Sys-Verilog Questions Review

Some Solutions to questions from ChipIO-Dev

Counter



Counter



Router



Connect (wire)









Recursion or calling the same hardware segment repeatedly

Log2(**5**)

Minimum number of bits represent 5?

$$Min = 2$$

$$5 > 2^2$$
 (increment) ____ ___

Log2(**4**)

Minimum number of bits represent 5?

$$Min = 2$$

$$4 \not< 2^2$$
 (perfect)

Log2 : Debug Results



Second Largest





Count	0	1	2	3	4	
Data_In	DO	D1	D2	D3	D4	
2nd Largest	0	0	2	2	2	3

-, **2** 3, **2** 3, **2** 3, **2** 7, **3**

Rounded Division





Generate Logic Blocks



Gray code

Vertical Delay:) $2^3 = 8cycle$ $2^2 = 4cycle$ $2^1 = 2cycle$ $2^0 = 1cycle$

$$2^3 = 8cycle$$

$$2^2 = 4cycle$$

$$2^1 = 2cycle$$

$$2^0 = 1$$
cycle

0	0	0	O
O	O	O	1
O	O	1	1
O	O	1	O
O	1	1	O



Parralel In -Serial Out



Serial to Parallel



Serial to Parallel (Simulation Concept)





Fibonacci



Count Ones

Architecture Similar to Linked List Gen Din_0 adderz $wire_0$ equal_width Gen Din_1 Din adderz data_width $wire_1$ equal_width $wire_{15}$ Dout (i.e. Count) adderz equal_width equal_width = log2(data_width)

Count Ones

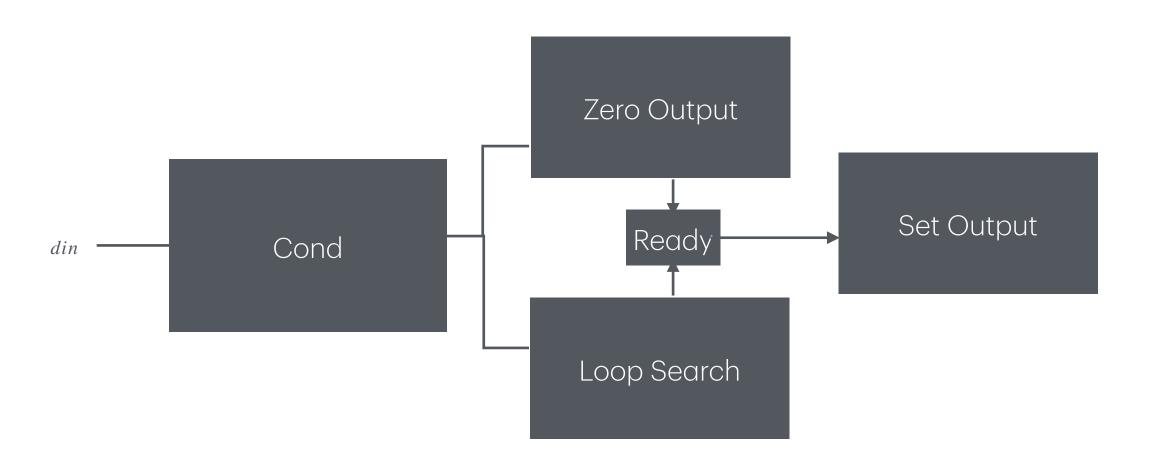
```
[2025-10-21 23:50:16 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv && unbuffer vvp a.out index - 0 input - 3 n_ones - 2 index - 1 input - 5 n_ones - 2 index - 2 input - 8 n_ones - 1 testbench.sv:44: $finish called at 9 (1s)

Done
```

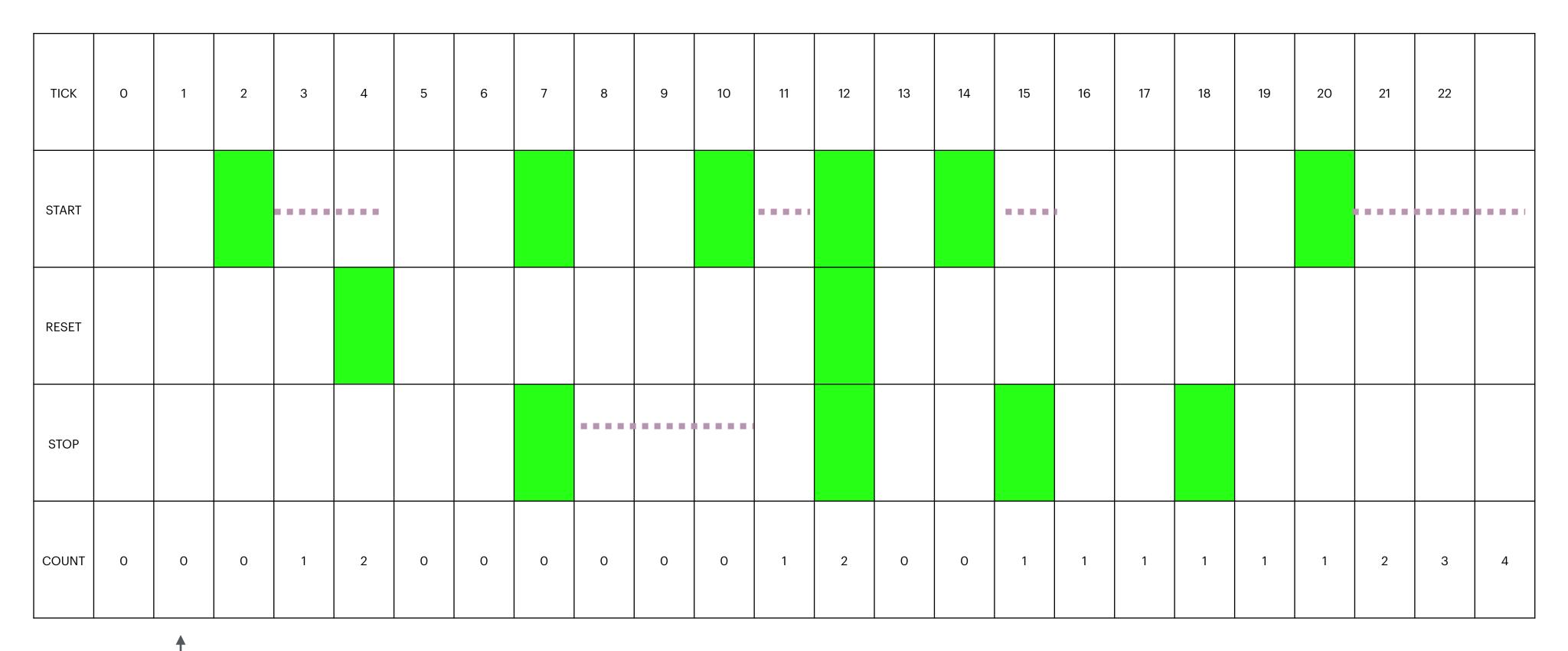
Gray Code to Binary (Width = 3)



Trailing Ones



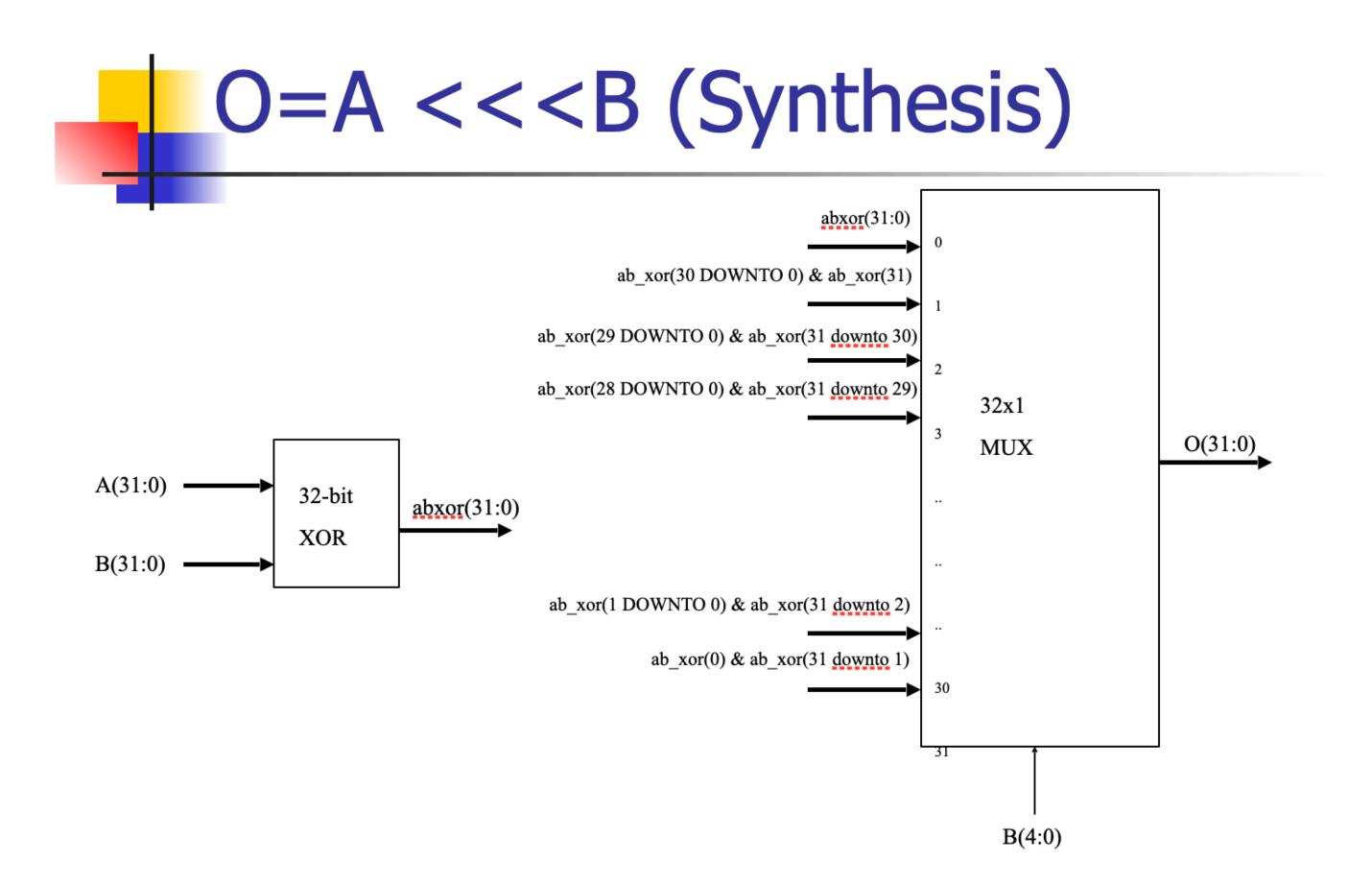
StopWatch Timer



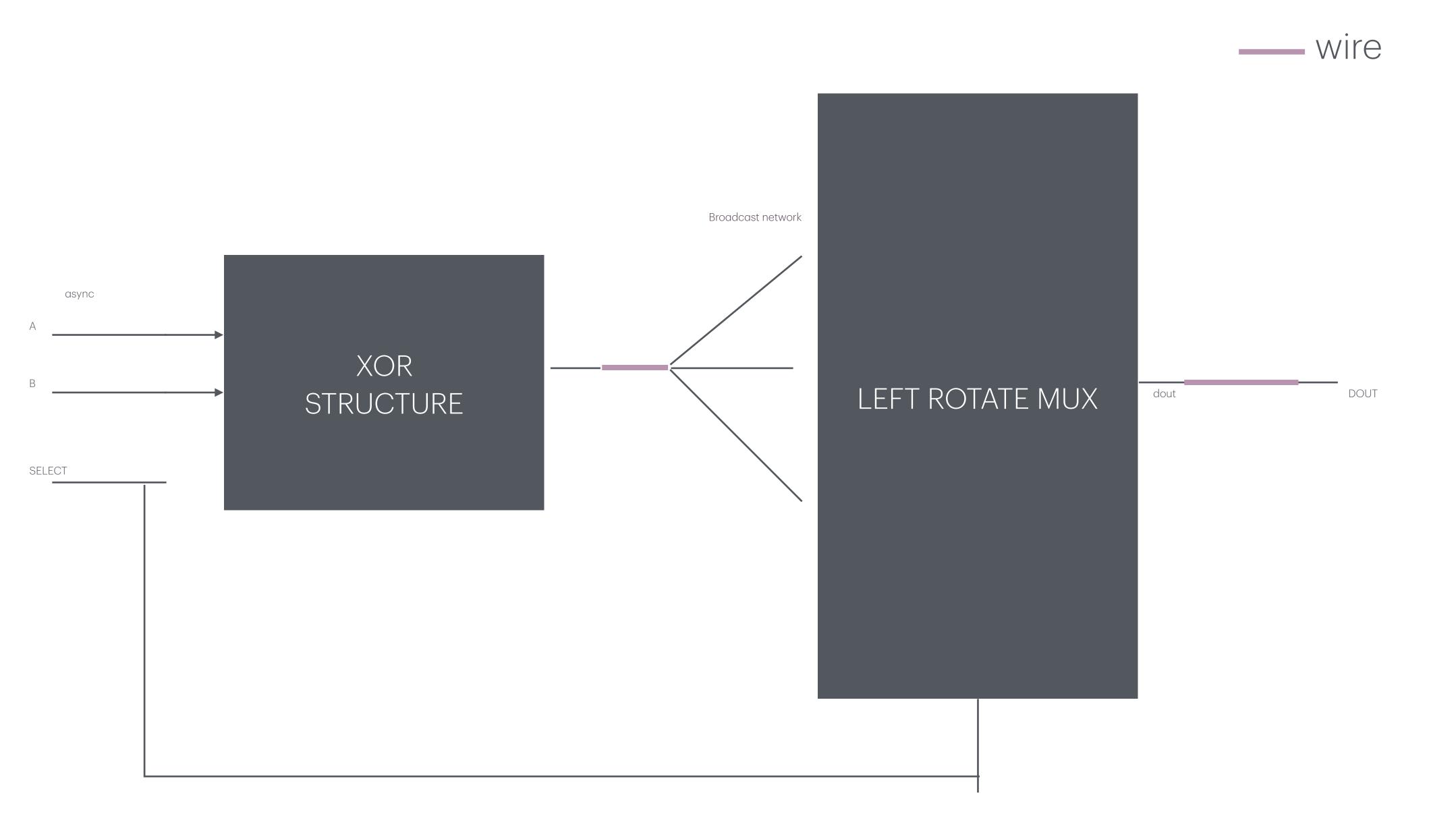




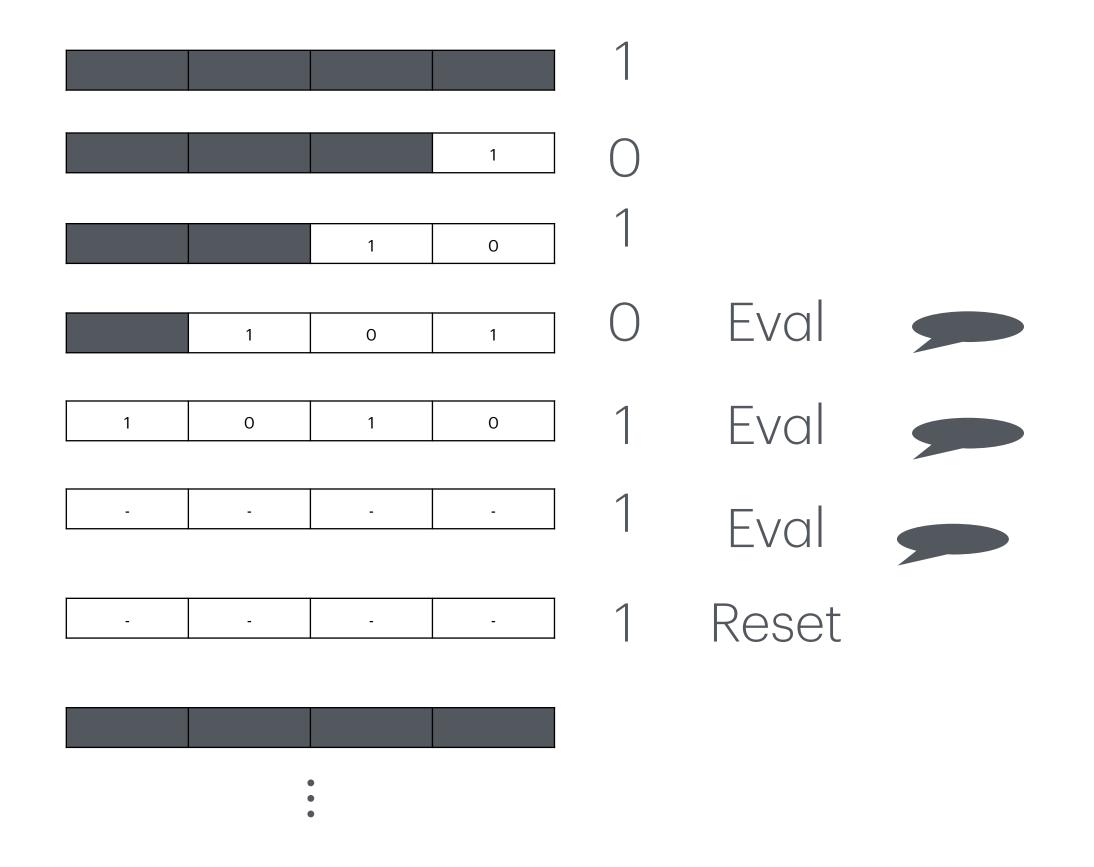
```
[2025-10-23 19:15:00 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv && unbuffer vvp a.out
prev_button
              0] Button Press: reset[0] start[0] stop[0]
                                                           return -
prev_button Event
              1] Button Press: reset[0] start[0] stop[0]
                                                           return -
              2] Button Press: reset[0] start[1] stop[0]
                                                           return -
                 Button Press: reset[0] start[0] stop[0]
                                                           return -
              4] Button Press: reset[1] start[0] stop[0]
                                                           return -
              5] Button Press: reset[0] start[0] stop[0]
                                                           return -
              6] Button Press: reset[0] start[0] stop[0]
                                                           return -
              7] Button Press: reset[0] start[1] stop[1]
                                                           return -
              8] Button Press: reset[0] start[0] stop[0]
                                                           return -
              9] Button Press: reset[0] start[0] stop[0]
                                                           return -
           [ 10] Button Press: reset[0] start[1] stop[0]
                                                           return -
           [ 11] Button Press: reset[0] start[0] stop[0]
                                                           return -
           [ 12] Button Press: reset[1] start[1] stop[1]
                                                           return -
           [ 13] Button Press: reset[0] start[0] stop[0]
                                                           return -
           [ 14] Button Press: reset[0] start[1] stop[0]
                                                           return -
           [ 15] Button Press: reset[0] start[0] stop[1]
                                                           return -
           [ 16] Button Press: reset[0] start[0] stop[0]
                                                           return -
           [ 17] Button Press: reset[0] start[0] stop[0]
                                                           return -
           [ 18] Button Press: reset[0] start[0] stop[1]
                                                           return -
           [ 19] Button Press: reset[0] start[0] stop[0]
                                                           return -
           [ 20] Button Press: reset[0] start[1] stop[0]
                                                           return -
           [ 21] Button Press: reset[0] start[0] stop[0] return -
           [ 22] Button Press: reset[0] start[0] stop[0] return -
           [ 23] Button Press: reset[0] start[0] stop[0] return -
           testbench.sv:136: $finish called at 114000 (1ps)
           Done
```



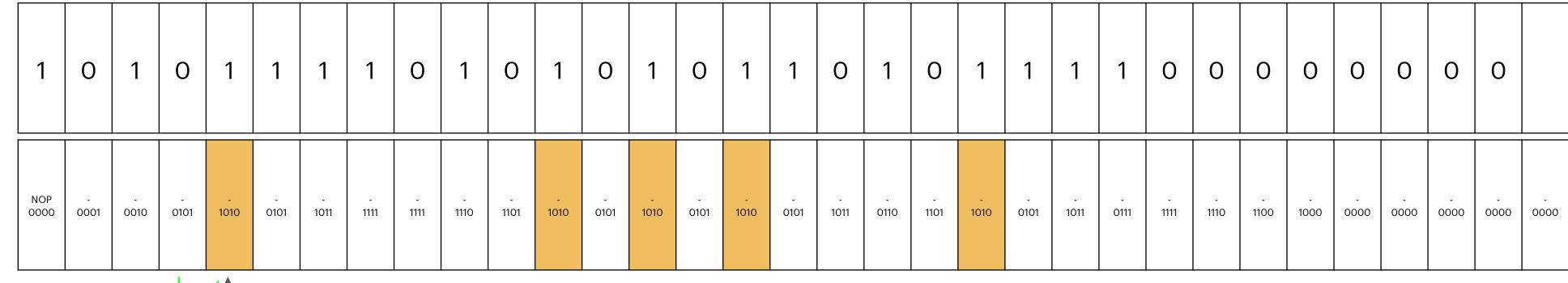
Circuit from OLD grad school slides:). Might as well build it in SV



Sequence Detector



Sequence Detector

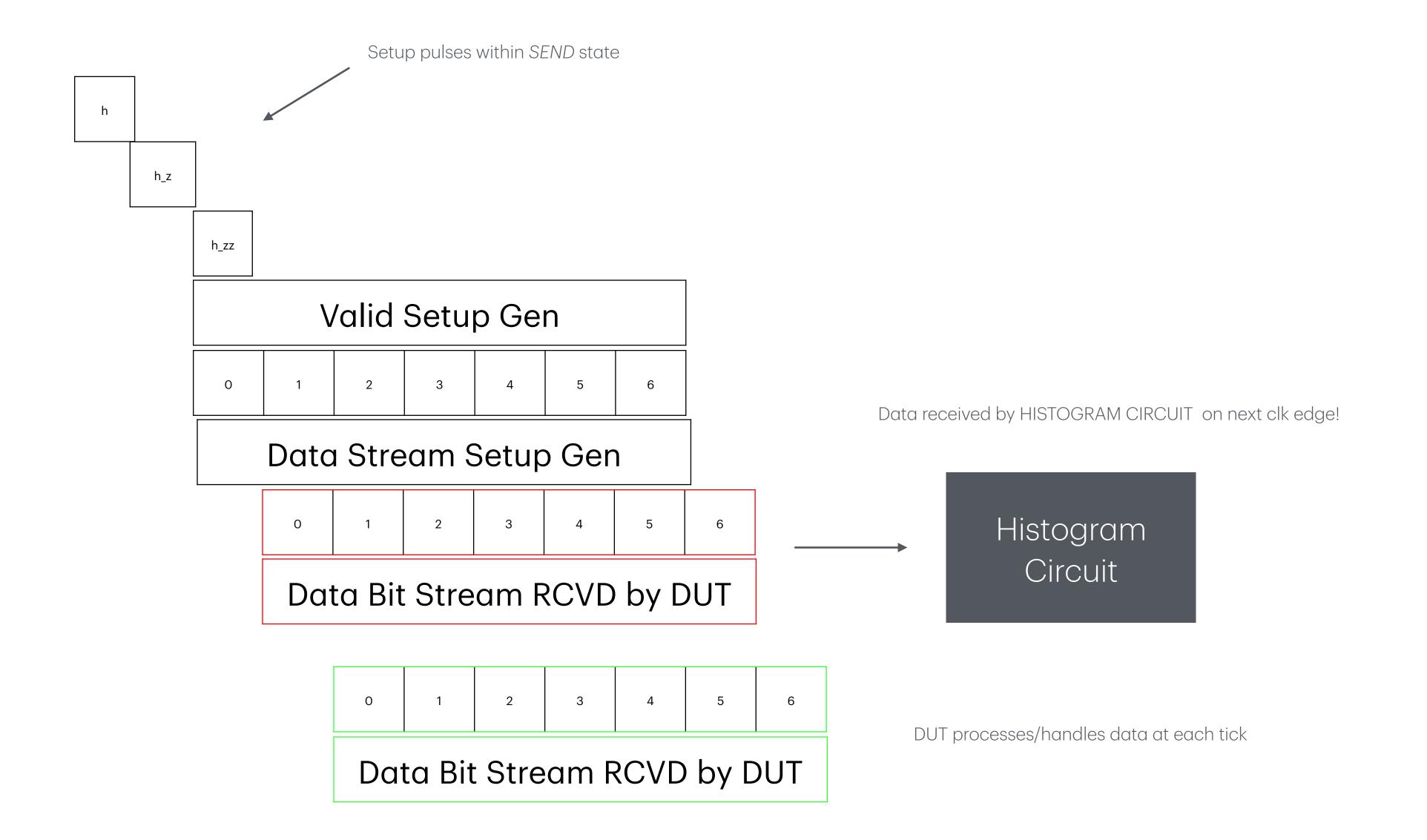


FIFO FULL

Evaluations of sequence are valid

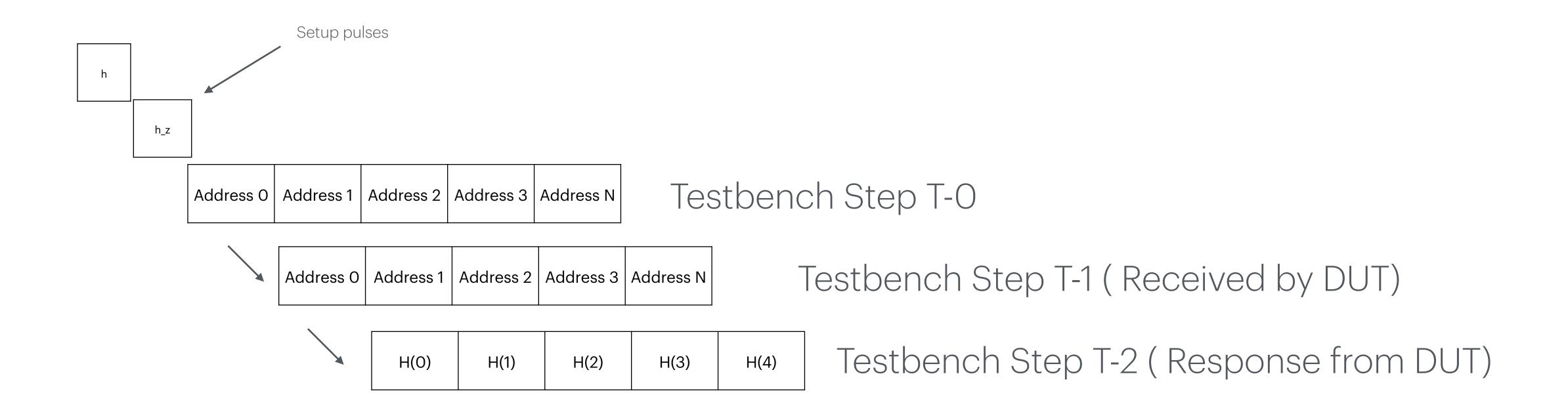
Histo Amazon

Send Mechanism



Histo Amazon

Receive Mechanism (Test Address)



Divisible By Three

