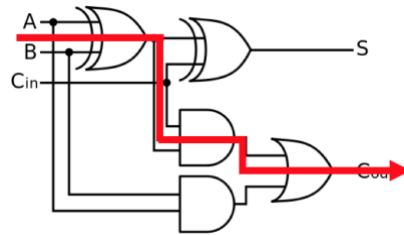


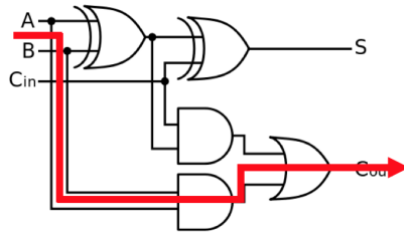
## 1. 8-bit Carry Ripple Adder

### (3) Critical path of 8-bit carry ripple adder (CRA)

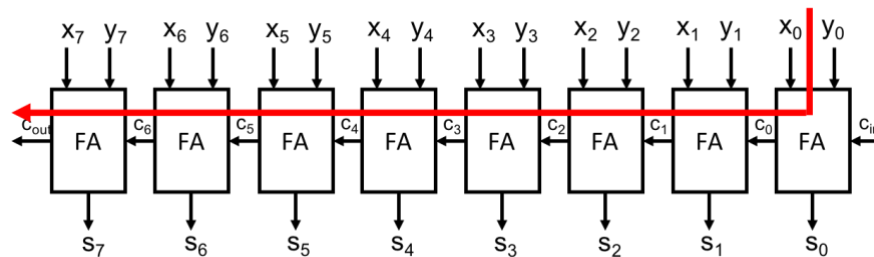
首先考慮 LSB Full adder，一般來說 Full adder 的 critical path 上有三個 gate，如下：



不過在本題中 LSB 的 carry in 固定為 0，故 LSB 的 critical path 為下圖兩個 gate：



接著考慮整個 CRA 的 critical path，如下圖：



本題的條件下 ( $c_{in} = 0$ , gate delay = 2ns)，第一個 FA 的 propagation delay 為 2ns，其他 FA 由  $c_{in}$ - $c_{out}$  的 delay 也為 2ns，總和為 **16ns**。

#### ■ Simulation

要使 CRA 有 worst-case propagation delay，使用的測資為  $x = 0xFF$ ,  $y = 0x01$ 。

當 clock cycle 為 16ns 時，可以通過 testbench：

```
===== Congratulation! You Pass! =====
```

當 clock cycle 為 15ns 時，無法通過上述 worst-case：

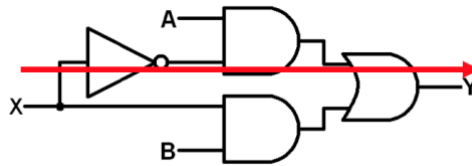
```
An ERROR occurs at no.      11 pattern: {carry out, output} 000 != answer 100.

There are      1 errors.
```

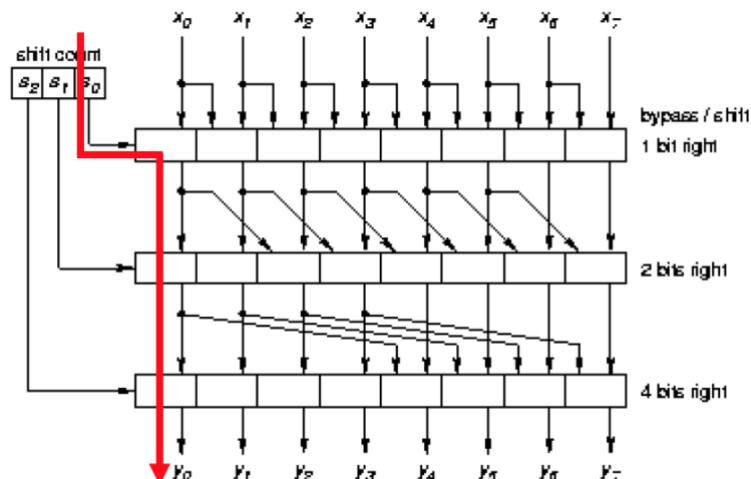
## 2. 8-bit Barrel Shifter

### (3) Critical path of 8-bit barrel shifter

首先考慮第一層中的 1-bit MUX 的 critical path，經過以下三個 gate：



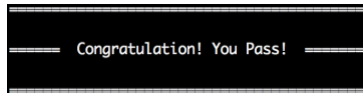
則 8-bit barrel shifter 的 critical path 為：



第一層 delay 為 3ns，其他層為 2ns，總和為 **7ns**。

### ■ Simulation

當 clock cycle 為 7ns 時，可以通過 testbench：



當 clock cycle 為 6ns 時則無法通過：

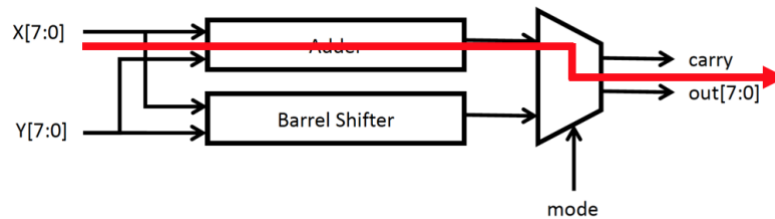
```
An ERROR occurs at no.      1 pattern: Output x0x0x0x0 != answer 10101010.
An ERROR occurs at no.      2 pattern: Output 11111110 != answer 01010100.
An ERROR occurs at no.      3 pattern: Output 00000000 != answer 10101000.
An ERROR occurs at no.      4 pattern: Output 11111000 != answer 01010000.
An ERROR occurs at no.      5 pattern: Output 00000000 != answer 10100000.
An ERROR occurs at no.      6 pattern: Output 11100000 != answer 01000000.
An ERROR occurs at no.      7 pattern: Output 00000000 != answer 10000000.
An ERROR occurs at no.      8 pattern: Output 10000000 != answer 00000000.

There are      8 errors.
```

### 3. Adder-Shifter Unit

#### (2) Critical path of adder-shifter unit (ASU)

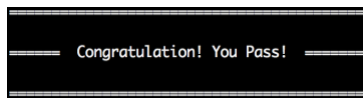
由前面實驗可以知道 ASU 的 critical path 應該會通過 adder，如下：



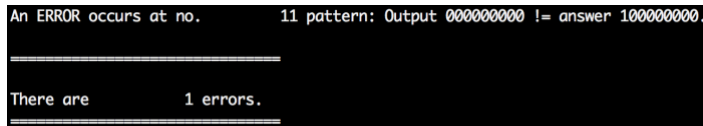
其中 adder 的 delay 為 16ns，mode multiplexer 為 2.5ns，總和為 **18.5ns**。

#### ■ Simulation

當 clock cycle 為 18.5ns 時，可以通過 testbench：



當 clock cycle 為 18ns 時則無法通過：



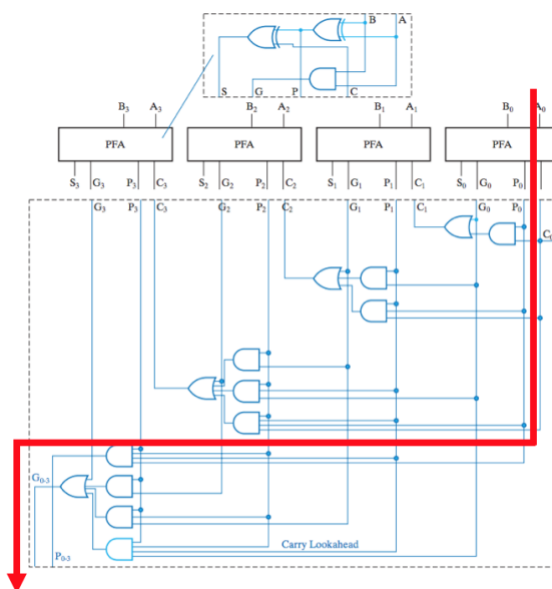
#### (3) Adder optimization

由於 adder 造成 critical path，故優化 adder。

以下實作兩種不同的 fast adder 架構進行比較

#### ■ Carry lookahead adder (CLA)

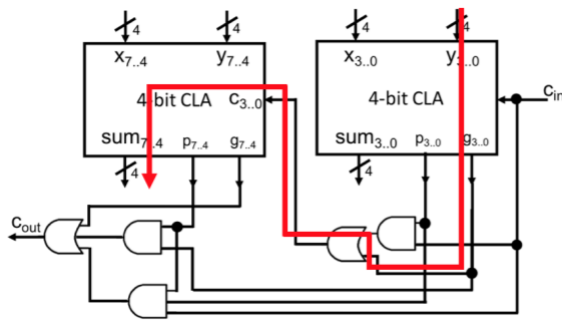
首先實作 4-bit carry lookahead adder<sup>1</sup>：



其 Propagation delay 為 3ns。

接著在兩個 4-bit CLA 的輸出再加上 carry lookahead logic，如下：

<sup>1</sup> Source : [http://www.eng.ucy.ac.cy/theocharides/Courses/ECE210/Carrylookahead\\_supp4.pdf](http://www.eng.ucy.ac.cy/theocharides/Courses/ECE210/Carrylookahead_supp4.pdf)

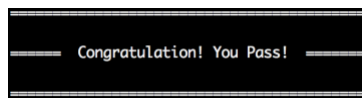


此為  $c_{in} = 0$  的條件下的 critical path，其 propagation delay 為：

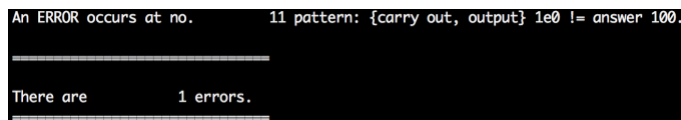
$$3ns (y_{3..0}-g_{3..0}) + 1ns (OR) + 3ns (c_{3..0}-sum_{7..4}) = 7ns \circ$$

### - Simulation

當 clock cycle 為 7ns 時，可以通過 testbench：

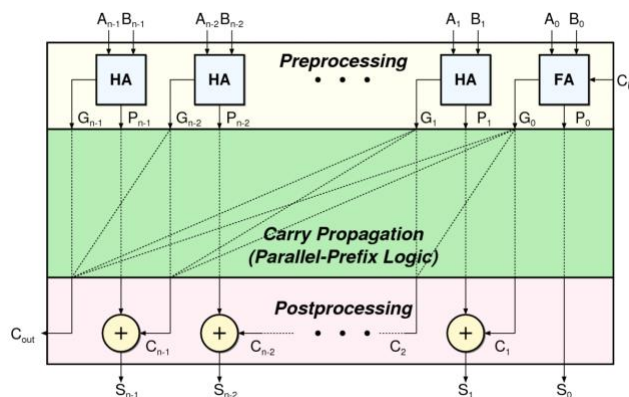


當 clock cycle 為 6ns 時則無法通過 worst-case：

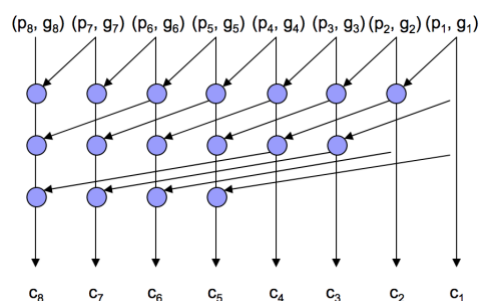


## ■ Kogge-Stone parallel prefix adder (KS)

Parallel prefix adder 架構如下<sup>2</sup>：



其中第二階段的 parallel-prefix logic 可有不同架構，KS 為其中一種：



Propagation delay 為  $x_0-c_{out}$ ：1ns (half adder) + 6ns (parallel-prefix) = 7ns  $\circ$

<sup>2</sup> [http://www.syssec.ethz.ch/content/dam/ethz/special-interest/infk/inst-infsec/system-security-group-dam/education/Digitaltechnik\\_14/13\\_AdvancedAdders.pdf](http://www.syssec.ethz.ch/content/dam/ethz/special-interest/infk/inst-infsec/system-security-group-dam/education/Digitaltechnik_14/13_AdvancedAdders.pdf)

#### - Simulation

當 clock cycle 為 7ns 時，可以通過 testbench：

```
===== Congratulation! You Pass! =====
```

當 clock cycle 為 6ns 時則無法通過：

```
An ERROR occurs at no.      3 pattern: {carry out, output} 182 != answer 102.
An ERROR occurs at no.      11 pattern: {carry out, output} 0d0 != answer 100.

There are      2 errors.
```

#### ■ Applying fast adders to ASU

本次實作兩種 fast adder 的 propagation delay 皆為 7ns，和 barrel shifter 相同，故 ASU 的 propagation delay 應為 9.5ns。

#### - Simulation

當 clock cycle 為 9.5ns 時，可以通過 testbench：

```
===== Congratulation! You Pass! =====
```

當 clock cycle 為 9ns 時則無法通過：

```
An ERROR occurs at no.      1 pattern: Output 0xxxxxxx != answer 010101010.
An ERROR occurs at no.      2 pattern: Output 010101010 != answer 001010100.
An ERROR occurs at no.      3 pattern: Output 001010100 != answer 010101000.
An ERROR occurs at no.      4 pattern: Output 010101000 != answer 001010000.
An ERROR occurs at no.     11 pattern: Output 100011110 != answer 100000000.

There are      5 errors.
```

#### (4) Unsigned multiplier using ASU

兩個 8-bit 的正整數可以視為 8 個位移和 8 個加法運算，例如：

$$\begin{aligned} & 10010110 \times 01101110 \\ = & \begin{aligned} & 10010110 \ll 1 \\ & + 10010110 \ll 2 \\ & + 10010110 \ll 3 \\ & + 10010110 \ll 5 \\ & + 10010110 \ll 6 \end{aligned} \end{aligned}$$

這些步驟可以分為 16 個 clock cycle 完成，在奇數 cycle 時進行位移運算，當第 k 個位元為 1 時將  $x \ll k$  存入 register，否則存入 8'b0。在偶數 cycle 時將目前累加的值與上個 cycle 計算的  $x \ll k$  相加，再存回 register。如此就能做出乘法器。

注意到位移運算的部分需要一個計數器（counter），可以利用三個 flip-flop 來實現 ripple counter。