

## Lab Exercise #4 -- Sequential Circuits

In a combinational circuit, the outputs of the circuit are determined by the current inputs; in a sequential circuit, the current inputs and the current state of the circuit determine the next state (and the outputs).

The fundamental building block for sequential circuits is the D flip-flop. Commonly used components include parallel load registers, serial load registers, counters, and sequencers.

### A. D Flip-Flops

1. The file "`~cse320/Labs/lab04.dff.c`" contains a C++ module which is a test bed for experimenting with the D flip-flop ("Dff") component. Bring the function into simulation using the UNIX command:

```
<prompt> sim ~cse320/Labs/lab04.dff.c
```

Experiment with the circuit and answer the following questions.

a) Give the initial value of all inputs to the D flip-flop.

Set: \_\_\_\_\_ Data: \_\_\_\_\_ Clock: \_\_\_\_\_ Reset: \_\_\_\_\_

b) What is the initial value of the output from the Dff? Explain.

c) Toggle (assert, then de-assert) the Data signal three times. What happens to the output from the Dff? Explain.

d) Assert the Data signal, then toggle the Clock signal. What happens to the output from the Dff? Explain.

e) With the Data signal asserted, toggle the Clock signal three times. What happens to the output of the Dff? Explain.

f) Assert the Clock signal, then toggle the Data signal three times. What happens to the output from the Dff? Explain.

g) Assert the Set signal, de-assert the Data signal, then toggle the Clock signal several times. What happens to the output from the Dff? Explain.

h) De-assert the Set signal, assert the Reset signal and the Data signal, then toggle the Clock signal several times. What happens to the output from the Dff? Explain.

2. Which of the inputs to a D flip-flop are asynchronous signals? Explain.

## B. Parallel Load Registers

A parallel load register is composed of flip-flops; it captures the data inputs when the enable signal is asserted and the next clock pulse occurs.

The file "`~cse320/Labs/lab04.register.c`" contains a C++ module which is a test bed for experimenting with the "Register" component.

1. Bring the function into simulation, then experiment with the circuit and answer the following questions.

a) Press the 'F1' key to guarantee that the simulation starts fresh. What do the probes show as the initial contents of the register? Explain.

b) Assert one of the data input signals. What happens to the value displayed by the probes on the output of the register? Explain.

c) Set the data inputs to 1010, then cycle the clock pulse by touching the 'c' key. What happens to the value displayed by the probes on the output of the register? Explain.

d) Leave the data inputs as 1010, assert the enable signal, and then cycle the clock pulse by touching the 'c' key. What happens to the value displayed by the probes on the output of the register? Explain.

e) De-assert the enable signal, set the data inputs to 1111, and then cycle the clock pulse by touching the 'c' key. What happens to the value displayed by the probes on the output of the register? Explain.

f) Assert the enable signal, then cycle the clock pulse. What happens to the register output? Explain.

2. Summarize the purpose of each of the input signals: enable signal, clock signal, and data signals.

## C. Counters

A counter uses flip-flops to retain the current state (value in the count sequence); it moves to the next state (next value in the count sequence) when the counter is strobed. It always outputs the current state.

The file "`~cse320/Labs/lab04.counter.c`" contains a C++ module which is a test bed for experimenting with the "Counter" component.

1. Bring the function into simulation, then experiment with the circuit and answer the following questions.

a) Press the 'F1' key to guarantee that the simulation starts fresh. What do the probes show as the initial outputs of the counter? Explain.

b) Touch the 'r' key to reset the counter. What values do the probes display? Explain.

c) Touch the 's' key three times to make the counter move through the sequence {0, 1, 2, 3}. What values do the probes display? Explain.

d) Touch the 'r' key to reset the counter. What values do the probes display? Explain.

e) What is the largest value in the count sequence? What happens when you reach the largest value, then touch the 's' key again? Explain.

f) The reset signal for the Counter component is "active low". Explain.

2. The Counter used in the test bed is a 4-bit binary counter: it outputs the sequence <0000, 0001, 0010, 0011, 0100, ..., 1111, 0000, ...>.

Recall that the Gray code sequence is <000, 001, 011, 010, 110, 111, 101, 100, 000, ...> for 3 bits.

Describe (in general terms) what would have to be done to convert a 3-bit binary counter into a 3-bit Gray code counter.

#### D. Sequencers

It is often necessary to ensure that certain steps occur in a particular order (sequence). If there are  $N$  steps in the sequence,  $N$  flip-flops (synchronized by a common clock signal) can be used to trigger the steps in order. Only one of the flip-flops stores a '1' at any given moment in time, and the output of that flip-flop triggers the action for that step.

The file "`~cse320/Labs/lab04.part_D.c`" contains an incomplete implementation of a four-bit sequencer in a test bed. Modify that circuit as specified in the comments.