

● Standard Instruction Set 教學

- Hackmd: <https://hackmd.io/h8TpNAIYTPyrJDj-6uFl w?view>

- 影片:

https://www.youtube.com/watch?v=t04LTsOuWgU&feature=share&utm_source=EJGixlgBCJiu2KjB4oSJEQ

● 基礎題 (70%):

- 題目敘述:請實作一個 32bits 的加法器。
- 測資和範例: 如右圖中第一列的值為

Address	00	01	02	03
000	10	11	12	FF
010	20	21	22	FF
020	30	32	35	FE

101112FF，第二列的值為 202122FF，將兩者相加的值存入第三列。

- 評分標準:

- 請將兩個初始化的 32bits 個別放在如圖示的 0X000~0X003 和 0X010~0X013 並將結果儲存在 0X020~0X023。

- 請使用如圖的測資。

- 不能使用 ADDWFC 指令，若使用將斟酌扣分。

● 進階題 (30%):

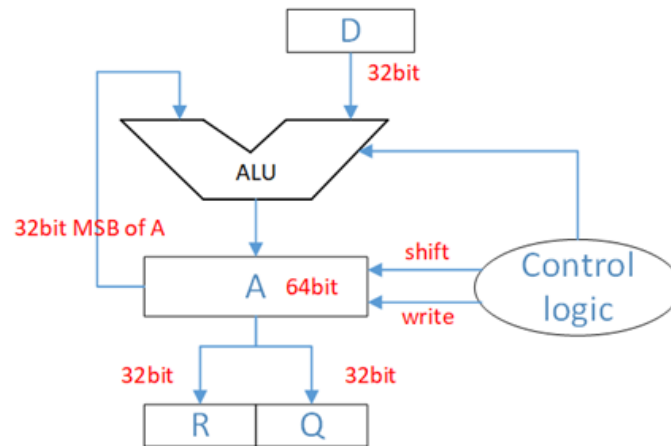
- 題目敘述:請實作一個 8bits 除以 4bits 的除法器並且輸出商到 0X000，將餘數輸出到 0X001。

- 測資和範例: 0X17 除以 0X06 的商為 0X03 餘為 0X05。則將 0X03 存到 0X00，0X05 存到 0X01 如右圖。

Address	00	01
000	03	05

- 評分標準:

- 測資請以 0X17 除以 0X06，並將結果放在如上述的位置。
- 請以除法器實作，例如下圖(圖片中 bit 數僅供參考，請依據題目要求的 bit 數去實作)



- 不能使用連減，用連減的方式直接沒分。
- 請不要使用 goto 和 BRA 指令，使用了將斟酌扣分。

● 加分題 (20%):

- 題目敘述:給定兩個 unsigned 8-bit 非零整數 A, B，分別儲存在 0X010 和 0X011。請找出兩數的最小公倍數，結果可能是 8bits 或者 16bits 請將結果儲存在 0X000 和 0X001。
- 測資和範例: 0XF5 和 0X5A 的最小公倍數為 0X113A。請將 0X11 存在 0X000，0X3A 存在 0X001。
- 評分標準:請使用範例的測資
- 備註:加分題這裡若有需要使用除法可以使用連減，不一定要使用除法器。

● Standard Instruction Set Tutorial

- Hackmd: https://hackmd.io/h8TpNAIYTPyrJDi-6uFl_w?view
- Video: https://www.youtube.com/watch?v=t04LTsOuWgU&feature=share&utm_source=EJGixlgBCJiu2KjB4oSJEQ

● Basic (70%):

- **Description:** Please implement a 32 bits adder.
- **Example and test data :**
 1. Store data values 0x10, 0x11, 0x12, and 0xFF on the memory addresses 0x000, 0x001, 0x002, and 0x003 respectively, as shown in the right figure.
 2. Store data values 0x10, 0x11, 0x12, and 0xFF on the memory addresses 0x010, 0x011, 0x012, and 0x013 respectively, as shown in the right figure.
 3. Add two 32 bits numbers and save the answer into 0x020~0x023.
- **Standard of grading:**
 - Please use the sample data as the picture.
 - You cannot use ADDWFC instruction. Otherwise, there will be a penalty of minus 35% .

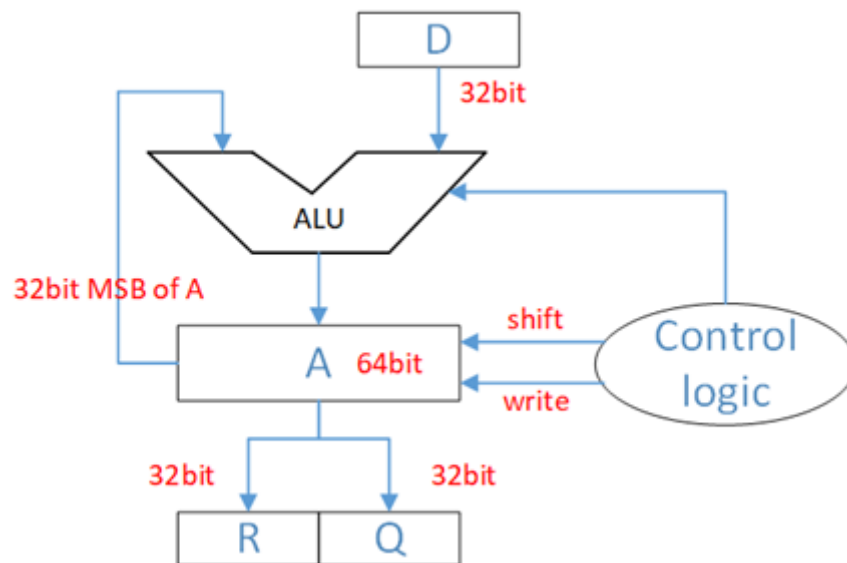
Address	00	01	02	03
000	10	11	12	FF
010	20	21	22	FF
020	30	32	35	FE

● Advanced (30%):

- **Description:** Please implement a divider which allows 8-bit divisor and 4-bit dividend. Store the quotient and remainder on the memory address 0x000 and 0x001.
- **Example and test data:** please use 0x17 to divide by 0x06 as the picture shown on the right. Save the quotient into 0x000, and save the remainder into 0x001.

Address	00	01
000	03	05

- **Standard of grading:**
 - Please implement with a divider, like the picture below. (The number of bits in the picture is just an example, please implement your own divider according to the description.)



- Do not use continuous decrease. Otherwise, you will get **no point** in this section.
- Do not use BRA and goto instruction. Otherwise, we will take some points off.

● **Bonus** (20%):

- **Description:** Given two unsigned 8-bit nonzero integers A and B, stored at 0X010 and 0X011. Please find their least common multiple, the result might be 8bits or 16bits. Please save your answer at 0X000 and 0X001.
- **Example and test data:** The least common multiple of 0XF5 and 0X5A is 0X113A. Please save 0X11 at memory address 0X000 and save 0X3A at 0X001.
- **Standard of grading:** Please use the sample data 0XF5 and 0X5A.
- If you need to division in this section, you can use continuous decrease.