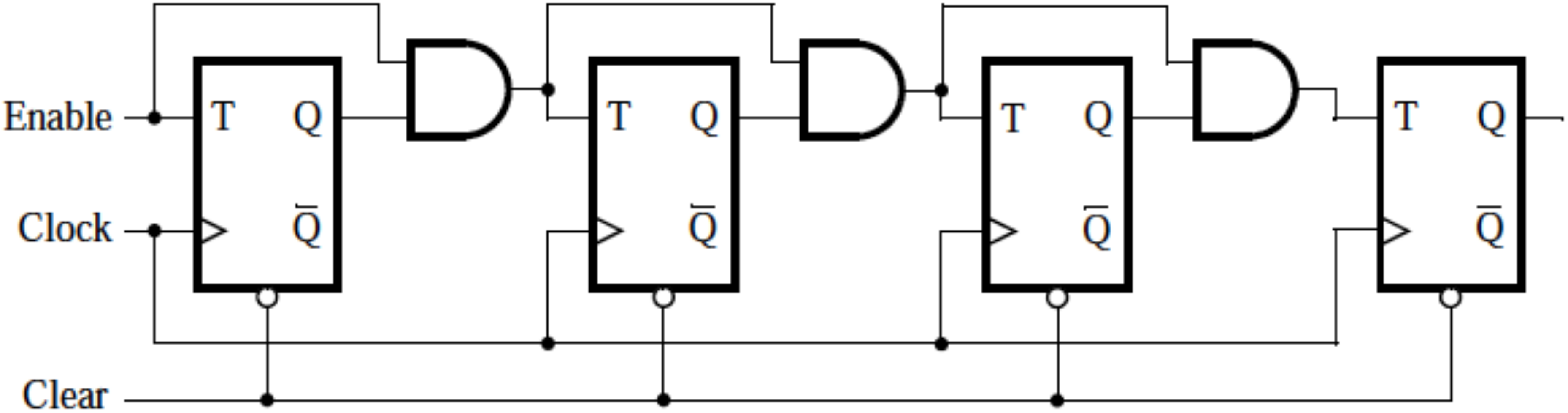
# DE0 Lab6

counters

# Part 1
# Design a 8-bit synchronous counter


Enable
Clock
Clear
T
Q
Q̄
T
Q
Q̄
T
Q
Q̄
T
Q
Q̄

# Steps

1. 1. Write a Verilog file that defines a 8-bit counter. Your code should include a T flip-flop module that is instantiated 8 times to create the counter. Compile the circuit. Howmany logic elements (LEs) are used to implement your circuit? What is the maximum frequency, **F**max , at which your circuit can be operated?
2. Simulate your circuit to verify its correctness.
3. Augment your Verilog file to use the pushbutton **KEY**0 as the Clock input, switches **SW**1 and **SW**0 as Enable and Clear inputs, and 7-segment displays HEX1-0 to display the hexadecimal count as your circuit operates. Make the necessary pin assignments needed to implement the circuit on the DE0 board, and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.

# Verilog code (1)

```
module Lab6_part1 (SW, KEY, HEX0);
        input [1:0] SW ;
        input [0:0] KEY ;
        output [0:6] HEX0;
        wire Clock = KEY[0];
        wire Resetn = SW[0];

        // 4-bit counter based on T-flip flops
        wire [3:0] Count;
        wire [3:0] Enable;
```

# Verilog code (2)

```
    assign Enable[0] = SW[1];
    ToggleFF tff0(Enable[0], Clock, Resetn, Count[0]);
    assign Enable[1] = Count[0] & Enable[0];
    ToggleFF tff1(Enable[1], Clock, Resetn, Count[1]);
    assign Enable[2] = Count[1] & Enable[1];
    ToggleFF tff2(Enable[2], Clock, Resetn, Count[2]);
    assign Enable[3] = Count[2] & Enable[2];
    ToggleFF tff3(Enable[3], Clock, Resetn, Count[3]);
    // drive the displays
    hex7seg digit0 (Count[3:0], HEX0);
endmodule
```

# Verilog code (3)

```
module ToggleFF(T, Clock, Resetn, Q);
        input T, Clock, Resetn;
        output reg Q;

        always @(posedge Clock)
                if (Resetn  == 1'b0)      // synchronous clear
                        Q <= 1'b0;
                else if(T)
                        Q <= ~Q;
endmodule
```

# Part 2

Design and implement a circuit that successively flashes digits 0 through 9 on the 7-segment display HEX0 . Each digit should be displayed for about one second. Use a counter to determine the one second intervals. The counter should be incremented by the 50-MHz clock signal provided on the DE0 board. Do not derive any other clock signals in your design–make sure that all flip-flops in your circuit are clocked directly by the 50-MHz clock signal.

# Verilog code

```
module Lab6_part2 (CLOCK_50, HEX0);
        input CLOCK_50;
        output [0:6] HEX0;
        wire [3:0] bcd;
        parameter m = 25;
        reg  [m-1:0] slow_count;
        reg[3:0] digit_flipper;
        // Create a 1Hz 4-bit counter
        // A large counter to produce a 1 second (approx) enable from the 50 MHz Clock
        always @(posedge CLOCK_50)
                slow_count <= slow_count + 1'b1;
        // four-bit counter that uses a slow enable for selecting digit
```

# Verilog code

```
    always @ (posedge CLOCK_50)
        if (slow_count == 0)
            if (digit_flipper == 4'h9)
                digit_flipper <= 4'h0;
            else
                digit_flipper <= digit_flipper + 1'b1;

    assign bcd = digit_flipper;
    // drive the display through a 7-seg decoder
    bcd7seg digit_0 (bcd, HEX0);

endmodule
```