Lab1:
Sorting Design

Goal

- Implement a Verilog module for a sorting network capable of arranging eight 4-bit unsigned numbers in descending order.
- Submit your Verilog module to E3 with Student ID (ie. Lab1_112550001.V)
- Deadline: 5/1(WED), 10:00 AM.
- No plagiarism !!
- Please remove all the delays before upload files to E3!

Requirements

Lab 1 Descriptions

Block diagram:

{ A[31:28], A[27:24], … ,A[3: 0]}

1st 2nd 8th

{ B[31:28], B[27:24], … ,B[3: 0]}

1st 2nd 8th

Input and Output format

- The input array A is packed with eight 4-bit unsigned numbers
- Each bit is between range of 0 to 16 (< 16 && >= 0)
- It will be better for you to unpack input while sorting
- Your output should be a descending array with eight 4-bit unsigned numbers
- Other format will not be accepted

Unpack Input

Unpack the input A into an array signal using the bit-field extraction operator(+: or -:) and a for-loop.

[<start_bit> +: <width>] // part-select increments from start-bit
[<start_bit> -: <width>] // part-select decrements from start-bit

eg:
assign tmp = A[0 +: 4]; // Same as A[3:0]
assign tmp = A[31 -: 4]; // Same as A[31:27]

Sorting

- All sorting algorithms are welcome
- Just make sure bits are arranged in descending order
- You can try to observe difference of different sorting algo. in vvp file
- Example:

```
function bubble_sort (array, length)
    var i, j;
    for (i from 0 to length - 1)
        for (j from 0 to length - 1 - i )
            if (array[ j ] < array[ j + 1 ])
                swap(array[ j ], array[ j + 1 ])
```

Time Delay

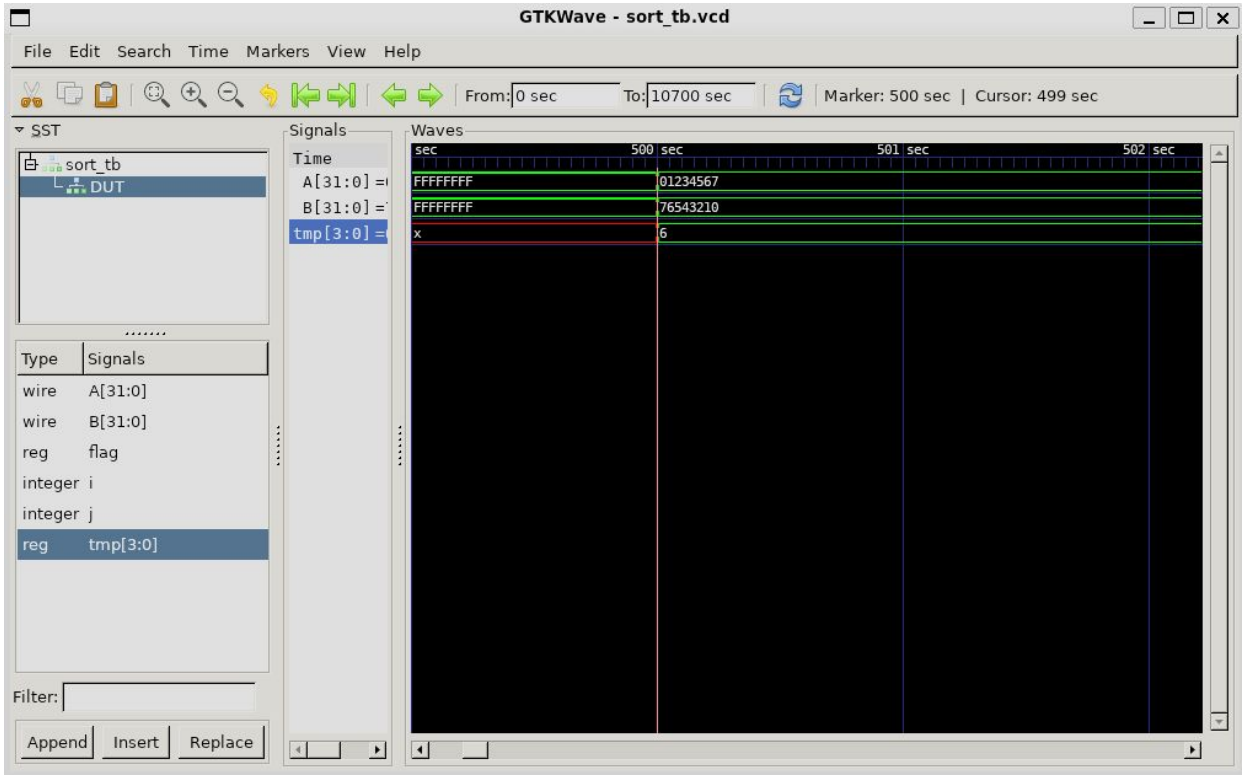
- Notice that output wire will not change instantly
- Adding time delay to “testbench” is necessary to have modified output

```
A = 32'b10001001101010111100101111101111;
#100;
$display("Output B: %b", B);
$finish;
```

Generate VCD File

- You can use provided testbench to check your answer
- Modify delay and input data as you want
- Hint:
  - iverilog -I include <your module> -o <output file> <testbench>
  - vvp <output file>
  - gtkwave <vcd file>

Example: Gtkwave output



You can observe the wave output while debugging

Notice

Declare the module as follows, and upload the sort() module and its supporting modules(if needed) to E3.

```
1   module sort(input [31:0] A, output [31:0] B);
2
3       reg [3:0] I[8:0], tmp;
4
5   /* Implement your design here. */
6
7       genvar k;
8       generate
9           for (k = 0; k < 8; k = k + 1) begin
10              assign B[(31-(k*4)) -: 4] = I[k];
11          end
12      endgenerate
13  endmodule
```

Submission format

{student_id}.v

Don’t upload the testbench module to E3!

References

- Bubble sort wiki
- bit-field extraction operator
- verilog tutorial