

Digital Integrated Circuits homework 5 電子所 陳柏翔 313510156

1. Design a 20-bit 2's complement multi-operand adder ($A+B+C+D$) using **Carry-Save Adder + Carry skip adder**. To further optimize the delay, each module can have different size. The goal is to have **minimum critical path delay** time with the **least gate count**. Assume there is no overflow during the additions. The internal word length of each module is **20 bits**.

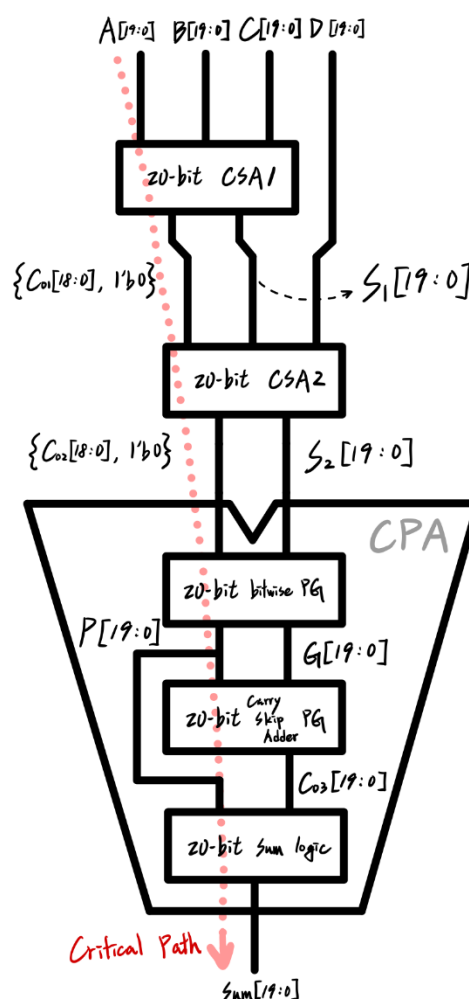
(a) Show your block diagram like that shown in Fig.1. The CPA in Fig.1 shall be the carry skip adder with variable group size (Fig.2). Indicate the critical path. Explain your design concept.

• **Block Diagram (右圖由上至下) :**

根據題目所述，此題不需要考慮 Overflow 的情況，因此每條內部連線皆為 20-bit 的寬度。

為了將四個數值($A[19:0]$, $B[19:0]$, $C[19:0]$, $D[19:0]$)相加，首先用一級 Carry-Save Adder 將 A, B, C 相加起來輸出 $S_1[19:0]$ 與左移 1 bit 的 Carry $\{C_0[18:0], 1'b0\}$ 接到第二級的 Carry-Save Adder 輸入，再將 S_1, C_0, D 相加起來，得到 $S_2[19:0]$ 與同樣左移 1 bit 的 $\{C_2[18:0], 1'b0\}$ 接到 CPA 的輸入。

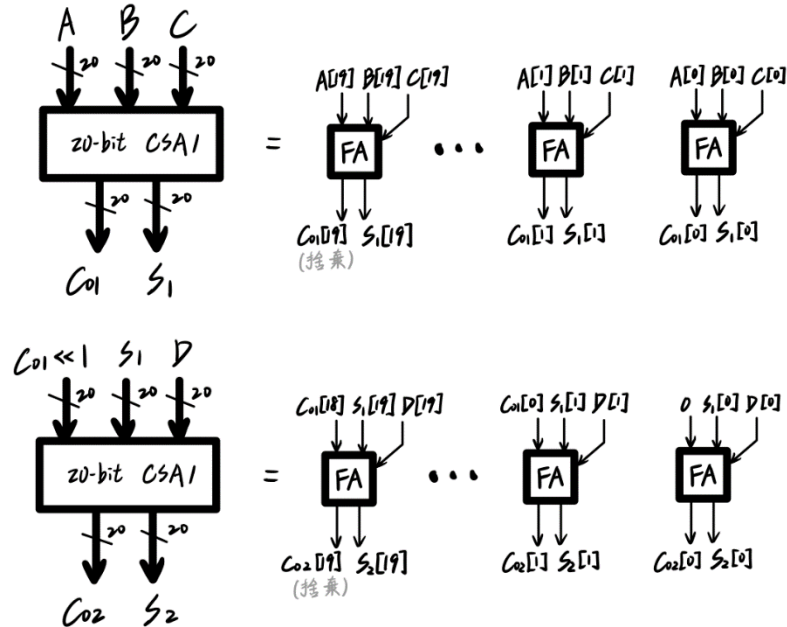
而 CPA 如題目所述是由 Carry-Skip Adder 所實現的，內部首先經過 Bitwise PG logic、再經過 Carry-Skip Adder 內部的 PG network、最後再經過 Sum logic 得到最終的相加結果。
(同題目要求，輸出的數值也是 20-bit)



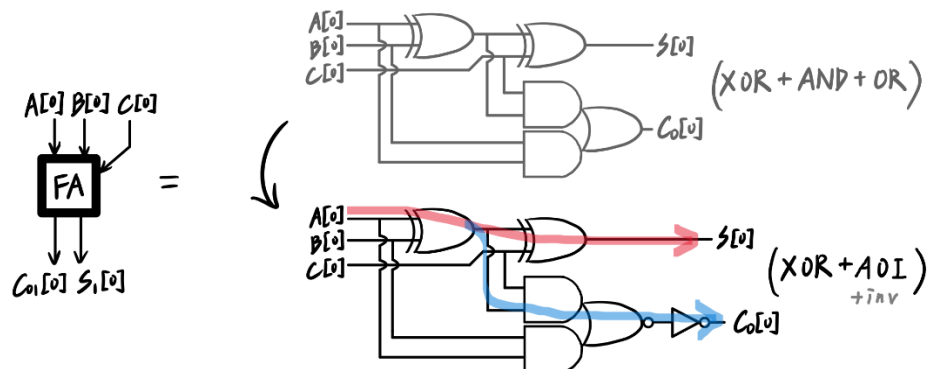
另外，據題目所述需要對 Carry-Skip Adder 進行 variable group size 的安排，以及拆解每一個 Block 中的 logic design，此部分都將在下一小題進行說明。

(b) For each block, you shall show its logic design diagram. Indicate the critical path. Explain your design concepts.

• Carry-Save Adder (CSA) Blocks :



對於兩級Carry-Save Adders我都採用相同的架構，使用20個1-bit Full Adders (FAs)平行的計算出Carry與Sum並輸出。由於FAs之間是平行的，因此Critical path有複數條且延遲都一致，而FA的Gate-level電路如下：



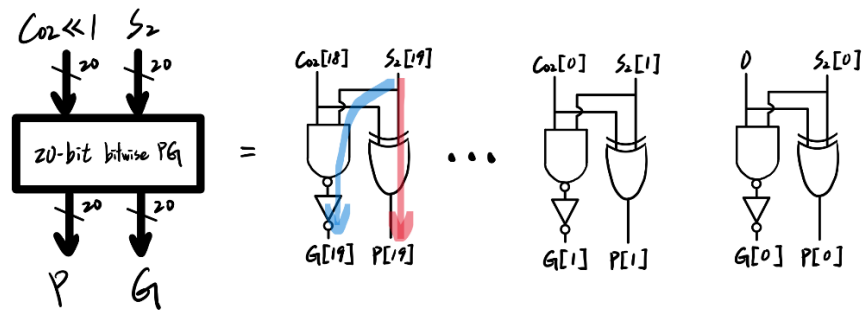
應老師上課所述的作業要求，展示出內部邏輯時要更貼近現實(不能直接用AND, OR, ... gates)：

因此在實作上，Carry路徑原本是2個AND gates + 1個OR gate，我將其改為一個(AOI) Compound gate，並在輸出端加上一個Inverter將邏輯翻正；至於另一條Sum的路徑則是XOR閘，輸出不需要再進行翻轉。

而FA電路中的Critical path如上圖的兩個箭頭路徑所示，由於XOR gate內部邏輯較為複雜，包含有Inverters與AND-OR的結構，因此XOR的延遲相較於下面路徑的AOI加上Inverter延遲其實差不多。

如果 $t_{XOR} > t_{AOI} + t_{inv}$ 其Critical path就會是上面的Sum路徑；如果 $t_{XOR} < t_{AOI} + t_{inv}$ 則Critical path就會是下面的Carry路徑。

• (Carry-Skip Adder) Bitwise PG Logic Block :



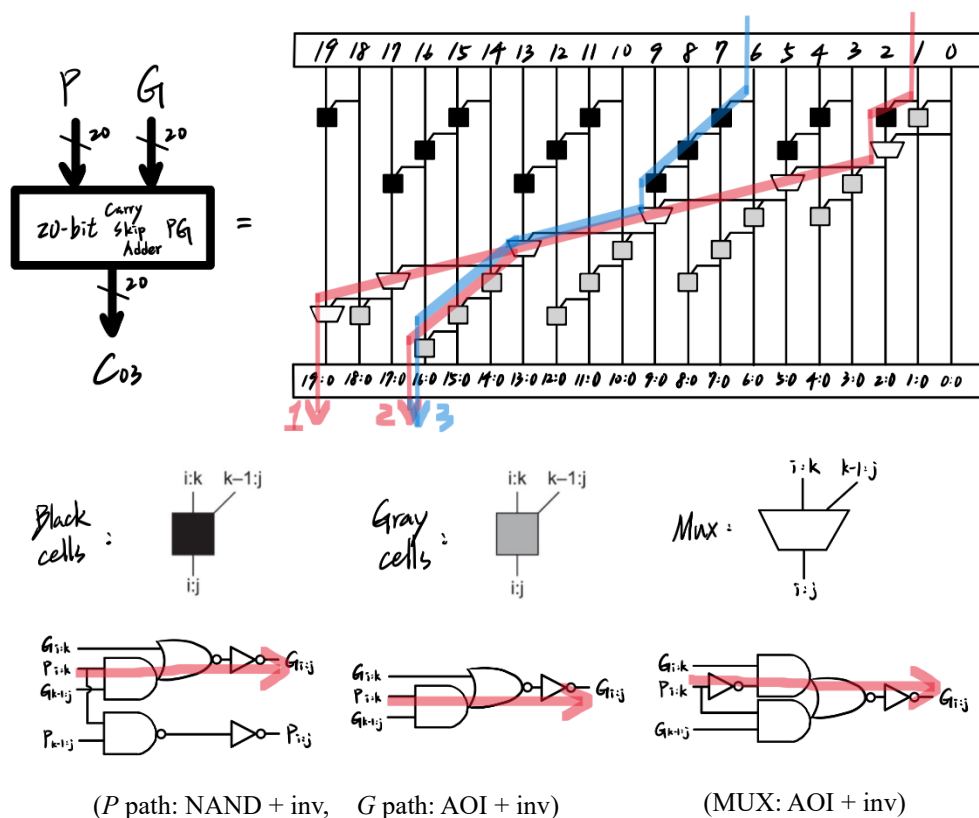
在Bitwise PG的邏輯電路中，與Carry-Save Adder相似，採用複數個PG子電路平行計算的方式，因此Critical path也是複數個延遲相同的路徑。

而每個子電路的Critical path與前面Full Adder的說明相似，XOR的邏輯較為複雜，內部本身就含有Inverter與AND的結構，因此其路徑延遲與NAND + Inverter的延遲差不多(補充：但是XOR除了AND也還有OR的結構在內，因此路徑上的電容可能較NAND更大，延遲通常可能也較大)。

如果 $t_{XOR} > t_{NAND} + t_{inv}$ 其Critical path就會是右邊的Propagate路徑；如果 $t_{XOR} < t_{NAND} + t_{inv}$ 則Critical path就會是左邊的Generate路徑。

• (Carry-Skip Adder) PG Network Block :

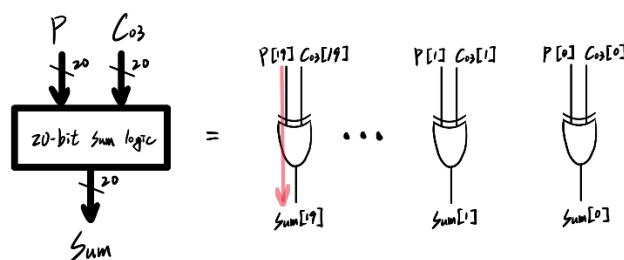
在PG Network中為了降低Critical path的延遲，主要注意第一個與最後一個group的長度(Valency-2 cells)以及MUX數量。我在假設Valency-2 cells與MUX的延遲都一樣的情況下，將MUX的數量盡量減少，還有盡量減少第一個與最後一個group的長度來達到minimum delay，如下圖所示：



如上圖所示，PG Network中的Critical path有多種可能：

- (1) 如果 $t_{MUX2} > 1.5 \cdot t_{Valency-2\ cell}$: Critical path是從輸入端口 1 到輸出端口 19 的路徑。
- (2) 如果 $1.5 \cdot t_{Valency-2\ cell} > t_{MUX2} > t_{Valency-2\ cell}$: Critical path是從輸入端口 1 到輸出端口 16 的路徑。
- (3) 如果 $t_{MUX2} < t_{Valency-2\ cell}$: Critical path是從輸入端口 6 到輸出端口 16 的路徑。

• **(Carry-Skip Adder) Sum Logic Block** :



與前面討論過的情況相似，在Sum Logic中也是複數個平行且相同的路徑，這邊就使用多個XOR gates來實現Sum的邏輯。Critical path就是一個XOR gate的延遲路徑。

- (c) Calculate the critical path delay in terms of the sum of **1-bit FA**, **1-bit PG**, **Mux2**, **Valency-2 cells** and **XOR2** delay, etc.

同前面討論PG network提到的多種可能性，會影響模組的Critical path，因此這邊假設MUX的延遲大於Valency-2 cell的延遲但小於1.5倍，也就是假設： $1.5 \cdot t_{Valency-2\ cell} > t_{MUX} > t_{Valency-2\ cell}$ ，這樣PG network的Critical path就是前面討論的第2條路徑。因此最終的Total critical path為：

$$T_{critical} = 2 \cdot t_{FA} + t_{PG} + (4 \cdot t_{Valency-2\ cell} + 4 \cdot t_{MUX2}) + t_{XOR2}$$

- (d) Indicate the overall module used in terms of the number of 1-bit FA, 1-bit PG, Mux2, Valency-2 cells and XOR2.

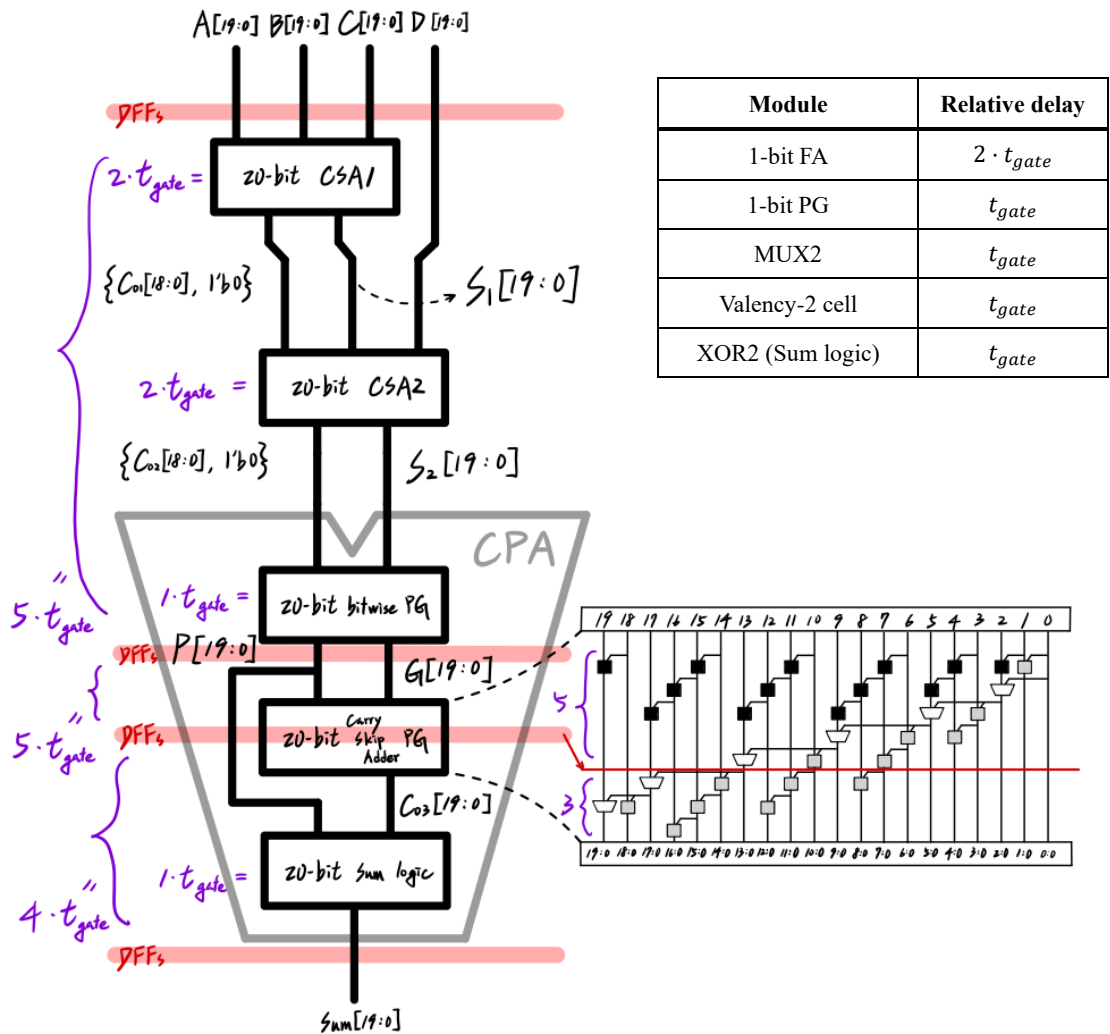
Module	Number of modules
1-bit FA	$2 \times 20 = 40$
1-bit PG	20
MUX2	6
Valency-2 cell	26
XOR2 (Sum logic)	20

2. Pipeline design of 1

- (a) Show the block diagram of the design with three pipelining stages (with DFFs) within the multi-operand adder.

為了適當的切Pipeline stages，我首先假設 $t_{XOR2} = t_{PG} = t_{MUX2} = t_{Valency-2\ cell} = t_{gate}$, $t_{FA} = 2t_{gate}$ ，再帶入回前一題計算的Critical path delay中，得到： $(2 \cdot 2 + 1 + (4 + 4) + 1) \cdot t_{gate} = 14 \cdot t_{gate}$

根據題目要求，如果希望切 3 Pipelining stages 的話因為無法整除，因此這邊採用5+5+4的架構，Block diagram如下：



上圖中紅色標線的地方就是加上DFFs的位置(所有紅線與黑線交疊的位置)，這部分據助教於E3討論區所述不需要提供DFFs的架構圖。

(b) List the number of DFF used.

- Input DFFs:

$$20\text{-bit A} + 20\text{-bit B} + 20\text{-bit C} + 20\text{-bit D} = 20 \times 4 = 80$$

- Pipeline stage-1 DFFs:

$$20\text{-bit P} + 20\text{-bit G} = 20 \times 2 = 40$$

- Pipeline stage-2 DFFs:

$$20\text{-bit P (P}_i) + 7 \text{ Black cells (P}_{i:0}, \text{G}_{i:0}) + 6 \text{ Gray cells (G}_{i:0}) + \\ 4 \text{ MUXs (C}_i) + 3 \text{ Wires (G}_i) = 20 + 7 \times 2 + 6 + 4 + 3 = 47$$

- Pipeline stage-3 (Output) DFFs:

$$20\text{-bit Sum} = 20$$

$$\text{Total number of DFFs: } 80 + 40 + 47 + 20 = 187$$

(c) List the clock cycle time of this pipelining design with tpd of the module used in the critical path.

$$t_{pd1} = 2 \cdot t_{FA} + 1 \cdot t_{PG} = 5 \cdot t_{gate}$$

$$t_{pd2} = 4 \cdot t_{MUX2} + 1 \cdot t_{Valency-2 \text{ cell}} = 5 \cdot t_{gate}$$

$$t_{pd3} = 3 \cdot t_{Valency-2 \text{ cell}} + 1 \cdot t_{XOR2} = 4 \cdot t_{gate}$$

上方推算可以發現Pipeline後的前兩級是影響Cycle time的主要Critical paths，因此在不考慮第三級的情況下，取前兩級的Delay來決定Clock cycle time就會得到：

$$T_c \geq t_{pd1} + t_{setup} + t_{pcq} = (2 \cdot t_{FA} + 1 \cdot t_{PG}) + t_{setup} + t_{pcq}$$

$$T_c \geq t_{pd2} + t_{setup} + t_{pcq} = (4 \cdot t_{MUX2} + 1 \cdot t_{Valency-2 \text{ cell}}) + t_{setup} + t_{pcq}$$