

CSE 320 Spring 2012

Computer Project #1 -- Combinational Circuits

Assignment Overview:

This assignment develops familiarity with combinational circuits, as well as the "sim" software simulation package. You will design the combinational circuits described below, then implement and test the circuits using "sim".

It is worth 30 points (3% of course grade), and must be completed no later than 11:59 PM on Thursday, January 26.

Assignment Specifications:

Your circuits will accept four input signals (the four bits in a hexadecimal digit) and will produce four output signals: "Valid", "Present", "Even", and "Prime".

The "Valid" signal will be asserted (true) when the four input signals represent a hexadecimal digit which is valid in an MSU PID; it will be deasserted (false) for all other cases. That is, the function will recognize the hexadecimal digits in the set {0-9,A}.

The "Present" signal will be asserted when the four input signals represent a hexadecimal digit in your personal MSU PID; it will be deasserted for all other cases. That is, the function will recognize the hexadecimal digits which are present in your MSU PID.

For the hexadecimal digits which are present in your personal MSU PID, the "Even" signal will be asserted when the four input signals represent a hexadecimal digit which is an even number; it will be deasserted if the digit is not even. For all hexadecimal digits which are not present in your MSU PID, the value of the signal is irrelevant.

For the hexadecimal digits which are present in your personal MSU PID, the "Prime" signal will be asserted when the four input signals represent a hexadecimal digit which is a prime number; it will be deasserted if the digit is not prime. For all hexadecimal digits which are not present in your MSU PID, the value of the signal is irrelevant.

Your design will be formalized by completing the requested information in the file "~cse320/Projects/project01.design".

Your implementation will be formalized by creating a C++ source code file which represents the minimized version of your circuits and serves as input to the "sim" package.

You will execute the simulator using "sim proj01.netlist.c".

Assignment Deliverables:

The deliverables for this assignment are:

```
proj01.design    -- the text file for your design
proj01.netlist.c -- the implementation file for your solution
```

Be sure to use the specified file names, and to submit your files for grading via the "handin" program.

Assignment Notes:

An even number is defined to be an integer of the form $n = 2k$, where k is an integer.

A prime number is defined to be a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself.

To implement your circuits, you will create two functions: "simnet" and "circuits". A skeleton for the C++ source code for those two functions is available on the system:

```
/user/cse320/Projects/project01.netlist.c
```

Function "simnet" will serve as a test fixture which allows the user to enter hexadecimal digits and observe the resulting function outputs. The input signals will be entered using four switches, and the output signals will be displayed using four probes.

The four switches will be placed vertically on the left side of the "sim" window. The most significant bit of the input will be on the top, and the least significant bit will be on the bottom.

The four probes will be placed vertically on the right side of the "sim" window, and they will appear in the order "Valid" (on the top), "Present", "Even", and "Prime" (on the bottom).

Function "circuits" will serve as the implementation of your four Boolean functions ("Valid", "Present", "Even" and "Prime"). You will use "Not", "And" and "Or" components to construct your four Boolean functions.

Since function "circuits" uses a "Module" component, it will appear as a "black box" in the initial circuit diagram. To move down one level in the module hierarchy, use the "Page Down" key and then redraw the circuit diagram using the "F12" key. To move back up in the module hierarchy, use the "Page Up" key (and then redraw the circuit diagram with the "F12" key).

If the circuit diagram becomes difficult to read due to the lines connecting the various gates and components, you can remove them by using the "F2" key to toggle the drawing of lines and the "F12" key to redraw the screen.