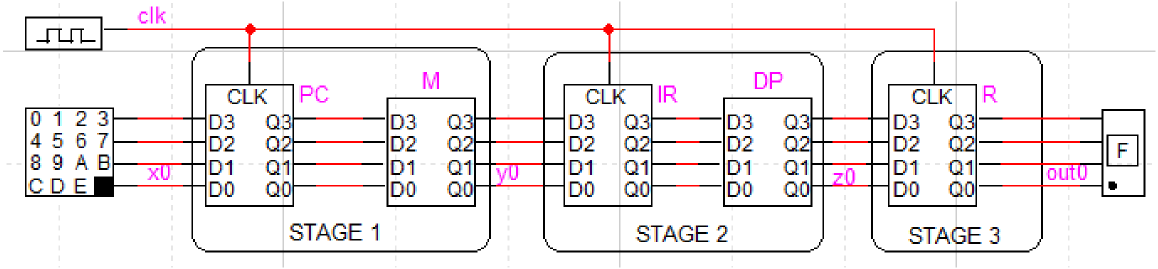
CMPT 295: Signal Propagation

In this lab you will be constructing a circuit that illustrates the role of registers in propagating a signal through a synchronous digital system. A synchronous system is one where all registers are edge-triggered and share a common clock. In such a system, during one clock period, signals can only advance from one register to the next; that is, signals cannot advance beyond the "next register" in one clock period, although it may take more than one clock period depending on the propagation delay encountered between registers.

Setting up the test circuit :

Construct the following circuit, consisting of 3 registers, 2 buffers, a hex keyboard, a hex display and a clock. The grouping of components into "stages" can be ignored for now):



The components labelled "PC", "IR", and "R" are positive edge-triggered parallel load registers, implemented using the "Reg-4 wo/CLR" component found in the Designworks library "Simulation logic.clf."

The components labelled "M" and "DP" are implemented with the "Buffer-4" component, also found in the "Simulation logic.clf" library.

The I/O components, including the "CLOCK" can be found in the library, "Simulation IO.clf."

1. Assign labels to the signal lines as indicated in the diagram. As each label is introduced, you should observe the label appearing in a list on the left hand side of the window at the bottom of the screen called the "Timing Window". This window will be used in order to generate a "waveform diagram" that shows the values on each labelled signal line as a function of time.
2. Highlight the buffer labelled, "M". From the Simulation pull-down menu select "simulation parameters" and then set the propagation delay of "'M" to 94 ns.
3. Highlight the buffer labelled "DP". In the same way as in the previous step, set the propagation delay to 14 ns.
4. In a similar manner, set the propagation delay of all 3 registers to 5 ns.
5. Expand the time scale on the timing window so that the space between "timing ticks" is 10 ns. You can do this by clicking on the "<>" icon in the tool bar at the top of the window.
6. NOTE: The clock component should have a default clock cycle period of 20 ns. After placing your clock in the circuit, verify this as follows:
   (a) Highlight the CLOCK entity in the schematic.
   (b) Click on the Simulation pull-down menu and select "Simulation Parameters".
   (c) The small window will show one period of a waveform and values corresponding to the length of time the signal is logic-0 and the length of time it is logic-1. If necessary, edit these values so that each value is "10 ns".

Experiment 1 :

1. In the Simulation pull-down menu select "Show values" in order to display the values on all signal lines.

2. Start the simulation by moving the "run speed" slider about 1/4 way toward the "running man". Set the hex keyboard to 0. Allow the simulation to run until all registers and the output are "0000".

3. While the simulation is running, set the Hex keyboard to "F". Allow the simulation to run until the hex display changes to "F". Then pause the simulation (click on the "sitting man") and examine the waveform diagram. Observe the value of the outputs, `Q0`, from each register on the signals labelled `y0, z0` and `out0` and note the time when each output changed. These times represent the earliest time an output is available as input to a register. However, it cannot be stored until the next positive edge of the register's clock input.

4. Beginning with the first positive edge after `x0` changes to "1", count the number of clock cycles that occur until the first positive edge after `y0` changes to "1". Then, from the change in `y0`, on the next positive edge, count the number of clock cycles until the first positive edge after `z0` changes to "1" . Finally count the number of clock cycles beginning with the first positive edge following the change to `z0` until the first positive edge after `out0` changes to 1 and a value is displayed on the hex display.

   These values define the number of clock cycles required to perform each "stage" of the digital system. A *stage* represents a subsystem of the digital system between two registers.

   In this example, there are 3 stages with the following propagation delays:

stage	components	propagation delay
1	PC → M	t_{pd}(stage 1) = t_{pd}(PC) + t_{pd}(M) = 5 + 94 = 99 ns
2	IR → DP	t_{pd}(stage 2) = t_{pd}(IR) + t_{pd}(DP) = 5 + 14 = 19 ns
3	R	t_{pd}(stage 3) = t_{pd}(R) = 5 ns

5. Compare your observed results of the number of clock cycles × the clock cycle period with the propagation delays of each stage. Since the clock cycle period is currently 20 ns, multiply this value by your observed number of cycles to obtain the time provided by the system for each stage. Why is this not the same as simply summing the propagation delays of the components that define each stage as given in the table?

Experiment 2 :

1. Highlight the clock by clicking on it. In the Simulation pull-down menu select "Simulation parameters" and set the time the signal is "0" to 20 ns, and the time the signal is "1" to 20 ns. This will result in the clock generating a signal alternating between 0 and 1 every 20 ns; that is, the clock cycle period now 40 ns.

2. Repeat the simulation in Experiment 1 and determine how many clock periods were required to propagate the value "F" from the hex keyboard to the hex display. Compare this waveform diagram with the one obtained in the previous experiment.

Experiment 3 :

Each positive edge of the clock signal represents an opportunity for a change in input to change the output of a stage. It is usually undesirable to have the input or output of a stage change more than once per clock cycle. Therefore, the clock cycle period is chosen so that a single clock cycle is long enough to insure that the propagation delay, t_{pd}, of any stage is less that the clock cycle period.

1. By experimenting with different clock cycle periods, find the minimum clock cycle period such that the value supplied by the Hex keyboard can be propagated to the Hex display in three clock cycles. What is this clock cycle period?
2. Modify the clock cycle period to be 300 ns. How many clock cycles are required to transmit the hex keyboard value to the hex display? Can the hex display be changed in less than 3 clock cycles after the hex keyboard is changed? Explain your answer.