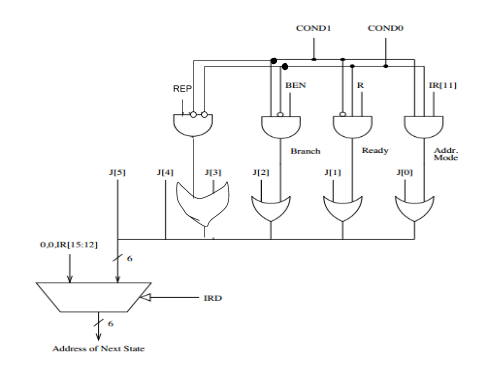
18-447 Homework 3

``1. a)



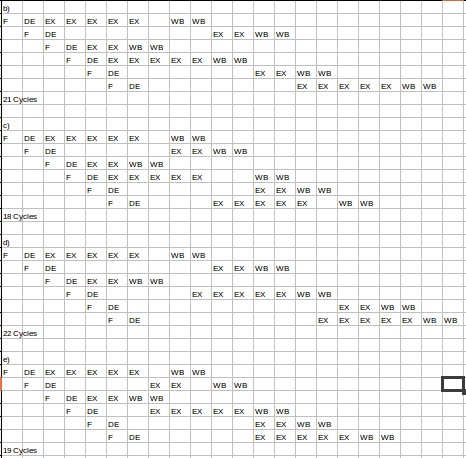
b)

c)



2.

a) This takes 45 cycles



3.

a)This would take 2 delay slots.

b) Delay slots underlined.

(i) ADD R5 R4, R3

J X

OR R3 R1, R2

SUB R7 R5, R6

LW R10 (R7)

ADD R6 R1, R2

X:

(ii) ADD R5 R4, R3

SUB R7 R5, R6

BEQ R5 R7, X

OR R3 R1, R2

NOP

LW R10 (R7)

ADD R6 R1, R2

X:

(iii) OR R5 R1, R2

SUB R7 R5, R6

BEQ R5 R7, X

ADD R2 R4, R3

NOP

LW R10 (R7)

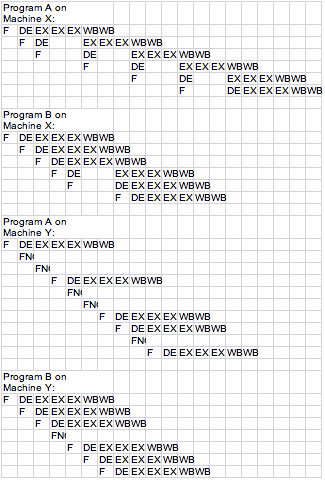
ADD R6 R1, R2

X:

c) I would delay the start of an instruction by one cycle. This would let the previous instruction decode, then while the previous instruction starts its executing, the next instruction is fetched, which would be the one delay slot. When the target address is calculated in the previous instruction’s execute stage, the following instruction would begin one cycle later, being the branch target instruction. Therefore, only one instruction would begin before the branch target address was resolved, making one delay slot instead of two.

4.

a)



Cycle count:

Program A on Machine X: 20 cycles

Program B on Machine X: 14 cycles

Program A on Machine Y: 16 cycles

Program B on Machine Y: 13 cycles

b) Machine Y takes less cycles to execute on both Program A and Program B.

c) Machine Y takes less on both. This is because when there is a data dependency, Machine X has to stall the pipeline for the whole execution of an instruction for each dependence within a proximity of , which is 2 cycles. Machine Y on the other hand has to stall AT MOST 2 cycles for each dependence within a proximity of 2 instructions. The ability for the compiler in Machine Y to rearrange the instructions also eliminates these stalls as much as possible, yielding a fuller pipeline.

d) I would say that Machine Y does have higher performance. Since both machines are multi cycle, they probably have similar clock cycle lengths. They are both running the same programs, so the number of instructions are the same for both machines. The only thing differentiating them in performance is the cycles per instruction, which on both programs, Machine Y has less. So it is safe to say that Machine Y has higher performance.

e)

Program A on Machine X: 4\*5 = 20 bytes

Program B on Machine X: 4\*6 = 24 bytes

Program A on Machine Y: 4\*10 = 40 bytes

Program B on Machine Y: 4\*28 bytes

f) Machine X has a lower code size for both Program A and Program B because there are no nop insertions.

g) Yes, Machine X has a lower code size for both. Again, this is because it does not insert any instructions like Machine y does.