Digital IC Design

Exercise 2: Comparison of Logics Family

Project Report

Professor: Po-Tsang Huang

Student/ID: 尤彥捷/310591036

Exercise 2.1: Synthesis 4-bit Adder

Point	Incr	Path
input external delay	0.00	0.00 f
A[0] (in)	0.00	0.00 f
U13/Y (NAND2x1_ASAP7_75t_R)	15.89	15.89 r
U24/Y (OR2x2_ASAP7_75t_R)	22.85	38.75 r
U25/Y (NAND3xp33_ASAP7_75t_R)	11.18	49.92 f
U26/Y (NAND2xp5_ASAP7_75t_R)	20.70	70.62 r
U27/Y (XNOR2xp5_ASAP7_75t_R)	18.27	88.89 r
sum[3] (out)	0.00	88.89 r
data arrival time		88.89
max_delay	90.00	90.00
output external delay	0.00	90.00
data required time		90.00
data required time		90.00
data arrival time		-88.89
slack (MET)		1.11

Figure 1. Timing Report of 4-bit Adder

Critical Path:	A[0] (f) -> U13/Y (r) -> U24/Y (r) -> U25/Y (f) -> U26/Y (r) -> U27/Y (r) -> sum[3] (r)
Data Required time:	88.89ps

- Pattern of critical path:

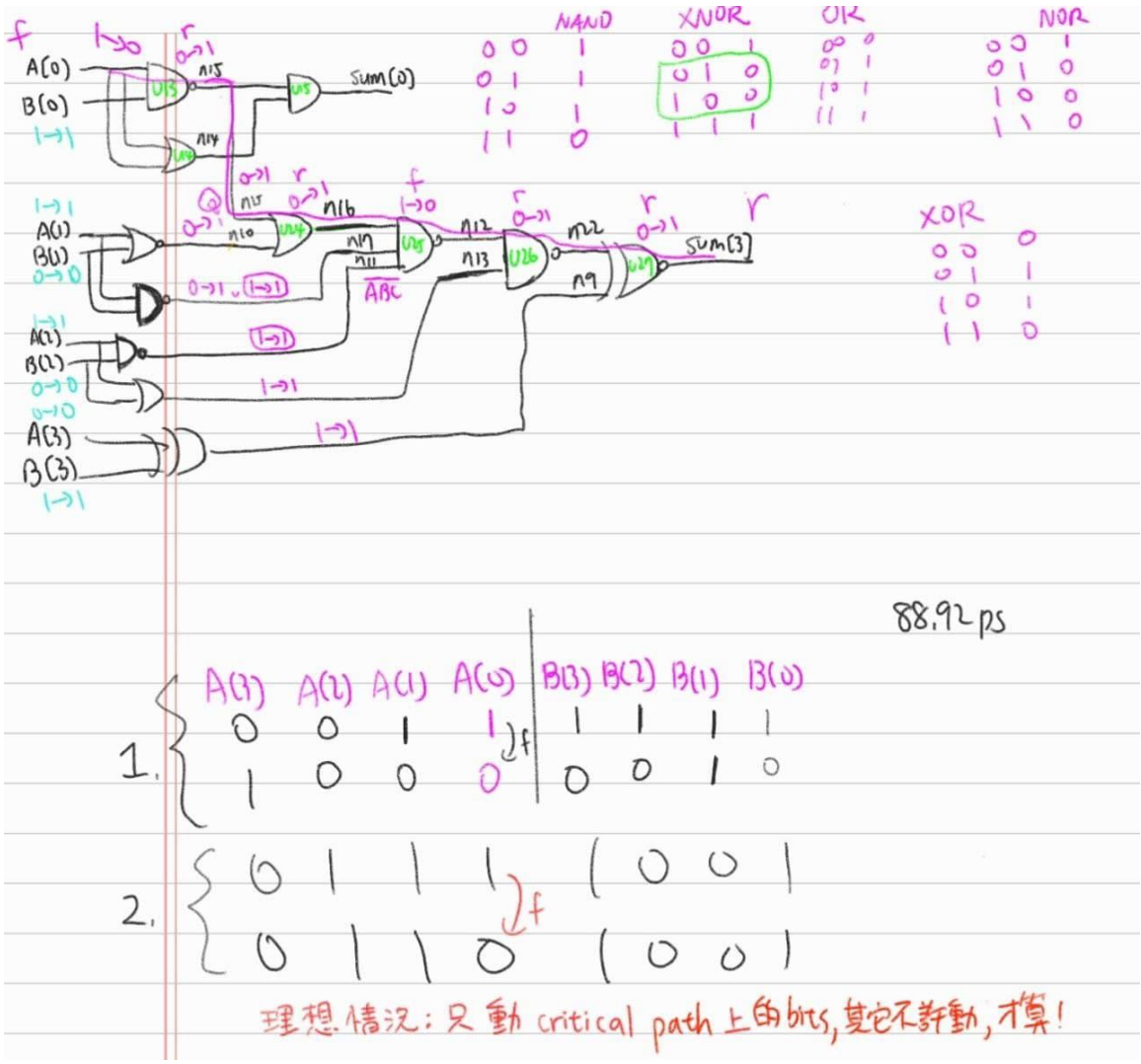


Figure 2. My note for calculating the critical path (In the button of the note are the two patterns I found)

Result:

First Critical Path		
First state	A[3:0]=0011	B[3:0]=1111
Second State	A[3:0]=1000	B[3:0]=0010
Second Critical Path		
First state	A[3:0]=0111	B[3:0]=1001
Second State	A[3:0]=0110	B[3:0]=1001

Exercise 2.2: CMOS Logics for 4-Bit Adder

```
.SUBCKT Adder_4bit VSS VDD  A[3] A[2] A[1] A[0] B[3] B[2] B[1] B[0] carry_out sum[3] sum[2] sum[1] sum[0]
XU13 VSS VDD  A[0] B[0] n15 NAND2x1_ASAP7_75t_R
XU14 VSS VDD  A[0] B[0] n14 OR2x2_ASAP7_75t_R
XU15 VSS VDD  n14 n15 sum[0] AND2x2_ASAP7_75t_R
XU16 VSS VDD  B[3] n20 INVx8_ASAP7_75t_R
XU17 VSS VDD  A[3] n21 INVx8_ASAP7_75t_R
XU18 VSS VDD  A[2] n19 INVx8_ASAP7_75t_R
XU19 VSS VDD  A[3] B[3] n9 XOR2xp5_ASAP7_75t_R
XU20 VSS VDD  B[2] A[2] n11 NAND2xp33_ASAP7_75t_R
XU21 VSS VDD  B[1] A[1] n17 NAND2xp33_ASAP7_75t_R
XU22 VSS VDD  B[2] A[2] n13 OR2x2_ASAP7_75t_R
XU23 VSS VDD  B[1] A[1] n10 NOR2xp33_ASAP7_75t_R
XU24 VSS VDD  n10 n15 n16 OR2x2_ASAP7_75t_R
XU25 VSS VDD  n16 n17 n11 n12 NAND3xp33_ASAP7_75t_R
XU26 VSS VDD  n13 n12 n22 NAND2xp5_ASAP7_75t_R
XU27 VSS VDD  n9 n22 sum[3] XNOR2xp5_ASAP7_75t_R
XU28 VSS VDD  B[1] A[1] n15 A0  sum[1] FAx1_ASAP7_75t_R
XU29 VSS VDD  n17 n16 n18 NAND2xp5_ASAP7_75t_R
XU30 VSS VDD  B[2] n18 n19 A1  sum[2] FAx1_ASAP7_75t_R
XU31 VSS VDD  n20 n22 n21 carry_out MAJIxp5_ASAP7_75t_R
.ENDS
```

Figure 3. Convert gate-level netlist to HSPICE netlist

- Measuring delay and power consumption:

  (Using the same testing environment with synthesis)

  1. Vdd = 0.7V
  2. Output Loading = Capacitor 5f
  3. Temperature = 25V
  4. Pattern (vih = 0.7V, vil = 0V, slope = 0.07V/ps, period = 200ps [5Ghz])

#1. First input pattern

A[3:0]=0011 B[3:0]=1111 ⟶ A[3:0]=1000 B[3:0]=0010

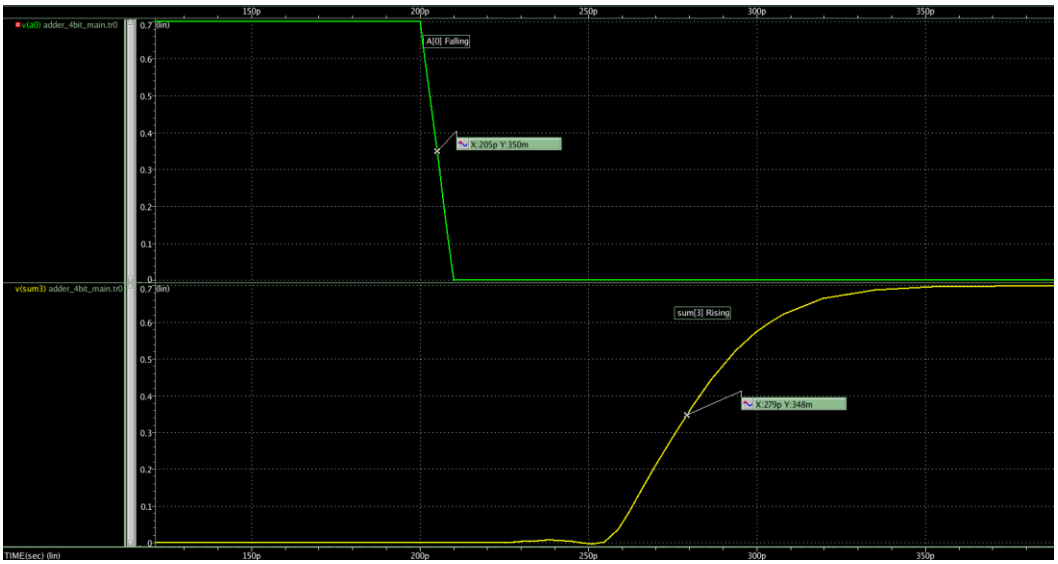


Figure 4. Waveform of the first input pattern (A[0] Falling to sum[3] Rising)

```
******
.title 4-bit adder

****** transient analysis tnom=  25.000 temp=  25.000 ******
pwr=  15.6364u  from=   0.             to= 800.0000p
tplh=  74.3126p  targ= 279.3126p   trig= 205.0000p

         ***** job concluded
```

Figure 5. Transient Analysis of First input pattern

	Delay	Power Consumption
Result	74.3126ps	15.6364uW

#2. Second input pattern

A[3:0]=0111 B[3:0]=1001 ⟹ A[3:0]=0110 B[3:0]=1001

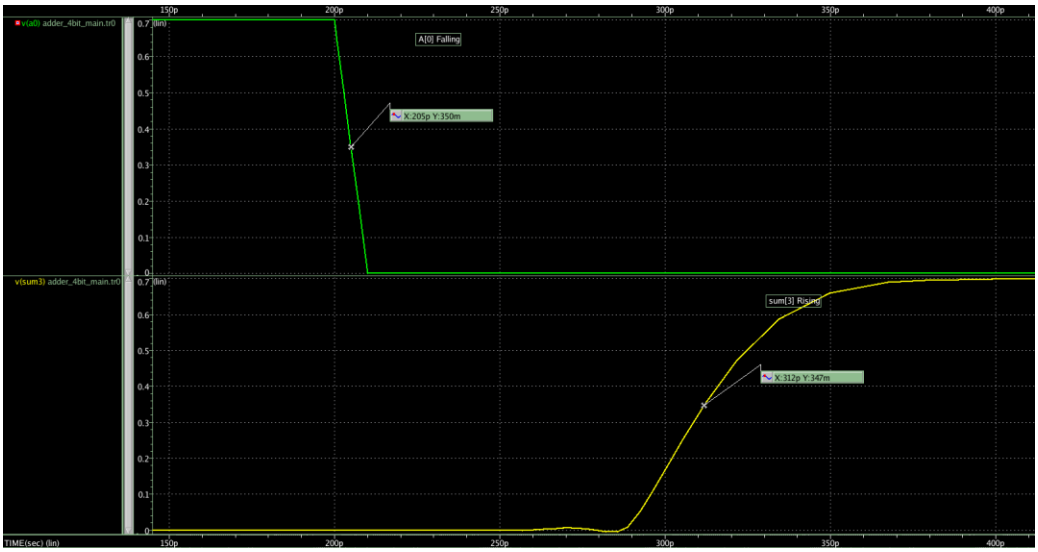


Figure 6. Waveform of the first input pattern (A[0] Falling to sum[3] Rising)

```
******
.title 4-bit adder

****** transient analysis tnom=  25.000 temp=  25.000 ******
pwr=  13.8502u  from=   0.             to= 800.0000p
tplh= 107.0063p  targ= 312.0063p   trig= 205.0000p

         ***** job concluded
```

Figure 7. Transient Analysis of Second input pattern

	Delay	Power Consumption
Result	107.0063ps	13.8502uW

- Discussion:

In this exercise, our goal is to find the critical path for the 4-bit adder. The criteria is that the delay of the pattern we found must take longer than the delay we got from synthesis, 88.89ps, since it uses many optimization methods during the process.

The two set of critical paths I found both follow the logic mentioned in the timing report. However, they have totally different time delay and power consumption. For time delay, before the simulation, I assume that the first pair of input pattern is going to bear longer time delay since every gate in the first pattern does more state changing (1>0 or 0>1) than the second pattern. Nevertheless, the delay in second pattern is way longer than the first one. On top of that, it also satisfies the criteria we mentioned (107.0063ps > 88.89ps). Although, I do not fully understand how can this happen, I can make an assumption based on my discovery. The less we change the state of other logic gates, except the gate in our critical path, the more time delay we will get.
(TA explanation: If the gate which is not on the critical path toggled, it may make the circuit to send signal earlier and shorten the time delay of the critical path.)

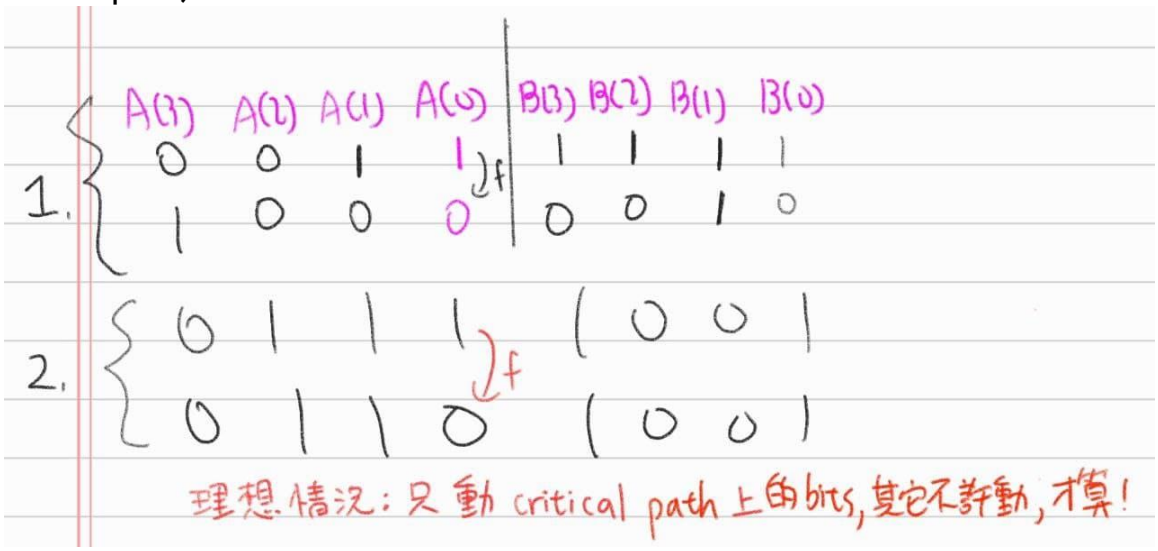


Figure 8. Two patterns of critical path

Final Solution: The second pattern is the critical path

Exercise 2.3: Comparisons of different logic families

I. Theoretically Discussion

Logic Family	Pros	Cons
CMOS Logic (2N)	• high stability & robustness • high noise margin • static power consumption = 0 • powerful driving ability (Rin->∞)	• more transistor may take higher power consumption • having larger design area
Pass Transistor Logic (N)	• less transistor count	• No Driving Ability • could cause signal degradation
Dynamic Logic (N+2)	• Tplh = 0 (high speed) • powerful driving ability (Rin->∞)	• noise sensitive • charge sharing • lower bound on freq. due to charge leakage • higher power consumption due to requirement of clock • NM_L=Vt

From the diagram above, we can decide the ranking of each logic by two factors, the transistor count and the location in the circuit.
In the discussion of the transistor count, we assume the logic gate is in the middle of the circuit (neither output or input stage).

⇨ Ranking: PTL > Dynamic Logic > CMOS

Considering the location in the circuit which means the logic gate is at input or output stage. We will take CMOS having the priority due to its robustness and low static power consumption.

⇨ Ranking: CMOS > Dynamic >>>>> PTL

II. Implementation

- Test pattern

```
radix   1111 1111
vname   A[3:0] B[3:0]
io      iiii iiii
tunit ps
slope 10
vih 0.7
vil 0
period 200

0111 1001
0110 1001
0111 1001
0110 1001
1010 1010
0000 0000
```

Figure 9. Testing pattern for delay (critical path)

```
radix   1111 1111
vname   A[3:0] B[3:0]
io      iiii iiii
tunit ps
slope 10
vih 0.7
vil 0
period 200

0000 0000
0000 0001
0000 0010
0000 0011
0000 0100
0000 0101
0000 0110
0000 0111
0000 1000
.....
1111 1100
1111 1101
1111 1110
1111 1111
```

Figure 10. 256*patterns for measuring power

➢ Standard 4-bit adder (only CMOS logic)

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
tplh= 132.6820p  targ= 337.6820p   trig= 205.0000p
```

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
pwr=  73.3931u  from=   0.             to=  52.0000n
```

```
******  Circuit Statistics  ******
# nodes       =    2041 # elements    =    4315
# resistors   =    1902 # capacitors  =    2181 # inductors   =       0
# mutual_inds =       0 # vccs        =       0 # vcvs        =       0
# cccs        =       0 # ccvs        =       0 # volt_srcs   =       2
# curr_srcs   =       0 # diodes      =       0 # bjts        =       0
# jfets       =       0 # mosfets     =     222 # U elements  =       0
# T elements  =       0 # W elements  =       0 # B elements  =       0
# S elements  =       0 # P elements  =       0 # va device   =       0
# vector_srcs =       8 # N elements  =       0
```

Average Power Consumption = 73.3931uW
Critical Path Delay = 132.682ps
Transistors Count = 222 MOSFETs

Figure 11. CMOS Logic critical path patterns

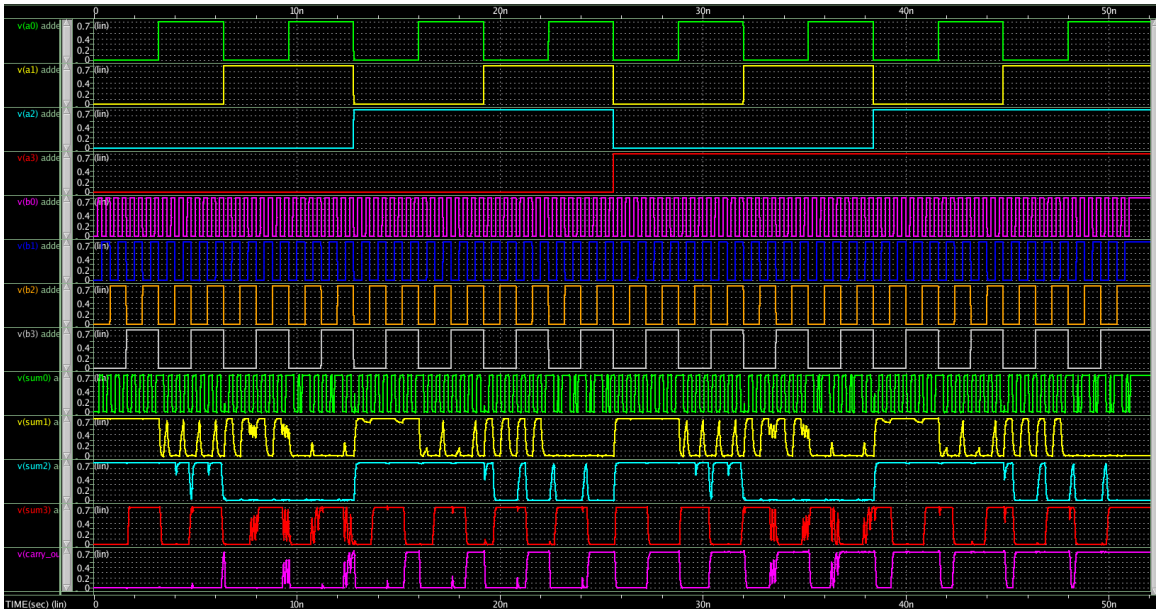


Figure 12. CMOS Logic 256 patterns

- Transistor Count of each logic gate implement by each logic family

Logic Family / Gate	CMOS	PTL	Dynamic
ANDx2	8 (MOSFETs)	4	6
XORx2	10	6	12
ORx2	8	4	6
NORx2	4	6	4
NANDx2	4	6	4
NANDx3	6	10	5
XNORx2	10	8	10

➢ Policy: CMOS: U13, U16, U17, U18, U20, U21, U23, U24, U26, U29
PTL: U14, U19, U22
Dynamic: U15, U25, U27

- Analysis Report

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
tplh=  25.7430p  targ= 230.7430p   trig= 205.0000p
```

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
pwr=  76.6857u  from=   0.             to=  52.0000n
```

```
******  Circuit Statistics  ******
# nodes       =    1474 # elements   =    3096
# resistors   =    1357 # capacitors =    1522 # inductors   =       0
# mutual_inds =       0 # vccs       =       0 # vcvs        =       0
# cccs        =       0 # ccvs       =       0 # volt_srcs   =       2
# curr_srcs   =       0 # diodes     =       0 # bjts        =       0
# jfets       =       0 # mosfets    =     207 # U elements  =       0
# T elements  =       0 # W elements =       0 # B elements  =       0
# S elements  =       0 # P elements =       0 # va device   =       0
# vector_srcs =       8 # N elements =       0
```

Average Power Consumption = 76.6857uW
Critical Path Delay = 25.743ps
Transistors Count = 207 MOSFETs

Figure 13. Mixed Logic critical path patterns

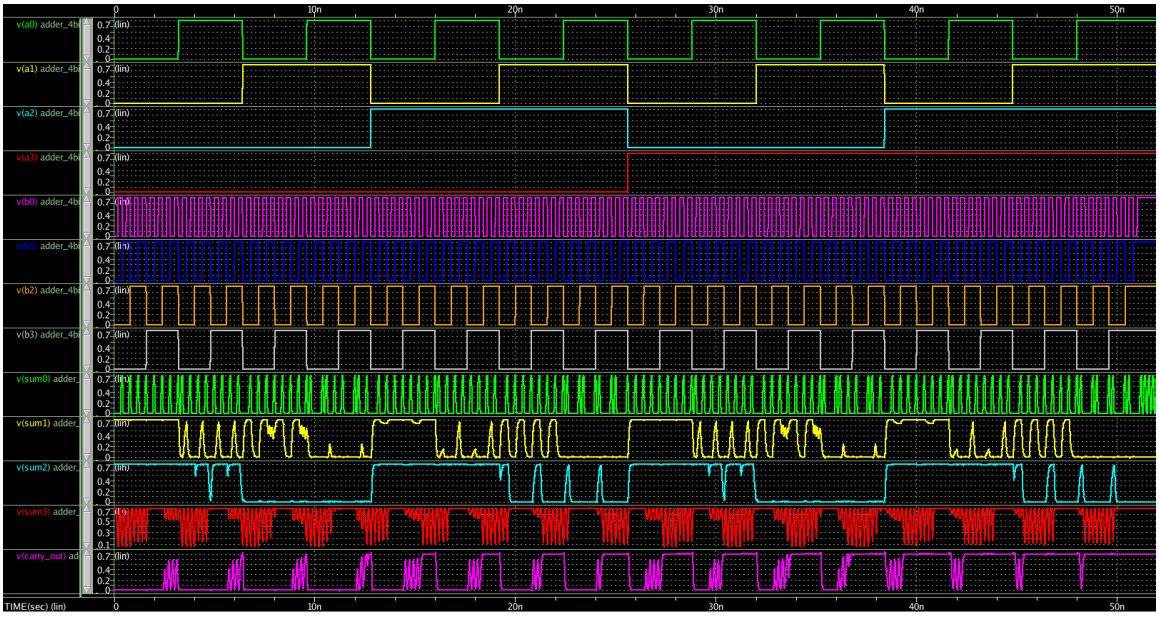
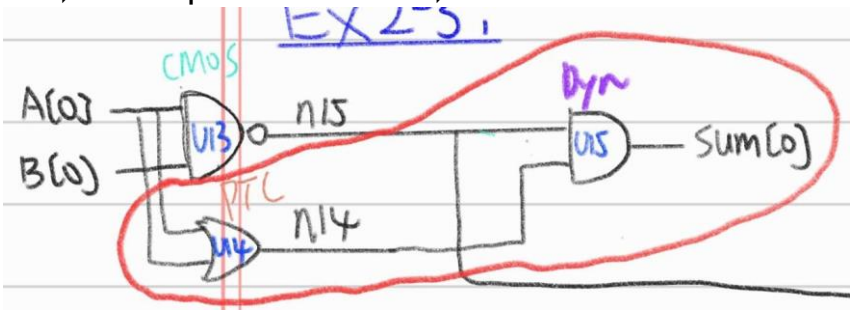


Figure 14. Mixed Logic 256 patterns

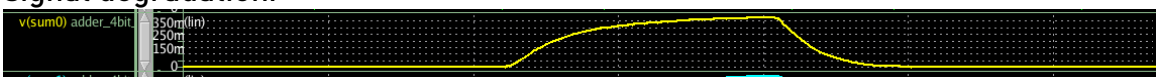
- Discussion:

I tried tons of policy to come up with this best one. I chose most of the logic based on the transistor count from the chart above. But, under some special circumstances, we should take other effect into consideration. In the following report, I will introduce every special consideration I took through the process in logic selection.

#1. First, for this part of the circuit,

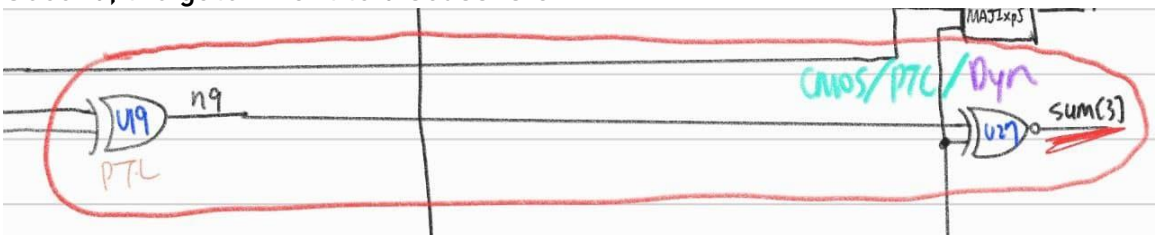


I plan to use PTL instead of Dynamic in U15 initially. This is a very classic situation exploit the biggest problem of cascading PTL, which is the signal degradation.



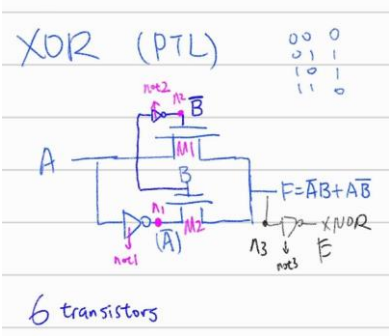
We can see the output signal sum[0] degrade from 0.7V to 0.35V, which is way out of our noise margin high. Although PTL have a big benefit in the transistor count, for the correctness of our circuit, I choose to use dynamic logic instead, which also have a strong driving ability but less transistors than CMOS.

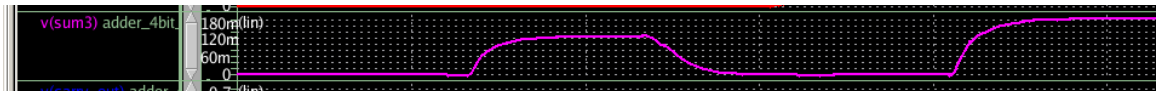
#2. Second, the gate I want to discuss is U27.



Initially, I plan to use PTL which have a smaller transistor count, 8. The reason why I think we can cascade PTL under this circumstance is because, I design XNOR by placing an inverter in the rear part of XOR.

I thought that the inverter may bring up the degraded signal. The reality is always cruel. Sum[3] is degraded from 0.7V to 0.18V. When the signal is degraded to much, it is certainly out of the NMh of the inverter.





Then, how about the selection between Dynamic and CMOS logic which both have the same transistor counts in XNOR? Here comes the strong trait of Dynamic logic, which is high speed

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
pwr=   38.4249u  from=   0.            to=   1.0000n
tplh=   22.2272p  targ= 227.2272p   trig= 205.0000p
area= 410.0002n
```

Figure 15. Analysis of Dynamic Logic(U27)

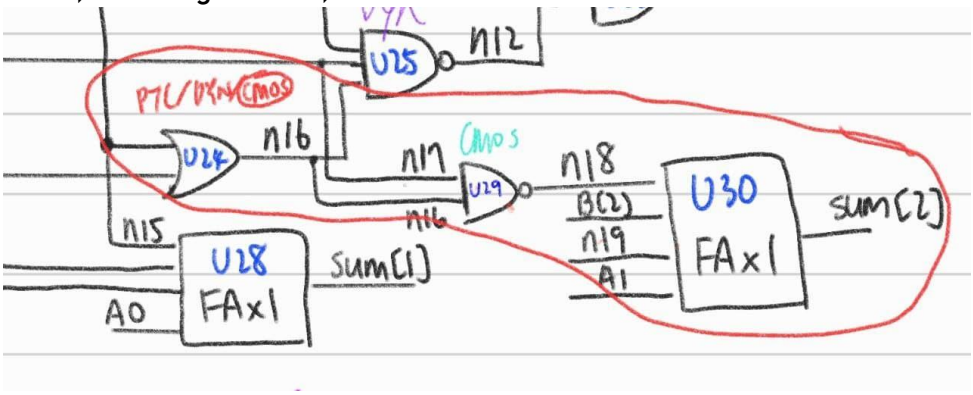
v.s.

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
pwr=   34.6328u  from=   0.            to=   1.0000n
tplh=   71.8000p  targ= 276.8000p   trig= 205.0000p
area= 310.0002n
```

Figure 16. Analysis of CMOS Logic (U27)

We can see the delay of the critical path from two different logics. Dynamic logic is much faster than CMOS logic. However, the higher speed we want, the circuit must take more power consumption. It is the trade-off. To summarize, it is a great way to design Dynamic logic in the output stage if there is no other problem with itself.

#3. Third, for the gate U24,



In the beginning, I tried to use PTL and Dynamic logic to get a less transistor count. But, sum[2] signal turned out to have a serious signal degradation. So, I can only use CMOS to implement it.

Final Circuit Diagram:

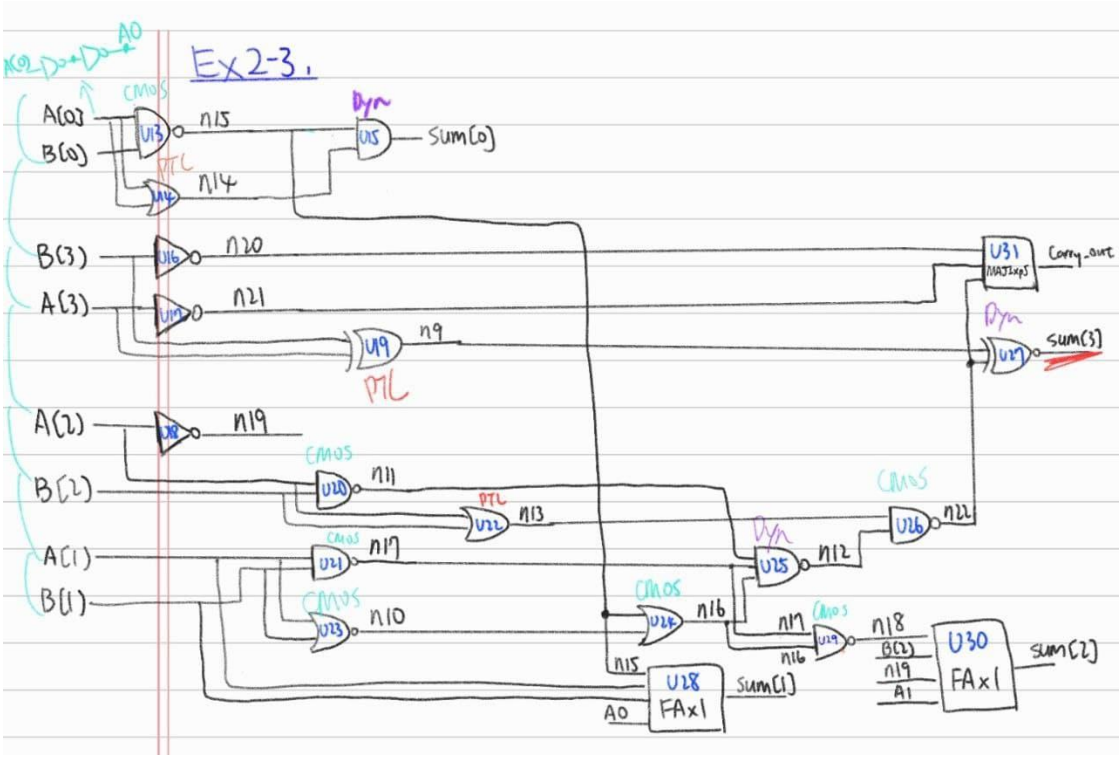


Figure 18. Final Digital Circuit

III. Appendix

(HSPICE code: logic_family.sp)