黃偉祥 X1136010

Limitations:

VDD = 1.8v
L = 0.18um
W >= 0.25um

Settings:
pFET(w=0.5u),nFET(w=0.25u), .tem=30
h1.sp : .tran 0.01n 260n
h1_bonus.sp : .tran 0.01n 520n

How to perform the simulation

1. Enter the command below to switch to hspice.chsrc, it will give you the hspice version
   - source /usr/cad/synopsys/CIC/hspice.cshrc
2. Use hspice to compile your .sh file and output as .lis file
   - hspice -i input.sh -o output.lis
3. If the last line showed hspice job concluded then you are good to go and check the waveview. But if showed aborted or something else then you need to check the error message in .lis file

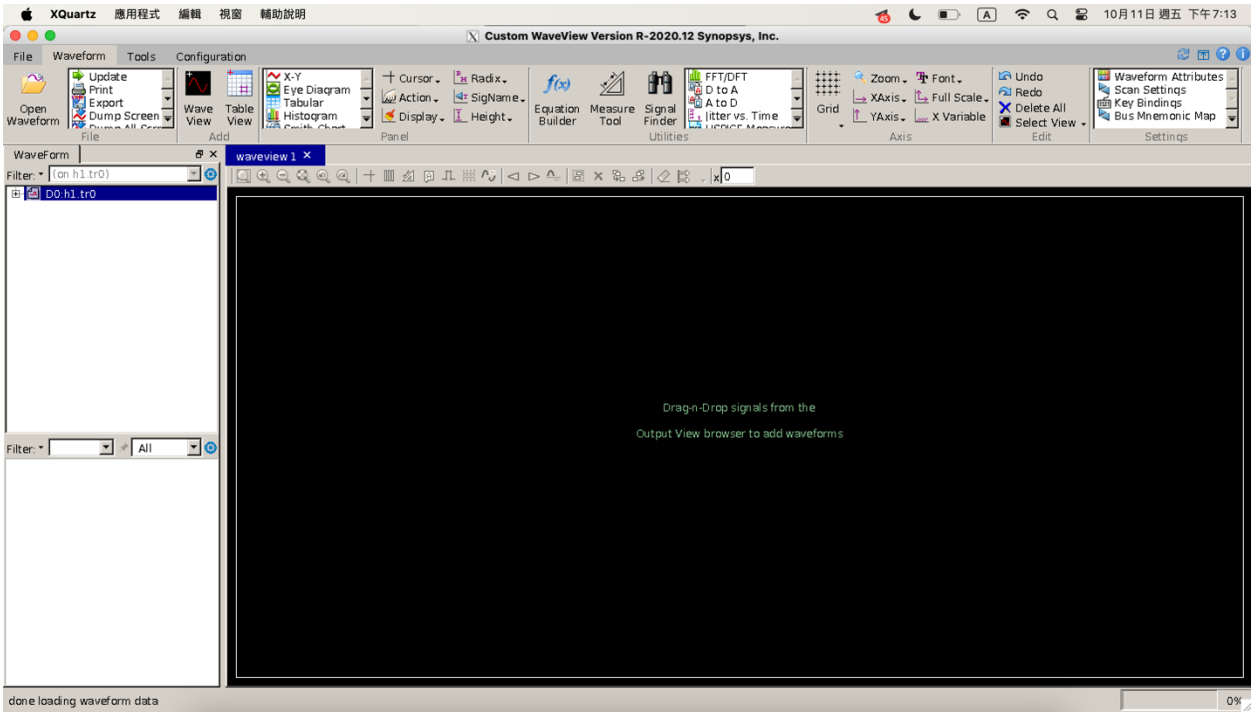
For example:

```
[tsx1136010@linuxcad30 ~]$ source /usr/cad/synopsys/CIC/hspice.cshrc
set hspice version: 2020.12 (default)
[tsx1136010@linuxcad30 ~]$ hspice -i h1
h1.sp        h1/          h1_bonus.sp
[tsx1136010@linuxcad30 ~]$ hspice -i h1.sp -o ./h1/h1.lis
Using: /home/tools/synopsys/hspice/2020.12/hspice/linux64/hspice  -i 'h1.sp' -o
./h1/h1.lis

>info:         ***** hspice job concluded
```

4. Enter the command below to switch to source customexplorer.cshrc, it will give you customerexplorer version
   - source /usr/cad/synopsys/CIC/customexplorer.cshrc
5. Use custom waveview to check your result wave, using wv target_file.tr0
   - wv file.tr0

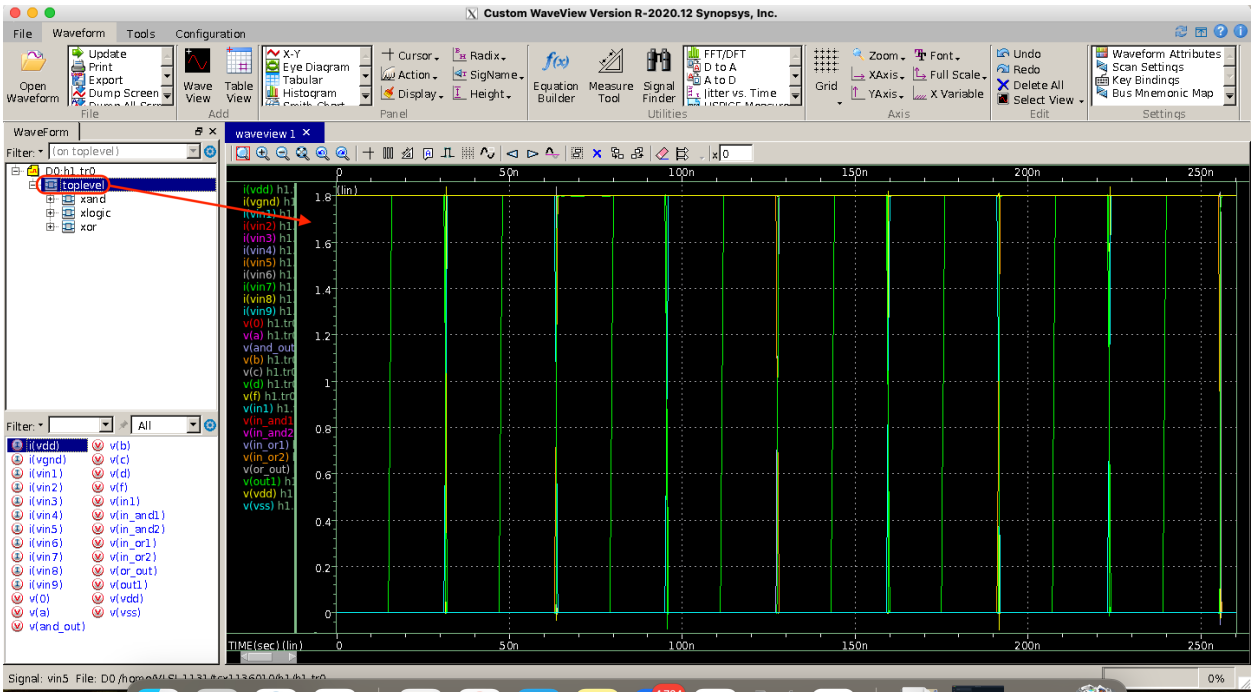
For example:

```
[tsx1136010@linuxcad30 ~]$ !76
source /usr/cad/synopsys/CIC/customexplorer.cshrc
set customexplorer version: 2020.12 (default)
[tsx1136010@linuxcad30 ~]$ ls
bonus  h1  h1_bonus.sp  h1.sp  mylib.sp
[tsx1136010@linuxcad30 ~]$ cd h1
[tsx1136010@linuxcad30 ~/h1]$ ls
h1.ic0  h1.st0   inv.lis  inv.tr0      sxcmd.log.2
h1.lis  h1.tr0   inv.pa0  sxcmd.log    sxcmd.log.3
h1.pa0  inv.ic0  inv.st0  sxcmd.log.1  sx_crash.sx
[tsx1136010@linuxcad30 ~/h1]$ wv h1.tr0
```



6. Drag the thing into waveview window then you can see all wave of your program, you can delete the useless wave and ungroup some of them to have a clean view.

For example:

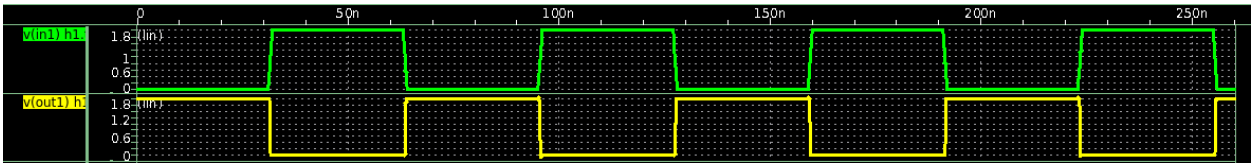


The completion of the assignment

- Done all including bonus

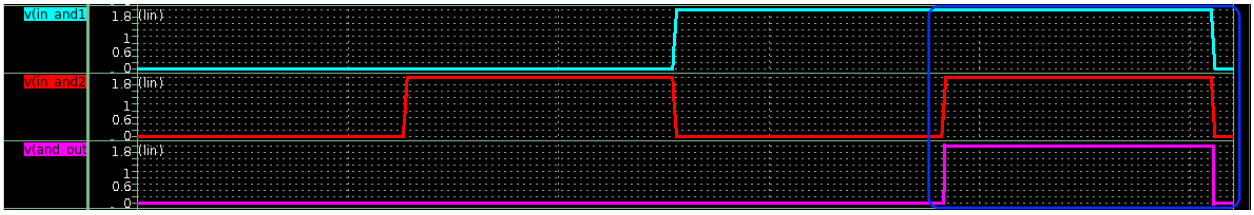
The waveform of OR gate, AND gate, inverter, and the specific logic function

1. Inverter waveform



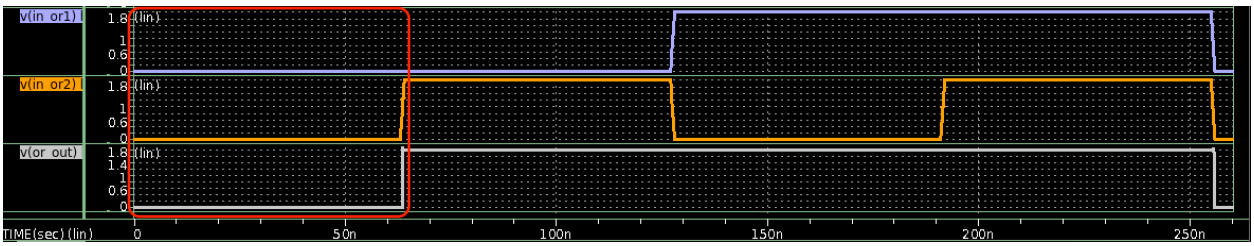
- Output is the inverse of input

2. AND2 gate waveform



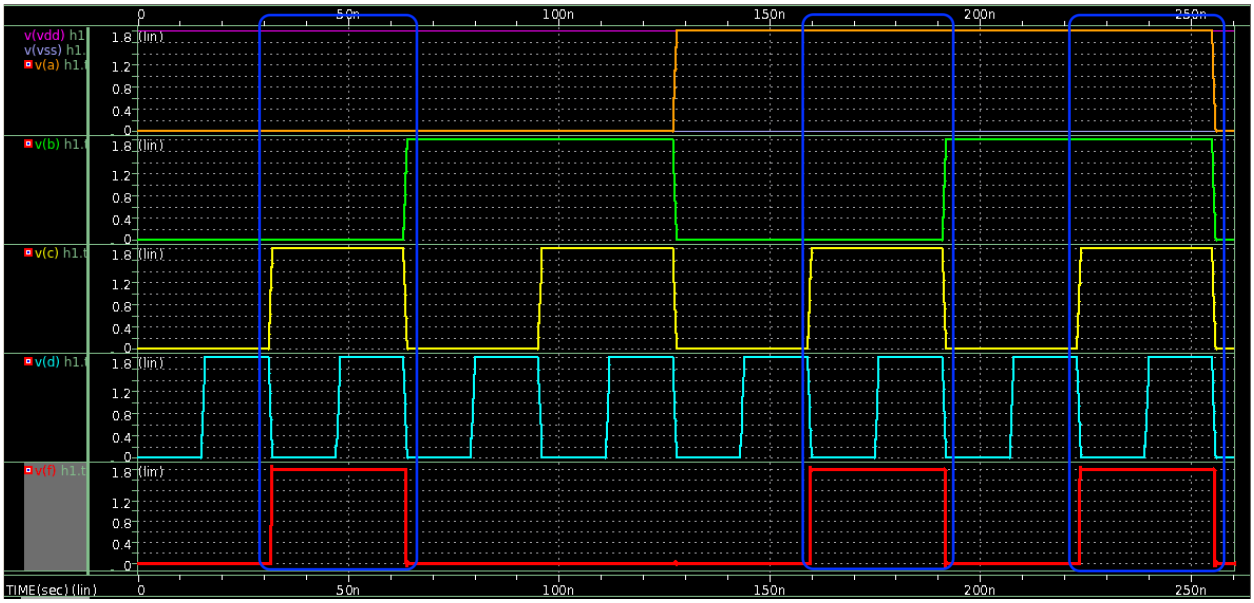
- We can see the output will be high only when both inputs are high, which is the AND gate function.

3. OR2 gate waveform



- We can see the output will be low only when both inputs are low, which is the OR gate function.

4. The specific logic function



a	b	c	(c ∧ (¬b ∨ a))
F	F	F	F
F	F	T	T
F	T	F	F
F	T	T	F
T	F	F	F
T	F	T	T
T	T	F	F
T	T	T	T

a	b	c	d	(((d ∧ ¬(b ∨ ¬c)) ∨ (c ∧ (¬b ∨ a)))
F	F	F	F	F
F	F	F	T	F
F	F	T	F	T
F	F	T	T	T
F	T	F	F	F
F	T	F	T	F
F	T	T	F	F
F	T	T	T	F
T	F	F	F	F
T	F	F	T	F
T	F	T	F	T
T	F	T	T	T
T	T	F	F	F
T	T	F	T	F
T	T	T	F	T
T	T	T	T	T

- From the true table, we can know that output will be high only when a,b,c = {(0,0,1),(1,0,1),(1,1,1)}
  - True table from Truth Table Generator (https://web.stanford.edu/class/cs103/tools/truth-table-tool/)

This is my prove:

$d + ab = a$

$a(1 + b) = a$

$$F = d\cdot\overline{(b+\bar{c})} + c(\bar{b}+a)$$
$$= d\cdot(\bar{b}\cdot\bar{\bar{c}}) + c(\bar{b}+a)$$
$$= d\cdot\bar{b}\cdot c + c\bar{b} + ac$$
$$= c(d\bar{b} + \bar{b} + a)$$
$$= c(\bar{b}+a)$$

The hardness of this assignment and how you overcame it

1. Mac m1 cannot use MobaXterm and default mac terminal can only connect to workstation but cannot open waveview

```
[tsx1136010@linuxcad30 ~/h1]$ source /usr/cad/synopsys/CIC/h
hercules.cshrc* hsimplus.cshrc* hspice.cshrc*
[tsx1136010@linuxcad30 ~/h1]$ source /usr/cad/synopsys/CIC/hspice.cshrc
set hspice version: 2020.12 (default)
[tsx1136010@linuxcad30 ~/h1]$ hspice -i h1.sp -o h1.lis
Using: /home/tools/synopsys/hspice/2020.12/hspice/linux64/hspice  -i 'h1.sp' -o h1.lis

>info:         ***** hspice job concluded
[tsx1136010@linuxcad30 ~/h1]$ source /usr/cad/synopsys/CIC/cus
customcompiler.cshrc* customexplorer.cshrc*
[tsx1136010@linuxcad30 ~/h1]$ source /usr/cad/synopsys/CIC/customexplorer.cshrc
set customexplorer version: 2020.12 (default)
[tsx1136010@linuxcad30 ~/h1]$ wv h1.tr0 &
[1] 21407
[tsx1136010@linuxcad30 ~/h1]$ Fatal: QXcbConnection: Could not connect to display  (:0, )
```

   - I found an alternative software for mac m1 which is XQuartz (https://www.xquartz.org/releases/XQuartz-2.8.5.html)

     i. ssh -X ts_stuID@server_ip -p 22

2. Logic function too complicated, so I simplified the logic function.

$$a + ab = a$$

$$a(1 + b) = a$$

$$\bar{F} = d \cdot \overline{(b + \bar{c})} + c(\bar{b} + a)$$

$$= d \cdot (\bar{b} \cdot \bar{\bar{c}}) + c(\bar{b} + a)$$

$$= d \cdot \bar{b} \cdot c + c\bar{b} + ac$$

$$= c(d\bar{b} + \bar{b} + a)$$

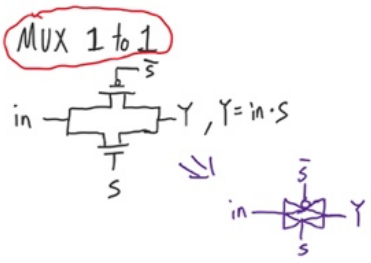
$$= c(\bar{b} + a)$$

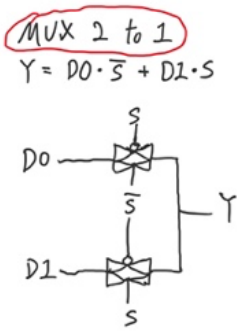
3. When I'm doing bonus I don't want to rewrite all the basic gates again, so I copy "h1.sp" without the simulation part as "mylib.sp".

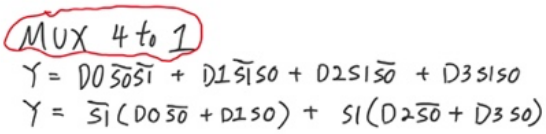
```
(base) wongweixiang@Wongs-MacBook-Air circuit_hw1 % cat mylib.sp
** X1136010_HW1 **
** Environment setting **
************************
.protect
.lib "/usr/cad/cic018.l" tt
.unprotect

Cdecap vdd vss 0.1u

*** Inverter ***
.subckt INV in1 out vdd vss
**pFET drain=output, gate=input, source and body = vdd **
mp1 out in1 vdd vdd P_18 w=0.5u l=0.18u
**nFET drain=output, gate=input, source and body = vss(gnd) **
mn1 out in1 vss vss N_18 w=0.25u l=0.18u
.ends

*** NOR ***
.subckt NOR2 in1 in2 NOR vdd vss
** 2 pFET in series **
mp1 pd1 in1 vdd vdd P_18 w=0.5u l=0.18u
mp2 NOR in2 pd1 vdd P_18 w=0.5u l=0.18u
** 2 nFET in parallel **
mn1 NOR in1 vss vss N_18 w=0.25u l=0.18u
mn2 NOR in2 vss vss N_18 w=0.25u l=0.18u
.ends

*** OR ***
.subckt OR2 in1 in2 OR vdd vss
Xnor in1 in2 NOR vdd vss NOR2
** NOR&INV = OR **
Xinv NOR OR vdd vss INV
.ends

*** NAND ***
.subckt NAND2 in1 in2 NAND vdd vss
** 2 pFET in parallel **
mp1 NAND in1 vdd vdd P_18 w=0.5u l=0.18u
mp2 NAND in2 vdd vdd P_18 w=0.5u l=0.18u
** 2 nFET in series **
mn1 NAND in1 ns1 vss N_18 w=0.25u l=0.18u
mn2 ns1 in2 vss vss N_18 w=0.25u l=0.18u
.ends

*** AND ***
.subckt AND2 in1 in2 AND vdd vss
Xnand in1 in2 NAND vdd vss NAND2
Xinv NAND AND vdd vss INV
.ends

*** logic function ***
.subckt logic A B C D F_out vdd vss
Xinv B B_ vdd vss INV
Xor A B_ f1 vdd vss OR2
Xand C f1 F_out vdd vss AND2
.ends
(base) wongweixiang@Wongs-MacBook-Air circuit_hw1 %
```

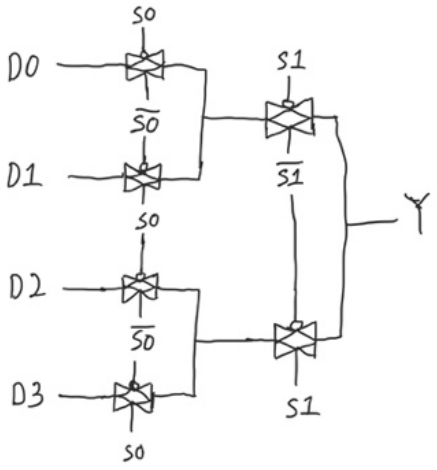
Bonus – MUX 4 to 1

- transistor-based schematic diagram



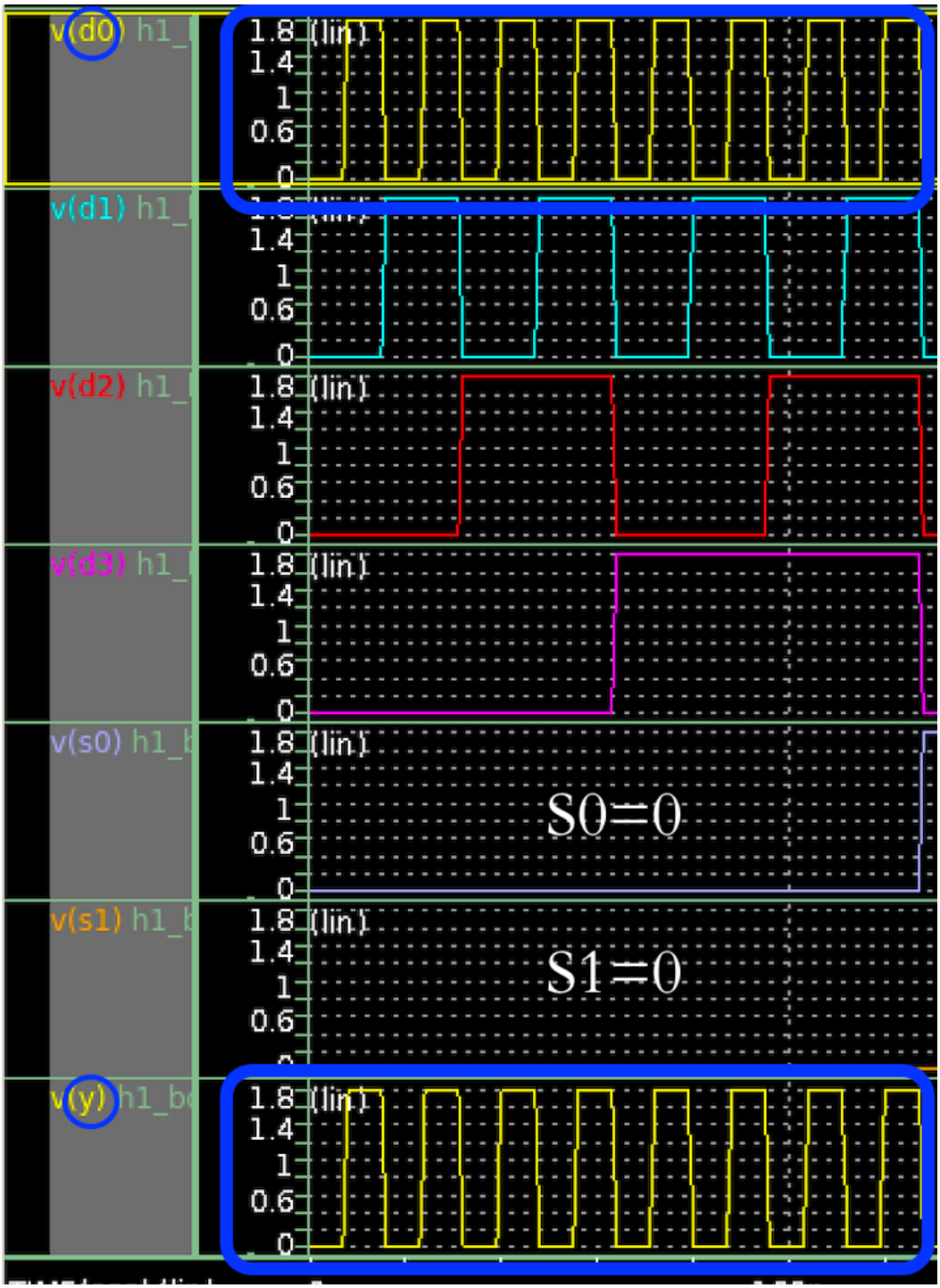






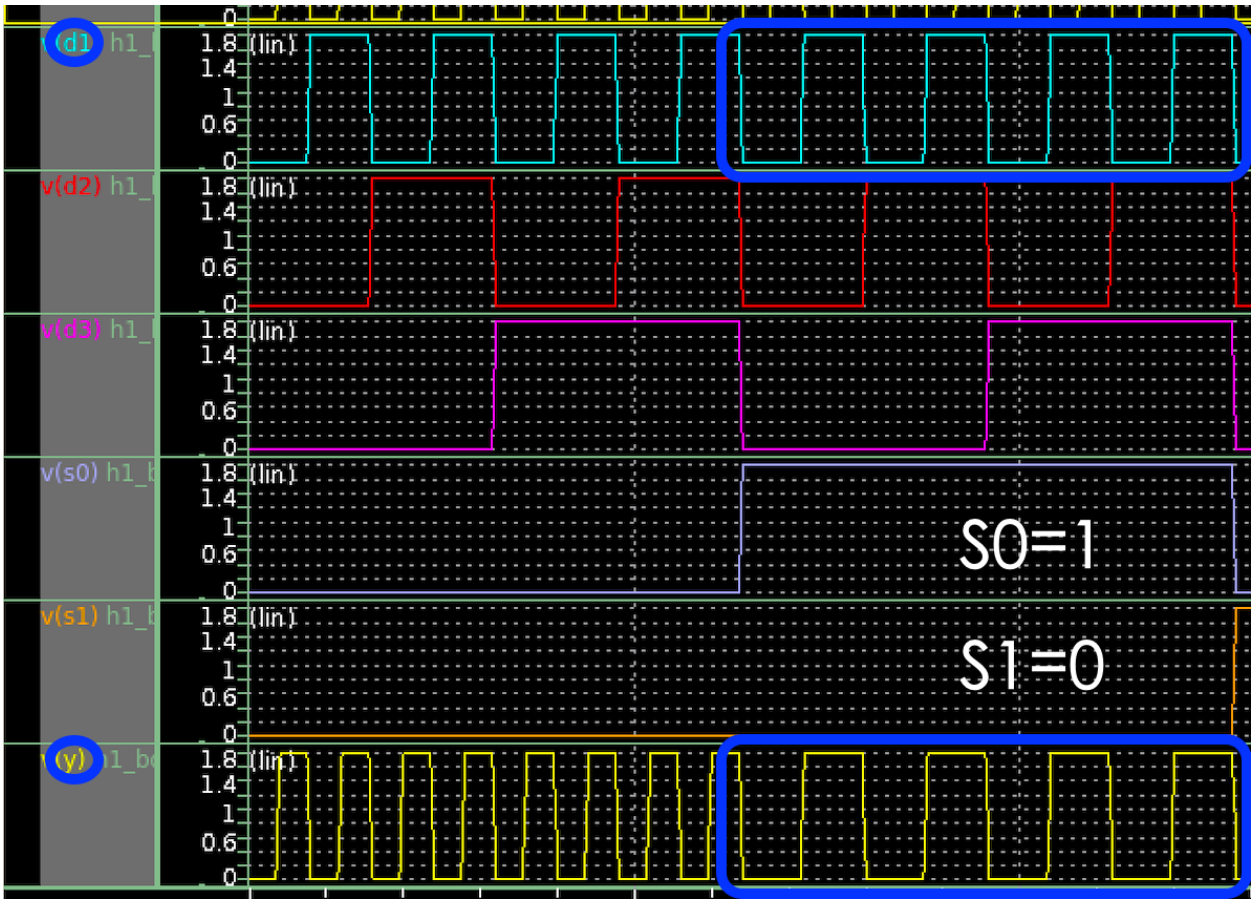
MUX 4 to 1 waveforms

1. S1=0,S0=0



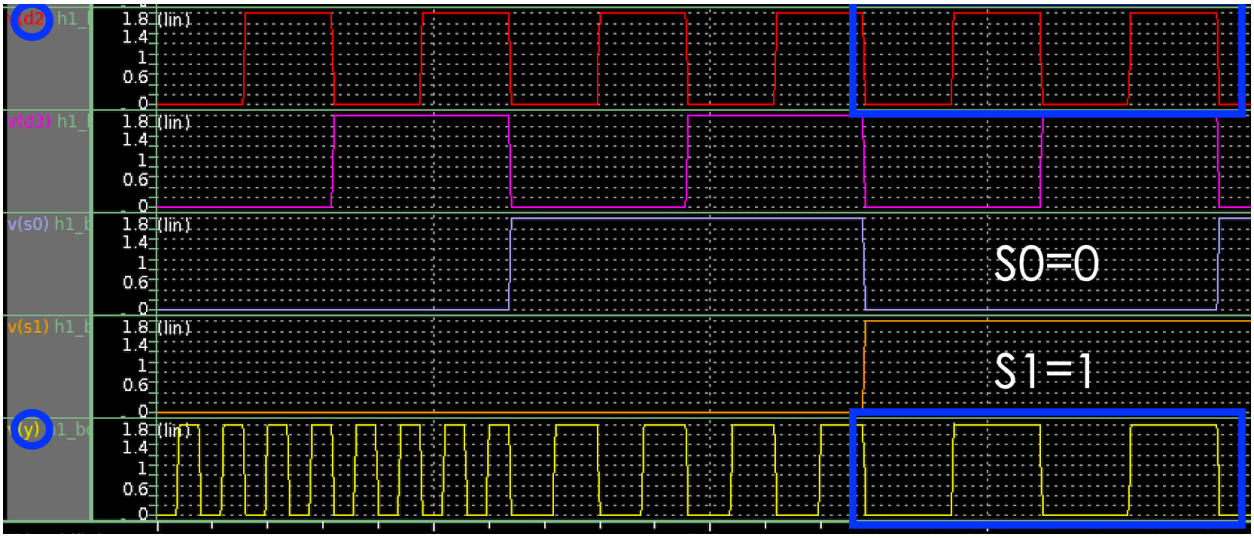
- When S0 and S1 are both equal to 0, output will same as D0.

2. S1=0,S0=1



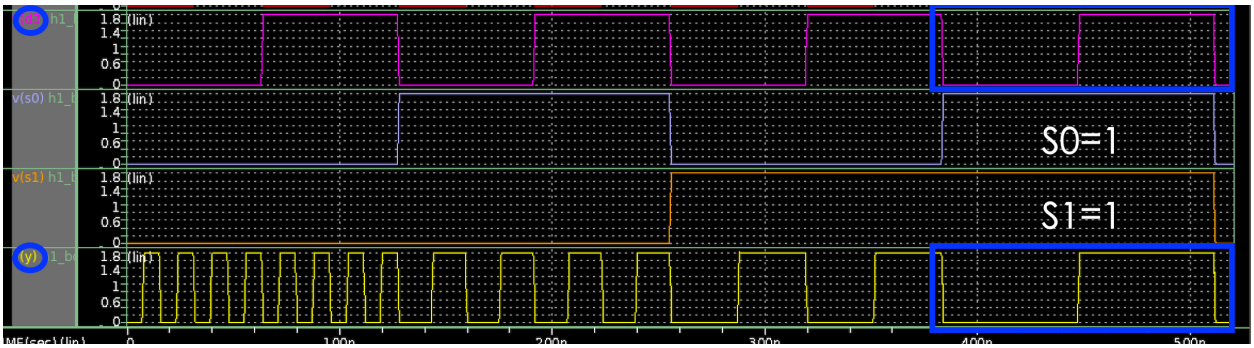
- When S1 = 0 and S0 = 1, output will same as D1.

3. S1=1,S0=0



- When S1 = 1 and S0 = 0, output will same as D2.

4. S1=1,S0=1



- When S1 = 1 and S0 = 1, output will same as D3.

Full wave for 520n