Laboratory 6

(Due date: **002/003/008**: Nov. 30th, **004/011/013/016**: Dec. 1st, **005**: Dec. 2nd, **007**: Dec. 3rd, **012/014**: Dec. 4th)

OBJECTIVES

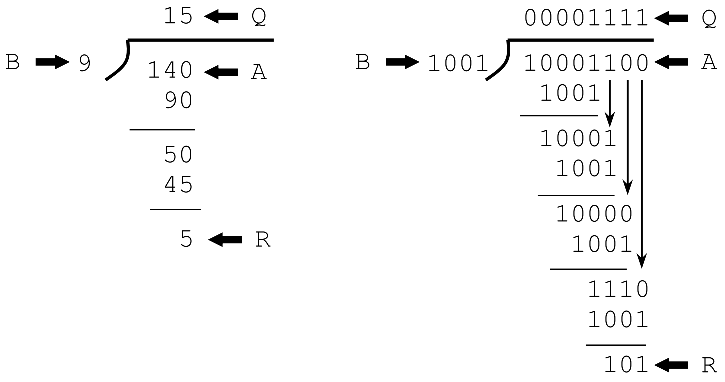
✓ Describe Finite State Machines (FSMs) in VHDL.
✓ Implement a Digital System: Control Unit and Datapath Unit.

VHDL CODING

✓ Refer to the Tutorial: VHDL for FPGAs for parametric code for: register, shift register, counter, adder.

ITERATIVE DIVIDER IMPLEMENTATION (100/100)

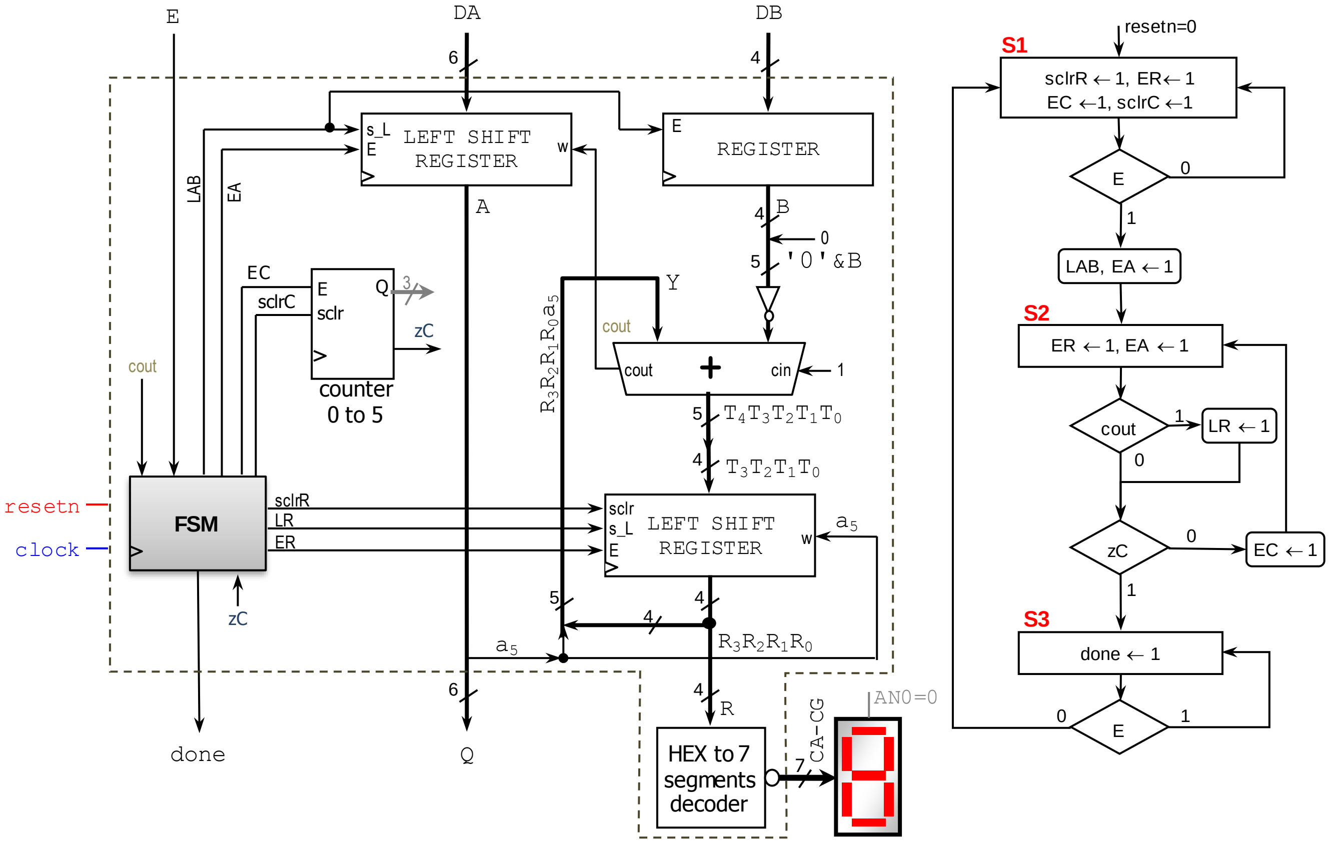
- Given two unsigned numbers A and B, we want to design a circuit that produces the quotient Q and a remainder R. $A = B \times Q + R$. The algorithm that implements the traditional long-hand division is as follows:



ALGORITHM

```
R = 0
for i = n-1 downto 0
    left shift R (input = a_i)
    if R ≥ B
        q_i = 1, R ← R-B
    else
        q_i = 0
    end
end
```

- An iterative architecture is depicted in the figure for DA with 6 bits and DB with 4 bits. The register R stores the remainder. A division operation is started when $E = 1$ (where DA and DB values are captured). Then, at every clock cycle, we either: i) shift in the next bit of A, or ii) shift in the next bit of A and subtract B. The signal $done$ is asserted to indicate that the operation has been completed and the result appears in Q and R.



- Modulo-6 counter: It includes: i) a synchronous input $sclr$ that clears the count when $E = sclr = 1$, and ii) an output zC that is asserted when the count reaches 5. The counter increases its value when $E = 1$ and $sclr = 0$. Note that Q is unused.
- Parallel Access Left-shift register: Note that one of the shift registers includes a synchronous input $sclr$ that clears the register outputs when $E = sclr = 1$. Refer to '*Notes – Unit 6*' for a description of the circuit and its operation.
- Each sequential component has $resetn$ and $clock$ inputs.

- The circuit is an example of a Digital System: It includes a Control Circuit (FSM) and a Datapath Circuit. The Datapath Circuit is made from combinational and sequential components. The circuit is also called a Special-Purpose Processor. In this case, the special purpose is the unsigned division.

PROCEDURE

- **Vivado: Complete the following steps:**

  ✓ Create a new Vivado Project. Select the corresponding Artix-7 FPGA device (e.g.: the XC7A50T-1CSG324 FPGA device for the Nexys A7-50T).

  ✓ Write the VHDL code for the given circuit. Suggestion: create a separate file for modulo-6 counter, shift Register, shift register with $sclr$ input, register, adder, hex to 7-segments decoder, FSM, and top file. Synthesize your circuit.

  ✓ Write the VHDL testbench (generate a 100 MHz input clock for your simulations) to test the following cases:
  - DA = 111011 (59), DB = 0111 (7)
  - DA = 100101 (37), DB = 1101 (13)
  - DA = 100111 (39), DB = 1001 (9)
  - DA = 010100 (20), DB = 0011 (3)
  - DA = 111110 (62), DB = 1011 (11)
  - DA = 101001 (41), DB = 0110 (6)

  ✓ Perform Behavioral Simulation and Timing Simulation of your design. **Demonstrate this to your TA**.
  - Behavioral Simulation: Add the internal signal R to the waveform view. Then, re-run the simulation.
  - For the following set of inputs, complete the expected values (when done=1) of the internal signal R as well as the outputs Q and CA-CG. Then, run the simulation and compare the values in the simulation waveform (when done=1) with the ones you computed. This will facilitate the debugging of this circuit.

DA	DB	R	Q	CA-CG
111011	0111			
100101	1101			
100111	1001			
010100	0011			
111110	1011			
101001	0110			

  ✓ I/O Assignment: Create the XDC file. Nexys A7-50T: Use SW0 to SW10 for the inputs (DA, DB, and E), CLK100MHZ for the input $clock$, CPU_RESETN push-button for $resetn$, LED15 for $done$, LED5 to LED0 for Q, CA-CG (7-segment display signals), and AN7-AN0 (anode enable for each 7-segment display; enable only one 7-segment display).
  * Note: If you are using the **Basys3 Board**, use SW11 for $resetn$, and AN3-AN0 (there are only 4 7-seg displays).

  ✓ Generate and download the bitstream on the FPGA and test. **Demonstrate this to your TA**.

- Submit (as a .zip file) all the generated files: VHDL code files, VHDL testbench, and XDC file to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

TA signature: ______________________________ **Date:** ________________________