Laboratory Exercise 6

Dedicated microprocessor for counting 1 to 10

The purpose of this exercise is to build and use Datapath and finite state machine. The designed circuits are to be implemented on an Altera DE0-CV, DE1-SoC, or DE2-115 Board.

Students are expected to have a basic understanding of combinational and sequential component and sufficient familiarity with the VHDL hardware description language.

Part I

From the algorithm shown in Figure 1 below, we see that there are three data manipulation instructions: lines 1, 3, and 4. Line 2 is not a data manipulation statement, but rather, it is a control statement.

```
1          i = 0
2          WHILE (i ≠ 10){
3             i = i + 1
4             OUTPUT i
5          }
```

Figure 1: algorithm for counting 1 to 10

We have derived the three corresponding control words for these three data manipulation instructions for controlling the dedicated datapath shown in Figure 2. These three control words are shown here in Figure 3.

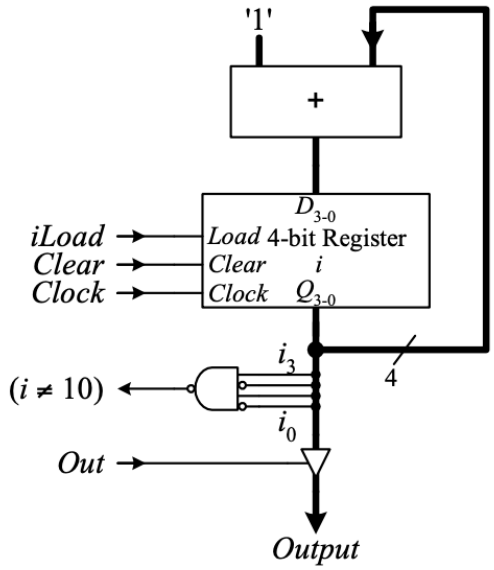

Figure 2: Dedicated data path for counting 1 to 10

Control Word	Instruction	*iLoad*	*Clear*	*Out*
1	$i = 0$	0	1	0
2	$i = i + 1$	1	0	0
3	OUTPUT i	0	0	1

Figure 3: Control words for counting 1 to 10

You are to implement a dedicated datapath for counting 1 to 10.

1. Write a VHDL file that defines a datapath using the structured depicted in Figure 2. Your code should include a 4-bit adder, 4-bit register with load and clear, a 4-bit comparator and a quad tri-state buffer. Compile the circuit. How many logic elements (LEs) are used to implement your circuit?
2. Simulate your circuit to verify its correctness.
3. Augment your VHDL file to use the pushbutton KEY_0 as the Clock input and switches SW_2, SW_1 and SW_0 as *iLoad*, *Clear* and *Out* inputs, LED_3 to LED_0 to display binary representing the 1 to 10 counts as your circuit operates, and $LEDR_9$ to display the condition ($i \neq 10$). Make the necessary pin assignments needed to implement the circuit on your DE-series board and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.

Part II

The state diagram of control unit is derived as shown in Figure 4.

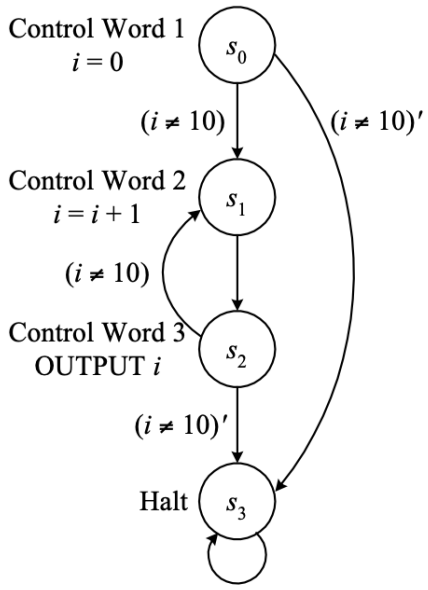

Figure 4: State diagram of control unit for counting 1 to 10

You are to implement a control unit that cooperate with the dedicated datapath.

1. Write a VHDL file that defines a finite state machine (FSM) depicted in Figure 4. Your code should include a state variable that hold the current state ($s_0 ... s_3$), a reset input to initialize the state to s_0 and the control signals (*iLoad*, *Clear* and *Out*) as outputs to the datapath. Compile the circuit. How many logic elements (LEs) are used to implement your circuit?
2. Simulate your circuit to verify its correctness.
3. Augment your VHDL file to use the pushbutton KEY_0 as the *Clock* input and switches SW_1 and SW_0 as *Reset* and $(i \neq 10)$ inputs, and three $LEDR_{2-0}$ to display *iLoad*, *Clear* and *Out* as your circuit operates. Make the necessary pin assignments needed to implement the circuit on your DE-series board and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.

Part III

The dedicated microprocessor consisting of a Control unit and a Datapath shown in Figure 5.

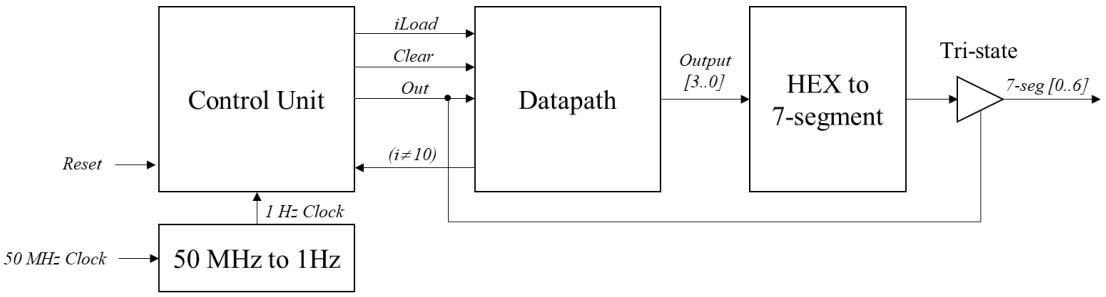

Figure 5: Dedicated microprocessor for counting 1 to 10.

You are to implement a dedicated microprocessor by assembling the Control unit and Datapath.

1. Write a VHDL file that defines component circuit (created in Part I and Part II), depicted in Figure 5. Modify the control unit to be able to operate with system internal clock (use a counter to transform 50 MHz to 1 Hz trigger or clock signal with period of 1 second). Compile the circuit. How many logic elements (LEs) are used to implement your circuit?
2. Simulate your circuit to verify its correctness.
3. Augment your VHDL file to use a pushbutton KEY_0 as *Reset*, and a 7-segment display HEX_0 to display the 1 to 10 ('A') count as your circuit operates. Make the necessary pin assignments needed to implement the circuit on your DE-series board and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.