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

`4'b1001`?

`5'd3`?

`6'h0_a`?

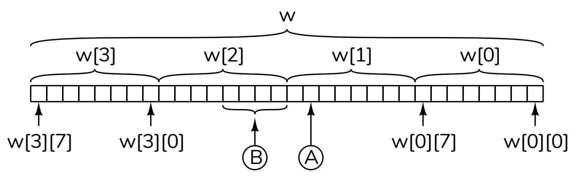
`3`?

Exercise 4: 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 5:



How would you specify the bit marked A in the diagram above?

The bits marked B?

The least-significant byte?

Exercise 6: What are the values of the following expressions: `!4'b010`?

`~4'b010`?

`0+~(!0)`?

Exercise 7: Use slicing and concatenation to compute the byte-swapped value of an array `n` declared as `logic [15:0] n`.

Exercise 8: If `n` has the value `16'h1234`, what is the value and width of:

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

Exercise 9: Use concatenation to shift **n** left by two bits.

Exercise 10: Use concatenation to assign the high-order byte of **n** to **a** and the low-order byte to **b**.

Exercise 11: 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 12: 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 13: 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 14:

```
// concatenation:
logic [3:0] x = { 2'b00, 2'b11 } ;

// array literal
logic [0:1] [3:0] z = '{ 2'b11, 3'b101 } ;
```

What are the dimensions and initial values of **x**, and **z** in the examples above?

Exercise 15: Write the truth table for a one-bit adder with 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**.

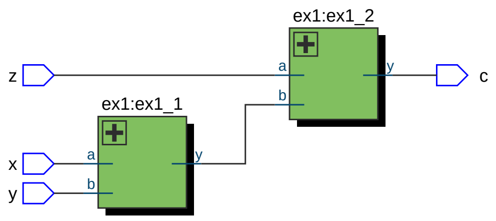
Exercise 16:

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

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

Exercise 17: Write an `always_ff` statement that toggles (inverts) its output on each rising edge of the clock.

Exercise 18:



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 19: Rewrite the `ex60` module using operators. Which version – “structural” or “behavioural” – is easier to understand?