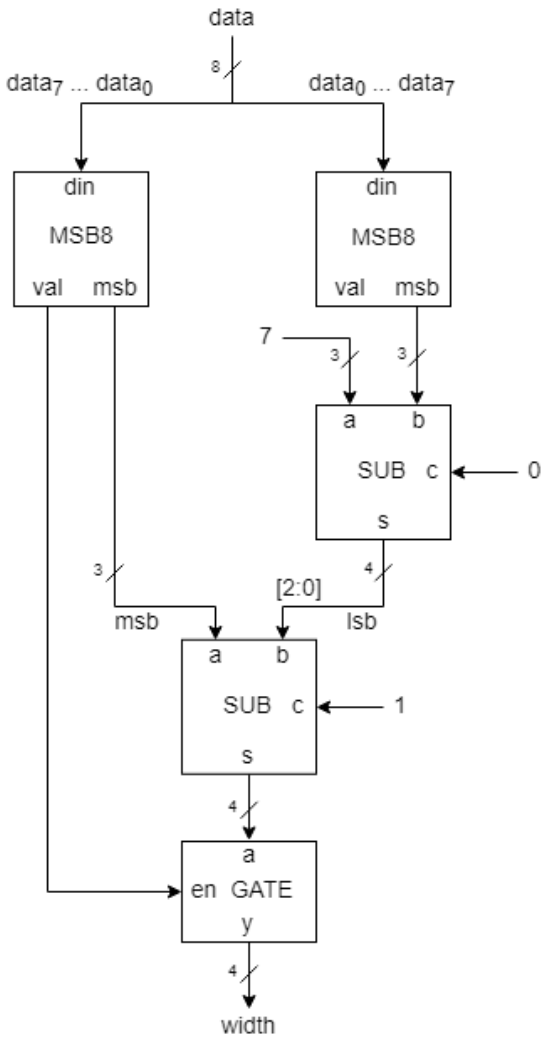
ASSIGNMENT 1

Description



This circuit, **width8**, computes the width of the substring between the most significant 1 (msb) and the least significant bit 1 (lsb) from the input 8 bit string. If there is not a single 1 in the input string, the width is 0.

Examples:

din	width
0 0 0 0 0 0 0 0	0
0 0 0 0 0 1 0 0	1
0 0 0 1 1 0 1 0	4
1 1 1 1 1 1 1 1	8

It uses two instances of **msb8**, a block that outputs the index of the most significant bit 1 from its input. If the input of the **msb8** block has no 1, its output is not valid (val = 0).

The left **msb8** instance outputs the msb index of the input data.

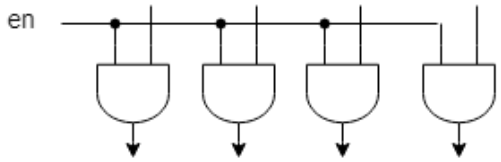
The right **msb8** instance has its input connected to the reflected data, such that data[0] is connected to din[7], data[1] to din[6], a.s.o. Its output is converted to the lsb index of the original data by subtracting it from 7. In the example above, if data is 00011010, its msb is 4, the msb of the reflected data, 01011000 is 6, and the lsb will be 1 = 7 – 6.

The second subtractor (of the same type, **sub**, as the first subtractor) computes the width according to the formula:

width = msb – lsb + 1

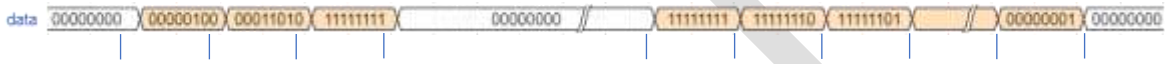
For the above example string the width will be 4 – 1 + 1 = 4

The final **gate** block ensures that the width is zero if there is no any 1 in the input data string. It's a simple array of gates:



Requirements

- The **sub** module is described behaviorally with a continuous assignment. Its c input is one bit wide, the other inputs have 3 bits, and the output has 4 bits.
- The **gate** module is described structurally with verilog gate primitives.
- The **msb8** module is described behaviorally using an always process for msb output (hint: use nested if-else statements starting with the leftmost bit of the input).
- The top-level design module, **width8**, is described structurally as in the given block schematics.
- Write a testbench for width, **width8_tb**, which instantiates the top-level design module with the name **dut**, and generates its input sequence as in the figure below:



The first 4 combinations are those from Examples, each one lasting 10 simulation steps, after which data resets to all 0. From simulation step 100, the input combination changes at each 10 simulation steps such that all combinations are presented to dut in the decreasing order of their numerical value. The sequence ends with all 0.

- All modules must have the names and pin names as in the list below. The widths of the pins are shown in the block schematics.

```
sub (a, b, c, s)
gate (en, a, y)
msb8 (din, val, msb)
width8 (data, width)
width8_tb
```

Grading

2 – sub

2 – gate

4 – msb8

7 – width8

4 – width8_tb

1 – coding style