

(A)

```

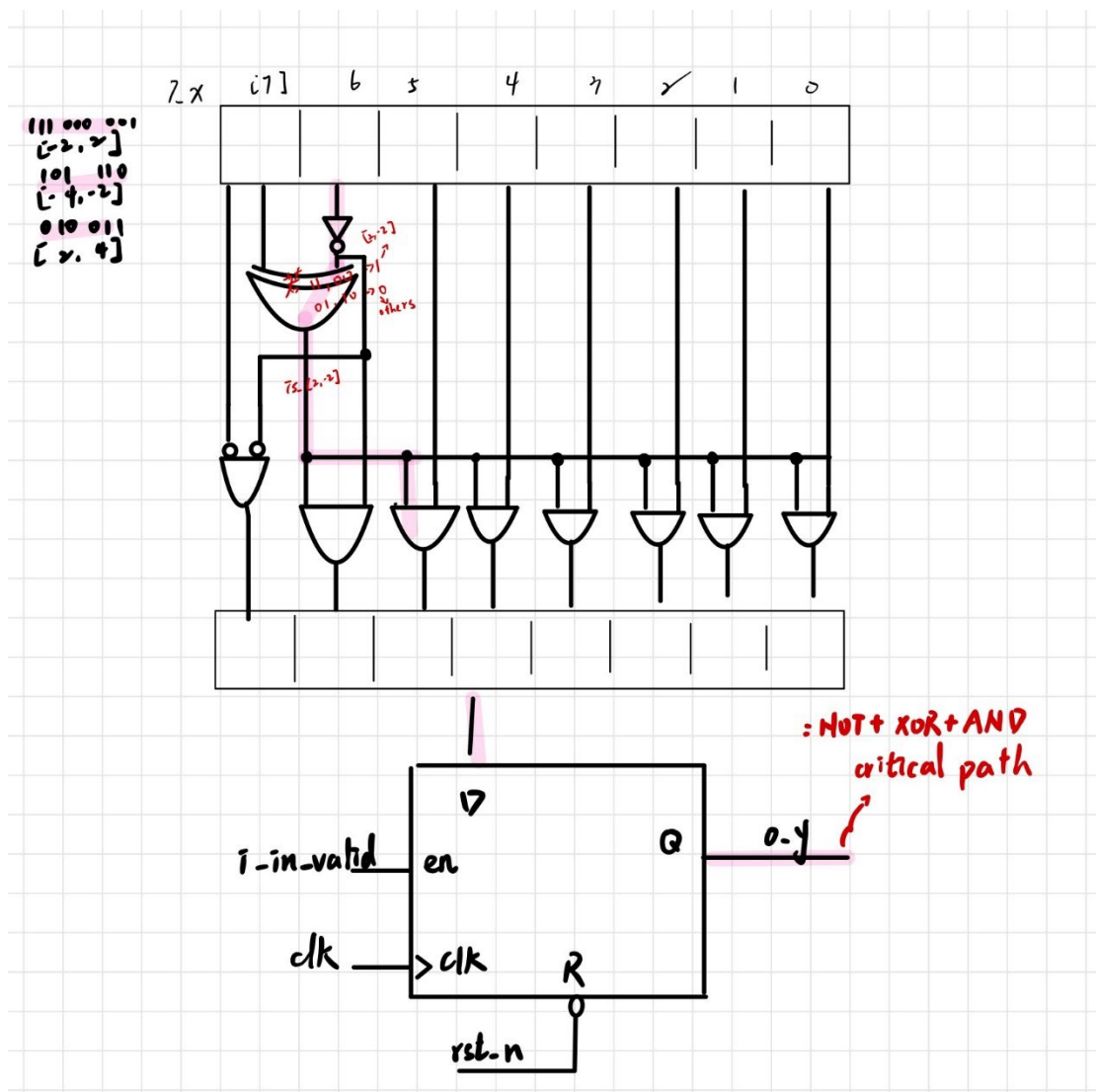
=====
Simulation finished
=====

Summary
=====
Clock cycle:           1.7 ns
Number of transistors:  357
Total excution cycle:  256
Approximation Error Score: 3284277.0
Performance Score:      155366.4
=====

$finish called from file "tb.v", line 131.
$finish at simulation time      442850
VCS Simulation Report
Time: 442850 ps
CPU Time:      0.430 seconds;      Data structure size:  0.0Mb
Mon Dec 11 16:13:47 2023
CPU time: .676 seconds to compile + .540 seconds to elab + .502 seconds to link +
b0505050@cad30 ~]$

```

(B)



(C)

1.

近似函數被分為三段

$$y = 0 \quad \text{for } -4 \leq x \leq -2$$

$$y = 1 \quad \text{for } -2 \leq x \leq 2$$

$$y = \frac{1}{4}x + 0.5 \quad \text{for } -2 \leq x \leq 2$$

由於此次作業會計算到 transistor 的數量，因此以避免使用 MUX、FA 等複雜的元件為主，且在判斷此三段時，分段的判斷也相對好解決，我們可以藉由檢查前兩位是否為相同(將一者 not 後 XOR = 1)，來確認該數是否屬於此範圍。而另外兩段則只需以前一判斷的結果與判定第一位為 1 或 0 便可以知曉。

而在輸出方面，由於斜率為  $\frac{1}{4}$ 、常數為 0.5，故我們得以用右移的方式進行，以達到避免使用乘法、加法器的結果。

2.

由於最開始分割線段的方式，我們得以在不使用複雜元件的情況下進行計算，這使得電路速度上升，而 critical path 最終也只有依序經過 NOT >XOR>AND>FF。

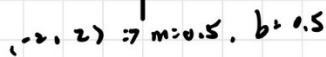
3.

在追求 transistor 數量與速度的情況下，我們須避免使用複雜的元件和盡量以平行計算為主，並試著在精準度與速度間取得平衡，而最後我選擇了設計輸出數值誤差較大的電路。不過雖然誤差較大，但換方面來說，transistor 的數量與電路計算速度也得到提升，而且由於自己本身從一開始就沒有打算用提升精度的方式來拿分數，所以設計的過程並無太多的取捨。

4.

原本是打算依照斜率的變化值來將線段切割為五段的，但後來由於時間與實行上的困難，且在大概思考了一下此電路設計的 critical path 和 transistor 數量後，發現沒有想出更好的簡化版電路，最終還是決定採用目前的方式進行。

range:  $[-4, 4]$

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$[-1, 1] \quad m = \frac{0.46 + 1191572}{1 - (-1)} = 0.2710585786 \quad b = 0.5 \quad y = 0.2710585786x + 0.5$$

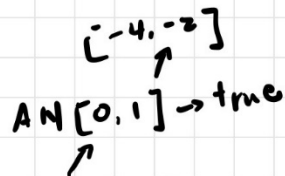
$$[1, 2], \quad m = \frac{0.880 \dots - 0.7 \dots}{2 - 1} = 0.1497815798 \quad b = 0.471815798 \quad y = 0.1497815798x + 0.471815798$$

$$[2, 1], \quad m = \frac{0.98 \times 10^{11} - 0.88 \times 10^{18}}{4 - 3} = 0.101 \times 10^{17} \quad y = 0.101 \times 10^{17} x + 0.678 \times 10^{25}$$

$$[-1, 2] \quad m = 0.1497384994 \quad b = 0.4186799208 \quad y = 0.1497384994x + 0.4186799208$$

$$[-2, -4] \quad m = 0.101 \times 1.6712 \quad b = 0.921676322 \quad y = 0.101 \times 1.6712 X + 0.921676322$$

Transfer float into n-bits



$$\begin{cases} [-4, -2] \\ [-2, 2] \\ [2, 4] \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$