

Computer Project #4

Assignment Overview

This assignment develops familiarity with sequential circuits, as well as the "sim" software package. You will design a circuit which serves as a specialized sequence counter, and then implement that circuit using "sim".

It is worth 40 points (4% of course grade) and must be completed no later than 11:59 PM on Thursday, 9/27.

Assignment Deliverables

The deliverables for this assignment are the following files:

`proj04.design` – the text file containing your design
`proj04.netlist.c` – the source code file containing your solution

Be sure to use the specified file names and to submit them for grading via the CSE handin system before the project deadline.

Assignment Specifications

Your circuit will repeatedly generate the unique hexadecimal digits in your MSU PID, in the order of their first occurrence. For example, assume that your PID is "A20019802". After removing all but the first occurrence of each digit, the remaining unique hexadecimal digits of this PID are "A20198". Thus, the circuit would generate the sequence "A20198A20198A..." (indefinitely).

Four D flip-flops will be used to store the current hexadecimal digit, which is the input to the combinational logic. The combinational logic will map the current hexadecimal digit to the next digit in the sequence. Since duplicate digits are not allowed, this mapping is unambiguous.

The circuit will use an initialization signal and a clock (strobe) signal to control the operation of the sequence counter. When the initialization signal is asserted, the four D flip-flops will be initialized to "1010" (the digit "A"). When the clock signal is asserted, the counter will advance to the next hexadecimal digit in the sequence.

Your design will be formalized by copying the file `/user/cse320/Projects/project04.design` into your account as the file `"proj04.design"`, then editing that file to include the requested information.

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

You will execute the simulator using `"sim proj04.netlist.c"`.

Assignment Notes

1. Whenever possible, your circuit design will take advantage of "don't care" conditions during the minimization process.
2. 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/project04.netlist.c
```

3. Function "simnet" will serve as a test fixture which allows the user to control the circuit and observe the sequence of hexadecimal digits generated by the circuit. The initialization and clock signals will be simulated using pulsers, and the results will be displayed using four probes.

The pulsers used to control the circuit will be positioned on the left side of the screen. Use the 'i' key for the initialization signal and the 'c' key for the clock signal.

The four probes will be used to display the current count. The probes will be positioned vertically on the right side of the screen, with the most significant bit at the top.

All pulsers and probes will be appropriately labeled in the "sim" window.

4. Function "circuits" will serve as the implementation of your specialized sequence counter. You will use "Not", "And" and "Or" components to construct your four Boolean functions, and you will use "Dff" components to store the current count.

5. The D flip-flop is a "sim" component with four inputs and a single output:

Dff("row-col", (set, D, clock, reset), Q);

The output "Q" takes on the value of the input "D" whenever the "clock" signal transitions from 0 to 1 (the rising edge of the pulse). The "reset" and "set" inputs asynchronously force the flip-flop into the "0" or "1" state (respectively). Whichever of the "set" and "reset" inputs is unused should be set to the constant "Zero".