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’.

Positive edge clocking:

1. The top-level input signal ‘sys\_clk’ shall be mapped to the internal signal ‘i\_sclk’.

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 set ‘o\_basetick’ to 0 on initialization.
3. Module shall utilize a parameter ‘MAX\_COUNT’ that is set to 500000 (instance value based on specification system clock speed).

Timer-enable conditions:

1. Module shall increment the internal signal ‘counter’ by 1 on the rising edge of ‘sys\_clk’ when ‘i\_timerenb’ and ‘i\_reset\_n’ are set to 1.
2. Module shall reset the value of the internal signal ‘counter’ to 1 when counter equals to ‘MAX\_COUNT’.
3. 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 ‘o\_basetick’ to 0 when ‘i\_reset\_n’ is set to 0.

Output 10ms clock:

1. ‘o\_basetick’ shall toggle every 5ms when ‘i\_timerenb’ and ‘i\_reset\_n’ are set to 1 and the ‘i\_sclk’ is set to 100Mhz.

**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’ increments to ROLLOVER\_COUNT + 1.
3. ‘o\_rolloverflag’ shall be set to 1 for one clock cycle when ‘o\_bcdcount’ increments to ROLLOVER\_COUNT + 1.

Internal count driven by trigger detection signals:

1. ‘o\_bcdcount’ shall be initialized to “0000”.
2. ‘o\_bcdcount’ shall increment on the rising edge of ‘i\_rtcclk’ when ‘i\_countenb’ and ‘i\_latchcount’ are set to 1.

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

1. Each consecutive digit’s input ‘i\_countenb’ shall be mapped to the less significant instance ‘o\_rolloverflag’ output signal.
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’.
8. ‘o\_count’ shall increment on the rising edge of ‘i\_rtcclk’ when ‘i\_countenb’ and ‘i\_latchcount’ are set to 1.
9. ‘o\_count’ shall reset to x”000000” when it increments past x”595999”.
10. ‘o\_count’ shall reset to x”000000” when ‘i\_resetn’ is 0.

**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 cycle through connecting the 8-bit inputs to the 8-bit output ‘o\_segments’ every 6 milliseconds.
4. ‘o\_segments’ shall be set to ‘i\_segout1’ during the first millisecond of every cycle.
5. ‘o\_segments’ shall be set to ‘i\_segout2’ during the second millisecond of every cycle.
6. ‘o\_segments’ shall be set to ‘i\_segout3’ during the third millisecond of every cycle.
7. ‘o\_segments’ shall be set to ‘i\_segout4’ during the fourth millisecond of every cycle.
8. ‘o\_segments’ shall be set to ‘i\_segout5’ during the fifth millisecond of every cycle.
9. ‘o\_segments’ shall be set to ‘i\_segout6’ during the sixth millisecond of every cycle.
10. 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 be set to 0 when ‘i\_reset\_n’ is 0.
2. ‘o\_countinit’ shall always be set to 1 when ‘i\_reset\_n’ is 0.
3. ‘o\_countinit’ shall always be set to 0 when ‘i\_reset\_n’ is 1.
4. ‘o\_latchcount’ and ‘o\_countenb’ shall be toggled when a rising edge signal on ‘i\_trigger’ is held high for two consecutive cycles of ‘i\_sclk’.