Assignment 4

Exercise 1. Implement each of the following functions using both active-high and active-low output decoders and logic gates.

a) $F = \Sigma\ (0,1,3,7,9,12,15)$

b) $G = \Pi(0,1,2,5,6,10,11)$

c) $H = (x_3+x_2)\ (\overline{x_2}+x_1+x_0)$

Exercise 2. Using only DEC 2:4, desing the following decoders:

a) DEC 1:2 b) DEC 3:8 c) DEC 4:16

Exercise 3. Using only MUX 4:1, desing the following multiplexers:

a) MUX 2:1 b) MUX 8:1 c)MUX 16:1

Exercise 4. Implement the following functions using multiplexers and inverters (if necessary).

$f(a, b, c, d) = \sum(0, 2, 4, 5, 6, 8, 9, 10, 12, 15)$

$g(a, b, c, d, e) = \sum(0, 1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15, 17, 20, 22, 23, 25, 28, 29, 30, 31)$

a) f using MUX 8:1 b) f using MUX 4:1

c) g using MUX 4:1 d) g using MUX 8:1

Exercise 5. Solve problem 19c of the course's problem collection 3: analysis of circuits with combinational subsystems. To solve the problem obtain the truth table of function f.

Exercise 6. Design a circuit that takes two 4-bit numbers A and B and outputs two 4-bit numbers X and Y so that X is the greatest of A and B and Y is the lowest of A and B. Give two solutions:

a) Draw a circuit using combinational subsystems and logic.

b) Write a Verilog description and an appropriate test bench to check its operation.

Exercise 7. Design a circuit that takes three 4-bit numbers A, B and C; and outputs two 4-bit numbers X and Y so that X is the greatest of A, B and C; and Y is the lowest of A, B and C. Give two solutions:

a) Draw a circuit using the circuit designed in exercise 4 as a basic block.

b) Write a Verilog description and an appropriate test bench to check its operation.

Exercise 8. Starting with the initial design of the BCD-7s converter of example 4.2 in the Verilog course[1]:

a) Complete the description and simulate the converter. What happens when the input is not a BCD code.

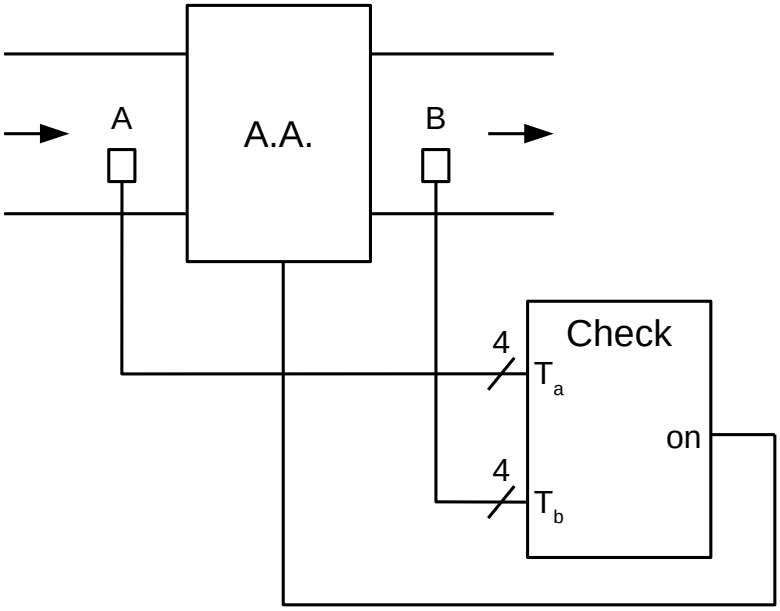
b) Extend the design (and the testbench if necessary) in order to implement an hexadecimal 7-segment code converter, that is, the circuit takes a binary number from 0 to 15 and represents the number as an hexadecimal digit using 7 segments. You will have to figure out how to build digits A to F using the 7 segments.

Exercise 9. An air conditioning (AC) system works by cooling the air that enter through an input pipe and goes out through an output pipe. In in the input pipe there is temperature sensor A that measures the temperature of incoming air, and another sensor B measures the temperature of outgoing air through the output pipe. Both sensors generate a digital value for the temperature in the range 0 to 15ºC using 4 bits.

We want to design a circuit “Check” that ensured that the temperature in B is always less or equal than the temperature in A. The circuit does so by comparing T_a and T_b and activating de AC system when necessary through the control signal 'on' that is active-high. Propose a solution

1 https://gitlab.com/jjchico/verilog-course.v

for circuit “Check” using standard combinational subsystems.



Exercise 10. Describe the circuit in unit's example 11 using Verilog. Use the the BCD-7s converter designed in exercise 8 and use the paraty detection code shown in the unit. Write a testbench and check that the circuit operates correctly.

Exercise 11. A wind turbine generates energy from wind force. Its control system measures the speed of the wind and its direction. Speed come in a 4-bit input S and direction comes in a 4-bit input D, both encoded in Gray code. Speeds from 0 to 3 are considered low, from 4 to 12 are considered moderate and from 13 to 15 are considered high. Direction 0 is north, 4 is east, 8 is south, 12 is west, and so on. The system's outputs are 'off' (disconnect power generation) and 'brake' (stop the generator). Both are active high.

a) Design the control system so that power generation is stopped when wind speed is low, to prevent power consumption by the turbine; and the turbine is halted (and power generation stopped) when wind speed is high, to avoid damage.

b) Improve the design so that the turbine is also halted (and power generation stopped) whenever the wind comes from any direction between (and including) south-est and south-west, since wind from those directions are irregular and not appropriate for a correct homogeneous generation.

c) Write a Verilog description and an appropriate test bench to test the operation of the circuit.

Exercise rationale

- Exercises 1 to 4 are useful to understand how decoders and multiplexers can be used to implement arbitrary logic functions. Also to understand that, due to their modularity, any decoder or multiplexer can be implemented just with decoders of multiplexers of a different size. Although the practical application of these exercises is limited, they help the student to get familiar with the functionality of decoders and multiplexers, which are basic blocks in many other digital applications.
- Exercise 5 is just to show that analyzing a circuit with combinational subsystems is not more difficult than analyzing a circuit with gates, once you understand the logic function implemented by these subsystems.
- Exercises 6 to 11 are some “realistic” applications of combinational blocks studied in the unit. The intention is to show that relatively complex combinational problems that are not

feasible to solve using K-maps and logic gates, can be easily solved (with some practice) if we map the functionality of the circuit to that of standard combinational subsystems.

- Exercises 7, 8, 10 and 11 also ask for a solution in Verilog. HDL descriptions are done at a similar level of abstraction than modular design with combinational subsystems so it is convenient to study them in parallel. In the real world, most digital design is actually done using HDLs, but it is good for the designer to know what kind of blocks will be derived from the HDL code.