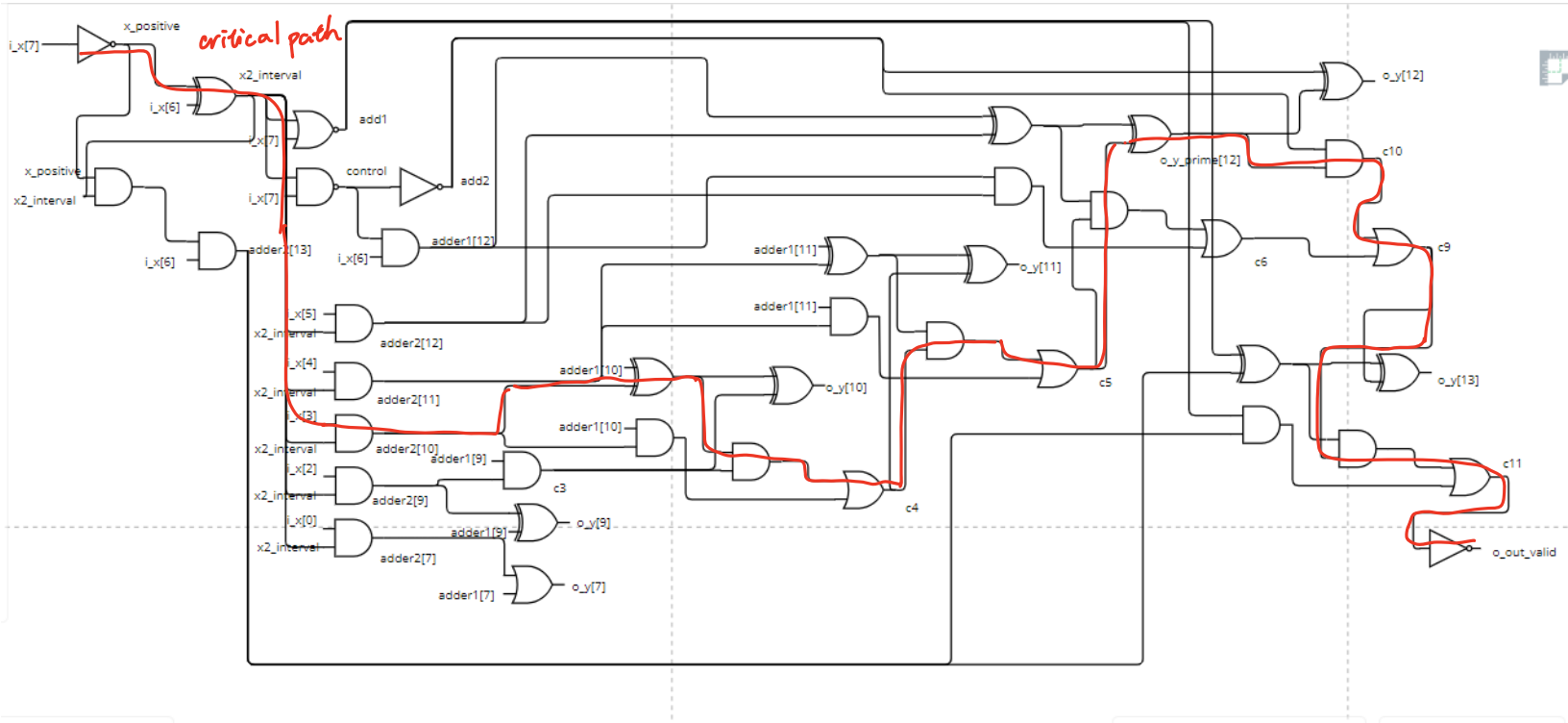
(a) the minimum cycle time is 7.4ns

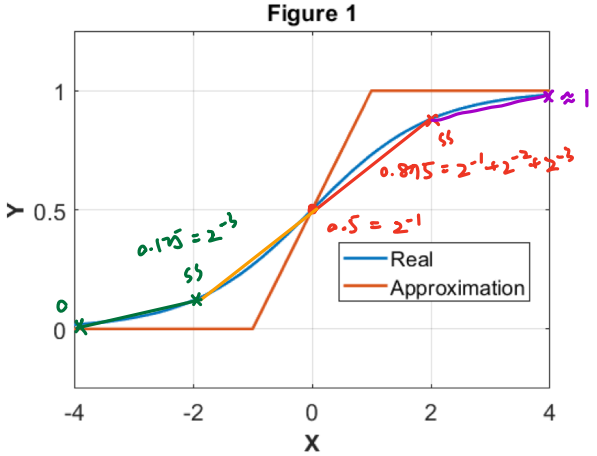
```
==========================================
            Simulation finished
==========================================

==========================================
                 Summary
==========================================
  Clock cycle:                7.4 ns
  Number of transistors:      210
  Total excution cycle:       255
  Approximation Error Score:  662950.0
  Performance Score:          396270.0
==========================================
```

(b)



(C)



use piecewise linear approximation (4 segments)

- slope: 2^{-4} ($y = 2^{-4}x + \underbrace{2^{-2}}_{add1} + 2^{-1}$) $2 \le x < 4$
- slope: $2^{-3}+2^{-4}$ ($y = 2^{-4}x + 2^{-3}x + 2^{-1}$) $0 \le x < 2$
- slope: $2^{-3}+2^{-4}$ ($y = 2^{-4}x + 2^{-3}x + \underbrace{2^{-1}}_{add2}$) $-2 < x < 0$
- slope: 2^{-4} ($y = 2^{-4}x$) $-4 < x \le -2$

(1) for $x \geq 0$, x_positive = 1, o_y[14] = x_positive = 1 ($y \geq 0.5$)

(2) for $-2 < x < 2$, x2_interval = 1, indicates that the slope will be $2^{-3} + 2^{-4}$ (else slope = 2^{-4})

(3) for $2 \leq x < 4$ (x_positive = 1 and x_2_interval = 0), I need to add 2^{-2} (add1 = 1, else add1 = 0)

(4) for $-2 < x < 0$ (x_positive = 0 and x_2_interval = 1), I need to add 2^{-1} (add2 = 1, else add2 = 0)

∵ $x < 0$ is 2's complement ∴ for $-2 < x < 0$, the equation $y = 2^{-4}x + 2^{-3}x + 2^{-1}$, x only use i_x[5:0]

and we use control to represent $x > 0$ or $x \leq -2$ (x can use i_x[6:0])

(5) we use adder1[15:0] to represent $2^{-4} \cdot x$. adder2[15:0] to represent $2^{-3}x$

(6) according to different condition, use (1) ~ (5), half adder, full adder to complete the circuit *

To improve the critical path and the number of transistors, we use NAND, NOR gates instead of AND, OR gates. Besides, I sacrifice the accuracy to improve both the critical path and the number of transistors. (give up using full adder in LSB calculation)

HA1 ha1 (c1, o_y[1], adder1[1], adder2[1])

FA1 fa2 (c_2, o_y[8], adder1[8], adder2[8], c_1)

FA1 fa3 (c_3, o_y[9], adder1[9], adder2[9], c_2)

↓ after trade-off with accuracy

OR2 or5 (o_y[1], adder1[1], adder2[1])

assign o_y[8] = adder1[8]

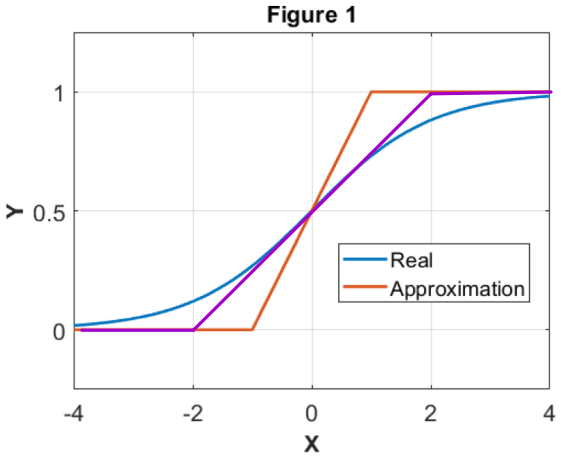
HA1 ha2 (c_3, o_y[9], adder1[9], adder2[9]) *

Carry lookahead Adder
(used in $2^{-3}x + 2^{-4}x$)
more area, less delay

after trade-off between area and speed ⟺

Ripple Carry Adder
(used in $2^{-3}x + 2^{-4}x$)
less area, more delay *

original design



three segments

higher approximation error

lower critical path and number of transistors ☆