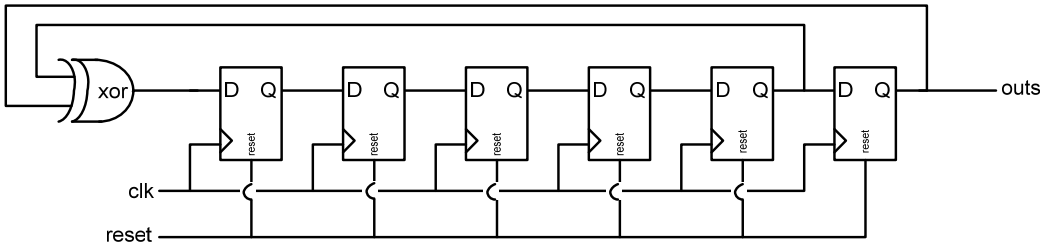
UNIVERSITY OF BRITISH COLUMBIA
ELECTRICAL AND COMPUTER ENGINEERING
CPEN 311: Digital Systems Design

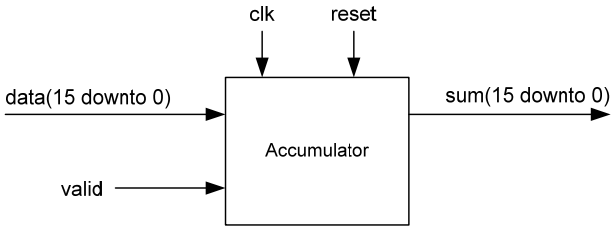
Practice Assignment 2: More Verilog/VHDL and Datapaths

This assignment will NOT be handed in, however, answers will be posted.

1. Consider the following Linear-Feedback Shift Register (LFSR). The reset line is an asynchronous reset. Write a synthesizable Verilog/VHDL description of this circuit. Your Verilog/VHDL code must contain exactly one process/always. Do not use a structural description. *Your code must be synthesizable.*



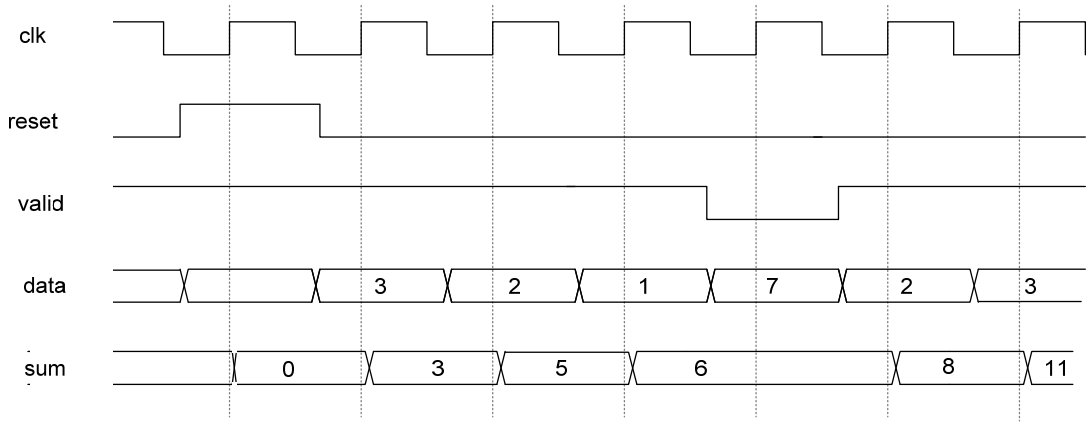
2. Consider the following circuit. The circuit receives a sequence of data elements, one per clock cycle, and maintains a running total of these data elements. More precisely, each rising clock edge, if "valid" is 1, the input "data" is sampled and added to the accumulator. If "valid" is 0, the contents of the accumulator is unchanged. The contents of the accumulator is always available on the "sum" output lines. The circuit has a synchronous reset.



The following diagram (on the next page) explains the operation of the circuit more clearly. In the following timing diagram, on the first rising clock edge, the reset input is high, so the accumulator is reset to 0. Note that when the reset input is high, the data on the data line is ignored. On the second rising clock edge, the data 3 is sampled and added to the accumulator (giving a total of 3). On the third rising clock edge, the data 2 is sampled and added to the accumulator (giving a total of 5). On the fourth rising clock edge, the data 1 is sampled and added to the accumulator (giving a total of 6). On the fifth rising clock edge, the data is not added to the accumulator, since "valid" is low.

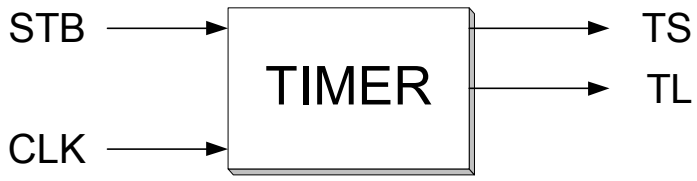
Your circuit will continue to add numbers as long as the clock keeps going. Assume overflow does not happen, and that all addition is unsigned.

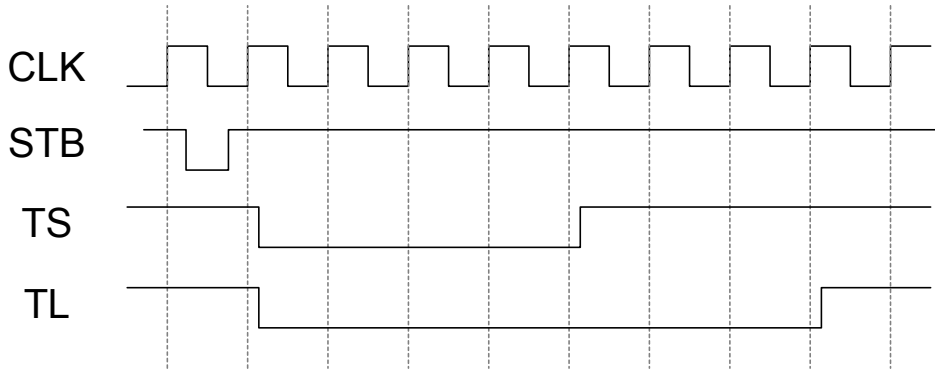
(continued on next page…)



Write <u>synthesizable</u> Verilog/VHDL code to specify the behaviour of this circuit. *Your code must be synthesizable.*

3. Write <u>synthesizable</u> Verilog/VHDL code to specify the following behaviour. When input STB goes low, outputs TL and TS go low at the next rising clock edge. When input STB goes high, TS goes high after 4 clock ticks, at TL goes high after 7 clock ticks. If STB goes low again before TL goes high, the count should reset (start counting again at 0). (If this was being marked, marks would be deducted for solutions which are excessively complex or long.)





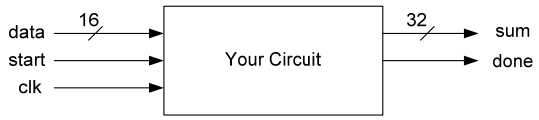
(continued on next page…)

4. A divider can be built using the algorithm below:

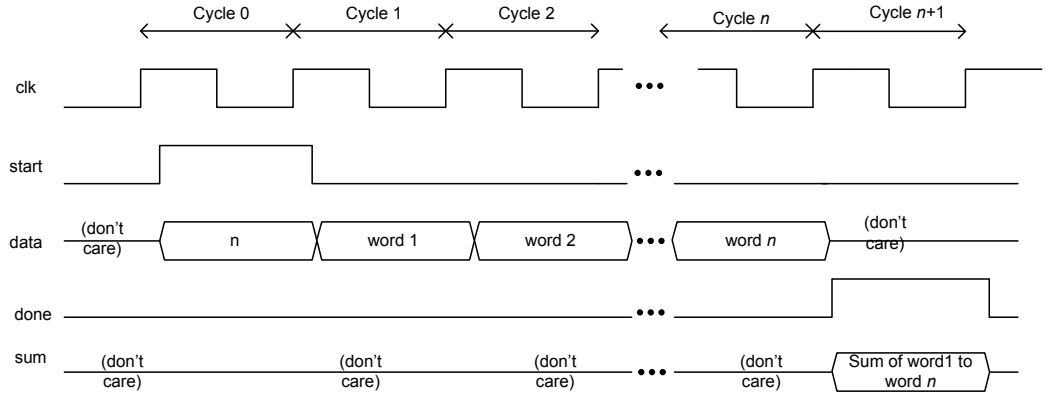
```
Q = 0;
R = A;
while (R >= B) {
        R = R – B;
        Q = Q + 1;
}
```

Design a circuit that performs division using this algorithm. There is an input *s* and an output *done*. Give your answer in terms of one or more schematics and one or more state machines.

5. In this question, you are to design an adder that adds an arbitrary number of 16-bit words. The inputs and outputs of your circuit are as follows:



During the first cycle (which we will call cycle 0), the user asserts **start**, and sends the number of words to be added on input *data*. Suppose *n* words are to be added. On each of the next *n* cycles (which we will call cycle 1 through cycle *n*), the user sends one word on input **data**. In cycle *n*+1, your circuit will assert output **done**, and send the sum of the *n* words on output **sum**. This is shown graphically below: Note that, *except* in cycle *n*+1, we don't care what value is driven on **sum**.



You can assume that *n* will not be zero, but your circuit should work for any other value of *n* (that can be specified using 16 bits).

Give your answer in terms of one or more schematics and one or more state machines,