IC Design

Homework # 4

Due 2023/12/14, 9:00, Verilog code and report upload to NTUCool, no hardcopy

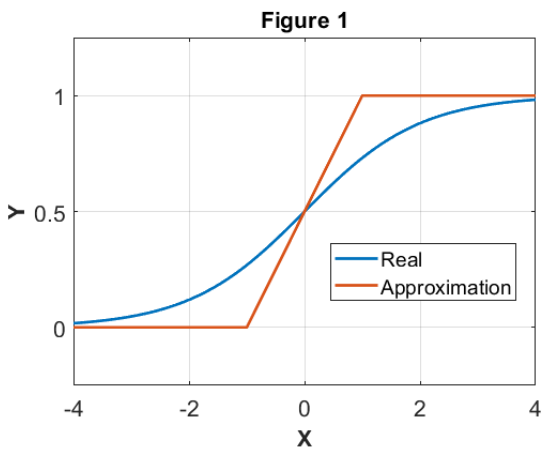
Plagiarism is not allowed. 10% penalty for each day of delay.

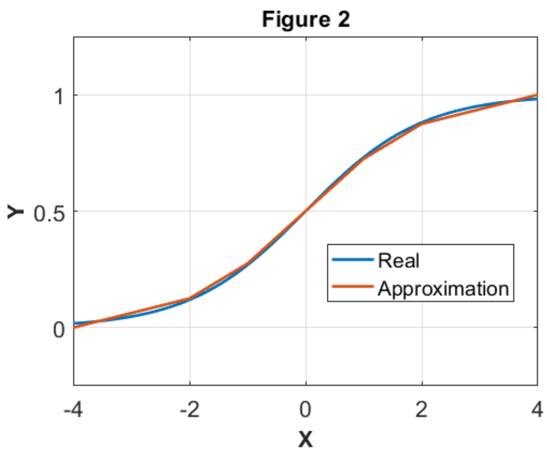
Problem Specification

Design a circuit with reset that computes the **approximation of the sigmoid function**. There are two input signals for the circuit, i.e., i_x with 8 bits, and a 1-bit i_in_valid. The i_x is a fixed-point format with a 1-bit sign, 2-bit integer, and 5-bit fraction. The circuit contains two output signals, i.e., up to 16-bit o_y for the approximation output value, and a 1-bit o_out_valid. The o_y is a fixed-point format with 1-bit integer and up to 15-bit fraction, and you don't need to use all 15 bits for your design's output. Note that **the input signal is signed,** and **the output signal is unsigned.** The relation between the input and the output signals is

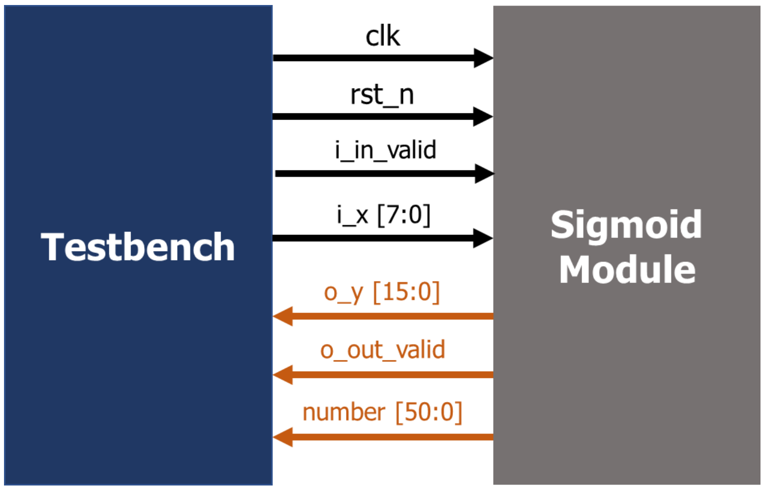
$$\mathrm{Y} \cong \mathrm{sigmoid}(\mathrm{x}) = \frac{1}{1+\mathrm{e}^{-\mathrm{X}}}$$

You are encouraged to use piecewise-linear approximation to design your circuit. For example, Figure 1 uses three segments to approximate the real function, while Figure 2 uses 5 segments. Figure 2 is more accurate than Figure 1, but its cost is using more transistors in the circuit because it needs to calculate more slopes. This is a tradeoff, so it is important to think about what your approximation function is before designing the circuit. Note that this homework only considers the range of [-4, 4).



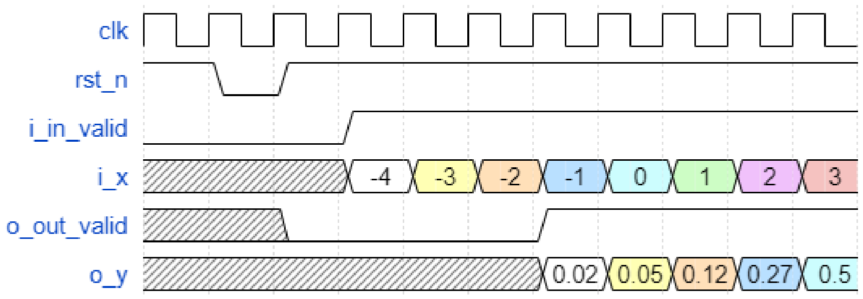


The following diagram illustrates the relation between your design and the testbench. Pull up "o_out_valid" and output your answer to "o_y[15:0]" once your circuit finishes the calculation, the testbench will then check your answers. Note that the output signals: "o_y[15:0]" and "o_out_valid" must be registered, i.e., they are outputs of DFFs **(use module FD2 (positive edge) in lib.v)**.



Timing Diagram

When your design is simulated in testbench, the circuit would only be reset once. After reset, **new values of i_x would be input at every cycle**. Besides, **o_out_valid should remain high once it is pulled up**, and the order of input i_x and their corresponding output o_y should not be changed (**First in, first out**.).



Signals Description

Signal name	I/O	Width	Simple description
clk	Input	1	Clock signal.
rst_n	Input	1	Active low asynchronous reset.
i_in_valid	Input	1	Indicate that the input is valid.
i_x	Input	8	Input X.
o_y	Output	16	Approximate output Y. Up to 15 bits fraction.
o_out_valid	Output	1	Indicate that the calculation was finished.
number	Output	51	The number of transistors.

Design Rules

Those who do not design according to the following rules will not be graded.

- **LUT-based designs are not allowed.**
- There should be a **reset signal** for the register.
- You are free to add pipeline registers.
- You can loosen your simulation timing first, (i.e., **`define CYCLE XXXX** in the tb.v), then shorten the clock period to find your critical path.
- Your design should be based on the **standard cells in the lib.v**. All logic operations in your design **MUST consist of the standard cells** instead of using the operands such as "+", "-", "&", "|", ">", and "<".
- Using "assign" to concatenate signals or specify constant values is allowed.
- Design your homework in the given "sigmoid.v" file. **You are NOT ALLOWED to change the filename and the header of the top module (i.e. the module name and the I/O ports)**.
- If your design contains more than one module, **don't create a new file for them**, just put those modules in "sigmoid.v."

Grading Policy

1. *Gate-level design using Verilog (70%)*

Your score will depend on both the approximation error and performance of your design.

(a) Approximation Error Score (35%)

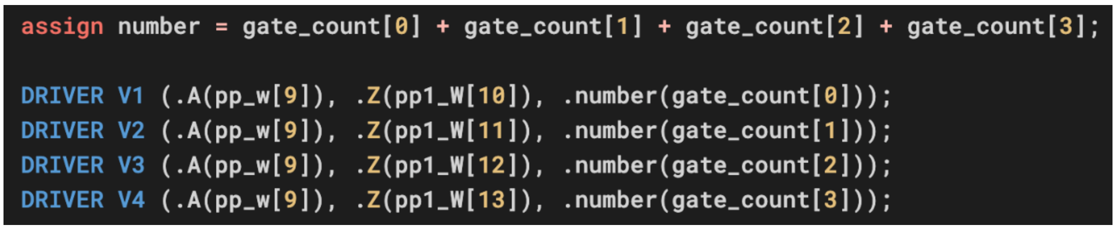
At this stage, we will evaluate how much the value of the sigmoid module approximates the real one. Time and area are not considered. We provide a testbench with a total of 256 patterns, whose distribution is in [-4, 4), and 16-bit format. There will be a ranking according to mean square error (MSE):

$$\sum_{i=0}^{255} \frac{(Y_{approximate,i} - Y_{real,i})^2}{256}.$$

(b) Performance Score (35%)

At this stage, you need to **add up the number of transistors of all used cells in the div module and connect it to number [50:0]**.

Only in this section, you are allowed to use "+" to help with calculations.



```
assign number = gate_count[0] + gate_count[1] + gate_count[2] + gate_count[3];

DRIVER V1 (.A(pp_w[9]), .Z(pp1_W[10]), .number(gate_count[0]));
DRIVER V2 (.A(pp_w[9]), .Z(pp1_W[11]), .number(gate_count[1]));
DRIVER V3 (.A(pp_w[9]), .Z(pp1_W[12]), .number(gate_count[2]));
DRIVER V4 (.A(pp_w[9]), .Z(pp1_W[13]), .number(gate_count[3]));
```

We will rank all students according to **A*T**, where A represents the **number of transistors** and T represents the **total execution time**.

Your approximation error score and performance score will be graded by your ranking according to the table below, respectively.

Percentage of students	**Approx. Error Score**	**Performance Score**
If your ranking is> 90 %	35	35
80% ~ 90%	32	32
70% ~ 80%	28	28
60% ~ 70%	25	25
50% ~ 60%	21	21
40% ~ 50%	18	18
30% ~ 40%	14	14
20% ~ 30%	11	11
10% ~ 20%	7	7
0% ~ 10%	4	4
Not using standard cell logic	0	0
Plagiarism	0	0
Incorrect number connection	0	0

2. *Report (30%)*

(a) Simulation (0%)

Specify your **minimum cycle time**. If you do not provide this information, **You will not get any score** for part 1 (70%)”**. This minimum cycle time would be verified by TAs.** Also, please put the screenshot of the summary provided by the testbench in the report.

(b) Circuit diagram (10%)

You are encouraged to use software to draw the architecture **instead of hand drawing**. Plot it **hierarchically** so that readers can understand your design easily. **All of the above will improve your report score.**

(5%) Plot the gate-level circuit diagram of your design.

(5%) Plot the critical path on the diagram above.

(c) Discussion (20%)

Discuss your design.

- (5%) Introduce your design.
- (5%) How do you improve your critical path and the number of transistors?

- (5%) How do you trade-off between area and speed?
- (5%) Compare with other architectures you have designed (if any).

Notification

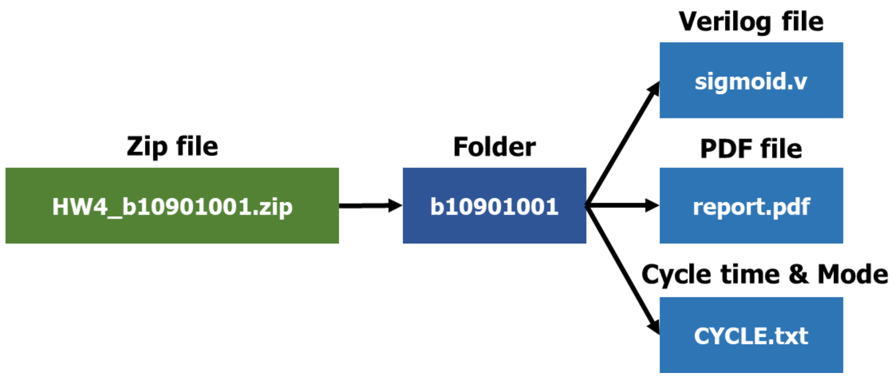
Following are the files HW4.zip includes

- **HW4_2023.pdf:** This document.
- **HW4_tutorial_2023.pdf:** Tutorial in class.
- **sigmoid.v:**
  Dummy design file. Program the design in this file.
  The header of the top module and the declaration of the I/O ports are predefined in this file and you are not allowed to change them.
- **lib.v:**
  Standard cells.
- **tb.v:**
  The testbench for your design.
- **pattern/Inn.dat:**
  Input patterns for the testbench. Please keep the hierarchy when simulation.
- **pattern/Gol.dat:**
  Patterns of the real hyperbolic tangent values for the testbench. Please keep the hierarchy when simulation.

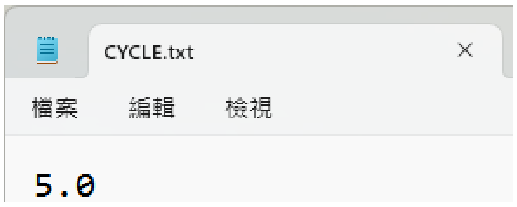
Submission

All students who do not submit files according to the rules **will get a 20% penalty.**

- You should upload a **zip file** to **NTUCool**, the file name is "HW4_Student ID", e.g.,   HW4_b10901001.zip
- Your file must conform to the following structure.



➢ Specify your cycle time in CYCLE.txt, e.g.,



```
5.0
```

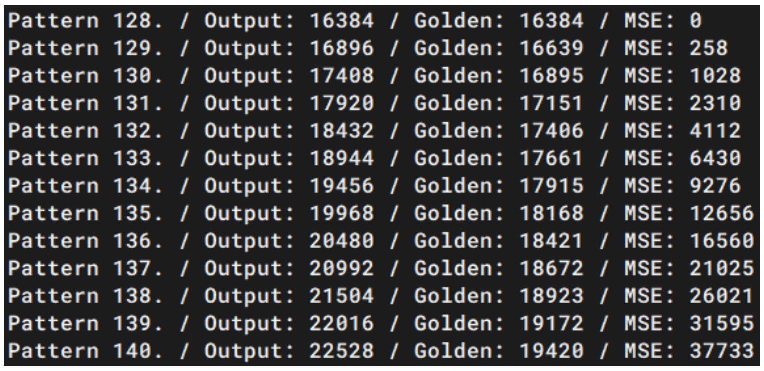
Testbench

1. Description

➢ The output waveform will be dumped to file “sigmoid.fsdb”, you can use nWave to examine it.

➢ You can change the number of test data to debug, but the final error will still test 256 data. (`**define PATTERN 256**)

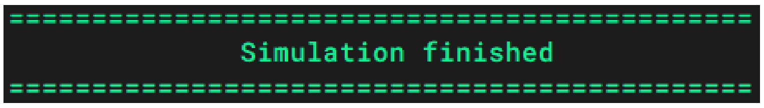
➢ You can enable the debug function, which will display the data received.



```
Pattern 128. / Output: 16384 / Golden: 16384 / MSE: 0
Pattern 129. / Output: 16896 / Golden: 16639 / MSE: 258
Pattern 130. / Output: 17408 / Golden: 16895 / MSE: 1028
Pattern 131. / Output: 17920 / Golden: 17151 / MSE: 2310
Pattern 132. / Output: 18432 / Golden: 17406 / MSE: 4112
Pattern 133. / Output: 18944 / Golden: 17661 / MSE: 6430
Pattern 134. / Output: 19456 / Golden: 17915 / MSE: 9276
Pattern 135. / Output: 19968 / Golden: 18168 / MSE: 12656
Pattern 136. / Output: 20480 / Golden: 18421 / MSE: 16560
Pattern 137. / Output: 20992 / Golden: 18672 / MSE: 21025
Pattern 138. / Output: 21504 / Golden: 18923 / MSE: 26021
Pattern 139. / Output: 22016 / Golden: 19172 / MSE: 31595
Pattern 140. / Output: 22528 / Golden: 19420 / MSE: 37733
```

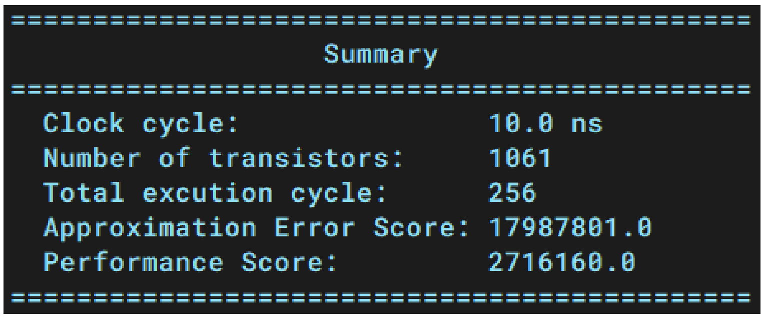
Note that the testbench discards the least 16 bits of the error in order to shrink the its length.

➢ If you pass the simulation, you should see:



```
============================================
            Simulation finished
============================================
```

You will also see the summary:



```
============================================
                 Summary
============================================
  Clock cycle:                 10.0 ns
  Number of transistors:       1061
  Total excution cycle:        256
  Approximation Error Score: 17987801.0
  Performance Score:           2716160.0
============================================
```

2. *Simulation Command*

```
> vcs tb.v sigmoid.v lib.v –full64 –R –debug_access+all +v2k
> vcs tb.v sigmoid.v lib.v –full64 –R –debug_access+all +v2k +define+DEBUG
```

If you have any questions or problems with this homework, feel free to contact TA.

TA: 陳丕全 Email: r11943013@ntu.edu.tw, EE2-329

HW4 Office hours: 12/12 Tue 14:00~16:00 @EE2-329
12/13 Wed 14:00~16:00 @EE2-329

If you are not available during office hours, you can email the TA to make an alternative appointment.