

Design Description :

Hamming code 也稱為海明碼，是(7,4)漢明碼推廣得到的一種線性錯誤更正碼，由理察·衛斯里·漢明於 1950 年發明。遵循下面的範例中的步驟，將可以檢測到一位元(1bit)的錯誤並進行修正。

1. Hamming code decode

首先會接收到一組 7 位元的訊息，其中四位元為原始信息(X1, X2, X3, X4)，剩下的三位元為奇偶校驗碼(P1, P2, P3)。

P1	P2	P3	X1	X2	X3	X4
----	----	----	----	----	----	----

其中(X1, X2, X3, X4)和(P1, P2, P3)會遵守以下的關係:

$$P1 = X1 \oplus X2 \oplus X3$$

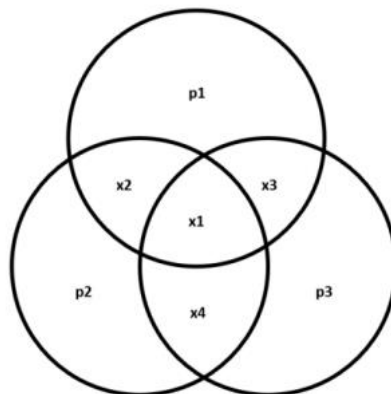
$$P2 = X1 \oplus X2 \oplus X4$$

$$P3 = X1 \oplus X3 \oplus X4$$

\oplus 是 XOR，真值表如下:

p	q	$p \oplus q$
True	True	False
True	False	True
False	True	True
False	False	False

步驟 1:將每個位元填入下列的圖形



步驟 2: 檢查每個大圓的總和是否為偶數

如果總和為偶數，則表示接受的訊息為正確。

如果總和為奇數，則需要找出錯誤的單位元(1bit)並進行修正。

範例: Circle1: $P1 \oplus X1 \oplus X2 \oplus X3 = 0$;

Circle2: $P2 \oplus X1 \oplus X2 \oplus X4 = 0$;

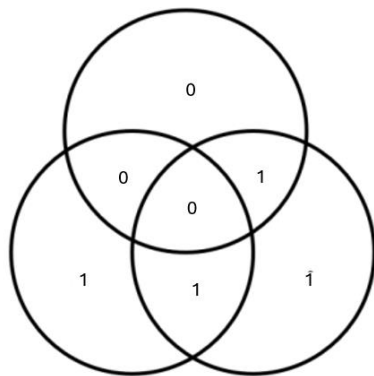
Circle3: $P3 \oplus X1 \oplus X3 \oplus X4 = 0$;

步驟 3: 修正兩個訊息中錯誤的位元，並得到錯誤位元和原始訊息。

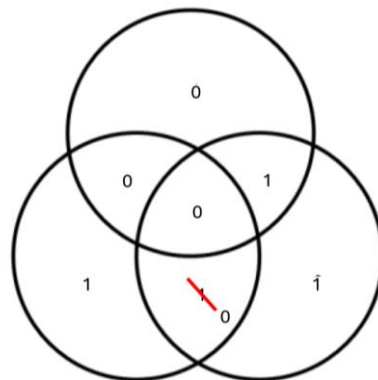
範例:

receive code word = $(51)_{10} = (011\ 0011)_2$;

correct code word = $(50)_{10} = (011\ 001\mathbf{0})_2$;



receive code word



correct code word

2. Code calculator

$\text{opt}[1:0] = \{\text{code_word1's error bit, code_word2's error bit}\}$

$c1 = \text{code_word1's correct original information (signed) } (X1, X2, X3, X4)$

$c2 = \text{code_word2's correct original information (signed) } (X1, X2, X3, X4)$

$\text{opt} = (0,0); \quad \text{out_n} = 2*c1 + c2;$

$\text{opt} = (0,1); \quad \text{out_n} = 2*c1 - c2 ;$

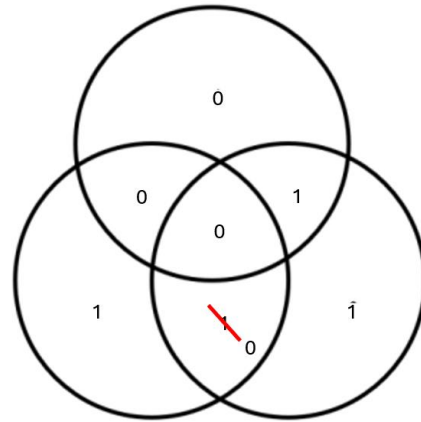
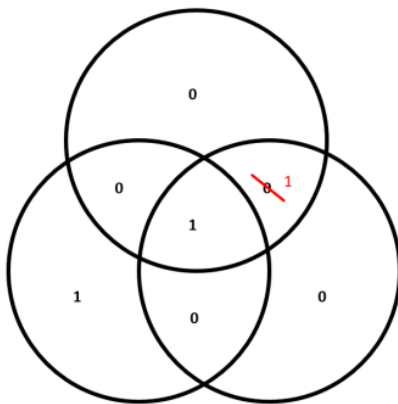
$\text{opt} = (1,0); \quad \text{out_n} = c1 - 2*c2 ;$

$\text{opt} = (1,1); \quad \text{out_n} = c1 + 2*c2;$

Example:

code_word1 is $(40)_{10} = (011\ 10\mathbf{00})_2$

code_word2 is $(51)_{10} = (011\ 001\mathbf{1})_2$



1. 使用漢明碼修正後，正確的訊息:

code_word1 為 $(011\ 10\mathbf{10})_2$ ，code_word2 為 $(011\ 001\mathbf{0})_2$.

2. code_word1 修正後的正確數值為 $c1 = (1010)_2 = (-6)_{10}$.

code_word2 修正後的正確數值為 $c2 = (0011)_2 = (3)_{10}$.

3. $\text{opt}[1:0] = \{\text{code_word1's 錯誤的位元}(\mathbf{0}), \text{code_word2 錯誤的位元}(\mathbf{1})\} = (01)_2$.

4. $\text{opt}[1:0]$ 為 $(01)_2$ 時， $\text{out_n} = 2*c1 - c2$.

5. $\text{out_n} = 2*c1 - c2 = 2*(-6) - (3) = -15$.

輸入訊號	Signed/Unsigned	Bit Width	定義
code_word1	unsigned	7	包含一個錯誤的 7 位元漢明碼。
code_word2	unsigned	7	包含一個錯誤的 7 位元漢明碼。

輸出訊號	Signed/Unsigned	Bit Width	定義
out_n	signed	6	$\text{opt}[1:0] = \{\text{code_word1's error bit, code_word2's error bit}\}$ $\text{opt} = (0,0); \text{ out_n} = 2*c1 + c2;$ $\text{opt} = (0,1); \text{ out_n} = 2*c1 - c2 ;$ $\text{opt} = (1,0); \text{ out_n} = c1 - 2*c2 ;$ $\text{opt} = (1,1); \text{ out_n} = c1 + 2*c2;$

NOTE: (請詳細閱讀後再進行作業撰寫，如果來詢問以下已

告知的事情會斟酌扣分，謝謝)

1. 撰寫 RTL code 時請注意以下兩點

A. 須將檔名儲存成 **HD.v**

B. Module name 請設為 **HD**，如下圖：

```
module HD(  
    //input signals pattern  
    code_word1,  
    code_word2,  
    //output signals  
    out_n  
);
```

C. 需有簡單註釋並將註釋連同 RTL code 截圖放入報告內或**直接在報告內進行 RTL code 解釋**。

2. 寫完 RTL code 後請在已架設好環境的伺服器(請參考本檔案最下面)中打開 Terminal 並來到你儲存檔案的位置，並確保該資料夾下有以下檔案：

A. HD.v

B. TESTBED.vp

C. PATTERN.vp

D. output.txt

E. input_a.txt

F. input_b.txt

除了 HD.v 為自己撰寫外，**請勿修改其他檔案**以免造成驗證錯誤

3. 驗證 Verilog 的方式請在 Terminal 中輸入

ncverilog TESTBED.vp -define RTL -debug -notimingchecks

請注意各個指令間的空格，耐心等候驗證後，如出現下圖代表實驗成功：

