

# LAB 01 - Setup and Instruction Set

- MPLAB X IDE 的下載及安裝：

- Link: <https://www.youtube.com/watch?v=3FtxzCHZzQM>

- Introduction to Instruction Set：

- Link: <https://youtu.be/Qf0pMBAEvjg>

- Lab requirements：

- 基本題 (70%)：

- 題目敘述：

將四個數(以下稱作x1,x2,y1,y2)分別存入[0x000], [0x001], [0x002], [0x003], 且分別求得 $x1 + x2$ 並存入[0x010],  $y1 - y2$ 存入[0x011], 最後判別[0x010], [0x011]中的數字是否相等, 若相等將[0x020]設為 0xFF, 反之設為 0x01。

(測資中 $x1 + x2$ 不會大於 0xFF, 且 $y1 > y2$ 。)

- 範例測資：

[0x000]	[0x001]	[0x002]	[0x003]	[0x010]	[0x011]	[0x020]
0x04	0x02	0x0A	0x04	0x06	0x06	0xFF
0x07	0x08	0x0D	0x0C	0x0F	0x01	0x01

- 評分標準：

1. 需使用 CPFSEQ 指令, 加減法的部分可以使用 ADDWF, SUBWF 等, 細節自行設計。
2. 結果必須存放至[0x020]中。

● 進階題 (30%) :

➤ 題目敘述 :

將兩數存放於[0x000], [0x002] , 並設計一迴圈 , 計算[0x000]中的數字是否為 2 的倍數 , 是的話把[0x002]加上一 , 否則減一 , 之後再將[0x000]中的數字向右移 , 反覆做至[0x000]為原本數字為止。

➤ 範例測資 :

假設[0x000]的數為b'00001010', [0x002]為 0x10。

Step	[0x000]	[0x002]
1	00001010	0x11
2	00000101	0x10
3	10000010	0x11
4	01000001	0x10
5	10100000	0x11
6	01010000	0x12
7	00101000	0x13
8	00010100	0x14
9	00001010	Do nothing

➤ 評分標準 :

1. 請使用迴圈完成此題 , 不能使用任何暴力手段 , 善用 GOTO, DECFSZ 等指令。
2. 必須使用指令 RRNCF, DECF, INCF。
3. 結果必須存放至[0x002]中。

● 加分題 (20%)：

➤ 題目敘述：

請利用 **ANDWF**、**IORWF**、**COMF** 指令，實作出一個 8bits 的格雷碼的轉換器，**不能使用 XORWF**。

➤ 範例測資：

以下是 0~7 的二進位轉格雷碼的表：

十進位	二進位	格雷碼
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

例: 0b01100010 轉換後為 0b01010011

➤ 評分標準：

1. 運算邏輯只能使用 **ANDWF**、**IORWF**、**COMF** 指令(可以只選擇一種或兩種使用，不一定要三個都使用到)，其餘部分可以自行設計。
2. 不能使用 **XORWF** 指令。
3. 轉換後結果必須存放至[0x001]中。

提示: 善用 AND、OR、NOT 邏輯運算

# LAB 01 - Setup and Instruction Set

- MPLAB X IDE Download and Install

- Link: <https://www.youtube.com/watch?v=3FtxzCHZzQM>

- Introduction to Instruction Set

- Link: <https://youtu.be/Qf0pMBAEvjg>

- Lab requirements:

- **Basic (70%) :**

- **Description :**

Store four numbers (x1, x2, y1, y2) in memory locations **[0x000]**, **[0x001]**, **[0x002]**, and **[0x003]** respectively. Calculate the sum of x1 and x2, storing the result in **[0x010]**. Similarly, compute the difference between y1 and y2, and store the result in **[0x011]**. Finally, check if the values in **[0x010]** and **[0x011]** are equal. If they are, set the value of **[0x020]** to **0xFF**; otherwise, set the value of **[0x020]** to **0x01**.

(Note: The sum of x1+x2 will not exceed 0xFF in the test data, and y1 will always be greater than y2.)

- **Sample test data :**

<b>[0x000]</b>	<b>[0x001]</b>	<b>[0x002]</b>	<b>[0x003]</b>	<b>[0x010]</b>	<b>[0x011]</b>	<b>[0x020]</b>
0x04	0x02	0x0A	0x04	0x06	0x06	<b>0xFF</b>
0x07	0x08	0x0D	0x0C	0x0F	0x01	<b>0x01</b>

- **Criteria :**

1. You must use the **CPFSEQ** instruction. You can use **ADDWF**, **SUBWF**, and other similar instructions for addition and subtraction operations in your design accordingly.
2. The result must be stored in **[0x020]**.

• **Advanced (30%) :**

➤ **Description :**

Save a pair of numbers in [0x000] and [0x002], then create a loop to check if the number in [0x000] is **even**. If it is, add one to the value in [0x002]; if not, subtract one. Then, **shift the number** in [0x000] **to the right** until you return to the original number in [0x000].

➤ **Sample test data :**

Assuming the value in [0x000] is b'00001010', and [0x002] is 0x10

Step	[0x000]	[0x002]
1	00001010	0x11
2	00000101	0x10
3	10000010	0x11
4	01000001	0x10
5	10100000	0x11
6	01010000	0x12
7	00101000	0x13
8	00010100	0x14
9	00001010	Do nothing

➤ **Criteria :**

1. Please use a **loop** to complete this task **without using any brute force methods**. Make good use of instructions like **GOTO**, **DECFSZ**.
2. You must use the instructions **RRNCF**, **DECF**, **INCF**
3. The result must be stored in [0x002]

● **Bonus (20%) :**

➤ **Description :**

Please implement an 8-bit Gray code converter using the **ANDWF**, **IORWF**, and **COMF** instructions. Using **XORWF** instruction is not allowed.

➤ **Sample test data :**

This table shows the conversion of binary numbers from 0 to 7 into Gray code.

Decimal	Binary	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Example: The binary number “0b01100010” will be converted to “0b01010011” in the Gray code format.

➤ **Criteria :**

1. You can only employ the **ANDWF**, **IORWF**, and **COMF** instructions for logical operations. You have the option to choose one or two of these instructions; it's not mandatory to use all three. The remaining instructions can be custom-designed.
2. **XORWF** instruction is not allowed.
3. The converted result must be stored in **[0x001]**.

**Hint: Make good use of AND, OR, and NOT logical operations.**