The goal of the assignment is to develop an understanding for the different Verilog concepts regarding switch-level design utilizing transistor primitives.

***Homework 7: PART 1***

The Verilog code for this section can be found in the Pt1 folder of the ZIP file submitted:

*transistorCircuit.v*

*transistorCircuitTester.v*

In looking at the given function in this part of the assignment, simply turning it into a Karnaugh map yielded the following minimization:

**f (a,b,c,d) = a’b + ac + c’d’**

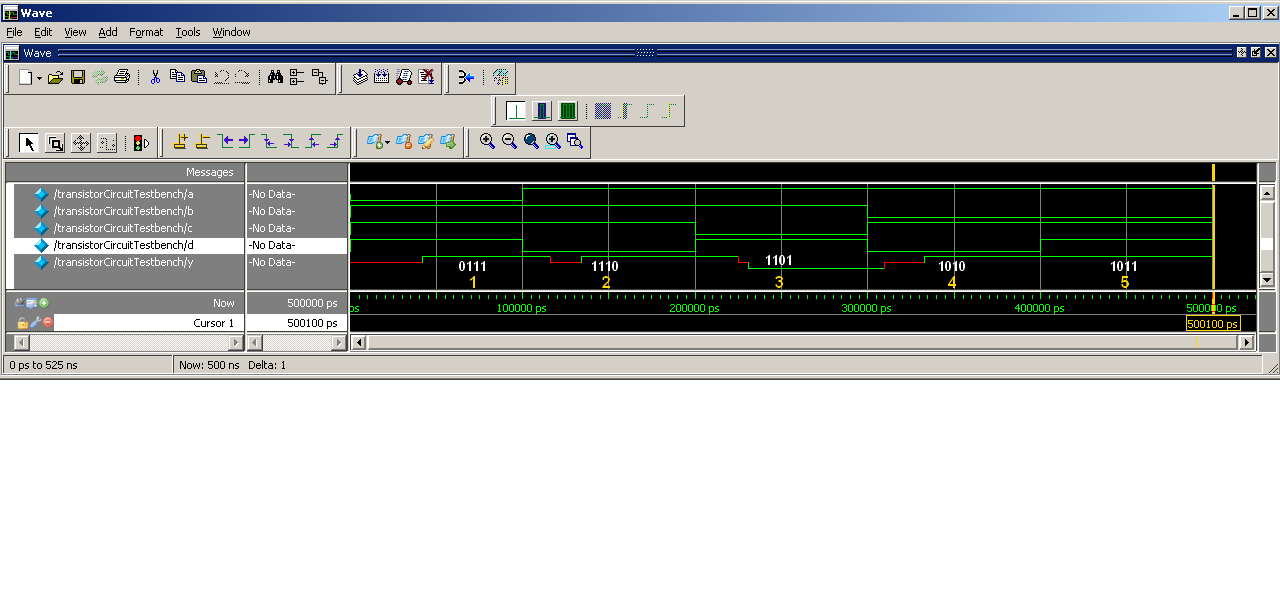
In order to further minimize at the transistor level, the newly-minimized function can be re-written as:

**f (a,b,c,d) = (a + b’)’ + (a’ + c’)’ + (c + d)’**

This allows the use of 4-transistor NOR gates for the individual AND operations, since the complements of all input signals are available. The OR operations will be implemented as NOT (NOR) functions, for a total of 6 transistors for each. The overall complex gate (CMOS) contains a total of **24 transistors**.

In order to test this circuit, 5 random variations of f(a,b,c,d) were generated and input into the system, with the changes separated by a total of 100ns. The output and analysis is seen below:

**Simulation Waveform Output of transistorCircuitTester**



**Scenario 1 CORRECT**

* a = 0, b = 1, c = 1, d = 1
* f (a,b,c,d) = a’b + ac + c’d’ = (1\*1) + (0\*1) + (0\*0) = 1
* Expected Output = 1 Actual Output = y = 1

**Scenario 2 CORRECT**

* a = 1, b = 1, c = 1, d = 0
* f (a,b,c,d) = a’b + ac + c’d’ = (0\*1) + (1\*1) + (0\*1) = 1
* Expected Output = 1 Actual Output = y = 1

**Scenario 3 CORRECT**

* a = 1, b = 1, c = 0, d = 1
* f (a,b,c,d) = a’b + ac + c’d’ = (0\*1) + (1\*0) + (1\*0) = 0
* Expected Output = 0 Actual Output = y = 0

**Scenario 4 CORRECT**

* a = 1, b = 0, c = 1, d = 0
* f (a,b,c,d) = a’b + ac + c’d’ = (0\*0) + (1\*1) + (0\*1) = 1
* Expected Output = 1 Actual Output = y = 1

**Scenario 5 CORRECT**

* a = 1, b = 0, c = 1, d = 1
* f (a,b,c,d) = a’b + ac + c’d’ = (0\*1) + (1\*1) + (0\*0) = 1
* Expected Output = 1 Actual Output = y = 1

This concludes the analysis for Homework 7, Part 1.

***Homework 7: PART 2***

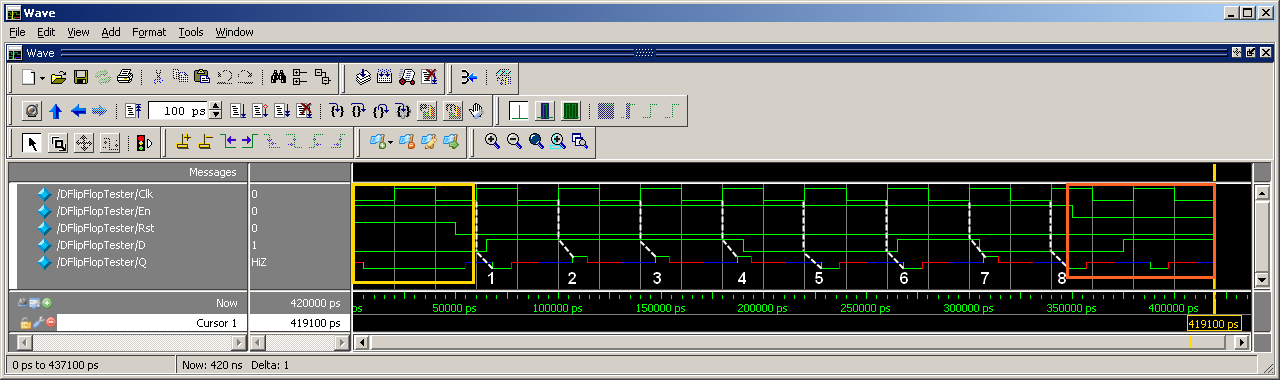
The Verilog code for this section can be found in the Pt2 folder of the ZIP file submitted:

*DFlipFlop.v*

*DFlipFlopTester.v*

The circuit was developed using a master-slave dynamic latch layout, with essentially *two-stages* of flip-flop logic. A capacitor was used to store the value of *Q* for the *En* to work correctly. In order to test this circuit, a racetrack of input changes was developed, all integrated with a periodic clock of 40ns. All functionality of the D Flip-Flop was tested. The output and analysis is seen below:

**Simulation Waveform Output of DFlipFlopTester**



In looking at the **YELLOW BLOCK**: **CORRECT**

The Rst variable is instantiated as: Rst = 1

Given the problem statement, we expect Q = 0 when Rst = 1.

In looking at the **MIDDLE BLOCK**:

With En = 1 and Rst = 0, when the Clk changes from 0 to 1 *(@posedge* effectively), the circuit takes a look at the value currently on D and propagates it through the transistors until it reaches Q.

In looking at the **ORANGE BLOCK**: **CORRECT**

The En variable is now set to 0: En = 0

Given the problem statement, we expect D = Q when En = 0.

This means that the next clock should see the previous Q as its current Q: Prev\_Q = 0, Curr\_Q = 0

This concludes the analysis for Homework 7, Part 2.