For all Beta related questions, you should make use of the Beta documentation, the Beta Instruction Summary, the Unpipelined Beta Diagram and the Pipelined Beta

Pipelined Beta: 1

0.990/1960784313726(f point (ungraded)
Assume a 5-stage Beta with full bypass paths, as presented in lecture. Remember that, in the pipelined Beta, the data read by loads becomes available in the writeback stage.

Fill in the pipelining diagram below using the command names (e.g., LD, ST) that live in each stage. If no operation happens in a particular slot, enter the word EMPTY. In steady state, how many cycles does this code consume in the pipelined Beta?

LD(R0,0,R2) LD(R1,0,R3) ST(R2,0,R1) ST(R3,0,R0) 3 10 ST ST ΙD ΙD ST FMDTV FMDTV FMDTV FMDTV FMDTV FMDTV FMDTV LD LD ST ST ST EMPTY EMPTY EMPTY EMPTY EMPTY EMPTY ~ LD LD NOP ST ST EMPTY EMPTY EMPTY EMPTY EMPTY EXE • LD LD NOP ST EMPTY EMPTY EMPTY EMPTY . . LD NOP ST LD ST EMPTY EMPTY EMPTY WB

If this code was executed repeatedly, how many cycles would it take in steady state?

Number of Cycles in Steady State:



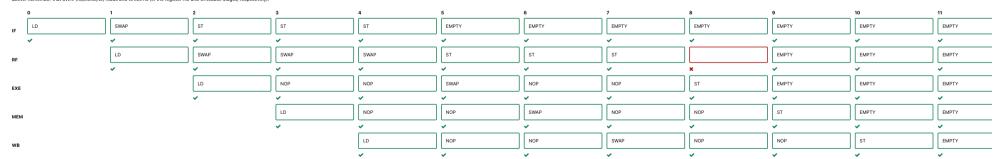
Suppose a new instruction SWAP, SWAP(Ra, literal, Rc), was added to the Beta. SWAP swaps a register value and a memory location.. The behavior of this instruction is as follows:

tmp <-- Mem[Reg[Ra] + SEXT(literal)]
Mem[Reg[Ra] + SEXT(literal)] <-- Reg[Rc]
Reg[Rc] <-- tmp</pre>

Using this new instruction, the code sequence above can be rewritten as:

LD(R0,0,R2) SWAP(R1,0,R2) ST(R2,0,R0)

Assume that the SWAP instruction is implementable in the fully bypassed 5-stage Beta, and also returns data in the writeback stage. Fill in the pipelined timing diagram as above. Remember that SWAP(Ra,literal,Rc) reads and writes Rc (in the register file and writeback stages, respectively).



If this code was executed repeatedly, how many cycles would it take in steady state?

Number of Cycles in Steady State:



Submit

6.004.3x Courseware | edX 25/07/24, 07:44