**Purpose:**

Create a stop-watch module in Verilog that runs on the Nexys A7 development board.

**Scope:**

A stop-watch that runs from 00:00:00 to 59:59:99 formatted as MM:ss:mm (M: minute, s: second, m: milli-second) , with one button to reset and one button to stop/start count.

**General Requirements (Tag prefix: RTC)**

Asynchronous, low active reset

1. The top-level input signal ‘reset\_n’ shall be mapped to the internal signal ‘i\_resetn’.
2. All modules shall be set to their initial values when ‘i\_reset\_n’ is 0.

Positive edge clocking

1. The top-level input signal ‘sys\_clk’ shall be mapped to the internal signal ‘i\_sclk’.
2. All synchronous operations shall be done on the positive edge of ‘sys\_clk’.

Active high trigger

1. The top-level input signal ‘trigger\_in’ shall be mapped to the internal signal ‘i\_trigger’.

**10 Millisecond Timer (Tag prefix: TIMER)**

**Initial Conditions:**

* 1. Timer\_INT\_001: Internal counter shall be set to value of 1.
  2. Timer\_INT\_002: Module shall output “o\_basetick” to logic low when initializing.
  3. Timer\_INT\_003: Module shall have a constant value “MAX\_COUNT” that is set to 500000.

**Timer-Enable Conditions:**

* 1. Timer\_ENB\_001: Module shall take 100MHz “i\_sclk” clock input.
  2. Timer\_ENB\_002: Module shall be enabled when “i\_timerenb” and “i\_reset\_n” are set to logic high
  3. Timer\_ENB\_003: Module shall increment internal counter by 1 at rising edge of clock signal until counter equals to “MAX\_COUNT”.
  4. Timer\_ENB\_004: Module shall reset the value of internal counter to 1 when counter equals to “MAX\_COUNT”.
  5. Timer\_ENB\_005: Module shall toggle the output of “o\_basetick” when counter equals to “MAX\_COUNT”.

**Reset condition**

* 1. Timer\_RST\_001: Module shall reset value of the internal counter to 1 when “i\_reset\_n” is set to logic low.
  2. Timer\_RST\_002: Module shall set output of “o\_basetick” to logic low when “i\_reset\_n” is set lo logic low.

**24-Bit BCD Up Counter (Tag prefix: COUNTER)**

4-bit counter submodule with asynchronous reset and parameterizable rollover value.

1. ‘o\_bcdcount’ shall be set to “0000” when ‘i\_resetn’ is 0.
2. ‘o\_bcdcount’ shall be reset to “0000” when ‘o\_bcdcount’ is equal to rollover\_count + 1.

Internal state machine driven by trigger detection signals.

1. All state transitions shall occur on the positive edge of ‘i\_rtcclk’.
2. Module shall be initialized in the idle state.
3. Module shall transition from idle state to count state when ‘i\_countenb’ and ‘i\_latchcount’ are set to 1.
4. Module shall transition from count state to idle state when ‘i\_latchcount’ is 0.]]
5. When in the idle state, ‘o\_bcdcount’ shall keep its current value.
6. When in the count state, ‘o\_bcdcount’ shall increment on the positive edge of ‘i\_rtcclk’.

Top level module sends the individual BCD digits to a single bus output.

1. The module shall instantiate six 4-bit counter submodules corresponding to each displayed digit.
2. ‘o\_count (3:0)’ shall be set to the corresponding bits from ‘o\_bcdcount0’.
3. ‘o\_count (7:4)’ shall be set to the corresponding bits from ‘o\_bcdcount1’.
4. ‘o\_count (11:8)’ shall be set to the corresponding bits from ‘o\_bcdcount2’.
5. ‘o\_count (15:12)’ shall be set to the corresponding bits from ‘o\_bcdcount3’.
6. ‘o\_count (19:16)’ shall be set to the corresponding bits from ‘o\_bcdcount4’.
7. ‘o\_count (23:20)’ shall be set to the corresponding bits from ‘o\_bcdcount5’.

**7-Segment Display (Tag prefix: Seg\_Disp)**

1. The module shall have a 24-bit Binary-Coded-Decimal count input ‘i\_count’.
2. The module shall have five 8-bit output vectors representing each digit in the 7-segment display: ‘o\_segout1’ – ‘o\_segout6’.
3. Every four bits of the input shall represent a decimal digit in binary to be displayed on the 7-segment display.
4. The minimum decimal value every digit (4 bits) shall take is 0.
5. The maximum decimal value every digit (4 bits) shall take is 9.
6. The module shall convert every 4 bits of the 24-bit input signal to the corresponding 7-segment code to light the segments appropriately according to the FPGA datasheet.
7. The output signals shall be continually driven according to the input signal.
8. Each one of the outputs shall represent one digit to be displayed on the 7-segment display.

**7-Segment Adapter (Tag prefix: Seg\_Adap)**

1. The module shall have a system clock input ‘i\_sclk’.
2. The module shall have an active low reset input ‘i\_reset\_n’.
3. The module shall implement an asynchronous reset.
4. When the ‘i\_reset\_n’ is set to 0, ‘o\_segments’ shall be set to 00000000.
5. When the ‘i\_reset\_n’ is set to 0, ‘o\_digits’ shall be set to 00000000.
6. The module shall have six 8-bit digit code inputs: ‘i\_segout1’ – ‘i\_segout6’ that represent each digit to be displayed.
7. The module shall have an 8-bit output to control the anodes of the 7-segment display segments ‘o\_segments’.
8. The module shall have an 8-bit output to control the cathodes of the 7-segment display digits ‘o\_digits’.
9. The module shall cycle through connecting the 8-bit inputs to the 8-bit output ‘o\_segments’.
10. Each input shall be connected to the output ‘o\_segments’ for a period of 1 millisecond.
11. The module shall set the bit corresponding to the input to be displayed in ‘o\_digits’ to 0 and all the rest of the bits to 1.
12. Total period of every cycle shall be 6 milliseconds.

**Traceability:**