EE214: Combinational Circuits-1

Wadhwani Electronics Lab, IIT Bombay

February 11, 2021

Note: In this experiment you are expected to design and implement using basic gates in Gates.vhdl, 4-bit adder and mux designed previously for the following parts:

1 Part-1

1. Design a combinational circuit block to check if the 4-bit input number is a BCD digit ($a_3a_2a_1a_0$)that varies from 0-9. The number is to be inputted through the four on-board slide switches (S4-S1) with MSB bit on S4. Show the pen-paper design to your TA.(5)
   If the input 4-bit number is:
   (a) a BCD number, display the same number on four on-board LEDs: LED4-LED1.
   (b) not a BCD number, LED4-LED1 ($l_4l_3l_2l_1$) should be turned ON.
2. Describe the logic circuit in VHDL and construct a trace-file (you may generate tracefile manually) which tries all 16 input combinations. Simulate the circuit with the generic test-bench and confirm that your circuit functions correctly. (4)
   The format for tracefile should be
   $a_3a_2a_1a_0 < space > l_4l_3l_2l_1 < space > mask - bits$
3. Confirm that the post-synthesis gate level netlist functions correctly. (3)

2 Part-2

1. Design a block which adds two 2-digit BCD numbers D1 and D2. Note that each of them is 8 bit wide packed with 2 BCD digits. The output will be valid a 2-digit BCD number and carry.
2. Note that you need to **BCD adjust** the two digit sum obtained for BCD output for all possible cases of invalid BCD sum due to binary addition. Show your design to your TA.(6)
3. The entity declaration for this block is given as follows:
```
entity BCD_ADDER is
port (D1, D2:  in std_logic_vector(7 downto 0);
Sum:  out std_logic_vector(7 downto 0); Carry:  out std_logic);
end entity BCD_ADDER;
```
4. Note that D1 and D2, both are BCD numbers so that the input will range from (00-99 decimal means 0000 0000 to 1001 1001 BCD).
5. Write the VHDL code for your design(4)
6. Simulate your 2 digit BCD adder using generic testbench.
7. Test all cases. Show the gate-level simulation to your TA.(5)
8. You will use this BCD adder in the next part.

3 Part-3

1. Now design a logic circuit which will :

   (a) Accept a two digit BCD number (AB with A as most significant digit and B as lower significant digit) using eight on-board switches (S8-S5 for A :$< a_3a_2a_1a_0 >$)and (S4-S1 for B: $< b_3b_2b_1b_0 >$) respectively.

   (b) Check if the input number is a valid two digit BCD number AB. You are advised to use the block you designed in Part 1.

   (c) If the input number is a 2-digit BCD, using the block in Part 2 add this 2-digit BCD number to 29, a fixed number. (AB+29).

   Hint: Pass ”00101001” as input in **port map** for BCD Adder instantiation.

   (d) Final output should be displayed on eight on-board LEDs (LED8-LED1) ($l_8l_7l_6l_5l_4l_3l_2l_1$) and is the BCD addition result or obtained as mentioned in table below. You may ignore the final carry bit to be displayed as there are only eight LEDs on board.

Digit A	Digit B	Expected display on LEDs
Valid BCD	Valid BCD	(29+AB)
Valid BCD	not BCD	A 1111
not BCD	Valid BCD	1111 B
not BCD	not BCD	1111 1111

2. Show your design in the form of a block/logic diagram to your TA. (5)

3. Describe the logic circuit in VHDL. (5)

4. Construct a trace-file which tries all input combinations. You may modify the given tracefile appropriately.(5)

5. Simulate the circuit with the generic test-bench and confirm that your circuit functions correctly.

6. Map your logic circuit to the Krypton board.

7. Confirm that the post-synthesis gate level netlist functions correctly. (5)

8. Program the Krypton board and demonstrate to the TA that your implementation works correctly. (5)