

Inf2C Computer Systems

Tutorial 4, Week 7

Solutions

1. Modulo-6 Counter

Design a synchronous Modulo-6 counter that will be able to count up to 5 clock positive edges. When it reaches 5, it resets to 0 and it starts the whole process again. In order to design this counter you can use D flip-flops and any one, two or three input gate you like.

Answer *First note that we need to be able to count up to 6, which means that we need 3 bits to represent all the possible states of our FSM. The state machine can be seen in Figure 1. From this FSM we can derive the next-state transition table seen in Table 1 and from that we have the circuit shown in Figure 2. 'X's are used in the transition table to indicate when we don't care about the values of the next-state bits. In the circuit, we have chosen to make the values for nc1 and nc2 all 0s and for nc0 to follow the negation of c0.*

In general, finite state machines have both inputs and outputs, and these must also be taken into account when drawing up the table. The Logic Design lecture slides show an example of a table that includes both inputs and outputs.

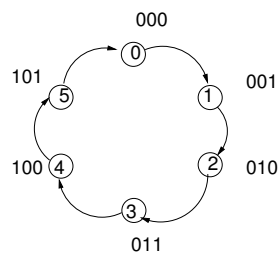


Figure 1: Modulo Counter FSM

c2	c1	c0	nc2	nc1	nc0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	X	X	X
1	1	1	X	X	X

Table 1: Modulo Counter transition table

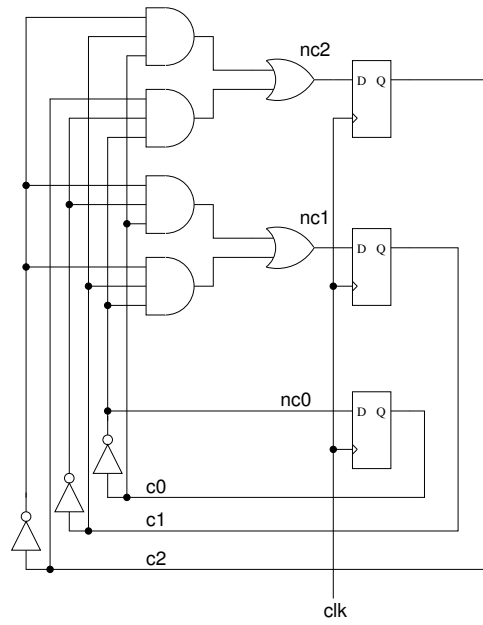


Figure 2: Modulo Counter Circuit

2. Single-Cycle Processor

Discuss the steps in executing the `sw` instruction in the single-cycle datapath presented in Figure 3.

Next discuss the changes needed to the datapath and control to support the `jr` instruction.

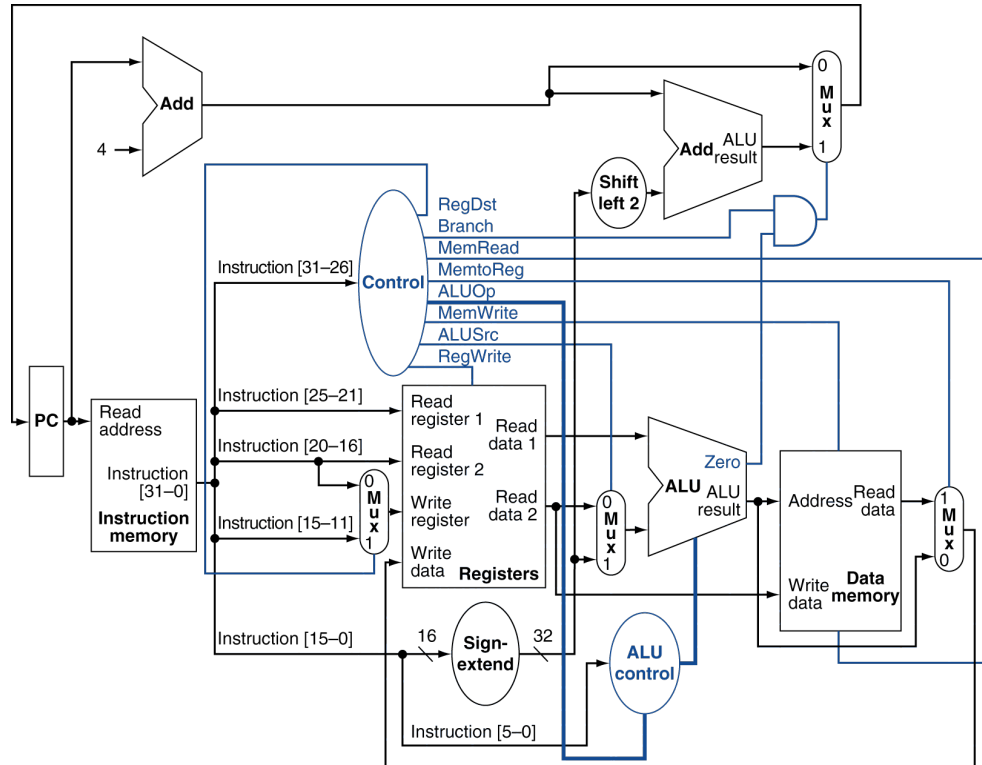


Figure 3: MIPS Single-Cycle Datapath

Answer

SW:

The execution of the SW instruction is shown in Fig 4. It is similar to the LW instruction that we saw in class. Reg 1 is added with the sign-extended immediate to form the memory address. Meanwhile, Reg 2 (write value) is bypassed from the register file directly to the Write Data port of the Data Memory. The Mem Write control signal is set to enable the memory to perform the write.

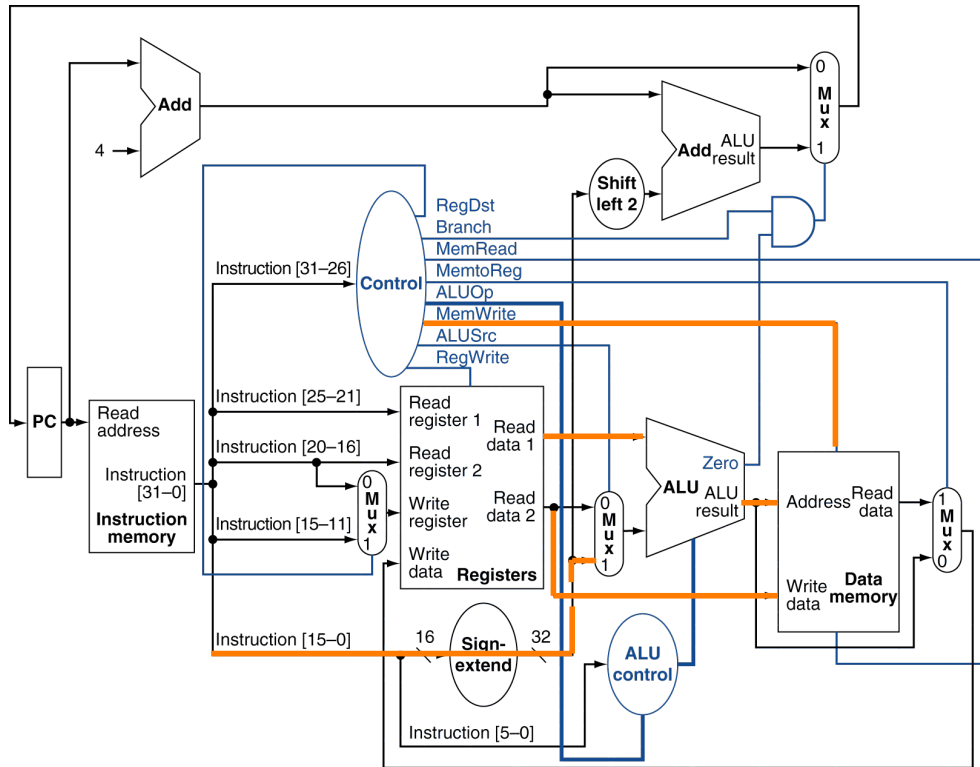


Figure 4: SW instruction

JR:

Fig 5 shows the datapath and control for the JR instructions. The required modifications include a new mux in the datapath (highlighted in red) to select between the branch PC and the jump register, as well as a new control signal, Jump, to choose the appropriate mux input.

The execution of JR starts by fetching and decoding the instruction as usual. Register \$rs specifies the jump target, whose value is read out

through the Reg 1 register file output. The Jump control signal is set to 1 to select the register as the new PC.

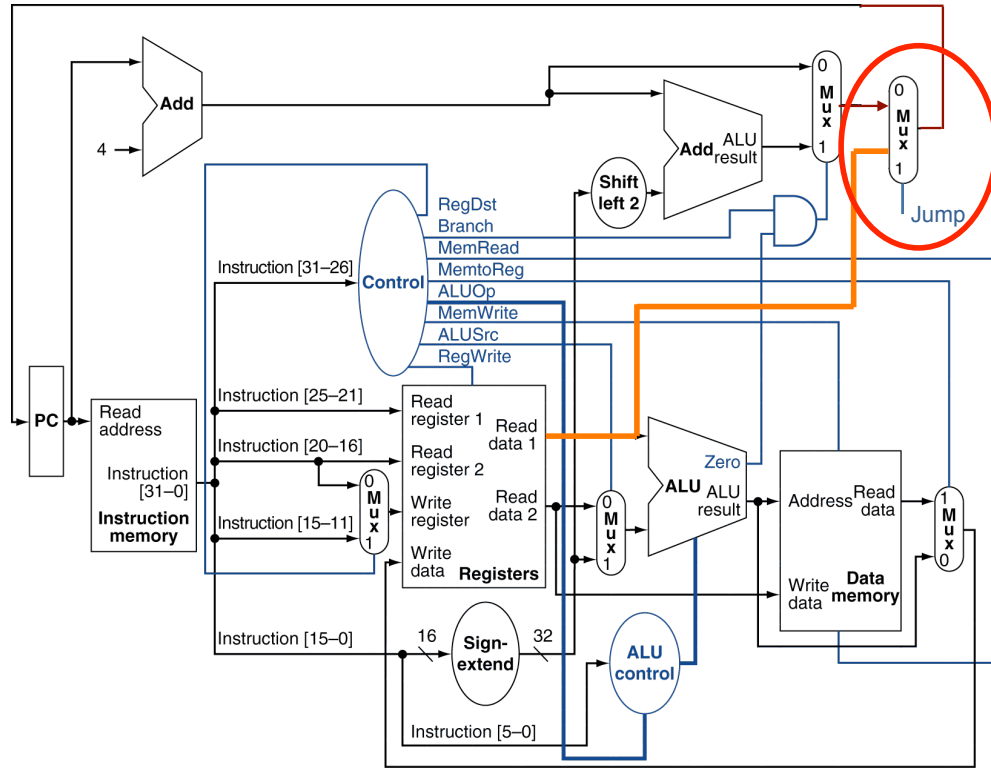


Figure 5: JR instruction