Introduction to Digital Design with Verilog HDL

Exercise 1: What changes would result in a 3-input OR gate?

Exercise 2: What schematic would you expect if the statement was `assign y = ( a ^ b ) | c ;`?

Exercise 3: If the signal `i` is declared as `logic [2:0] i;`, what is the 'width' of `i`?

If `i` has the value 6 (decimal), what is the value of `i[2]`?

Of `i[0]`?

Exercise 4: What are the widths and values, in decimal, of the following:

`4'b1001`?

`5'd3`?

`6'h0_a`?

`3`?

Exercise 5:

```
logic [3:0] z [0:1] = ' 3'b11, 3'b101 ;
```

Draw a two-dimensional table showing the values of the bits in **z**.

Exercise 6: Write the truth table for a one-bit adder with carry. There are three inputs (**a**, **b** and **carry**) and two outputs (**sum** and **carry**). Define an array that implements this function. Write an expression that uses this array to find the sum and carry of `logic` signals **a** and **b**.

Exercise 7: What are the values of the following expressions: `!4'b010`?

`~4'b010`?

`0+~(!0)`?

Exercise 8: An array declared as `logic [15:0] n;` and has the value `16'h1234`. What are the values and widths of the following expressions?

`n[15:13]`

`!n`

`~n[3:0]`

`n>>4`

`n + 1'b1`

`n[7:0] - n[3:0]`

```
n >= 16'h1234
```

```
n ^ '1
```

```
n && !n
```

```
 n * ( !n + 1'b1 )
```

Exercise 9: What are the width and value of the expression: `3 ? 16'd10 : 8'h20`?

If **x** has the value 0, what is the value of the expression: `x ? 1'b1 : 1'b0`?

If **x** has the value -1?

Exercise 10: Draw the schematics corresponding to:

```
y = a ? ( b ? s1 : s2 ) : ( c ? s3 : s4 );
```

```
y = a ? s1 : b ? s2 : c ? s3 : s4;
```

```
y = a ? b ? c ? s3 : s4 : s2 : s1 ;
```

Exercise 11: Use slicing and concatenation to compute the byte-swapped value of an array **n** declared as **`logic [15:0] n`**.

Exercise 12: If `n` has the value `16'h1234`, what are the value and width of:

`{n[7:0],n[15:8],4'b1111}`?

Exercise 13: Use concatenation to shift **n** left by two bits.

Exercise 14: Use concatenation to assign the high-order byte of **n** to **a** and the low-order byte to **b**.

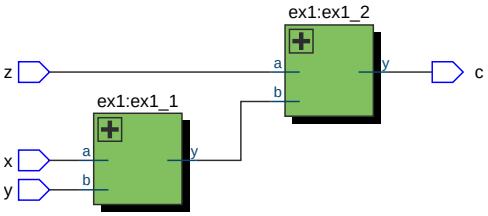
Exercise 15:

```
assign y = a + 1 ;
```

Some software warns about truncation. How could you re-write the **`assign`** statement to avoid such a warning?

Exercise 16: Write an **`always_ff`** statement that toggles (inverts) its one-bit output on each rising edge of the clock.

Exercise 17:



Identify the following in the diagram above: component names, component "instance names," component port names, module port names. Label the signal **t** in the schematic.

Exercise 18: Rewrite the **`ex60`** module using operators. Which version – "structural" or "behavioural" – is easier to understand?