Report on 4-bit Counter design
Report- 1, 2.

Report on D-Flipflop

Course project for ECE533
REPORT I

I. Objective:

The objective of this project is to design a 4-bit counter and implement it into a chip with the help of Cadence (custom IC design tool) following necessary steps and rules dependent on selected process technology.

II. Selection of Counter design:

The chosen design for the 4-bit counter is a simple 4-bit synchronous counter with synchronous set and reset option and input and output carry option. The reasons behind choosing this design are

i. Synchronous counter is the most used and reliable counter design
ii. Synchronous design ensures that all the output bits change simultaneously at the edge of a clock signal and holds that output until the next clock signal
iii. Low propagation delay than asynchronous counter
iv. The set reset options become effective with clock edge signal, so the outputs do not change suddenly in mid-clock-period.
v. Set/ reset options allows to clear the count data to start new counting session.
vi. Carry in and carry out allows 8-bit or higher bits counters by cascading.

III. Selection of Flip-flop:

The basic building block of a counter is flip-flop. The choice of flip-flop depends on the logic function of the circuit. The logic function of the counter suggests a T flipflop as most appropriate for the design. But I chose to use a J K Fliflop for the following reasons

i. J K flipflop allows to include both set and reset feature in it which can also be synchronous with the clock.
ii. A T flipflop can be easily produced from a J K flipflop by applying the same input to both J and K pin.
IV. **Design Steps:**

Cadence is one of the most popular, efficient and commercial custom IC design tool widely used for transforming a design idea to fabrication. For fruitful production of an IC from a design concept a sequence of steps has to followed. The flow chart below describes these steps:-

![Diagram showing the design steps flowchart]
### Truth tables and K-maps:

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>J3</th>
<th>K3</th>
<th>J2</th>
<th>K2</th>
<th>J1</th>
<th>K1</th>
<th>J0</th>
<th>K0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0001</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>0001</td>
<td>0010</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>0011</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>0011</td>
<td>0100</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>0100</td>
<td>0101</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>0101</td>
<td>0110</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>0110</td>
<td>0111</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>0111</td>
<td>1000</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>1001</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1001</td>
<td>1010</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1010</td>
<td>1011</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1011</td>
<td>1100</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1100</td>
<td>1101</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1101</td>
<td>1110</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1110</td>
<td>1111</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1111</td>
<td>0000</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
</tr>
</tbody>
</table>

**K-maps:**

<table>
<thead>
<tr>
<th>Q3</th>
<th>Q2</th>
<th>Q1</th>
<th>Q0</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>10</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

\[ J_3 = Q_2Q_1Q_0 \]

\[ K_3 = Q_2Q_3Q_0 \]
\[ J_2 = Q_3Q_0 \]

\[ K_2 = Q_3Q_0 \]

\[ J_1 = Q_3 \]

\[ K_1 = Q_3 \]

\[ J_0 = 1 \]

\[ K_0 = 1 \]
Logic circuit:

**Schematic of 4-bit counter:**

![Schematic diagram of 4-bit counter](image)

**Figure 1:** Schematic diagram of 4-bit counter

**VI. Required Basic building blocks:**

**J K Flip-flop:** Master slave JK flip flop used in for this circuit for reliable operation and stability. The flip flop triggers at negative edge of the clock cycle. Truth table for the JK flip flop is given below:

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q_{n+1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Q_n</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Q_n'</td>
</tr>
</tbody>
</table>
Schematic of JK flip flop:

Figure 2: Schematic diagram of JK flip flop with set and reset assembly

Description: Master slave cross coupled NAND gate assembly is negative edge triggered. At the negative edge of the clock cycle the output \(Q_n\) and \(Q'_n\) changes depending upon the input at J and K. The AND OR gate assembly forces “0-1” or “1=0” input combination to the J-K input to set or reset output. The changes for set and reset also occur at the negative edge of clock.
Layout:

Layout was drawn in AMI 0.6 micron process. All the building blocks used in the layout has the same height to maintain symmetry. Since the inverter output for the inverted clock signal was not able to drive 4 MOSFET gates in the subsequent stage a 3 inverter stage buffer was used, which increased the area. NAND gate assembly is placed on the upper row while the AND-OR gate assembly for set/reset is laid in the bottom row. Minimum spacing rule was utilized to reduce the size of layout.

Figure 3: Layout of JK flip flop with set/reset
Simulated output:

Simulation output shows performance of J K flip flop

![Simulation output of J K flip flop](image)

Figure 4: Simulation output of J K flip flop with reset enabled during 170ns to 220ns

NAND gates:

![NAND gate size= 35.5umX14um](image)

![NAND gate size= 35.5umX12um](image)

Figure 5: NAND2 gate size= 35.5umX14um

Figure 6: NAND3 gate size= 35.5umX12um
AND- OR- Inverter gates:

AND gate and OR gate require inverted input at the MOSFET gates. So to create a AND gate NAND gates were complemented with an inverter to get the inverted AND output. Similarly for OR gate a NOR gate output was inverted with an inverter to get the OR output.

Figure 7: Layout of an Inverter size= 35.5uX9u

Figure 8: Layout of an AND gate size= 35.5uX19u

Figure 9: Layout of an OR gate size= 35.5uX19u
VII. Complete Layout of counter design:

Maximum effort was made to minimize the design area on chip. Four J K flip flop was stacked on top of each other retaining minimum distance from n-well. Layout blocks are rotated to minimize interconnect requirement. AND gates used at input of each flip flop is placed at closest proximity to the J-K input and Q_n and Q_n’ output. Two bus of metal1 is used for supplying set and reset to each block.

The complete layout occupies an area $= 282.9\mu m \times 138.75\mu m = 39,252$

Figure 10: Layout of 4-bit up counter
Figure 11: Extracted layout of 4-bit up counter
VIII. Pre-layout simulation output:

Figure 12: Extracted layout including padframe

Figure 13: Prelayout simulation output, clock frequency= 10MHz
Post layout simulation output:

Figure 14: Post layout simulation output, clock frequency = 10MHz

Post layout simulation with capacitive load = 5pF

Figure 15: Post layout simulation output, clock frequency = 10MHz, C_load = 5 pF
Post layout simulation with set enabled:

![Post layout simulation output with set enabled](image1)

Figure 16: Post layout simulation output with set enabled, clock frequency= 10MHz

Post layout simulation output with reset enabled:

![Post layout simulation output with reset enabled](image2)

Figure 17: Post layout simulation output with reset enabled, clock frequency= 10MHz
IX. Measurement of rise time and fall time and propagation delay:

Capacitive loading was varied from 1pF to 15pF and rise time, fall time and propagation delay was measured. During rise time and fall time measurement time required to change from 30% to 70% of maximum voltage was measured. To measure propagation delay following equation was used

\[ t_{\text{delay}} = \frac{(t_{\text{PLH}} + t_{\text{PHL}})}{2} \]

Time measurement were taken for all four output bits.

### Q3

<table>
<thead>
<tr>
<th>Load</th>
<th>Rise (ns)</th>
<th>Fall (ns)</th>
<th>Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1pF</td>
<td>0.5909</td>
<td>3.1973</td>
<td>3.8689</td>
</tr>
<tr>
<td>3pF</td>
<td>2.8730</td>
<td>9.6123</td>
<td>7.9792</td>
</tr>
<tr>
<td>5pF</td>
<td>7.2129</td>
<td>16.8680</td>
<td>14.0339</td>
</tr>
<tr>
<td>7pF</td>
<td>9.5157</td>
<td>24.1250</td>
<td>19.1275</td>
</tr>
<tr>
<td>10pF</td>
<td>10.7590</td>
<td>32.5180</td>
<td>25.7555</td>
</tr>
<tr>
<td>12pF</td>
<td>14.6780</td>
<td>42.2800</td>
<td>31.1905</td>
</tr>
<tr>
<td>15pF</td>
<td>17.8560</td>
<td>53.2450</td>
<td>38.6780</td>
</tr>
</tbody>
</table>

### Q2

<table>
<thead>
<tr>
<th>Load</th>
<th>Rise (ns)</th>
<th>Fall (ns)</th>
<th>Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1pF</td>
<td>1.2517</td>
<td>3.1973</td>
<td>2.8210</td>
</tr>
<tr>
<td>3pF</td>
<td>3.5376</td>
<td>9.6123</td>
<td>7.9792</td>
</tr>
<tr>
<td>5pF</td>
<td>7.3777</td>
<td>16.8680</td>
<td>13.2200</td>
</tr>
<tr>
<td>7pF</td>
<td>8.8490</td>
<td>24.1250</td>
<td>19.6255</td>
</tr>
<tr>
<td>10pF</td>
<td>11.4230</td>
<td>32.5180</td>
<td>26.1155</td>
</tr>
<tr>
<td>12pF</td>
<td>15.1090</td>
<td>40.1670</td>
<td>31.7415</td>
</tr>
<tr>
<td>15pF</td>
<td>16.3110</td>
<td>53.2450</td>
<td>39.1010</td>
</tr>
<tr>
<td>Load</td>
<td>Rise (ns)</td>
<td>Fall (ns)</td>
<td>Delay (ns)</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>1pF</td>
<td>0.9274</td>
<td>3.1973</td>
<td>3.7745</td>
</tr>
<tr>
<td>3pF</td>
<td>3.0383</td>
<td>9.6123</td>
<td>7.9792</td>
</tr>
<tr>
<td>5pF</td>
<td>7.0472</td>
<td>16.8680</td>
<td>13.8648</td>
</tr>
<tr>
<td>7pF</td>
<td>8.5163</td>
<td>24.1250</td>
<td>19.2495</td>
</tr>
<tr>
<td>10pF</td>
<td>10.9800</td>
<td>32.5180</td>
<td>25.8660</td>
</tr>
<tr>
<td>12pF</td>
<td>14.2960</td>
<td>41.4480</td>
<td>31.4620</td>
</tr>
<tr>
<td>15pF</td>
<td>16.2710</td>
<td>53.2450</td>
<td>38.3450</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load</th>
<th>Rise (ns)</th>
<th>Fall (ns)</th>
<th>Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1pF</td>
<td>1.0916</td>
<td>3.1973</td>
<td>3.4866</td>
</tr>
<tr>
<td>3pF</td>
<td>3.2058</td>
<td>9.6123</td>
<td>8.0624</td>
</tr>
<tr>
<td>5pF</td>
<td>5.6769</td>
<td>16.8680</td>
<td>13.7700</td>
</tr>
<tr>
<td>7pF</td>
<td>8.8847</td>
<td>24.6160</td>
<td>18.4618</td>
</tr>
<tr>
<td>10pF</td>
<td>12.0230</td>
<td>32.5180</td>
<td>24.7010</td>
</tr>
<tr>
<td>12pF</td>
<td>14.5470</td>
<td>41.4470</td>
<td>29.2770</td>
</tr>
<tr>
<td>15pF</td>
<td>16.7380</td>
<td>53.2450</td>
<td>34.2685</td>
</tr>
</tbody>
</table>
Figure 18: Plot of Rise time, fall time Vs Capacitive load
**Figure 19: Plot of propagation delay Vs Capacitive load**

- **Q3 delay**
- **Q2 delay**
- **Q1 delay**
- **Q0 delay**
- Linear (Q3 delay)
- Linear (Q2 delay)
- Linear (Q1 delay)
- Linear (Q0 delay)
Chip Test Results:

Figure 20: Microphotograph of fabricated chip

Figure 21: Microphotograph of fabricated design of 4-bit counter
Test Result:

Figure 22: Output of 4-bit counter at 10 KHz Clock signal. Trace 1(yellow)- clock, Trace2 (green)- Q0, Trace3 (Purple) -Q1, Trace4(magenta) -Q2

Figure 23: Output of 4-bit counter at 10 KHz Clock signal. Trace 1(yellow)- clock, Trace2 (green)- Q2, Trace3 (Purple) -Q3, Trace4(magenta) -Carry_out
Plot of measured chip data:
Discussion:

Above plots show that the rise time, fall time and delay linearly varies with the capacitive loading. This indicates the linear behavior of the designed circuit. It may be noticed that the fall time higher than that of the fall time. This is because during fall time the capacitive loads discharges through the NMOS to reach ground potential. In this design a large number of NAND gates are used which has number of NMOS in series. The series connected NMOS has a larger resistance and has greater RC delay which leads to the higher value of fall time. Since in NAND gates PMOS are connected in parallel they quickly charge the load capacitor and give lower rise time.

All rise time and fall time was measured for 10MHz clock frequency. At this frequency capacitive loading beyond 15pF produces signal that cannot be recognized as ‘1’ or ‘0’. At lower clock frequency higher capacitive loading is possible.

X. Application:

Counter is used widely in number of digital and analog circuit. There are vast application of counter in the field of electronics. Some of them are listed below-

I. A counter can be used as frequency divider. This 4-bit counter can be efficiently used to get four different frequencies up to one sixteenth times of the original clock frequency. Since it is triggered only at negative edge of the clock, the first cycle gives output at half the frequency of that of the clock.

II. Counter can be used on building entrances to keep track of number of people entering or leaving.

III. Counter can be used in digital logic deigns to perform certain operation at certain steps.

IV. Counters are also applicable in timer circuit to turn on/ off a device after certain time with use of a proper logic function.
ABSTRACT:

For the final project in ECE-533 we had to design a four bit counter. This project was done with the aid of Cadence, in the AMI-0.6 micron process. The project involved the design of the schematic as well as the layout. In both cases, simulations were done. The counter is simulated under no-load condition as well as with few test loads. The rise time, fall time and delay are measured.

I designed an Asynchronous four bit counter using T flip flops.

INTRODUCTION:

A Flip-flop is the name given to two-state devices which offer basic memory for sequential logic operations. Flip-flops are synchronous bistable devices that operate as memory elements. A flip-flop circuit contains two outputs, one is for the normal value and the other is for the complement value of the stored bit. Flip-flops are used for digital data storage and transfer and are commonly used in banks called "registers" for the storage of binary numerical data.

Counters are designed using flip-flops. Counters can be classified as synchronous and asynchronous counters based on the application of clock to the flip-flops. A synchronous counter is clocked by a single clock for all the stages and the output for each stage changes at the same time. In an asynchronous counter the output from the previous stage is given as the clock for the next stage so that the output ripples across each stage to reach the final count.

The following were the steps involved in designing this project:

1. Design a flip flop using nand gates and an inverter.
2. Draw the schematic and layout using cadence
3. Design a four bit counter using the designed flip flop
4. Draw the schematic and layout of the counter using cadence
5. Measure the rise time and fall time of the various bits
6. Measure the propagation delay across various stages with different capacitive loads
I have designed a four bit asynchronous counter with the aid of T flip-flops.

This counter was designed to provide an output that is free of glitches.

**T FLIP FLOP:**

a) Introduction

The T or "toggle" flip flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. T flip flop is useful for constructing binary counters, frequency dividers, and general binary addition devices.

A T flip flop can be designed from a J K flip flop by shorting the J and K inputs.
Diagram of a T flip-flop

<table>
<thead>
<tr>
<th>Q</th>
<th>T</th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Transition table of T Flip Flop

b) Implementation:

The implementation of the flip flop consists of the following steps:

- Designing the schematic
- Performing pre layout simulations
- Designing the layout
- Obtaining the extracted view
- Performing LVS
- Performing post layout simulations
SCHEMATIC OF T FLIP FLOP:

PRELAYOUT SIMULATION RESULTS:
LAYOUT OF T FLIP FLOP:
COUNTER USING T FLIP FLOPS:

a) Introduction:

The counter was built with the help of T-Flip Flops. The gates used in realizing the design were two 3 Input Nand Gates, three 2 Input Nand gates and an inverter. The outputs of the Flip Flop are Q and QOb.

In an asynchronous counter the clock pulse is applied only at the first stage. The rest of the stages are driven by the output of the preceding flip flop. All the output bits do not change at the same time.
An asynchronous counter is also called as a ripple counter because only the first bit changes with the clock pulse, there exists some delay before the other bits change, so the output ripples across the counter.

A binary counter can be realized using T-Flip Flops by counting the number of toggles in the previous stage. The T input of each flip-flop is set to 1 to produce a toggle at each cycle of the clock input. For each two toggles of the first cell, a toggle is produced in the second cell, and so on down to the fourth cell. This produces a binary number equal to the number of cycles of the input clock signal.

Using a Master-slave configuration isolates the output from any glitches resulting from any changes happening in the input signal. The master-slave flip-flop is essentially two back-to-back JK flip-flops, but the feedback is to both to the master flip flop and the slave flip flop. In this configuration, the master flip flop sees the input when the clock is high and the output of the first stage holds the input for the next stage. During the clock ‘low’, the slave circuit is enabled to track the change in the input by using the output of the master Flip Flop. Thus the master-slave configuration eliminates any sharp change of state within a clock cycle and the flip flop is free of oscillations.
b) Implementation:

SCHEMATIC DIAGRAM OF A 4 BIT COUNTER USING T FLIP FLOPS:

PRELAYOUT SIMULATION RESULTS:
LAYOUT OF 4-BIT COUNTER:
EXTRACTED VIEW OF THE COUNTER:
### LAYOUT VS SCHEMATIC (LVS):

<table>
<thead>
<tr>
<th>Net-list summary for /home/michelle/schematic/schematic.netlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>rout</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

#### Personal correspondence points
- K1: K2: 81K
- K3: K4: 7
- K5: K6: 10
- K6: K7: 31
- K7: K8: 62
- K1: K2: 34
- K3: K4: 10
- K5: K6: 75
- K7: K8: 32
- K1: K2: 30K
- K3: K4: 10
- K5: K6: 50
- K7: K8: 32

#### Set lists

<table>
<thead>
<tr>
<th></th>
<th>layout</th>
<th>schematic</th>
</tr>
</thead>
<tbody>
<tr>
<td>unexpected</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>postnet</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mixed</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>total</td>
<td>132</td>
<td>132</td>
</tr>
</tbody>
</table>

#### nets

<table>
<thead>
<tr>
<th></th>
<th>layout</th>
<th>schematic</th>
</tr>
</thead>
<tbody>
<tr>
<td>unexpected</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>report</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mixed</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>total</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### transiently

<table>
<thead>
<tr>
<th></th>
<th>layout</th>
<th>schematic</th>
</tr>
</thead>
<tbody>
<tr>
<td>unexpected</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>added net</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>different type</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>total</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Files from /home/michelle/schematic/schematic:
- detail.txt
- netlist.txt
- portlist.txt
- terminal.txt
- transient.txt
- unconnected.txt
- unplaced.txt
- unreport.txt
- unroute.txt
- unused.txt
POST-LAYOUT SIMULATION RESULTS:

RISE TIME AND FALL TIME FOR VARIOUS OUTPUT BITS AT ZERO LOAD:

<table>
<thead>
<tr>
<th>RISE TIME (ns)</th>
<th>FALL TIME (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LOAD CAPACITANCE VS PROPAGATION DELAY:

\[
\begin{array}{|c|c|c|}
\hline
X0 & 0.3 & 0.45 \\
\hline
X1 & 0.25 & 0.4 \\
\hline
X2 & 0.2 & 0.43 \\
\hline
X3 & 0.15 & 0.39 \\
\hline
\end{array}
\]
APPLICATIONS OF COUNTERS:

- To count the number of times that a certain event takes place.
- To control a fixed sequence of actions in a digital system.
- To generate timing signals.
- To generate clocks of different frequencies.
- Used in various devices such as ATM’s, Watches etc.

SCHEMATIC OF A T FLIP FLOP:
SIMULATION RESULT OF THE T FLIP FLOP SCHEMATIC:
SCHEMATIC DIAGRAM OF A 4 BIT COUNTER USING T FLIP FLOPS:

SIMULATION RESULT OF THE COUNTER’S SCHEMATIC:
<table>
<thead>
<tr>
<th></th>
<th>RISE TIME (ns)</th>
<th>FALL TIME (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0</td>
<td>0.3</td>
<td>0.45</td>
</tr>
<tr>
<td>X1</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td>X2</td>
<td>0.2</td>
<td>0.43</td>
</tr>
<tr>
<td>X3</td>
<td>0.15</td>
<td>0.39</td>
</tr>
</tbody>
</table>
Flip-Flop Design

1. Objective

The objective of this project is to design and simulate a J-K Flip-Flop and a D-Flip Flop using Cadence (IC design software). The design will be in the form of a logic circuit and a layout.

2. Flip Flops

2.1 J-K Flip Flop

I will be designing a J-K flip flop with a set/reset option. The outputs will only change state on the falling edge of the CLK signal, and the J and K inputs will control the future output. If both the J and K inputs are held at logic 1 and the CLK signal continues to change, the Q and Q' outputs will simply change state with each falling edge of the CLK signal.

\[
\begin{array}{cccc}
R & J & K & Q_{n+1} \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & Q_n \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 1 & 1 & nQ \\
1 & d & d & 0
\end{array}
\]

Table 1: Truth Table for the J-K Flip-flop with Set/Reset

2.2 D Flip Flop

I also will be designing a negative edge triggered D Flip-Flop. The outputs’ states to change only when the clock signal falls from logic ‘1’ to logic ‘0’.
\[
\begin{array}{c|c|c|c}
D & Q & \text{CLK} & Q_{\text{next}} \\
\hline
0 & X & \text{Falling} & 0 \\
1 & X & \text{Falling} & 1 \\
\end{array}
\]

\(X\) – either logic ‘1’ or logic ‘0’

**Table 2**: Truth Table for the negative edge triggered D Flip-flop

### 3. Design Implementation and Simulation

I designed the J-K Flip Flop and the D Flip-Flop using Cadence IC Design Software with an AMI-0.6 micron process.

\(L_n=L_p=0.6\mu m\)

\(W_p=7.5\mu m\)

\(W_n=3.0\mu m\)

### 3.1 Schematic Design

The design process began with the schematics of the J-K flip flop (Figure 1) and the D flip flop (Figure 2), which I utilized nand gates and an inverter that I designed in my previous labs.

### 3.2 Schematic Simulation

After designing the schematics, I simulated using Spectre and the Cadence Analog Environment in order to generate waveforms that demonstrate the behavior of the J-K flip flop (Figure 3) and the D flip flop (Figure 4). In order to simulate realistic results, capacitances were added to the outputs in order to generate delays in the waveforms.

### 3.2 Create Symbol
Symbol creation was achieved by creating symbol cell view from schematic cell view

3.3 Layout Design

I then designed the layouts for the J-K flip flop (Figure 5) and the D flip flop (Figure 6) in which I utilized the layouts of nand gates and an inverter which I designed in my previous labs. After connecting all of the I/O pins, I made sure that I connected all of the vdd pins from each nand gate and inverter. I did the same for all of the ground (gnd) pins.

3.4 DRC Verify

After the design is complete, I used the DRC tool to verify that the layouts of the J-K flip flop and the D flip flop contained no. The DRC tool confirmed that the spacing between the various components in the layouts was not too close together and that pins were not duplicated on the same layout.

3.5 Extract

I created an extracted view of the layouts in which I utilized parasitic capacitances.

3.6 Verify LVS

I then utilized the LVS tool to verify that the layouts designs matched that of my schematics in terms of the number of nets, terminals, and instances (Figure 7 and Figure 8).

3.7 Extracted Layout Simulation

The simulation of the layouts is implemented exactly the same way as the schematic simulation. We also add capacitances to the outputs in the netlist to simulate delays and produce more realistic results. (Figure 9 and Figure 10)

4. Waveform Analysis
4.1 Results

JK Flip Flop Simulation Waveform – Figure 9

D Flip Flop Simulation Waveform – Figure 10

4.2 Rise Time, Fall Time, and Propagation Delay

The rise time/fall time were measured using the 10% - 90% method.

The propagation delay was measured using the 50% - 50% method.

4.3 Capacitive Load Effect Chart

<table>
<thead>
<tr>
<th>Capacitive Load (fF)</th>
<th>Rise Time (ns)</th>
<th>Fall Time (ns)</th>
<th>Propagation Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.41</td>
<td>0.7</td>
<td>0.64</td>
</tr>
<tr>
<td>20</td>
<td>0.64</td>
<td>0.82</td>
<td>0.74</td>
</tr>
<tr>
<td>50</td>
<td>1.33</td>
<td>1.55</td>
<td>1.48</td>
</tr>
<tr>
<td>100</td>
<td>2.35</td>
<td>3.17</td>
<td>2.94</td>
</tr>
</tbody>
</table>

Table 1: Rise time, fall time, and propagation delay measured from the J-K flip flop waveform
4.4 Capacitive Load Effect Graphs
5. Conclusion from Waveform Analysis

The capacitive load and the rise/fall times and delay correlate in a linear fashion. Based on the analysis of the rise/fall times and delay we see that if the capacitive load increases, then the rise time, fall time, and delay also increase.

6. Application

J-K flip flops can be utilized in binary counters, shift registers, and sequence detectors. Because of the behavior of the D-Flip Flop described in the introduction, it can be utilized in shift registers, which are an essential part of many electronic devices.

7. Pad Frame

Pad frame Schematic (Figure 11) – No Errors

Pad frame Layout (Figure 12) – DRC Clean

The pad frame schematic and the pad frame layout contained no errors. However, the symbol of the JK flip flop and contained no vdd or ground pins, therefore my attempts to verify both pad frames using LVS was unsuccessful.
8. Appendix

Figure 1: J-K Flip Flop Schematic

Figure 2: D Flip Flop Schematic
Figure 3: J-K Flip Flop Schematic Simulated Waveform
Figure 4: Figure 3: D Flip Flop Schematic Simulated Waveform

Figure 5: J-K Flip Flop Layout

Figure 6: D Flip Flop Layout
Figure 7: J-K Flip Flop LVS Verification
Figure 8: D Flip Flop LVS Verification

Figure 9: J-K Flip Flop Layout Simulated Waveform
Figure 10: D Flip Flop Layout Simulated Waveform
Figure 11 – JK Flip Flop Pad Frame Schematic
Figure 12 – JK Flip Flop Pad Frame Layout
Microphotograph of the Chip: