Help

You will need to reference the LC-3 pipeline diagram for Homework 1.

Please click here to open in a new window.

HOMEWORK 1. PIPELINING (1/1 point)

In the non-pipelined LC-3 implementation, the data to be stored for an STR instruction is read from the register file (after the read of the register associated with the address calculation), passed through the ALU, and sent to the memory using the bus. However, in the pipelined LC-3, the ALU is used to form the memory address and cannot be simultaneously used to pass through the stored value. Moreover, in ID, both register values are supposed to be read at the same time, but they both come from SR1.

Which of the following describes how these two issues are resolved in the LC-3 pipeline?

- (•) SR1 is used to read the register for the address calculation, SR2 for the data to be stored, and that data goes around the ALU in EX through a separate datapath. ✔
- ( ) The register file is read in two consecutive cycles, the first for the register for the address calculation, and the second for the data, and these are pipelined serially through EX.
- ( ) The register file is read in two consecutive cycles, the first for the data, and the second for the register for the address calculation, and these are pipelined serially through EX.
- ( ) STR instructions cannot be supported in a pipelined implementation of the LC-3.

EXPLANATION

From the LC-3 pipeline diagram, we can trace a path from SR2 to the memory. Thus, in ID, we use SR1 to read the register for the address calculation and SR2 to read the data to be stored, which is different than the non-pipelined LC-3 implementation. In EX, the address calculation is performed with the SR1 data while the SR2 data goes around the ALU. In MEM, the address and the data are sent to the Memory.

Hide Answer *You have used 2 of 2 submissions*

Show Discussion New Post

HOMEWORK 2. PIPELINING (1 point possible)

Pipelining may require fetching an instruction from memory at the same time as data is read from, or written to, memory for a load or store instruction.

Which of the following describes how this memory conflict is resolved in the LC-3 pipeline?

Help

- The memory is accessed in two consecutive clock cycles, the first for instruction fetch, and the second for load or store.
- The memory has the capability to simultaneously fetch an instruction, read data for a load, and write data for a store.
- The memory has the capability to simultaneously fetch an instruction and either read data for a load or write data for a store. ✔
- (•) The LC-3 ISA is designed so that conflicts between instruction fetch and loads and stores cannot occur. ✖

EXPLANATION

From the LC-3 pipeline diagram, we see two address inputs on the right side of the Memory, one from the PC for instruction fetch and a second from the ALU for loads and stores. On the left, we see the instruction path to the IR, the load datapath back to the register file, and the store datapath from SR2.

With two address inputs, we are able to simultaneously fetch an instruction and either read data for a load or write data for a store. It's not necessary to be able to simultaneously load and store data, since only one of these instructions will be in the MEM stage in any given clock cycle.

Hide Answer *You have used 2 of 2 submissions*

Show Discussion New Post