JOINT INSTITUTE
交大密西根学院

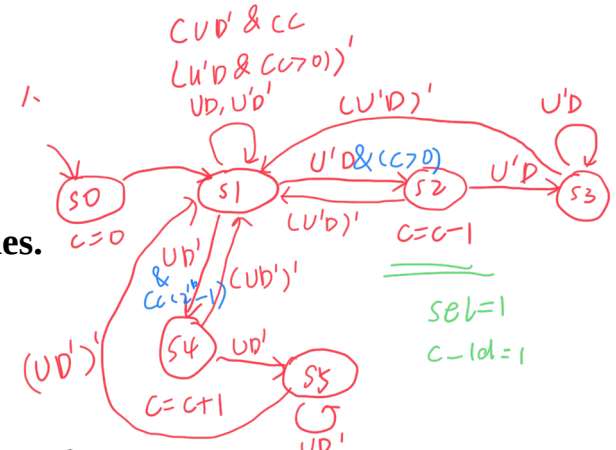
Ve270 Introduction to Logic Design

Homework 9

Assigned: November 22, 2018

Due: November 29, 2018, 4:00pm.

The homework should be submitted in hard copies.



1. Problem 5.3 (15 points)

5.3 Capture the following system behavior as an HLSM. The system has two single-bit inputs U and D each coming from a button, and a 16-bit output *C*, which is initially 0. For each press of U, the system increments *C*. For each press of D, the system decrements *C*. If both buttons are pressed, the system does not change *C*. The system does not roll over; it goes no higher than than the largest *C* and no lower than *C*=0. A press is detected as a change from 0 to 1; the duration of that 1 does not matter.

2. Problem 5.6 (15 points)

5.6 Create a high-level state machine for a simple data encryption/decryption device. If a single-bit input b is 1, the device stores the data from a 32-bit signed input *I*, referring to this as an *offset* value. If b is 0 and another single-bit input e is 1, then the device "encrypts" its input *I* by adding the stored offset value to *I*, and outputs this encrypted value over a 32-bit signed output *J*. If instead another single-bit input d is 1, the device "decrypts" the data on *I* by subtracting the offset value before outputting the decrypted value over *J*. Be sure to explicitly handle all possible combinations of the three input bits.

3. Problem 5.15. (20 points)

5.15 Use the RTL design process to design a reaction timer system that measures the time elapsed between the illumination of a light and the pressing of a button by a user. The reaction timer has three inputs, a clock input *clk*, a reset input *rst*, and a button input *B*. It has three outputs, a light enable output *len*, a 10-bit reaction time output *rtime*, and a *slow* output indicating that the user was not fast enough. The reaction timer works as follows. On reset, the reaction timer waits for 10 seconds before illuminating the light by setting *len* to 1. The reaction timer then measures the length of time in milliseconds before the user presses the button *B*, outputting the time as a 12-bit binary number on *rtime*. If the user did not press the button within 2 seconds (2000 milliseconds), the reaction timer will set the output *slow* to 1 and output 2000 on *rtime*. Assume that the clock input has a frequency of 1 kHz. Do not use a timer component in the datapath.

4. Problem 5.16 (25 points)

For problems in this section, unless otherwise indicated, allowable datapath components are from Figure 5.21 and Figure 5.27, and controller design can end after deriving the FSM. Use the RTL design process for problems that state the need to "design" a system.

5.16 Create an FSM that interfaces with the datapath in Figure 5.100. The FSM should use the datapath to compute the average value of the 16 32-bit elements of any array *A*. Array *A* is stored in a memory, with the first element at address 25, the second at address 26, and so on. Assume that putting a new value onto the address lines *M_addr* causes the memory to almost immediately output the read data on the *M_data* lines. Ignore overflow issues.

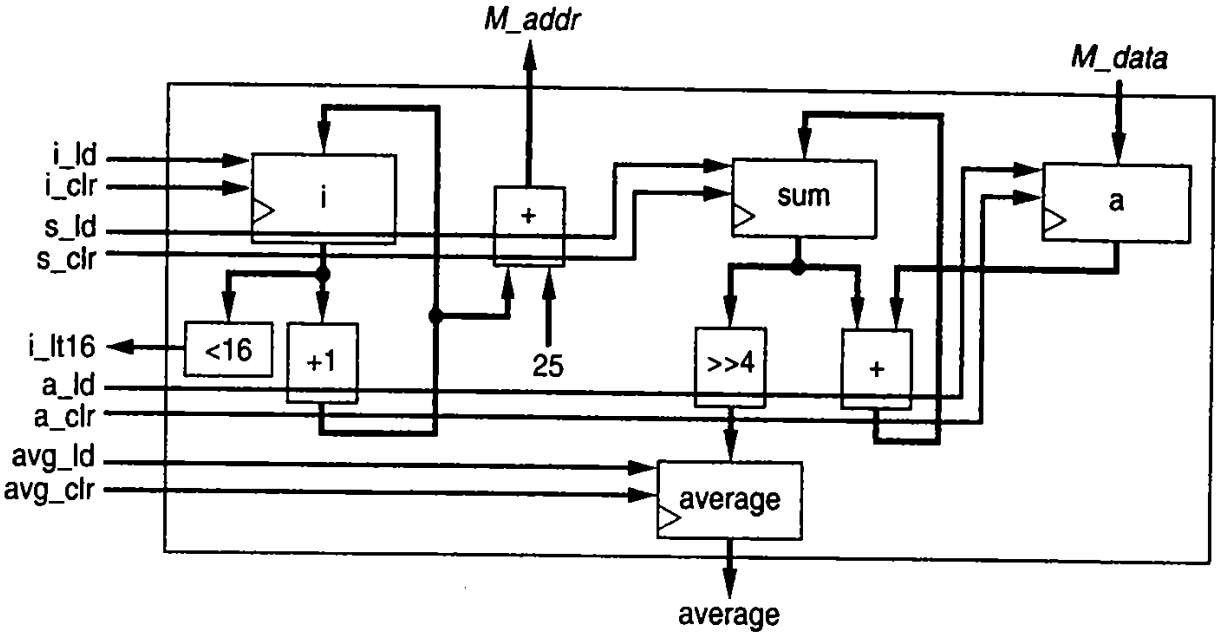


Figure 5.100 Datapath capable of computing the average of 16 elements of an array.

5. Design a circuit called ***Receiver*** that receives two single bit signals, **Valid** and **Data_in**, from another device called ***Transmitter***. The **Valid** signal sent from the ***Transmitter*** will be a 1-clock cycle pulse. After the ***Receiver*** receives the **Valid** pulse, it will start receiving 8 bits through port **Data_in**, bit by bit. After the 8 bits of data is received, they should be copied into an 8-bit register called **RxReg**. (25 points)