Name _____Andrew Quick_____ abc123_____qxt050_____

Homework #2

1. Write the Verilog code for a 4:1 MUX

```
module Mux4to1 (
    input wire I0, I1, I2, I3,  // 4 data inputs
    input wire S0, S1,          // 2 select lines
    output wire Y               // Output
);

assign Y = (~S1 & ~S0 & I0) |  // Select I0 when S1 = 0, S0 = 0
           (~S1 &  S0 & I1) |  // Select I1 when S1 = 0, S0 = 1
           ( S1 & ~S0 & I2) |  // Select I2 when S1 = 1, S0 = 0
           ( S1 &  S0 & I3);   // Select I3 when S1 = 1, S0 = 1

endmodule
```

2. There are a number of delays in Verilog. The two we will use most in this course are below. Explain the difference between these two delays:

    - Inertial Delay

      This is the default type of delay in Verilog. It filters out glitches, meaning only input pulses longer than the delay propagate to the output. Example: If #5 is used, pulses shorter than 5 time units will be filtered.

    - Transport Delay

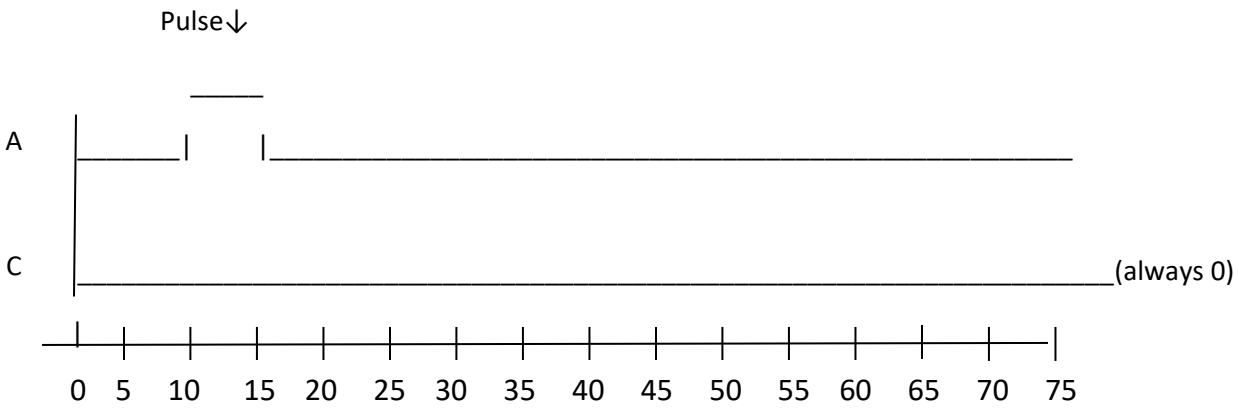
      This models a physical delay without filtering any input pulses. Even if an input changes rapidly, the signal is propagated to the output after the delay.

Name _____Andrew Quick_____ abc123_____qxt050_____

3. You apply a 5ns A=1 pulse to signal A where B=1. Draw the resultant timing diagram for A and C where C is driven by this code:

// Inertial delay therefore if pulse is under 10 time units, it'll be filtered out

assign #10 C = A && B



4. Write Verilog code for a 6 bit adder using a full 1 bit adder modules

```
module FullAdder (A, B, Ci, Co, S);

    input A, B, Ci;     // Inputs: A, B, and Carry In (Ci)
    output S, Co;       // Outputs: Sum (S) and Carry Out (Co)

    assign S = A ^ B ^ Ci;       // Sum calculation
    assign Co = (A & B) | (B & Ci) | (A & Ci);  // Carry Out calculation

endmodule
```

Name _____Andrew Quick_____ abc123_____qxt050_____

```
module Adder6 (S, Co, A, B, Ci);

    input [5:0] A, B;  // 6-bit inputs
    input Ci;          // Carry input
    output [5:0] S;    // 6-bit sum output
    output Co;         // Carry out
    wire [5:1] C;      // Internal carry signals

    // Instantiate 6 Full Adders
    FullAdder FA0 (A[0], B[0], Ci, C[1], S[0]);
    FullAdder FA1 (A[1], B[1], C[1], C[2], S[1]);
    FullAdder FA2 (A[2], B[2], C[2], C[3], S[2]);
    FullAdder FA3 (A[3], B[3], C[3], C[4], S[3]);
    FullAdder FA4 (A[4], B[4], C[4], C[5], S[4]);
    FullAdder FA5 (A[5], B[5], C[5], Co, S[5]);

endmodule
```