LTTS FPGA Training

Team: Time Travelers

**Requirements for RTC Stopwatch - Verilog Implementation**

**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. Internal counter shall be set to value of 1 on initialization.
2. Module shall output ‘o\_basetick’ to logic low when initializing.
3. Module shall have a constant value ‘MAX\_COUNT’ that is set to 500000 (value based on specification system clock speed).

Timer-enable conditions:

1. Module shall take 100MHz ‘i\_sclk’ clock input.
2. Module shall be enabled when ‘i\_timerenb’ and ‘i\_reset\_n’ are set to logic high
3. Module shall increment internal counter by 1 at rising edge of clock signal until counter is equal to ‘MAX\_COUNT’.
4. Module shall reset the value of internal counter to 1 when counter equals to ‘MAX\_COUNT’.
5. Module shall toggle the output of ‘o\_basetick’ when counter equals to ‘MAX\_COUNT’.

Reset condition:

1. Module shall reset value of the internal counter to 1 when ‘i\_reset\_n’ is set to 0.
2. Module shall set output of ‘o\_basetick’ to logic low when ‘i\_reset\_n’ is set lo 0.

**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.
3. ‘o\_rolloverflag’ shall be set to 1 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. Each consecutive digit’s instance ‘i\_countenb’ shall be mapped to the less significant instance ‘o\_rolloverflag’ output signal.
3. ‘o\_count (3:0)’ shall be set to the corresponding bits from ‘o\_bcdcount0’.
4. ‘o\_count (7:4)’ shall be set to the corresponding bits from ‘o\_bcdcount1’.
5. ‘o\_count (11:8)’ shall be set to the corresponding bits from ‘o\_bcdcount2’.
6. ‘o\_count (15:12)’ shall be set to the corresponding bits from ‘o\_bcdcount3’.
7. ‘o\_count (19:16)’ shall be set to the corresponding bits from ‘o\_bcdcount4’.
8. ‘o\_count (23:20)’ shall be set to the corresponding bits from ‘o\_bcdcount5’.

**7-Segment Display (Tag prefix: SEG\_DISP)**

1. ‘o\_segout1’ output shall be set to the BCD 7-segment code equivalent of the digit input ‘i\_count[3:0]’ according to the FPGA datasheet.
2. ‘o\_segout2’ output shall be set to the BCD 7-segment code equivalent of the digit input ‘i\_count[7:4]’ according to the FPGA datasheet.
3. ‘o\_segout3’ output shall be set to the BCD 7-segment code equivalent of the digit input ‘i\_count[11:8]’ according to the FPGA datasheet.
4. ‘o\_segout4’ output shall be set to the BCD 7-segment code equivalent of the digit input ‘i\_count[15:12]’ according to the FPGA datasheet.
5. ‘o\_segout5’ output shall be set to the BCD 7-segment code equivalent of the digit input ‘i\_count[19:16]’ according to the FPGA datasheet.
6. ‘o\_segout6’ output shall be set to the BCD 7-segment code equivalent of the digit input ‘i\_count[23:20]’ according to the FPGA datasheet.
7. The output signals shall be continually driven according to the input signal.

**7-Segment Adapter (Tag prefix: SEG\_ADAP)**

1. When the ‘i\_reset\_n’ is set to 0, ‘o\_segments’ shall always be set to 00000000.
2. When the ‘i\_reset\_n’ is set to 0, ‘o\_digits’ shall always be set to 00000000.
3. The module shall have an 8-bit output to control the anodes of the 7-segment display segments ‘o\_segments’.
4. The module shall have an 8-bit output to control the cathodes of the 7-segment display digits ‘o\_digits’.
5. The module shall cycle through connecting the 8-bit inputs to the 8-bit output ‘o\_segments’ every 6 millisecond.
6. ‘o\_segments’ shall be set to ‘i\_segout1’ during the first millisecond of every cycle.
7. ‘o\_segments’ shall be set to ‘i\_segout1’ during the second millisecond of every cycle.
8. ‘o\_segments’ shall be set to ‘i\_segout1’ during the third millisecond of every cycle.
9. ‘o\_segments’ shall be set to ‘i\_segout1’ during the fourth millisecond of every cycle.
10. ‘o\_segments’ shall be set to ‘i\_segout1’ during the fifth millisecond of every cycle.
11. ‘o\_segments’ shall be set to ‘i\_segout1’ during the sixth millisecond of every cycle.
12. 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.

**Trigger Detection (Tag prefix: TRIGGER)**

1. o\_latchcount and o\_counterenb shall always output low signals when i\_reset\_n is active low.
2. o\_countinit shall always get generate a high signal when i\_reset\_n is active low.
3. o\_countinit shall always generate a low signal when i\_reset\_n is high.
4. o\_latchcount and o\_countenb shall be toggled when a rising edge signal is sent to the i\_trigger input.
5. i\_sclk shall take input from a 100MHz system clock.