Laboratory 6

(Due date: **002/003**: April 2nd, **004**: April 3rd, **005**: April 4th)

OBJECTIVES

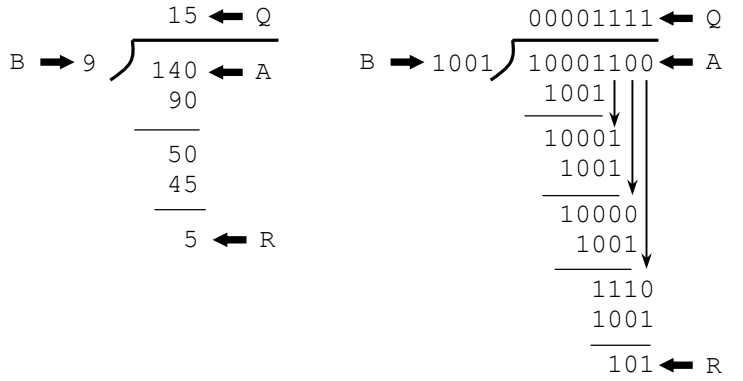
- ✓ Describe Finite State Machines (FSMs) in VHDL.
- ✓ Implement a Digital System: Control Unit and Datapath Unit.

VHDL CODING

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

ITERATIVE DIVIDER IMPLEMENTATION (100/100)

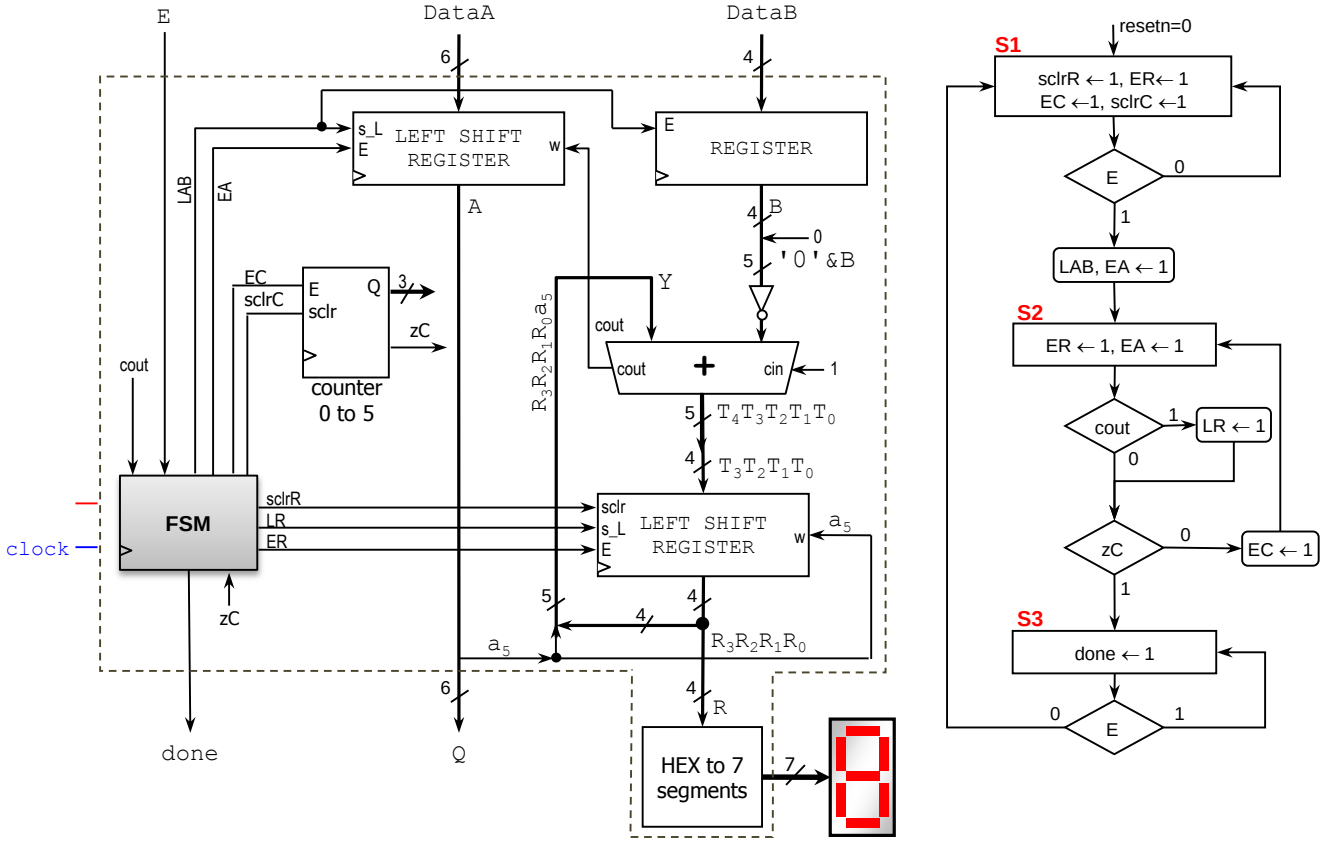
- Given two unsigned numbers A and B, we want to design a circuit that produces the quotient Q and a remainder R. $A = B \times Q + R$. The algorithm that implements the traditional long-hand division is as follows:




```
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 A with 6 bits and B with 4 bits. The register R stores the remainder. A division operation is started when $E = 1$ (where A and B 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.



Instructor: Daniel Llamocca
TAs: Jing Wu, Michael Lohrer, Saher Al-Khrayyef

- 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 $sclr = 0$.
- 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 out of combinational and sequential components. The circuit is also called a Special-Purpose Processor. In this case, the special purpose is the unsigned division.

  - ✓ Create a new Vivado Project. Select the **XC7A100T-1CSG324 Artix-7 FPGA** device.
  - ✓ 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.
  - ✓ Write the VHDL testbench (you must generate a 100 MHz input clock for your simulations) to test the following cases:
    - `DataA = 100111 (39), DataB = 0110 (6)`
    - `DataA = 110101 (53), DataB = 1101 (13)`
    - `DataA = 100101 (37), DataB = 1111 (15)`
    - `DataA = 010011 (19), DataB = 0100 (4)`
    - `DataA = 110011 (51), DataB = 0010 (2)`
    - `DataA = 011101 (29), DataB = 1001 (7)`

  - ✓ Perform Functional Simulation and Timing Simulation of your design. **Demonstrate this to your TA**.
  - ✓ I/O Assignment: Create the XDC file. Nexys-4-DDR: Use SW0 to SW10 for the inputs, CLK100MHZ for the input `clock`, CPU_RESET push-button for `resetn`, a LED for 'done', six LEDs for Q, and the 7-segment display for R.
  - ✓ 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:** _____