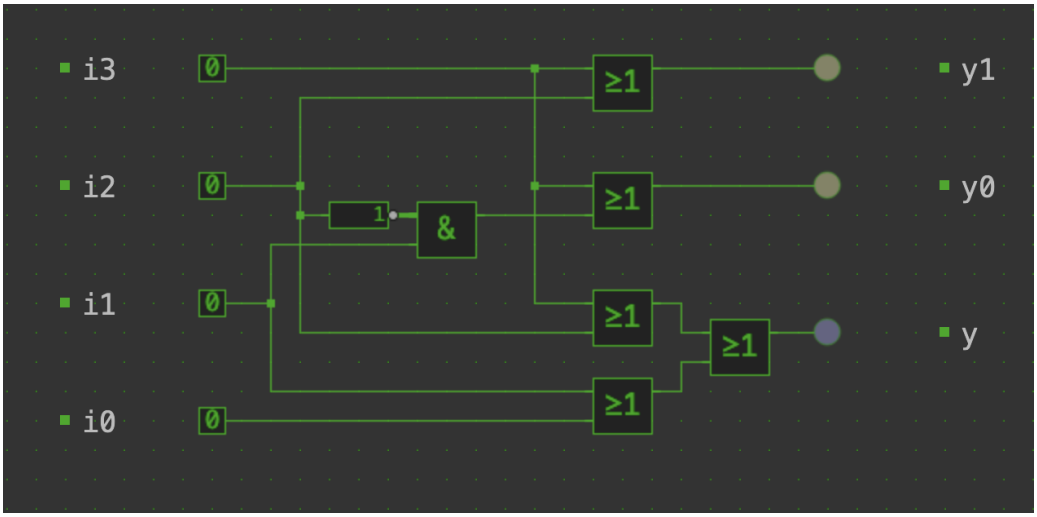
Introduction to Computer Science
Jacobs University Bremen
Dr. Jürgen Schönwälder

Module: CH-232
Date: 2022-10-28
Due: 2022-11-04

ICS 2022 Problem Sheet #8

Problem 8.1: *digital circuit analysis* (1+1+2 = 4 points)

You are given the following digital circuit. The circuit may as well be found online at `http://simulator.io/board/pu8qlKwg1J/3` (but there is no guarantee that it persists).



a) Write down the truth table defining the outputs y_0, y_1, and y.

b) Write down short boolean expressions defining y_0, y_1, and y.

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

Problem 8.2: *dice display* (2+1+1 = 4 points)

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:

```
a  b  c
   d
e  f  g
```

The numbers 0 to 7 are displayed as follows (a star indicates a LED producing light, a circle indices an LED currently off).

```
o  o  o      o  o  o      o  o  *      o  o  *
   o            *            o            *
o  o  o      o  o  o      *  o  o      *  o  o

*  o  *      *  o  *      *  *  *      *  *  *
   o            *            o            *
*  o  *      *  o  *      *  *  *      *  *  *
```

Your display is driven by three input lines x_2, x_1, x_0 indicating a binary number.

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

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

c) Create a digital circuit using https://simulator.io/.

Submit an image of your digital circuit and a link resolving to your digital circuit.

Problem 8.3: *decimal to binary and binary to decimal (haskell)* (1+1 = 2 points)

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

a) 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.

b) 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.

Submit your Haskell code as a plain text file. Below is a template file with a few unit test cases.

```haskell
module Main (main) where

import Test.HUnit

-- | Convert a non-negative integer number into a String providing a
--   binary representation of the number.
dtob :: Int -> String
dtob _ = undefined

-- | Convert a String representing a non-negative integer number as a
--   binary number into an integer number.
btod :: String -> Int
btod _ = undefined

-- Below are some test cases...

dtobTests = TestList [ dtob 0 ~?= "0"
                     , dtob 1 ~?= "1"
                     , dtob 2 ~?= "10"
                     , dtob 127 ~?= "1111111"
                     , dtob 12345 ~?= "11000000111001"
                     ]

btodTests = TestList [ btod "0" ~?= 0
                     , btod "1" ~?= 1
                     , btod "10" ~?= 2
                     , btod "1111111" ~?= 127
                     , btod "11000000111001" ~?= 12345
                     ]

main = runTestTT $ TestList [ dtobTests, btodTests ]
```