

Combinational Circuit 1

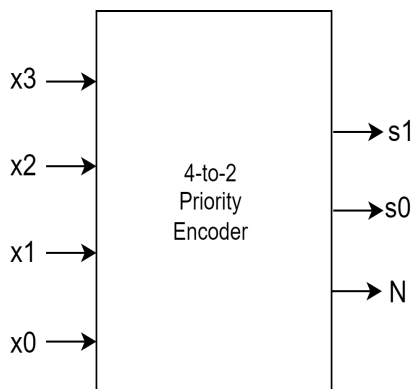
Wadhwani Electronics Lab, IIT Bombay

Monday 23rd May, 2022

[\[Download this handout as PDF\]](#)

Part-A: 4:2 Priority Encoder

Demonstrate the implementation of a priority encoder using the Krypton CPLD kit, and verify that the implementation is correct by using the on-board switches and LED's in the kit. The circuit to be implemented is an 4-to-2 encoder which has 4 inputs signals x_3, x_2, x_1, x_0 , and produces 2 bit encoded output s_1, s_0 and a signal bit N indicating whether the bits on s_1, s_0 are valid or not.



! INFO

If all the input bits to the encoder are 0, then $N=1$ and s_1, s_0 are don't-cares. If at least one of the input bits to the encoder is 1, then $N=0$, and the bits s_1, s_0 indicate the binary code for the lowest index I for which the corresponding input x_I is 1. So when multiple input bits are 1, the encoded bit s_1, s_0 represent the binary representation of lowest index I such that x_I is 1.

(i) Write the VHDL description of 4:2 Priority Encoder [10 Marks]

i NOTE

You can use either behavioural or structural modelling

(ii) Simulation [5 Marks]

Simulate the 4:2 Priority Encoder using the generic testbench to confirm the correctness of your description.

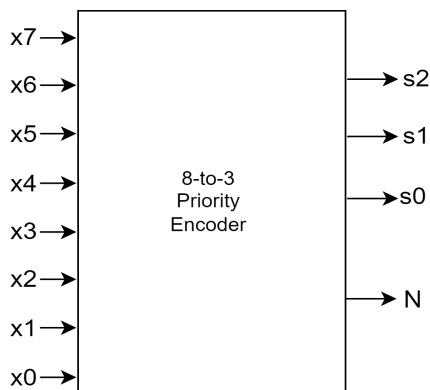
i NOTE

To do this, use the tracefile given below and modify the testbench given to you appropriately.

Tracefile format { $\langle x_3 \rangle \langle x_2 \rangle \langle x_1 \rangle \langle x_0 \rangle \langle s_1 \rangle \langle s_0 \rangle \langle N \rangle$ 111 } [Tracefile](#)

Part-B: 8:3 Priority Encoder

Repeat the above problem for 8:3 priority encoder



(i) Write the VHDL description of 8:3 Priority Encoder [10 Marks]

NOTE

You can use either behavioural or structural modelling

(ii) Simulation [5 Marks]

Simulate the 8:3 Priority Encoder using the generic testbench to confirm the correctness of your description.

NOTE

To do this, use the tracefile given below and modify the testbench given to you appropriately.

Tracefile format { <x7><x6><x5><x4><x3><x2><x1><x0> <s2><s1><s0><N> 1111 } [Tracefile](#) 