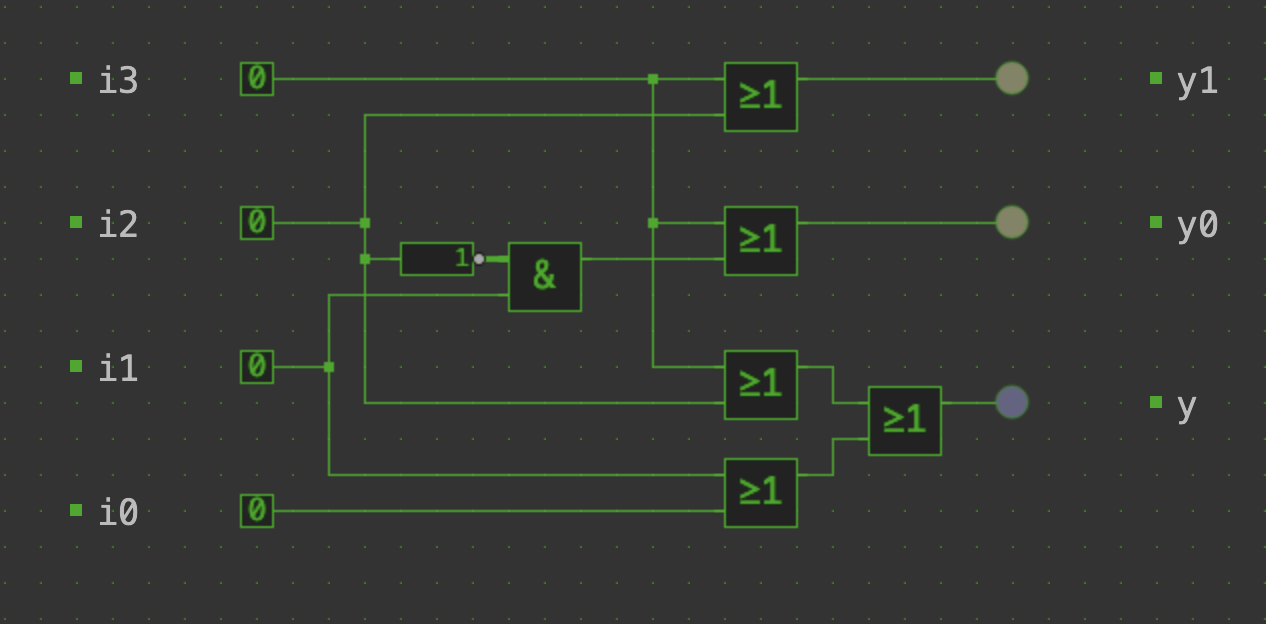
ICS 2022 Problem Sheet #8

**Problem 8.1:** digital circuit analysis

You are given the following digital circuit.



1. Write down the truth table defining the outputs y0, y1, and y.

y0 = i3 ( i2 i1 )

y1 = i3 i2

y = ( i0 i1 ) ( i2 i3 )

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I3 | I2 | I1 | i2 | ( i2 i1 ) | y0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| I3 | I2 | y1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| I3 | I2 | I1 | I0 | ( i0 i1 ) | ( i2 i3 ) | y |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

1. Write down short boolean expressions defining y0, y1, and y.

y0 = i3 ( i2 i1 )

y1 = i3 i2

y = ( i0 i1 ) ( i2 i3 )

1. Describe in your own words what the circuit is doing and how it might be used.

The circuit is a priority encoder, that compresses multiple [binary](https://en.wikipedia.org/wiki/Binary_code) inputs into a smaller number of outputs. The output of a priority encoder is the binary representation of the index of the most significant activated line, starting from zero. They are often used to control [interrupt requests](https://en.wikipedia.org/wiki/Interrupt_request) by acting on the highest priority interrupt input.

Priority encoders can be easily connected in arrays to make larger encoders, such as one 16-to-4 encoder made from six 4-to-2 priority encoders - four 4-to-2 encoders having the signal source connected to their inputs, and the two remaining encoders take the output of the first four as input. The priority encoder is an improvement on a simple encoder circuit, in terms of handling all possible input [configurations](https://en.wikipedia.org/wiki/Computer_configuration).

**Problem 8.2:** dice display

Too many students waiting inside the coffee bar to obtain drinks and snacks was found to be problematic and as a consequence the number of people waiting to be served got limited to seven. You got the task to create a display showing how many students are inside and you decided to build a display out of light emitting diodes (LEDs) that can be powered by a very tiny solar panel. Your display resembles the form of a dice with LEDs positioned as follows:

Your display is driven by three input lines x2, x1, x0 indicating a binary number.

1. Write a truth table defining the necessary boolean functions.

I started by writing the 8 numbers in binary, using three bits, that will represent the three input lines x2, x1, x0.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | x2 | x1 | x0 | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

1. Provide (simple) boolean expressions for the boolean functions.

a = g = x2

b = f = ( x2  x1 x0 ) ( x2  x1 x0 ) =

= ( x2  x1 )  x0  x0 ) =

= ( x2  x1 ) 1 =

= x1  x2

c = e = ( x2  x1 x0 ) ( x2  x1 x0 ) ( x2  x1 x0 ) ( x2  x1 x0 ) ( x1  x2 x0 ) ( x1  x2 x0 ) =

= ( x2  x1 )  x0  x0 ) ( x2  x1 )  x0  x0 ) ( x1  x2 )  x0  x0 ) =

= ( x2  x1 ) 1 ( x2  x1 ) 1 ( x2  x1 ) 1 =

= ( x2  x1 ) ( x2  x1 ) ( x2  x1 )

= ( x2  x1 ) x2 (x1 x1) =

= ( x2  x1 ) x2

= x2 x2) (x1 x2) =

= 1 (x1 x2) =

= x1 x2

d = x0

1. Create a digital circuit using <https://simulator.io/>.

<https://simulator.io/board/KFze8pskJF/1>

A screenshot of a computer

Description automatically generated with medium confidence

**Problem 8.3:** decimal to binary and binary to decimal (haskell)

Implement a functions to convert decimal numbers into binary notation and back.

1. Implement a function dtob :: Int -> String converting a non-negative integer number into a String (consisting of the characters ’0’ and ’1’) representing the integer number as a binary number. It is not necessary to handle negative integers in a meaningful way.

dtob :: Int -> String

dtob 1 = "1"

dtob 0 = "0"

dtob n

| even n = dtob(n `div` 2) ++ "0"

| otherwise = dtob(n `div` 2) ++ "1"

1. Implement a function dtob :: String -> Int converting a String (consisting of the characters ’0’ and ’1’) representing a binary number into the corresponding non-negative integer number. It is not necessary to handle unexpected strings in a meaningful way.

btod :: String -> Int

btod "0" = 0

btod "1" = 1

btod xs

| head xs == '1' = 2 ^ (length xs - 1 ) + btod(tail xs)

| otherwise = btod(tail xs)