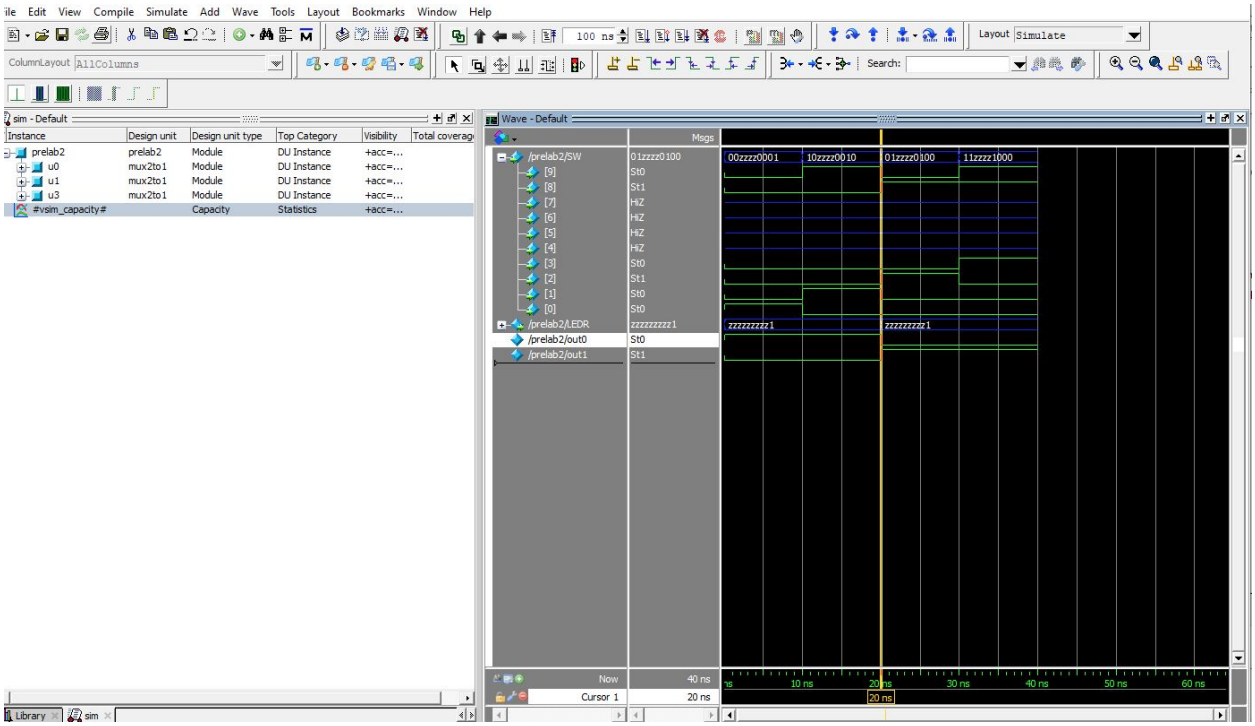
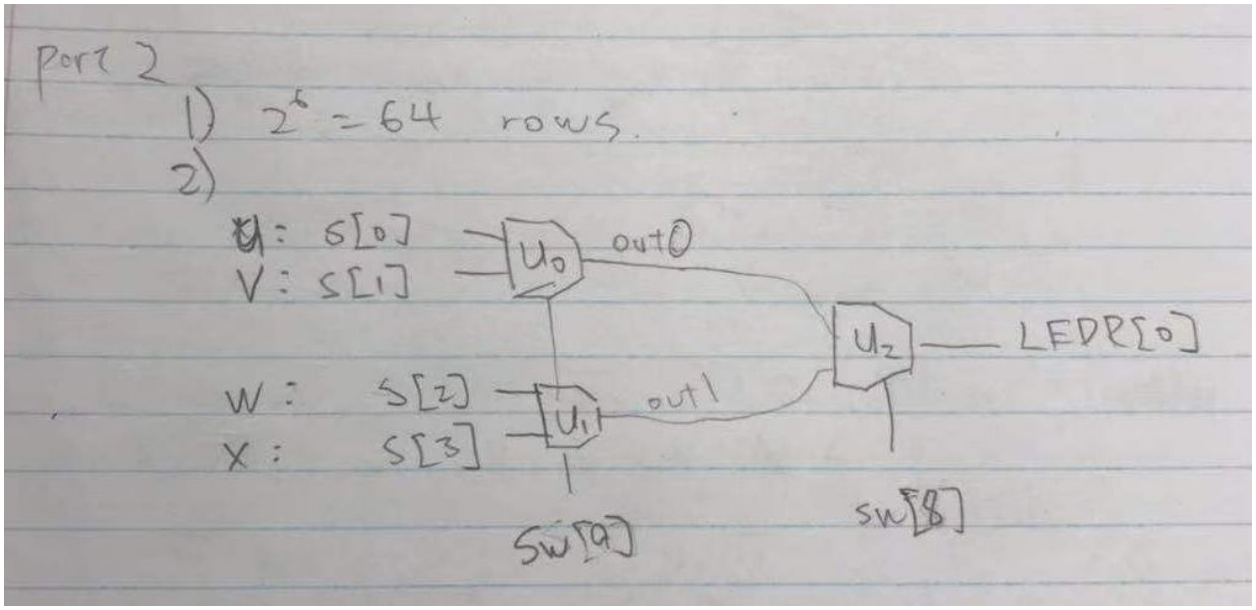
Ryan Chang
CSC258
Prelab 2

Part 1



Part 2



Part 2 3)

```
//SW[2:0] data inputs
//SW[9] select signal

//LEDR[0] output display

module mux(LEDR, SW);
    input [9:0] SW;
    output [9:0] LEDR;

            wire out0, out1;

    mux2to1 u0(
        .x(SW[0]),
        .y(SW[1]),
        .s(SW[9]),
        .m(out0)
        );
            mux2to1 u1(
        .x(SW[2]),
        .y(SW[3]),
        .s(SW[9]),
        .m(out1)
        );
            mux2to1 u3(
        .x(out0),
        .y(out1),
        .s(SW[8]),
        .m(LEDR[0])
        );
endmodule

module mux2to1(x, y, s, m);
    input x; //selected when s is 0
    input y; //selected when s is 1
    input s; //select signal
    output m; //output

    assign m = s & y | ~s & x;
    // OR
    // assign m = s ? y : x;
```

endmodule

Part 2 5): simulation code

```
# Set the working dir, where all compiled Verilog goes.
vlib work

# Compile all Verilog modules in mux.v to working dir;
# could also have multiple Verilog files.
# The timescale argument defines default time unit
# (used when no unit is specified), while the second number
# defines precision (all times are rounded to this value)
vlog -timescale 1ns/1ns prelab2.v

# Load simulation using mux as the top level simulation module.
vsim prelab2

# Log all signals and add some signals to waveform window.
log {/*}
# add wave {/*} would add all items in top level simulation module.
add wave {/*}

# First test case   u should light up
# Set input values using the force command, signal names need to be in {} brackets.
force {SW[0]} 1
force {SW[1]} 0
force {SW[2]} 0
force {SW[3]} 0
force {SW[9]} 0
force {SW[8]} 0
# Run simulation for a few ns.
run 10ns

# Second test case:  v should light up
# SW[0] should control LED[0]
force {SW[0]} 0
force {SW[1]} 1
force {SW[2]} 0
force {SW[3]} 0
force {SW[9]} 1
force {SW[8]} 0
run 10ns
```

```
# ...    w should light up
# SW[0] should control LED[0]
force {SW[0]} 0
force {SW[1]} 0
force {SW[2]} 1
force {SW[3]} 0
force {SW[9]} 0
force {SW[8]} 1
run 10ns

# x should light up
force {SW[0]} 0
force {SW[1]} 0
force {SW[2]} 0
force {SW[3]} 1
force {SW[9]} 1
force {SW[8]} 1
run 10ns
```

Part 3 1)

HEX[0]

0, 2, 3, 5, 6, 7, 8, 9, A, C, E, F

0000, 0010, 0011, 0101, 0110, 0111, 1000, 1001, 1010, 1100, 1110, 1101 should be 1 (cuz 1 is off).

this order

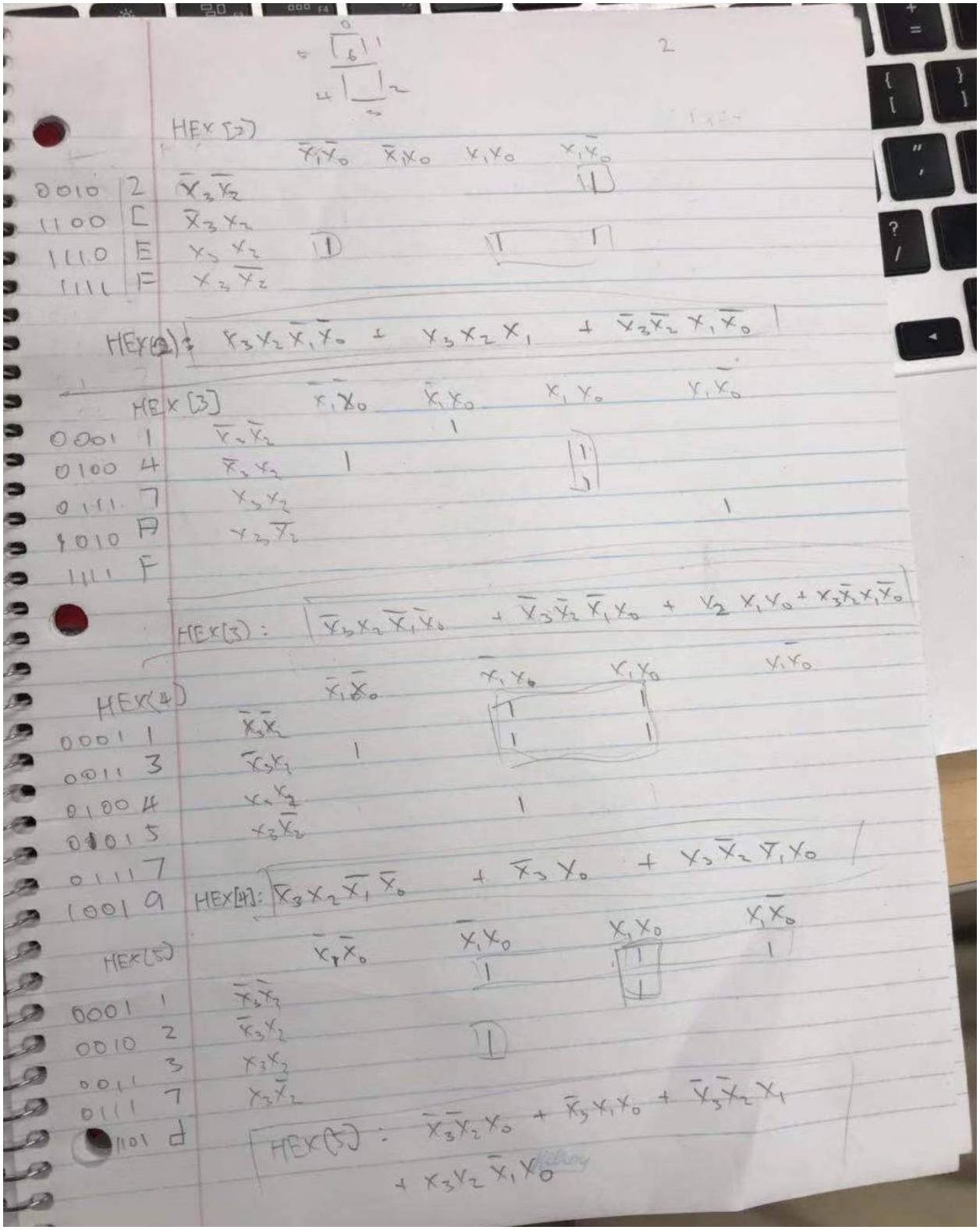
	$\bar{X}_1 \cdot \bar{X}_0$	$\bar{X}_1 \cdot X_0$	$X_1 \cdot X_0$	$X_1 \cdot \bar{X}_0$
$\bar{X}_3 \cdot \bar{X}_2$	0	1	0	0
$\bar{X}_3 \cdot X_2$	1	1	0	0
$X_3 \cdot X_2$	0	0	1	0
$X_3 \cdot \bar{X}_2$	0	0	0	0

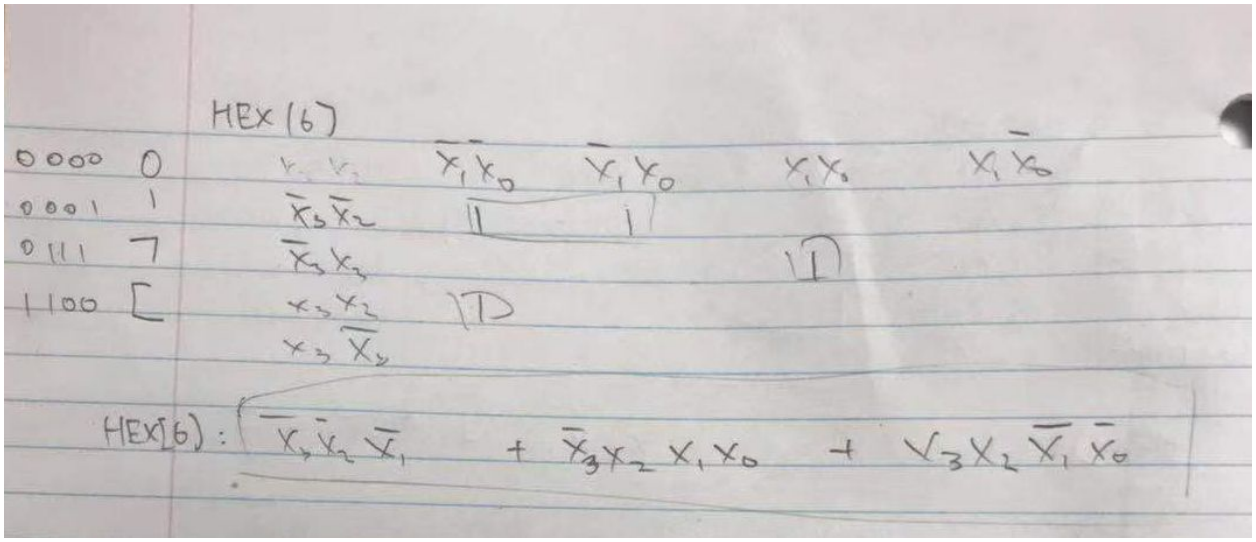
boolean: $HEX[0]: \bar{X}_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot X_0 + \bar{X}_3 X_2 \cdot \bar{X}_1 \cdot \bar{X}_0 + X_3 \cdot X_2 \cdot \bar{X}_1 \cdot X_0 + X_3 \cdot \bar{X}_2 \cdot X_1 \cdot X_0$

HEX[1]

		$\bar{X}_1 \cdot \bar{X}_0$	$\bar{X}_1 \cdot X_0$	$X_1 \cdot X_0$	$X_1 \cdot \bar{X}_0$
0000	0				
0001	1	$\bar{X}_3 \bar{X}_2$ 0	0	0	0
0010	2	$\bar{X}_3 \cdot X_2$ 0	1	0	1
0011	3	$X_3 X_2$ 1	0	1	0
0100	4	$X_3 \bar{X}_2$ 0	0	1	0
0111	7				
1000	8				
1001	9				
1010	A				
1101	D				

$HEX[1]: \bar{X}_3 \cdot X_2 \bar{X}_1 X_0 + X_3 X_2 \bar{X}_0 + X_3 X_1 X_0 + X_2 X_1 X_0$





Part 3 2)

```
module seven_seg_decoder(S,HEX0);
        input [9:0]S;
        output [6:0]HEX0;

        //x_0 is S[0]
        //x_1 is S[1]
        //x_2 is S[2]
        //x_3 is S[3]

        //hex0
        Assign HEX0[0]=(~S[3]&~S[2]&~S[1]&S[0])|(~S[3]&S[2]&~S[1]&~S[0])|
        (S[3]&S[2]&~S[1]&S[0])|(S[3]&~S[2]&S[1]&S[0]);

        //hex1
        Assign HEX0[1]=(~S[3]&S[2]&~S[1]&S[0])|(S[3]&S[2]&~S[0])|
        (S[3]&S[1]&S[0])|(S[2]&S[1]&S[0]);


        //hex2
        Assign HEX0[2]=(S[3]&S[2]&~S[1]&~S[0])|(S[3]&S[2]&S[1])|
        (~S[3]&~S[2]&S[1]&~S[0]);

        //hex3
        Assign HEX0[3]=(~S[3]&S[2]&~S[1]&~S[0])|(~S[3]&~S[2]&~S[1]&S[0])|
        (S[2]&S[1]&S[0])|(S[3]&~S[2]&S[1]&~S[0]);

        //hex4
        Assign HEX0[4]=(~S[3]&S[2]&~S[1]&~S[0])|(S[3]&S[0])|
```

```
        (S[3]&~S[2]&~S[1]&S[0]);

        //hex5
        Assign HEX0[5]=(~S[3]&~S[2]&S[0])|(~S[3]&S[1]&S[0])|
        (~S[3]&~S[2]&S[1])|(S[3]&S[2]&~S[1]&S[0]);

        //hex6
        Assign HEX0[6]=(~S[3]&~S[2]&~S[1])|(~S[3]&S[2]&S[1]&S[0])|
        (S[3]&S[2]&~S[1]&˜S[0]);


endmodule
```

Part 3 3)

```
# Set the working dir, where all compiled Verilog goes.
vlib work

# Compile all Verilog modules in mux.v to working dir;
# could also have multiple Verilog files.
# The timescale argument defines default time unit
# (used when no unit is specified), while the second number
# defines precision (all times are rounded to this value)
vlog -timescale 1ns/1ns prelab2.v

# Load simulation using mux as the top level simulation module.
vsim prelab2

# Log all signals and add some signals to waveform window.
log {/*}
# add wave {/*} would add all items in top level simulation module.
add wave {/*}

# First test case  THIS SHOULD DISPLAY: B
force {SW[0]} 1
force {SW[1]} 1
force {SW[2]} 0
force {SW[3]} 1

run 10ns
```

```
#   THIS SHOULD DISPLAY: A
force {SW[0]} 0
force {SW[1]} 1
force {SW[2]} 0
force {SW[3]} 1

run 10ns

#   THIS SHOULD DISPLAY: 3
force {SW[0]} 1
force {SW[1]} 1
force {SW[2]} 0
force {SW[3]} 0

run 10ns

#   THIS SHOULD DISPLAY: 1
force {SW[0]} 1
force {SW[1]} 0
force {SW[2]} 0
force {SW[3]} 0

run 10ns

#   THIS SHOULD DISPLAY: 4
force {SW[0]} 0
force {SW[1]} 0
force {SW[2]} 1
force {SW[3]} 0

run 10ns

#   THIS SHOULD DISPLAY: 5
force {SW[0]} 1
force {SW[1]} 0
force {SW[2]} 1
force {SW[3]} 0

run 10ns
```