JOINT INSTITUTE
交大密西根学院

## Ve270 Introduction to Logic Design

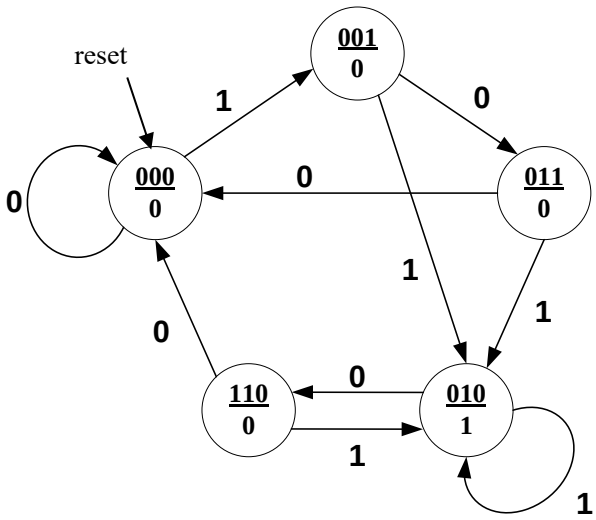
# Homework 8

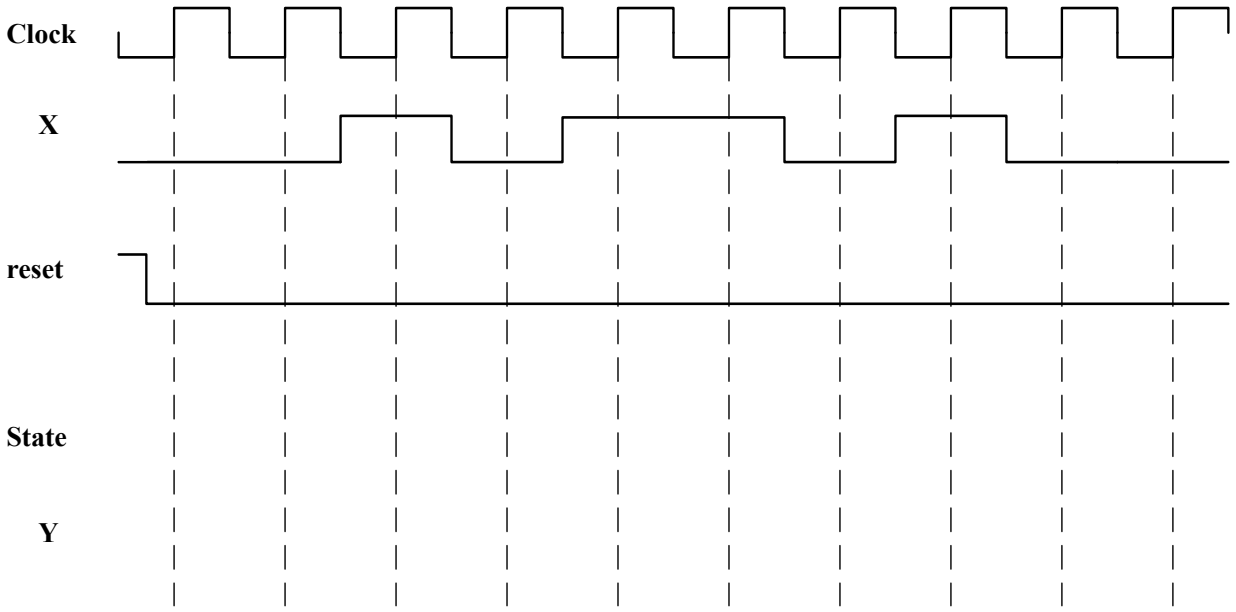
**Assigned: November 14, 2019**

**Due: November 21, 2019, 4:00pm.**

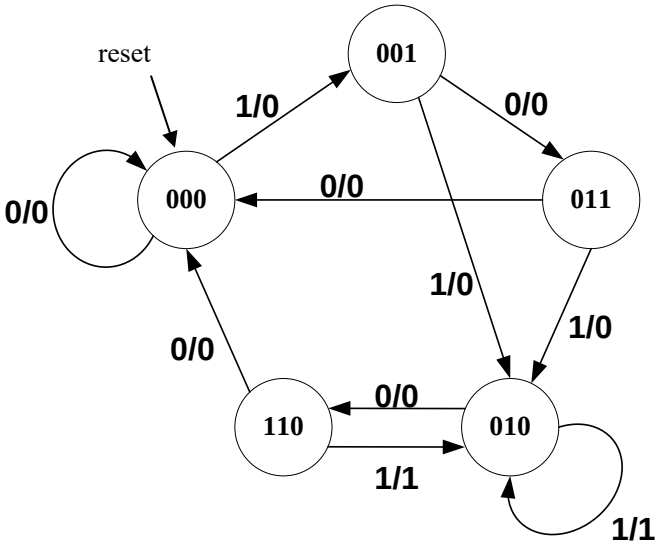
**The homework should be submitted in hard copies.**

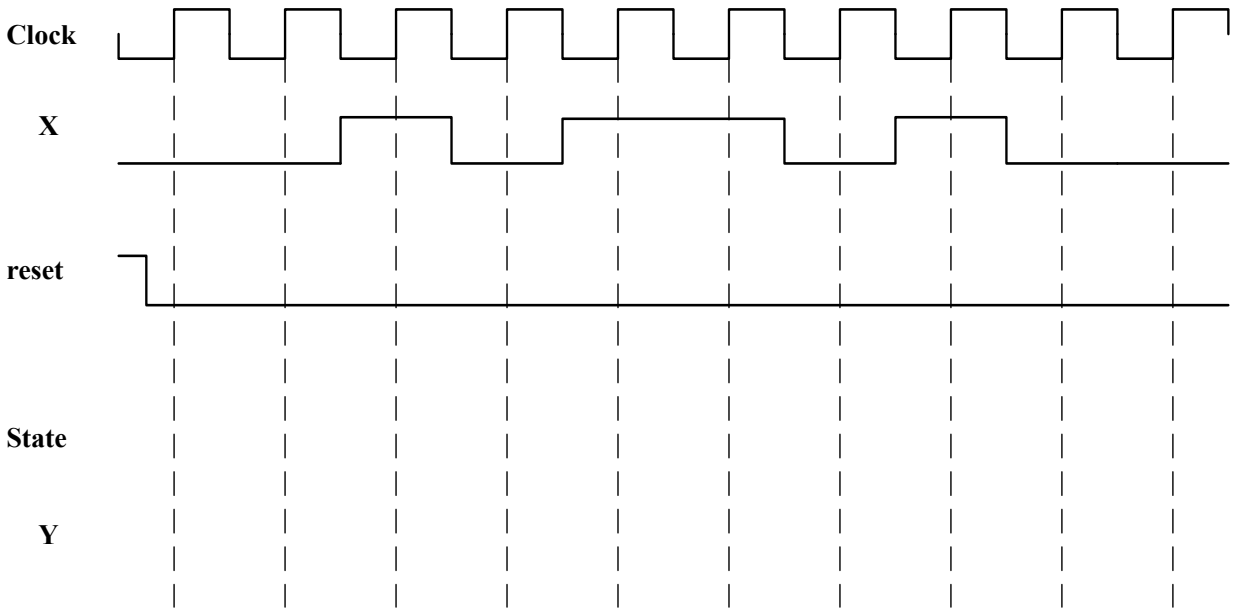
1. Given a finite state machine described as the following state diagram, complete the timing diagrams of states and output Y according to the given inputs X and reset. Ignore delays. (10 Points)



2. Repeat the same questions as in Problem 1 on the following state diagram. (10 Points)



3. Problem 5.3 (15 points)
4. Problem 5.4 (20 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.

5.4 Capture the following system behavior as an HLSM. A soda machine dispenser system has a 2-bit control input `C1 C0` indicating the value of a deposited coin. `C1C0 = 00` means no coin, `01` means nickel (5 cents), `10` means dime (10 cents), and `11` means quarter (25 cents); when a coin is deposited, the input changes to indicate the value of the coin (for possibly more than one clock cycle) and then changes back to `00`. A soda costs 80 cents. The system displays the deposited amount on a 12-bit output *D*. The system has a single-bit input `S` coming from a button. If the deposited amount is less than the cost of a soda, `S` is ignored. Otherwise, if the button is pressed, the system releases a single soda by setting a single-bit output `R` to `1` for exactly one clock cycle, and the system deducts the soda cost from the deposited amount.

5. Problem 5.14. (25 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.

6. Problem 5.16 (20 points)

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.

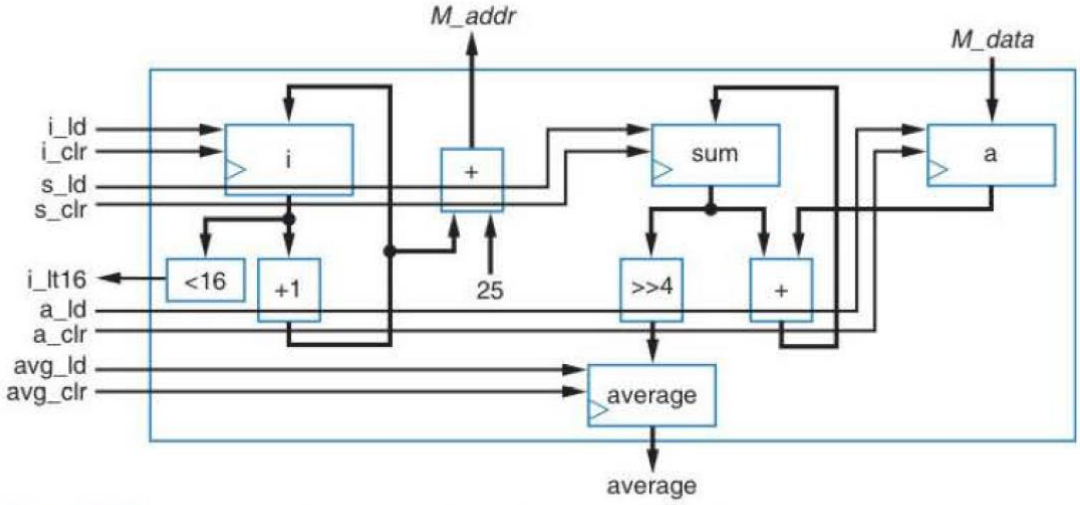



**Figure 5.100** Datapath capable of computing the average of 16 elements of an array.