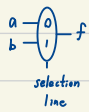
ALU設計

欲設計32-bit ALU，其功能為：add, sub, or, nor, slt, zero, overflow

設計方法為：bit slice，故先建構1個bit的ALU，再用1個bit的ALU去建構32個bit的ALU

建構1-bit ALU使用的邏輯元件為：and, or, not, mux, 1-bit full adder

① Mux：

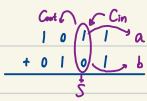


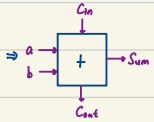
當selection line為0時，則用a作為output

當selection line為1時，則用b作為output

②

1-bit full adder





將所有可能的運算畫作 Truth Table ⟹ 將 Truth Table 化為 Boolean Function

a	b	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

用：• 表手布林代數 and

\+ 表手布林代數 or

$C_{out} = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$

$= \bar{a}bc + a\bar{b}c + ab\bar{c} + abc + abc + abc$ (∵ $abc + abc = abc$)

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

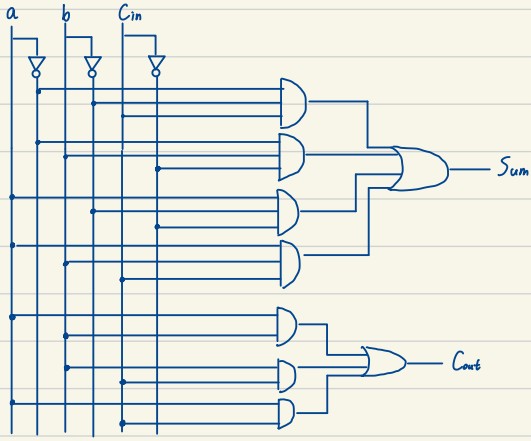
$= bc + ac + ab$

$Sum = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$

再將 Boolean function 實作成電路：

$C_{out} = ab + bc + ac$

$Sum = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$ ⟹



其中：以邏輯閘作為延遲時間基本單位

Sum 最長需通過 not ⟶ and ⟶ or

故 sum delay 為 3個 gate delay

Carry In 最長則需通過 and ⟶ or

故 carryin delay 為 2個 gate delay

③ 1-bit ALU

I. 先設計 and, or, add

利用Mux來控制輸出

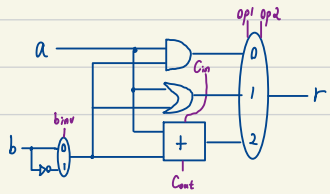
op1, op2 為 selection line

00：and 01：or 10：add

加入 sub

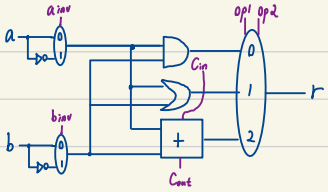
若以＝補數表示

則：$a - b = a + \bar{b} + 1$

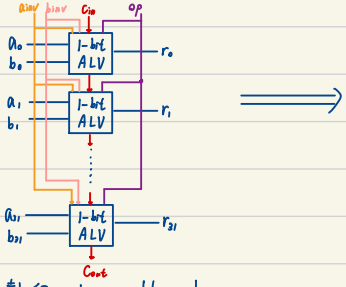


再來加入 nor 運算實現： $\overline{a+b} = \bar{a} \cdot \bar{b}$

II. and, or, nor, add, sub



III. 用 1 bit 之 II 的 ALU 建構 32 bit ALU

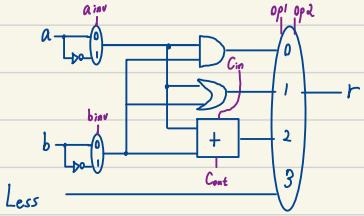


目前可執行 and, or, add, sub, nor

可執行 and, or, add, sub, nor, slt 之控制信号：

	ainv	binv	Cin	OP1	OP2
and	0	0	X	0	0
or	0	0	X	0	1
add	0	0	0	1	0
sub	0	1	1	1	0
nor	1	1	X	0	0
slt	0	1	1	1	1

但支援 slt 的 1 bit ALU 需重新設計



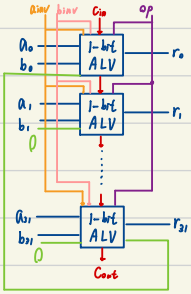
再來，實現 slt 運算， slt $s1, $s2, $s3

若 $s2 < $s3 則 $s1 為 1，反之為 0

⇒ 用減法運算： s2 - s3 若為負，則 s2 < s3

s2 - s3 結果的 r31 若為 1，則 s1 為： 000…01

反之，若為 0，則 s1 為 00…0，故直接傳結果的 r31 作為 r0



目前可執行 and, or, add, sub, nor, slt

其實也可以作 NAND ($\overline{a \cdot b} = \bar{a} + \bar{b}$)

再來，加入 Overflow 和 Zero 輸出線

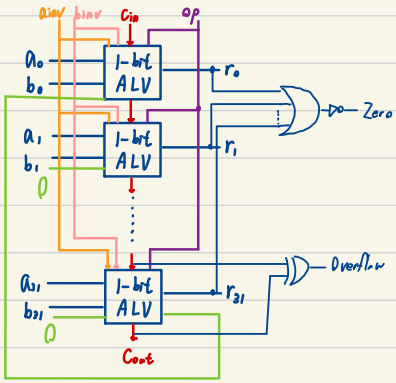
① Zero： NOR 所有的 r31 ~ r0 即可

② Overflow：

0110 6		0011 3	
+ 0101 5		+ 0100 4	
$C_{in}=1$, $C_{out}=0$		$C_{in}=0$, $C_{out}=0$	
1001 -7		1101 -3	
+ 1011 -5		+ 1110 -2	
$C_{in}=0$, $C_{out}=1$		$C_{in}=1$, $C_{out}=1$	

觀察可知： $C_{in} \otimes C_{out} = 1$ 時，代表有 Overflow

加入 Overflow & Zero 的 32 bit ALV



	ainv	binv	Cin	OP1	OP2
and	0	0	X	0	0
or	0	0	X	0	1
add	0	0	0	1	0
sub	0	1	1	1	0
nor	1	1	X	0	0
slt	0	1	1	1	1

∵ binv 和 Cin 的輸出相同

∴ 可合併成一個 b negate

計算机運算中，加、減、乘、除都是利用加法運算實現的

∴ 加法器效能大幅地影响整台計算机效能

Ripple Carry Adder：硬件成本最低，速度最慢

以 4-bit 加法為例：

C_4 a_3 a_2 a_1 a_0 C_0

0 1 1 1

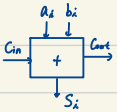
\+ 0 0 0 1

b_3 b_2 b_1 b_0

1 0 0 0

S_3 S_2 S_1 S_0

Review：1-bit full adder 設計

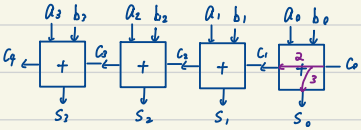


carry-out delay：2个 gate delay

sum delay：3个 gate delay

4-bit ripple carry adder 設計如下：

上一个位元的 full adder 的 carry out 會傳遞到下一个 bit 的 full adder 作為輸入



Critical Path Delay：從輸入到產生全部的進位所需的 gate delay

Sum delay：從輸入到產生全部的 sum 所需的 gate delay

Critical path delay：$2N$

Sum delay：$2N+1$（最後一个 full adder 需 $2N-2$ gate delay 得到 C_N，又需 3个 gate delay 得到 S_N）

加入一些 logic blocks，可完成 2's complement substraction（$S = A - B$）

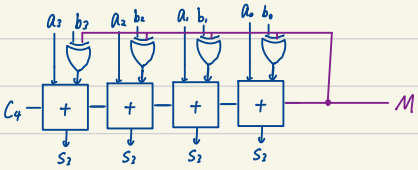
令 $M=0$ 時做 $S = A+B$，$M=1$ 時，$S = A-B$

又：$S = A - B = A + \overline{B} + 1$（設不支援 Mux）

則：$\overline{B} = B \otimes 1$ ∴ $S = A + (B \otimes 1) + 1$

$S = A + B = A + (B \otimes 0) + 0$

∴ $S = A + (B \otimes M) + M$ ⇒ 电路設計如右



無限硬件快速進位加法器

Carry out 的 Boolean Function 為：$C_i = a_{i-1}b_{i-1} + a_{i-1}C_{i-1} + b_{i-1}C_{i-1}$

即 $C_1 = a_0 b_0 + a_0 C_0 + b_0 C_0$

$C_2 = a_1 b_1 + a_1 C_1 + b_1 C_1$ ⇒ 可將 C_1 代入，變成：$C_2 = F(a_1, b_1, a_0, b_0, C_0)$

$C_3 = F(a_2, b_2, a_1, b_1, a_0, b_0, C_0)$

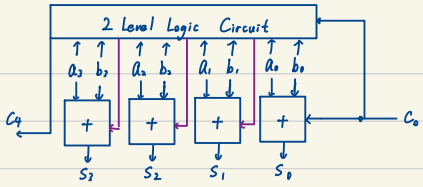
$C_4 = F(a_3, b_3, a_2, b_2, a_1, b_1, a_0, b_0, C_0)$

設在一 2-level logic circuit 中實現這4个 Boolean Function，就不用靠 ripple carry 來傳 carry out 了

又 2-level logic circuit 會為 sum of product：

⇒ ∴ 只需 2 gate delay 即可得到所有的 Carry

4 bit 的無限硬件快速進位加法器如下：



critical path delay : 2 gate delay

sum delay : 5 gate delay (2+3)

但：位元數越多，需 AND gate 越多（指數成長）

硬件成本很高

Carry Lookahead Adder (CLA)

概念：

a	b	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

觀察可得：當 a.b 為 0 時，Cout 為 0，和 Cin 無關 (kill)

當 a.b 為 1 時，Cout 為 1，和 Cin 無關 (generate)

⇒

a	b	Cin	
0	0	0	kill
0	1	Cin	propagate
1	0	Cin	propagate
1	1	1	generate

Boolean Function : $C_{out} = ab + (a+b)C_{in}$

$\Rightarrow C_{i+1} = a_i b_i + (a_i + b_i) C_i$

Define : $g_i = a_i b_i$, g_i 為 generate

$p_i = a_i + b_i$, p_i 為 propagate

則 : $C_{i+1} = g_i + p_i C_i$

已知：$C_{i+1} = g_i + p_i C_i$，以 4 bit adder 為例：

$C_1 = g_0 + p_0 C_0$

$C_2 = g_1 + p_1 C_1 = g_1 + p_1 g_0 + p_1 p_0 C_0$

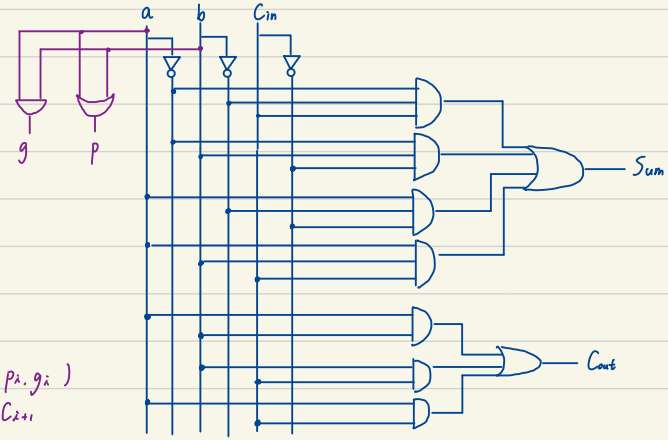
$C_3 = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 C_0$

$C_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 C_0$

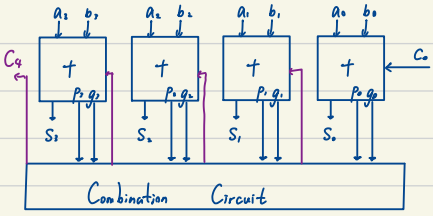
這樣一來，$C_{i+1} = F(p_i, g_i, C_0)$，一levelCombination Circuit，以 2-level logic 方式得到所有的 C_i

一樣為 2 Gate delay

1-bit adder 需加入 p_i, g_i 輸出設計：



而 4-bit 加法器電路設計如下



Critical path delay : 3 (1+2，先花 1 gate delay 產 p_i, g_i)

再花 2 gate delay 得到 C_{i+1}

Sum delay : 6 (3+3)