

Lab 3: Requirement Description

- Standard Instruction Set 教學

- 影片:

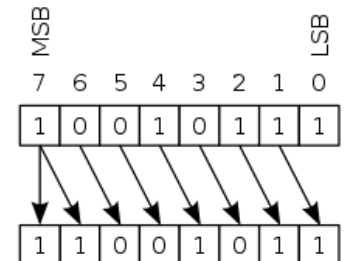
<https://youtu.be/kp7-VC0SKnU>

- 基礎題 (70%) :

- 題目敘述：利用 bitwise 指令 (NOT, AND, OR, XOR, ROTATE) 去實作算術右移 TRISA (right arithmetic shift TRISA)。

- 例如：

1. 原本 TRISA 的值是 0b10010111，算術右移後變為 0b11001011。
2. 原本 TRISA 的值是 0b00001111，算術右移後變為 0b00000111。



- 評分標準：

1. 請初始化 TRISA 為 0xC2。
2. 會檢查是否使用至少一個上述提到的 bitwise 指令。
3. 請將 TRISA 的內容向右移一個位元。
4. 並將結果存回 TRISA。
5. 不能使用 branch 相關的指令。

F92	TRISA	0xC2	194	11000010
-----	-------	------	-----	----------

Before

F92	TRISA	0xE1	225	11100001
-----	-------	------	-----	----------

After

- 進階題 (30%) :

- 題目敘述：實做一個 4x4 bits unsigned multiplier, 並將結果存到 TRISA, 給定兩個輸入分別存在 TRISB 和 TRISC (range from 0x01 to 0x0F)。

- 例如: TRISB 存入 0x08, TRISC 存入 0x0B, 經過運算後將結果 0x58 存入 TRISA。

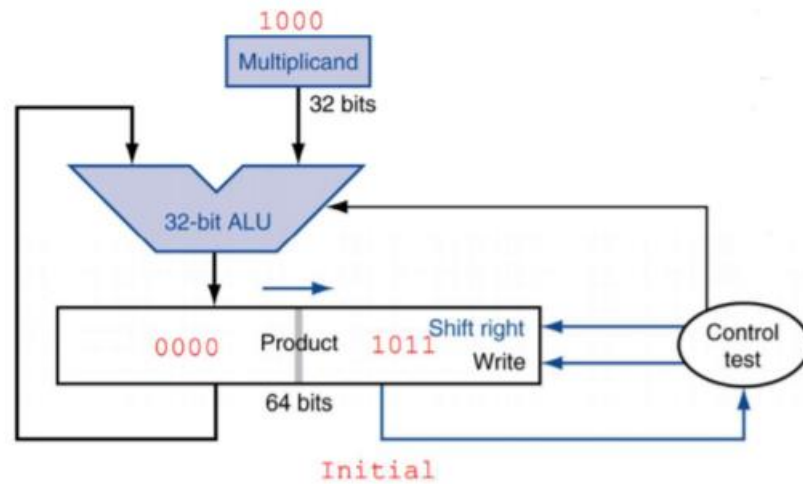
$$\begin{array}{r} 1000 \\ \times 1011 \\ \hline 1000 \\ 1000 \\ 0000 \\ 1000 \\ \hline 01011000 = 0x58 \end{array}$$

- 評分標準：

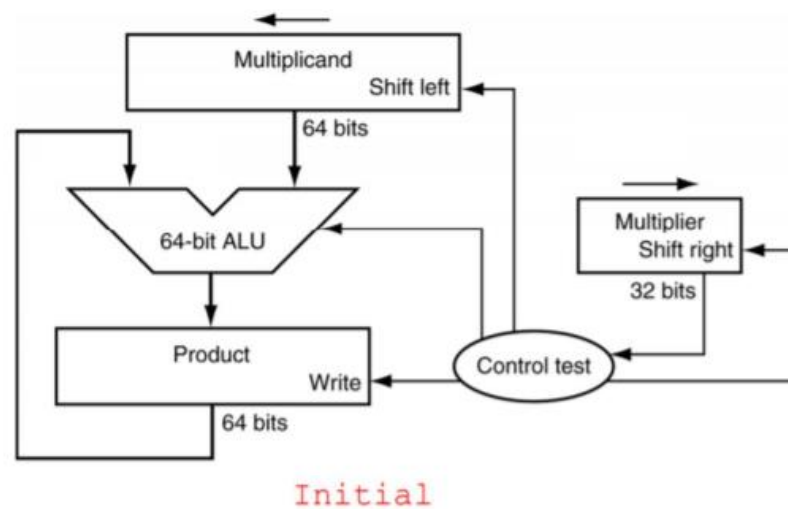
1. 請分別初始化 TRISB、TRISC 為 0x0D、0x06。並將結果存至 TRISA，會檢查 TRISA 的結果是否正確。

2. 請使用下列乘法器架構 (either **optimized multiplier** or **simple multiplier**) 去完成。乘法器架構請參考下圖：

- **optimized multiplier**



- **simple multiplier**



3. 請不要直接使用加法展開如下：
 $00001000 + 00010000 + 00000000 + 01000000$ (X)
4. 請不要使用 **MUL** 相關的指令。
5. 請使用 **ROTATE** 相關的指令。
 (RLCF/RLNCF/RRCF/RRNCF)
6. 請不要使用 **goto** 指令
 你可以用以下 **branch** 相關的 7 個指令 去完成此次 lab。.
 (BC/BN/BNC/BNN/BNOV/BNZ/BOV)

F92	TRISA	0x4E	78
F93	TRISB	0x0D	13
F94	TRISC	0x06	6

Result

● 加分題 (20%) :

- 題目敘述：給定兩個 unsigned 8-bit 非零整數 A, B (range from 0x01 to 0x7F) 分別儲存在 data memory 0x000 和 0x001 的位址 (A >= B)。請使用迴圈找出兩個數的 GCD (greatest common divisor) 並將結果儲存於 data memory 0x002 的位址。
- 例如：Data memory 0x000 的位址存入 0x40，data memory 0x001 的位址存入 0x28，經過運算將結果 0x08 存入 data memory 0x002 的位址。
- 評分標準：
 1. 同學需要在 demo 時呈現兩個輸入和其輸出結果，如下圖。
 2. 會檢查是否使用迴圈
- 提示：同學可以使用 COMPARE 指令去完成此加分題。

Address	00	01	02
000	40	28	08

Lab 3: Requirement Description

- Standard Instruction Set Guideline

- Video:
<https://youtu.be/kp7-VC0SKnU>

- Basic (70%) :

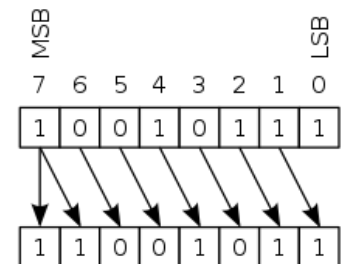
- Description: Use **bitwise instructions (NOT, AND, OR, XOR, ROTATE)** to implement **right arithmetic shift TRISA**.

- Example:

1. Before shifting, value of TRISA is 0b10010111, after right arithmetic shift, value of TRISA is 0b11001011.
2. Before shifting, value of TRISA is 0b00001111, after right arithmetic shift, value of TRISA is 0b00000111.

- Standard of grading:

1. Please initialize the value of **TRISA** to 0xC2.
2. We will check whether you use **at least 1 bitwise** instruction mentioned above.
3. The content of **TRISA** should be rotated **one bit** to the right.
4. The result should be stored back in **TRISA**.
5. **Branch** instructions are not allowed.



F92	TRISA	0xC2	194	11000010
Before				
F92	TRISA	0xE1	225	11100001
After				

- Advanced (30%) :

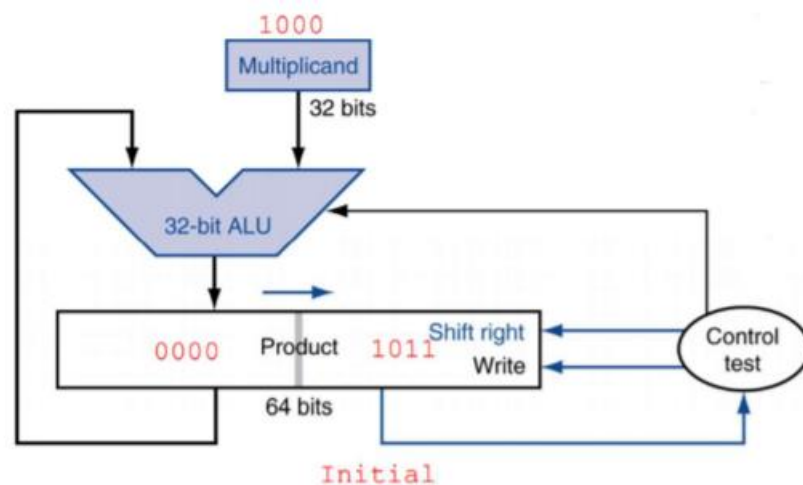
- Description: Write a **4x4 bits unsigned multiplier**, then store the result in register **TRISA**, given two number stored in **TRISB** and **TRISC (range from 0x01 to 0x0F)**.
- Example: Given two numbers 0x08, 0x0B stored in TRISB, TRISC, and store the result 0x58 in TRISA.

$$\begin{array}{r}
 1000 \\
 \times 1011 \\
 \hline
 1000 \\
 1000 \\
 0000 \\
 1000 \\
 \hline
 01011000 = 0x58
 \end{array}$$

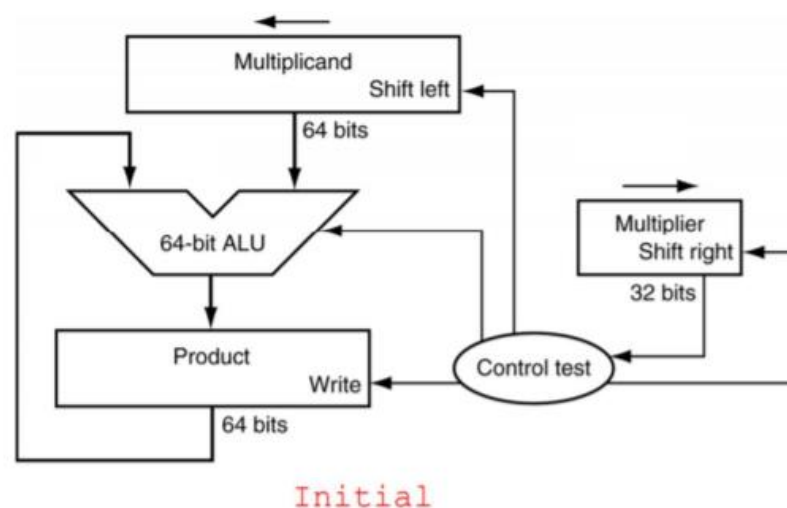
○ Standard of grading:

1. Please initialize the value of **TRISB**, **TRISC** to 0x0D, 0x06, respectively, and store the result in **TRISA**. We will check if the result stored in **TRISA** is correct or not.
2. Use the architecture of multiplier (either **optimized multiplier** or **simple multiplier**) to finish this Lab. Please refer to the following pictures.

- **optimized multiplier**



- **simple multiplier**



3. You are not allowed to do the addition of each **expansions** as following.

00001000+00010000+00000000+01000000 (X)

4. You are not allowed to use any instruction related to **MUL**.
5. **ROTATE** instructions are required in this lab.
(RLCF/RLNCF/RRCF/RRNCF)
6. **Goto** instruction is forbidden.
Instead, **7 commands** related to branch control are open for use.

(BC/BN/BNC/BNN/BNOV/BNZ/BOV)

F92	TRISA	0x4E	78
F93	TRISB	0x0D	13
F94	TRISC	0x06	6

Result

● **Bonus (20%) :**

- Description: Given two unsigned 8-bit non-zero numbers **A**, **B** (range from 0x01 to 0x7F) stored in registers 0x000 and 0x001 of data memory, respectively (**A >= B**). Use **loop** to find **GCD (greatest common divisor)** of the two numbers and store the result in 0x002 of data memory.
- Example: Given two numbers 0x40, 0x28 stored in address 0x000, 0x001 of data memory, and store the result 0x08 in address 0x002 of data memory.
- Standard of grading:
 - 1. You need to show two inputs and result during the demo. See figures below.
 - 2. You need to use **loop** to finish the lab.
- Hint: You can use **COMPARE** instructions to complete Bonus.

Address	00	01	02
000	40	28	08