

# Memory Control & Test Guide

Daniel 2017/4/17 Initial Version

## New PEB SRAM Memory 結構

每一片 PEB Channel Board 上有兩組 Memory，分別為 Instruction Memory 與 Pattern Memory。Instruction Memory 為儲存 OP Code、TimingSet Index、Operand 等等。Pattern Memory 則用來儲存輸出 waveform 或輸入比較條件的參照位置。在記憶體測試上我們可以先不理會儲存格式，使用一般測試 pattern 來測試 SRAM 的正確性即可。

Instruction Memory(寬度 32-bit)與 Pattern Memory(寬度 48-bit)的記憶體深度目前皆為 4M (Address = 22-bit)，之後正式 release 時為 8M (Address = 23-bit)。兩個 Memory 共用位址線控制(Register 0x03)，具有 Auto Increment 的功能(填入 Start Address 即可，不用反覆重新寫入位址)。

## Channel Board SRAM Control Register MAP:

W 0x03	Write First Time:	SRAM_ADDR[22:16]
	Write Second Time:	SRAM_ADDR[15:0]
R 0x03	Read First Time:	CURRENT_SRAM_ADDR[22:16]
	Read Second Time:	CURRENT_SRAM_ADDR[15:0]
W 0x04	Write First Time:	INSTR_SRAM_DATA[31:16]
	Write Second Time:	INSTR_SRAM_DATA[15:0] , SRAM_ADDR + 1
R 0x04	Read First Time:	INSTR_SRAM_DATA[31:16]
	Read Second Time:	INSTR_SRAM_DATA[15:0] , SRAM_ADDR + 1
W 0x06	Write First Time:	PAT_SRAM_DATA[47:32]
	Write Second Time:	PAT_SRAM_DATA[31:16]
	Write Third Time:	PAT_SRAM_DATA[15:0] , SRAM_ADDR + 1
R 0x06	Read First Time:	PAT_SRAM_DATA[47:32]
	Read Second Time:	PAT_SRAM_DATA[31:16]
	Read Third Time:	PAT_SRAM_DATA[15:0] , SRAM_ADDR + 1

因為對 0x04 R/W 兩次或對 0x06 R/W 三次皆會自動將 SRAM\_ADDR + 1，所以建議將 Instruction Memory 寫完再對 Pattern Memory 寫入。

## Instruction Memory 寫入範例

```
W 0x03  0x0000
W 0x03  0x0000  // Start Address = 0x000000
W 0x04  0xAAAA
W 0x04  0xBBBB  // Write 0xAAAABBBB to 0x000000
W 0x04  0xFFFF
W 0x04  0x0000  // Write 0xFFFF0000 to 0x000001
R 0x03  CUR_ADDR_MSB
R 0x03  CUR_ADDR_LSB  // 回讀目前的 Address，應該為 0x000002
```

### Instruction Memory 讀取範例

```
W 0x03 0x0000
W 0x03 0x0000 // Start Address = 0x000000
R 0x04 MSB_DATA
R 0x04 LSB_DATA // Read Address 0x000000 Data, Should be 0xAAAABBBB
R 0x04 MSB_DATA
R 0x04 LSB_DATA // Read Address 0x000001 Data, Should be 0xFFFF0000
R 0x03 CUR_ADDR_MSB
R 0x03 CUR_ADDR_LSB // 回讀目前的 Address，應該為 0x000002
```

### Pattern Memory 寫入範例

```
W 0x03 0x0000
W 0x03 0x0000 // Start Address = 0x000000
W 0x06 0xAAAA
W 0x06 0xBBBB
W 0x06 0xCCCC // Write 0xAAAABBBBCCCC to 0x000000
W 0x06 0xFFFF
W 0x06 0x5555
W 0x06 0x0000 // Write 0xFFFF55550000 to 0x000001
R 0x03 CUR_ADDR_MSB
R 0x03 CUR_ADDR_LSB // 回讀目前的 Address，應該為 0x000002
```

### Pattern Memory 讀取範例

```
W 0x03 0x0000
W 0x03 0x0000 // Start Address = 0x000000
R 0x06 DATA[47:32]
R 0x06 DATA[31:16]
R 0x06 DATA[15:0] // Read Address 0x000000 Data, Should be 0xAAAABBBBCCCC
R 0x06 DATA[47:32]
R 0x06 DATA[31:16]
R 0x06 DATA[15:0] // Read Address 0x000001 Data, Should be 0xFFFF55550000
R 0x03 CUR_ADDR_MSB
R 0x03 CUR_ADDR_LSB // 回讀目前的 Address，應該為 0x000002
```

### Memory Test 1 (Checker Board):

偶數位 Address 寫入 0xAAAAAAAA...

奇數位 Address 寫入 0x55555555...

全部位址寫完後回讀確認 SRAM 是否有 fail bit

### Memory Test 2 (反相 Checker Board)

偶數位 Address 寫入 0x55555555...

奇數位 Address 寫入 0xAAAAAAAA...

再次全部位址寫完後回讀確認 SRAM 是否有 fail bit

### Memory Test 3 (Data 填入 Address)

Address 0 寫入 Data = 0

Address 1 寫入 Data = 1

...

Address 0xFFFF... 寫入 Data = 0xFFFF...

全部位址寫完後回讀確認 SRAM 的位址控制是否正確