DIC HW4

313580059 李元睿

Using 7nm FinFET devices with VDD = 0.8V, FF process corner and medium Vt CMOS process. Rise time and fall time of input signals and clock are 0.02ns (0V 0.8V)

7nm tech. path: /RAID2/COURSE/dic/dicTA01/DIC_2024_Fall/7nm_files

A 4-bit ripple carry adder shown in Fig. 1 is designed with fully complementary static logic gate of full adder (FA). Input signals are A[3:0], B[3:0], and Cin. Outputs are SUM[3:0] and Cout.

(1) Try to design the fastest adder when each input signal is driven by one unit size inverter. For the output loads, 5 unit size inverters are used for each output signal. First, show your block diagrams in terms of the 1-bit FA. Second, show the circuit schematic of the FA. Use logic effort concepts (do not have to write down the procedure) to design **transistor widths**. **Describe your design concept**. (40%)

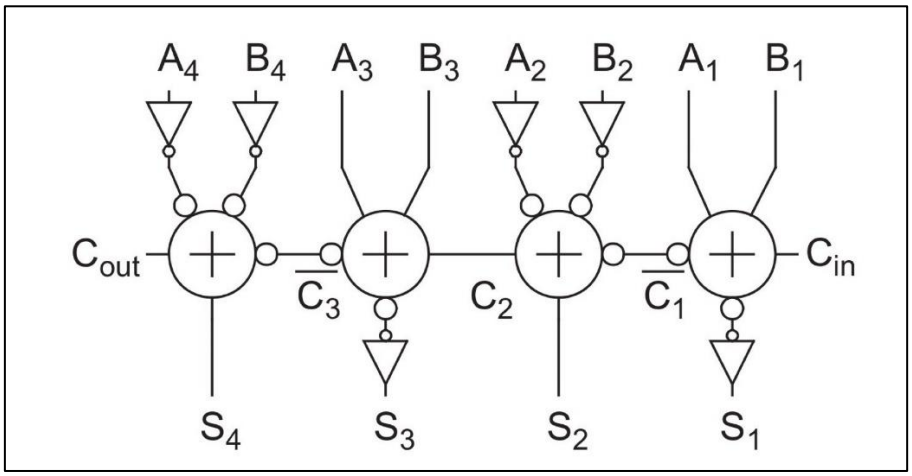

Fig1. 4-bit carry-ripple adder

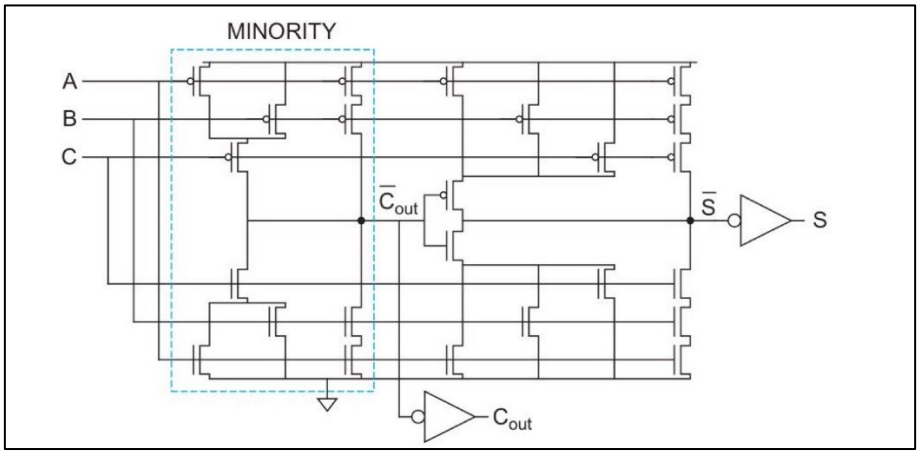

Fig2. transistor level of FA

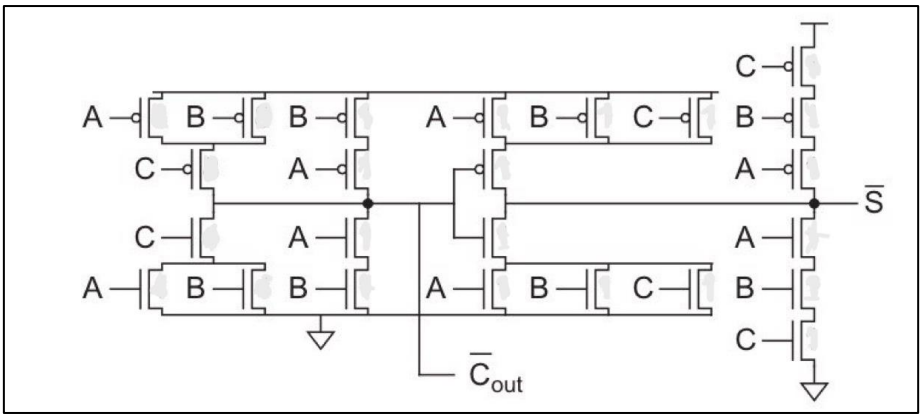

Fig3. transistor level of FA(optimized)

首先，題目要求為設計 fastest adder，因此參考 CMOS VLSI design 這本書裡的架構圖，如 Fig1 所示。因為在此架構中，雖然 critical path 上每個 Cout 是反向的，但可以透過串接 4 級的反向 FA 來改進原先使用 Fig2 架構在 critical path 上會多 4 個 inverter 的 delay。因此最後是採用 Fig3 的架構。此外，雖然在 Fig1 中，A2、B2、A4、B4、S1、S3 的 node 上都有多加 inverter 以確保邏輯正確，但是這樣並不會導致電路速度變慢，因為沒有影響到 critical path。

接著對 1-bit FA 進行 sizing(nfin_p:nfin_n=2:1)，如 Fig4 所示:

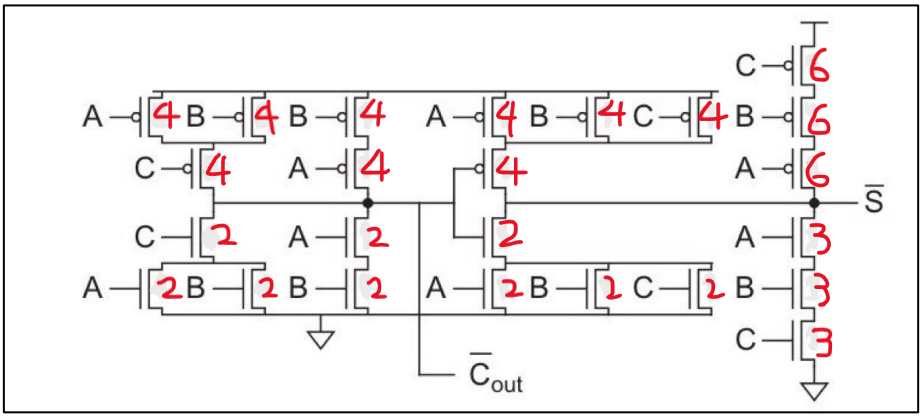

Fig4. sizing for transistor level of FA(optimized)

在 Fig4 中，$\overline{C_{out}}$又會接到下一級的 FA，因此可以視為在中間這點有

branch。此外在 on path 上還要考慮 input 的 unit size inverter 與 output 的 FO5 還有多出來的 input 與 output 的 DFF 一起納入 logic effort 的計算。最後算出的 size 如下所示:

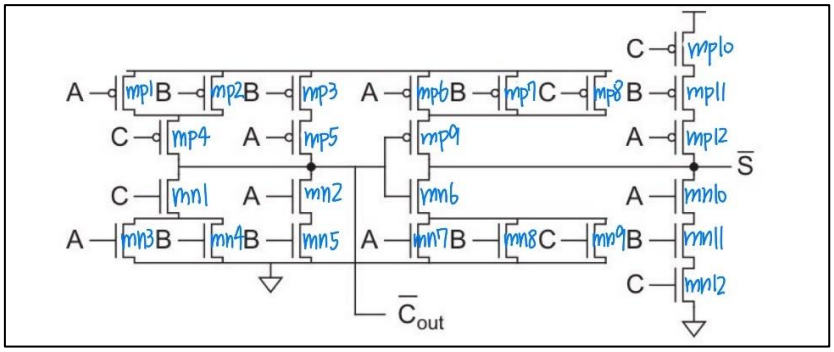

Fig5. the number of each transistor in Fig4

	mp1~mp9	mn1~mn9	mp10~mp12	mn10~mn12
FA1	nfin=5	nfin=2	nfin=7	nfin=3
FA2	nfin=6	nfin=3	nfin=9	nfin=5
FA3	nfin=7	nfin=3	nfin=11	nfin=5
FA4	nfin=8	nfin=4	nfin=12	nfin=6

Table1. sizing of FA(optimized)

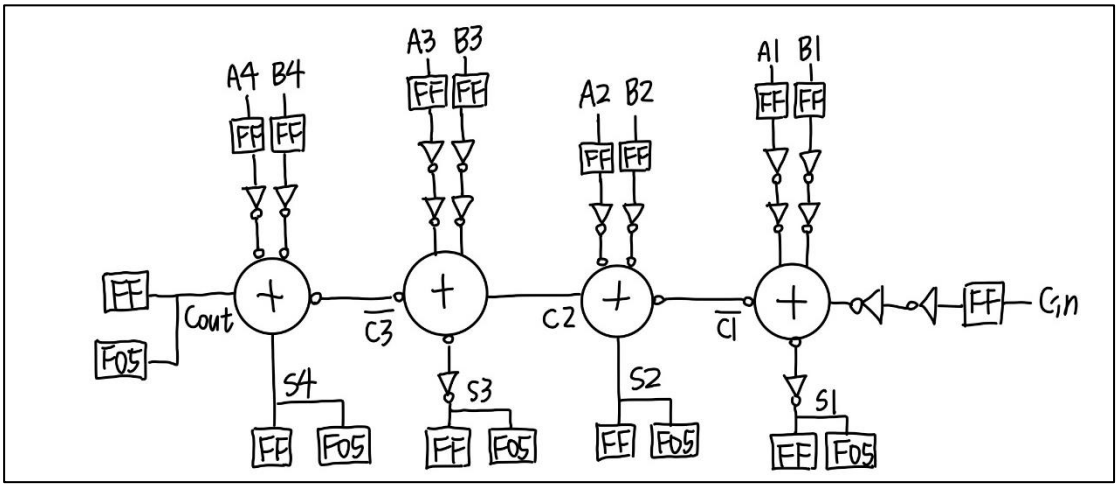

Fig6. final architecture of 4-bit ripple carry adder

上圖(Fig6)為最終設計的 4-bit ripple carry adder。在 input 端都會先經過一個 posedge 的 DFF，然後透過 unit size inverter 提供 input pattern 給各自的 FA。而在這 4 個 FA 中，輸出都是$\overline{C_{out}}$和$\overline{sum}$，由此來減少 critical path 上的

inverter propagation delay。也因此，需要將 S1 和 S3 掛上一個 inverter 來確保 function 是正確的。此外，在所有的輸出節點 S1~S4 與最後的C_{out}端，也都掛上 DFF 及 FO5 delay，其中 FO5 是由 5 個 unit size inverter 並聯組成的輸出負載。

(2) Based on the design of (1), run SPICE to find the propagation delay (As shown in Fig. 2, with pattern from **IA [3:0] = 4'b1111, IB[3:0] = 4'b0000, ICin = 1'b1** to **IA[3:0] = 4'b1111, IB[3:0] = 4'b0000, ICin = 1'b0**). Determine the maximum propagation delay of the Ripple Carry Adder (exclude the load inverters). (20%)

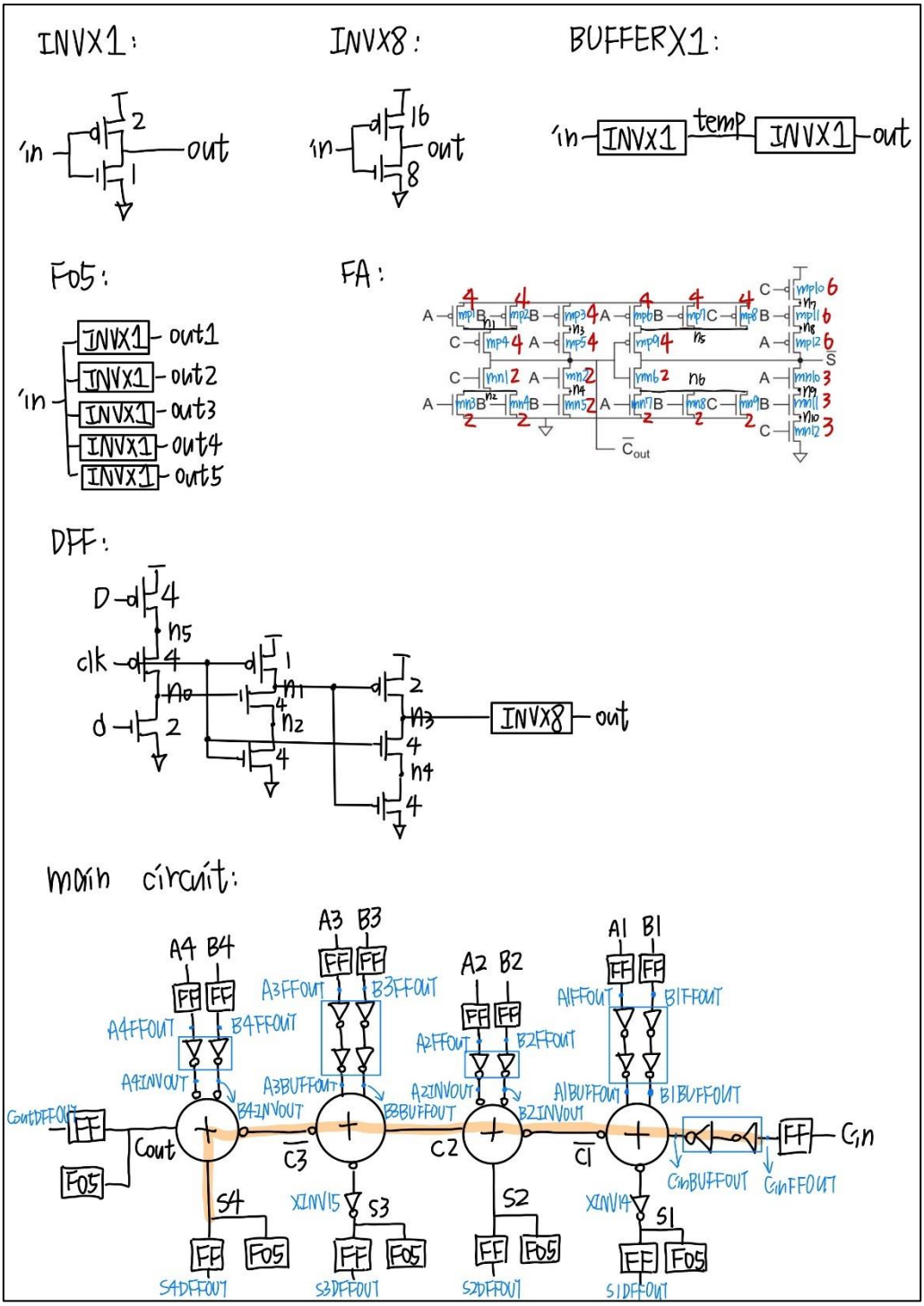

Fig7. detailed sub-circuits of 4-bit ripple carry adder

首先根據 Fig7 來實施 4-bit ripple carry adder 的 spice 模擬。量測 max propagation delay 可以透過 measure 此電路中的 critical path，也就是 Fig7 中的橘線，CinFFOUT 到 S4 的路線。而在題目敘述中，我們是使用 IA [3:0] =

4’b1111, IB[3:0] = 4’b0000, ICin = 1’b1 to IA[3:0] = 4’b1111, IB[3:0] = 4’b0000, ICin = 1’b0 的 input pattern，如此便可以用以界定若是要在一個 clock cycle 中運算完這個 combinational logic 最悲觀需要多少時間。

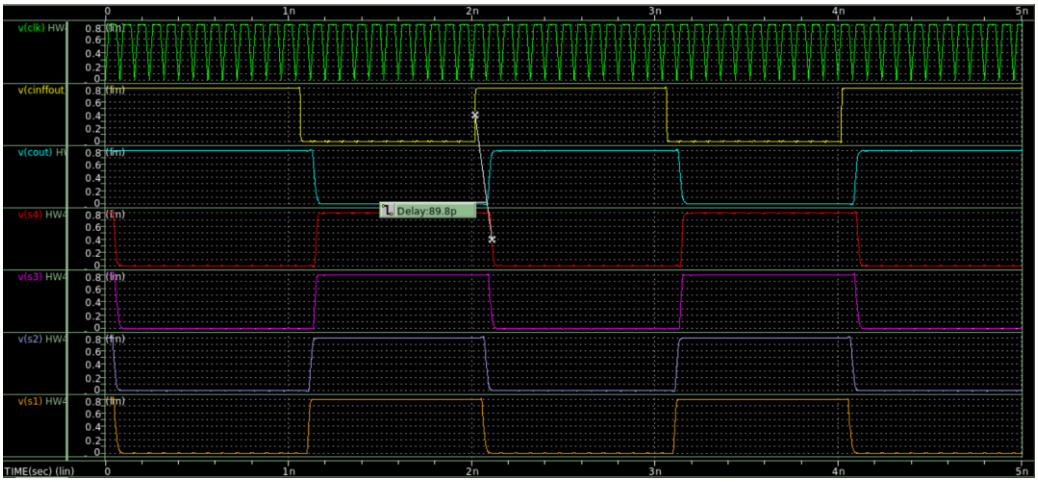

Fig8. simulation waveform for 4-bit ripple carry adder(logic effort version)

根據 Fig8，我們可以看到在 t = 0~1ns 時，輸入為 IA [3:0] = 4’b1111, IB[3:0] = 4’b0000, ICin = 1’b1，因此最後一級的C_{out}是 1’b1，S1~S4 是 1’b0。而當時間在 t = 1~2ns 時，輸入為 IA[3:0] = 4’b1111, IB[3:0] = 4’b0000, ICin = 1’b0，因此最後一級的C_{out}是 1’b0，S1~S4 是 1’b1，都與預期的結果相同。而透過指令: .meas tran tpd trig v(CinFFOUT) VAL='0.4' rise=1 targ v(s4) VAL='0.4' fall=2 及 WaveView 內建的 measure tool 都可以得到相同的t_{pd}為 89.8ps。但這僅是透過 logic effort concept 得到的架構，對於實際電路來講，還可以再進一步優化。

經過不斷嘗試後，我發現透過調小 Fig5 中 mp6~mp12 及 mn6~mn12 的 nfin(non-critical path)，並調大 critical path 上的電晶體的 nfin，可以有效減少 propagation delay。最後得到的 optimized version 如 Table2 所示。

	mp1~mp5	mn1~mn5	mp6~mp9	mn6~mn9	mp10~mp12	mn10~mn12
FA1	nfin=15	nfin=10	nfin=2	nfin=1	nfin=3	nfin=2
FA2	nfin=25	nfin=15	nfin=4	nfin=2	nfin=4	nfin=3
FA3	nfin=35	nfin=20	nfin=5	nfin=3	nfin=6	nfin=4
FA4	nfin=45	nfin=25	nfin=7	nfin=5	nfin=8	nfin=6

Table2. optimized sizing of FA

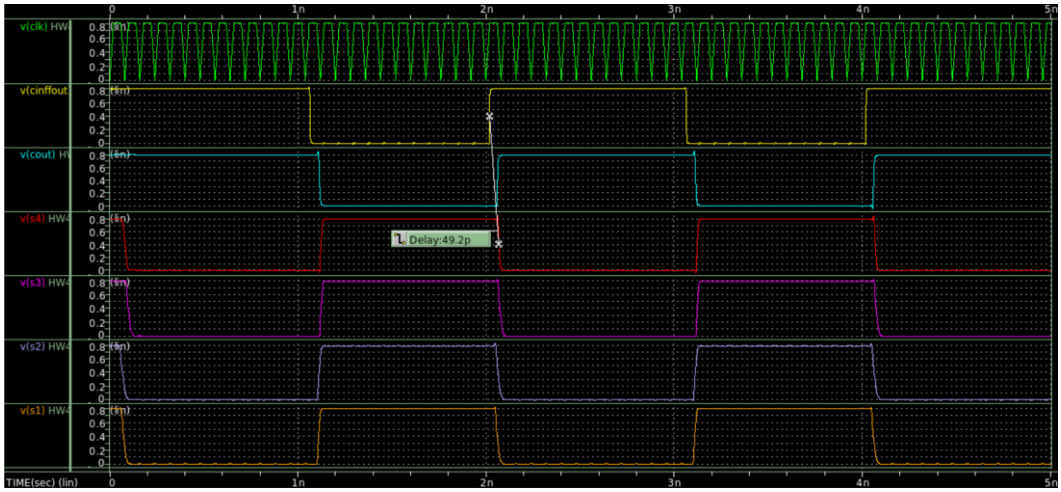

Fig9. simulation waveform for 4-bit ripple carry adder(optimized version)

根據上述的 sizing principle(Table2)，可以看到在 Fig9 中整體的功能還是正常且 propagation delay 減少許多(89.8ps→49.2ps)，而 max frequency 為 1/propagation delay = 20.325GHz。因此，最終選擇 optimized version 來實作 4-bit ripple carry adder。

(3) Run SPICE of the ripple carry adder with each output load equals to 4 unit size inverter. Determine the **<u>average, peak, and leakage</u>** power dissipation and energy per bit, respectively when simulating at the working frequency in (2). (20%)

在這裡我使用的 clock period 為 60ps，雖然 max propagation delay 為 49.2ps，但是為了不要有 setup time fail 及希望波型不要出現那麼多 glitch(因為 clk 變動太快，電容還沒充滿電就又放電，還沒放完電就又充電)，所以才將 clk period 調整為 60ps。但同時題目規定 rise time 和 fall time 為

0.02ns(20ps)，所以基本上 clk 會很接近三角波。

Fig10. simulation waveform for 4-bit ripple carry adder(optimized version)

透過 Fig10 可以看到，input A 和 input B 和C_{in}有照著題目要求去變化，如下表(Table3)所示:

input A	0000→0001→0010→…→1111
input B	1111→1110→1101→…→0000
C_{in}	0→1→0→…→1

Table3. input pattern for (3)

而 average power 和 peak power 則透過.meas 指令求得，如 Fig11 所示:

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
avgpower= 309.6070u  from=   0.            to=   5.0000n
peakpower=   1.2650m    at=   1.9985n
           from=   0.            to=   5.0000n
```

Fig11. average power and peak power

同理，leakage power 也是根據題目提供的 input pattern，然後利用.meas 指令去求得，如 Fig12 所示:

```
****** transient analysis tnom=  25.000 temp=  25.000 ******
leakagepower= 224.4233n  from=   0.            to=   5.0000n
```

Fig12. leakage power

energy per bit 則為(average power * simulation time)/4 = 309.6070uW * 5ns/4 = 387fJ。最後整理出的表格如 Table4 所示:

average power	peak power	leakage power	energy per bit
309.6070uW	1.2650mW	224.4233nW	387fJ

Table4. average power/peak power/leakage power/energy per bit for (3)

(4) Add pipelining stages as shown in Fig. 4 into the 4-bit ripple carry adder, with the D register given in Fig. 3. Run SPICE to find the propagation delay time (with pattern from **IA [3:0] = 4'b0000, IB[3:0] = 4'b1111, ICin = 1'b0** to **IA[3:0] = 4'b0000, IB[3:0] = 4'b1111, ICin = 1'b1**) between pipelining stages to determine the **maximum working frequency** of the clock.

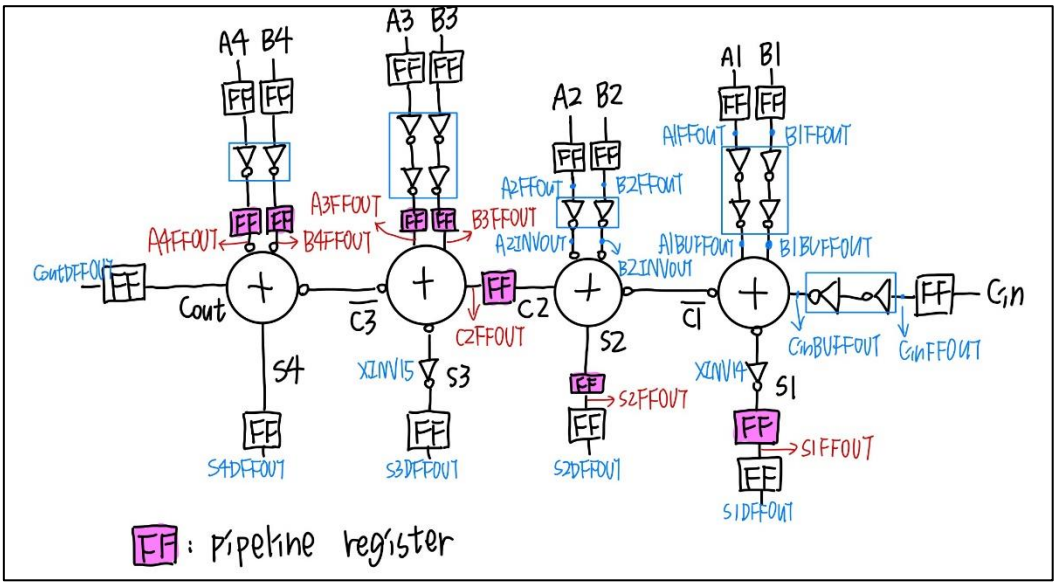

Fig13. pipeline architecture of 4-bit ripple carry adder

根據 Fig13，我把 pipeline 切在第二個與第三個 bit 中間，為求各級 pipeline 可以盡可能的平均，使 clk frequency 可以越大越好，亦即每級 pipeline stage 的 period 越小越好，這就是我選擇切在中間的原因。此外，因為 S1、S2 會先算好，因此要先用 DFF 存起來，而至於 A3、B3、A4、B4 也要等一下才進來，才可以正確得到 C2 的值，所以也要先用 DFF 存起來。待前一級算完後，得到正確的 C2 後，再將 A3、B3、A4、B4 的值拿進來計算最終的結果。因此，clk frequency 就可以透過量測兩級 propagation delay 取最大的值來決定。

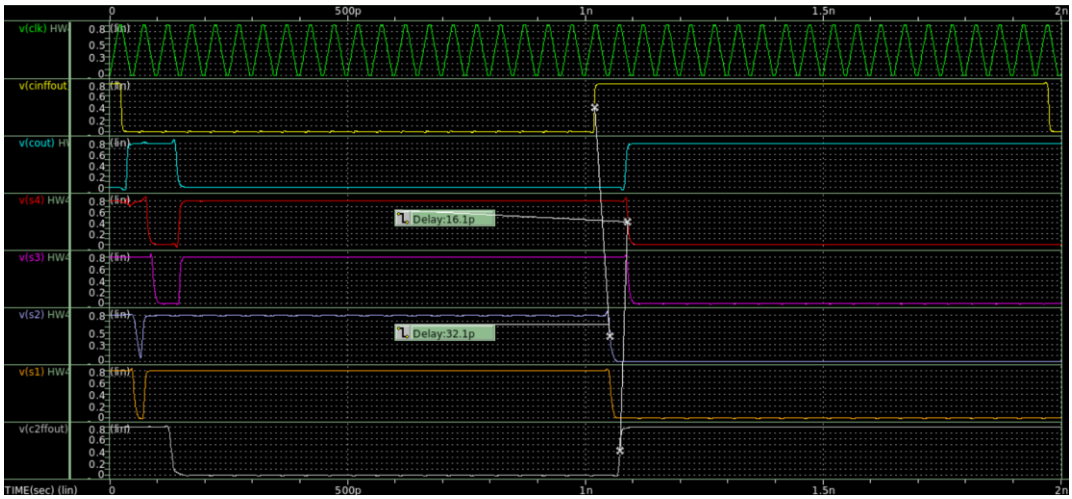

Fig14. simulation waveform for pipeline architecture of 4-bit ripple carry adder

根據 Fig14，第一級的 propagation delay 是 32.1ps，而第二級的 propagation delay 是 16.1ps。因此理論上，可以將 clk period 設為大約 40ps，但因題目要求 rise time 及 fall time 總共要花 40ps，因此我選定 clock period = 50ps。此外，透過 Fig14 也可以發現第二級的 propagation delay 較小，原因是我採用前面 optimized 的架構，FA3 和 FA4 的電晶體都有比較多的 nfin。因此也可以再 sizing 微調一次，讓兩級的 propagation delay 可以更平均，以得到最佳的 clk frequency。