JOINT INSTITUTE
交大密西根学院

Ve270 Introduction to Logic Design

Homework 9

Assigned: July 13, 2017

Due: July 20, 2017, at the beginning of the class.

The homework should be submitted in hard copies.

1. Problem 5.7. (15 points)

5.7 Create a datapath for the HLSM in Figure 5.98.

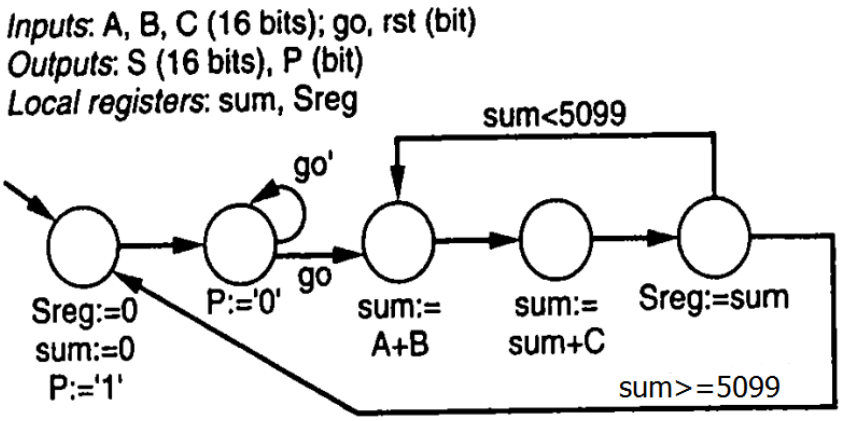

Figure 5.98 Sample high-level state machine.

2. Problem 5.14. (15 points)

5.14 Use the RTL design process to create an alarm system that sets a single-bit output `alarm` to `1` when the average temperature of four consecutive samples meets or exceeds a user-defined threshold value. A 32-bit unsigned input *CT* indicates the current temperature, and a 32-bit unsigned input *WT* indicates the warning threshhold. Samples should be taken every few clock cycles. A single-bit input `clr` when `1` disables the alarm and the sampling process. Start by capturing the desired system behavior as an HLSM, and then convert to a controller/datapath.

3. Problem 5.16 (35 points)

You can assume you only need to calculate the average once. (No need to return to the init state)

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.

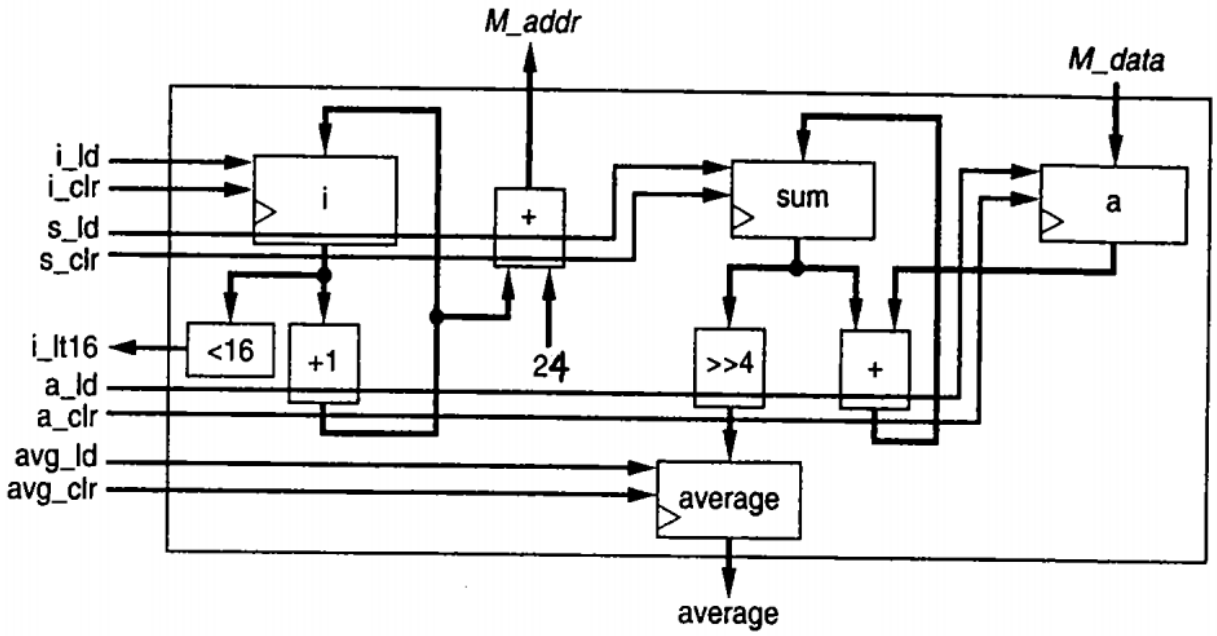

Figure 5.100 Datapath capable of computing the average of 16 elements of an array.

4. Repeat Problem 5 in HW6 with RTL design method. (35 points)
   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. The first bit of **Data_in** comes at the same time as **Valid**. Once the Receiver detects the **Valid** pulse, it will start receiving 8 bits through port **Data_in**, bit by bit. After the 8 bits of information is received, it should be copied into another 8-bit register called **RxReg.**
   (RxReg is the only output of this HLSM)
   (RxReg should save the 8-bit data until the next valid pulse)