

SOC LAB#D Test (lab-sdram)Report

Group no: 4

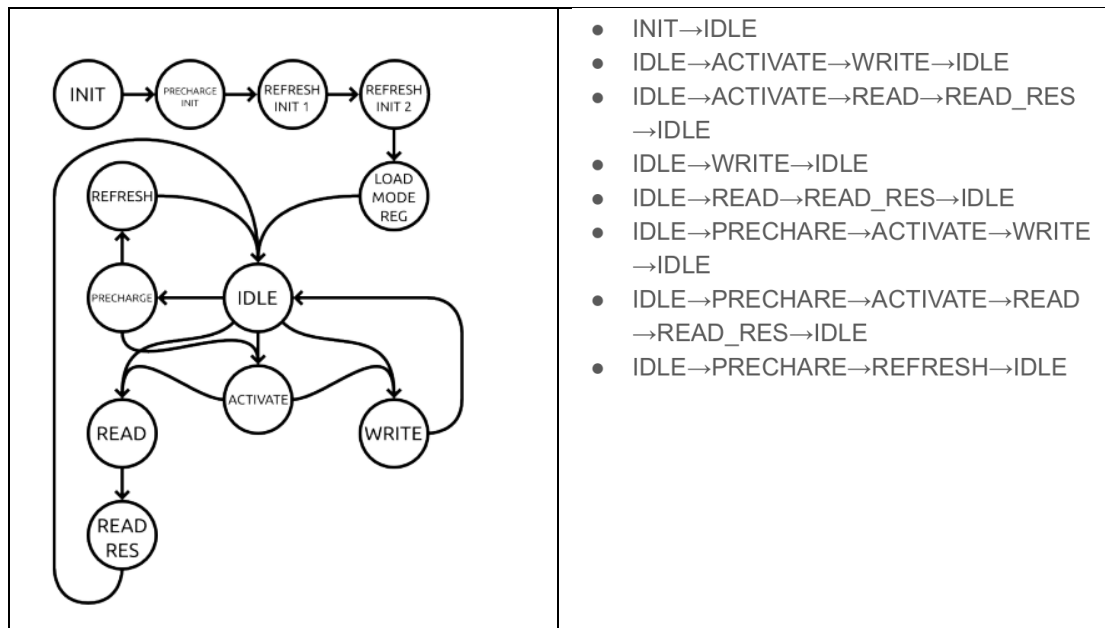
Members:

M11107410 羅善寬

M11107004 曹榮恩

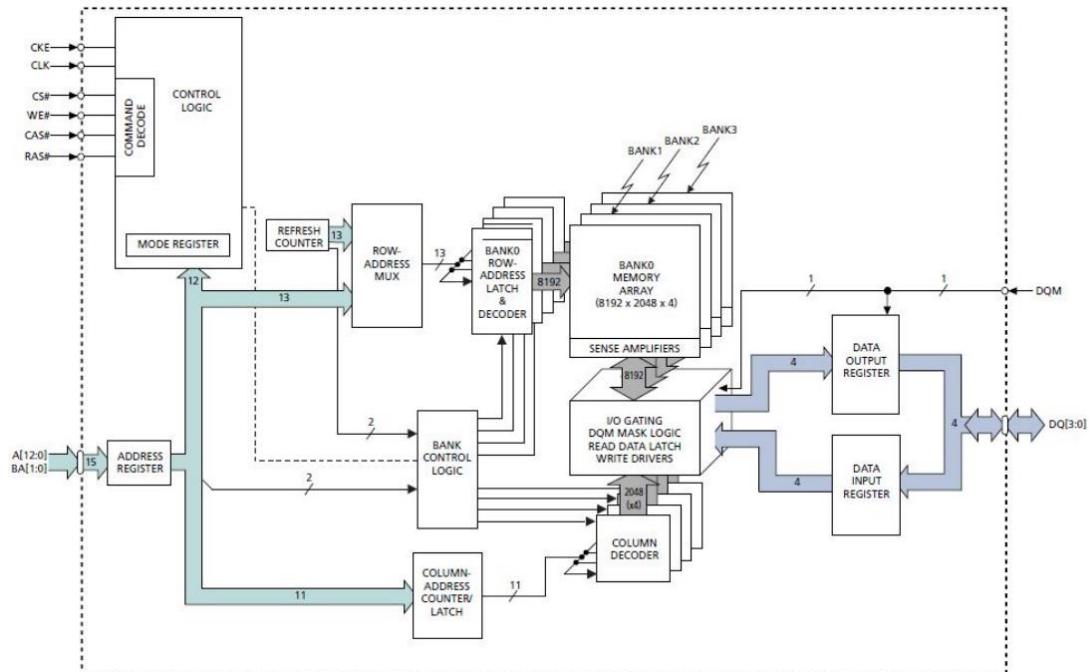
M11107409 陳昱碩

1. SDRAM controller design



Init	初始狀態設定
Wait	等待一段時間，delay
Idle	Init 做 reset 後，無人使用的狀態
Refresh	一定週期後進行資料的重新寫入
Activate	激活 bank 跟 row，等待是要 read 或 write
Read	激活 col，並做 data 的 read
Write	激活 col，並做 data 的 write
Precharge	想改變不同 row 或是 bank 時會使用到
Read_Res	read 並取回 data

2. SDRAM bus protocol



原本的 Dq 是 in-out pin，而現在 sdr 裡面則分為 Dqi 跟 Dqo，一個是 input，另一個是 output。

Rst_n	reset
Addr	col 或 row 的 address
Ba	bank 的選擇
clk	時脈
Cke	clk 的 enable 訊號
Dqm	選擇要 input Dqi 還是 output Dqo
CAS_n	行選擇
RAS_n	列選擇
Cs_n	enable command
We_n	write enable

```

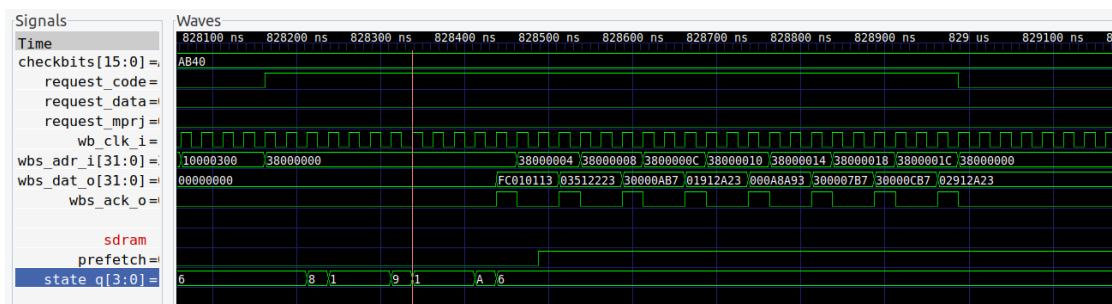
Active_enable   = ~Cs_n & ~Ras_n & Cas_n & We_n;
Aref_enable     = ~Cs_n & ~Ras_n & ~Cas_n & We_n;
Burst_term      = ~Cs_n & Ras_n & Cas_n & ~We_n;
Mode_reg_enable = ~Cs_n & ~Ras_n & ~Cas_n & ~We_n;
Prech_enable    = ~Cs_n & ~Ras_n & Cas_n & ~We_n;
Read_enable     = ~Cs_n & Ras_n & ~Cas_n & We_n;
Write_enable    = ~Cs_n & Ras_n & ~Cas_n & ~We_n;

```

Active_enable	enable row 並且等待 Read_enable 或 Write_enable
Aref_enable	auto-refresh
Mode_reg_enable	與 reset 一樣
Read_enable	讀取資料並激活 col
Write_enable	寫入資料並激活 col
Prech_enable	在需要執行不同的 bank 時，需先將目前開啟的 bank 關閉後，才可執行別的 bank
Burst_term	第一筆資料的地址，後續只傳數據，直到達到 burst length(1,2,4,8)或遇到終止命令。

3. Introduce the prefetch scheme

先看是否有 in_valid，若有的話，則把 prefetch_addr 改為 addr + 22'd4。
 如果今天 prefetch_addr 與 addr 相等且 rw 信號為 0 的時候，那就將 prefetch 改為 1。
 那在 IDLE 的時候，會去判斷是否要 prefetch，如果要的話，就直接把剛剛 dq_i_d 的值放入 data_d，並且 out_valid_d 改為 1，進行輸出。



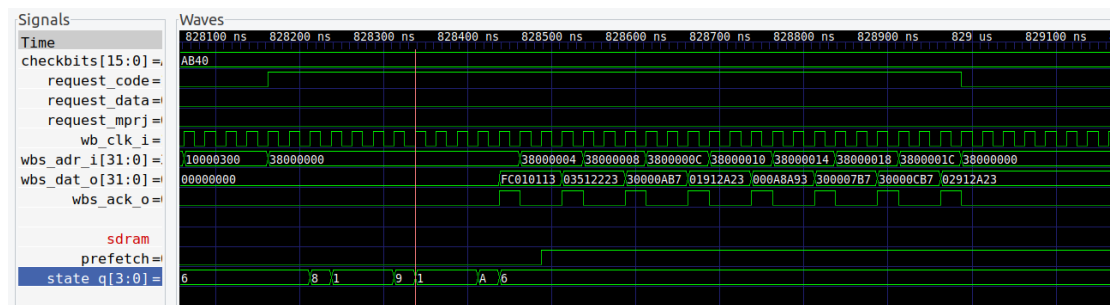
在資料讀取時經過 prefetch 之後，每 3 個 clk 就會 ack 回去。

4. Introduce the bank interleave for code and data

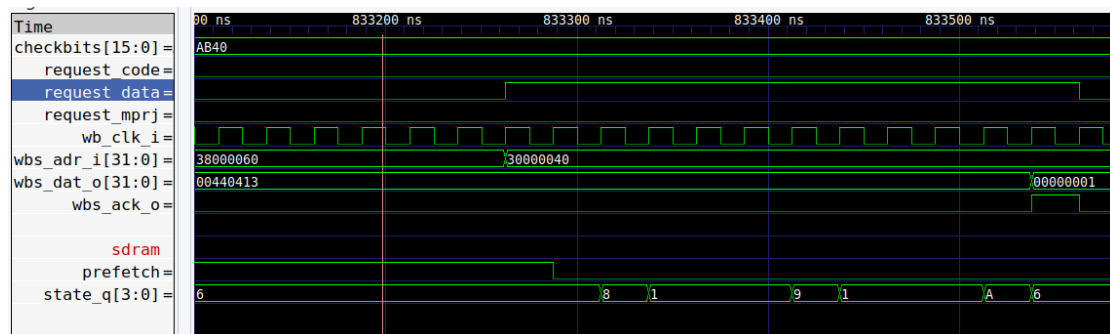
首先，先看 wbs_stb_i、wbs_cyc_i 跟 wbs_adr_i[31:24]來決定：

	wbs_adr_i	bank	addr
request_data	0x30xxxxxx	10 or 11	{wbs_adr_i[22:10], data_bank, wbs_adr_i[7:0]}
request_code	0x38xxxxxx	00 or 01	{wbs_adr_i[22:10], code_bank, wbs_adr_i[7:0]}

	wbs_adr_i[9:8] = 00 or 01	wbs_adr_i[9:8] = 10 or 11
data_bank	wbs_adr_i[9:8]	wbs_adr_i[9:8]-2'b10
code_bank	wbs_adr_i[9:8]+2'b10	wbs_adr_i[9:8]



code => wbs_adr_i[31:24] = 8'h38



data => wbs_adr_i[31:24] = 8'h30

5. Introduce how to modify the linker to load address/data in two different bank

```

47     .data :
48     {
49         . = ALIGN(8);
50         _fdata = .;
51         *(.data .data.* .gnu.linkonce.d.*)
52         *(.data1)
53         _gp = ALIGN(16);
54         *(.sdata .sdata.* .gnu.linkonce.s.*)
55         . = ALIGN(8);
56         _edata = .;
57     } > mprj AT > flash
58
59     .bss :
60     {
61         . = ALIGN(8);
62         _fbss = .;
63         *(.dynsbss)
64         *(.sbss .sbss.* .gnu.linkonce.sb.*)
65         *(.scommon)
66         *(.dynbss)
67         *(.bss .bss.* .gnu.linkonce.b.*)
68         *(COMMON)
69         . = ALIGN(8);
70         _ebss = .;
71         _end = .;
72     } > mprj AT > flash
73
74     .mprjram :
75     {
76         . = ALIGN(8);
77         _fsram = .;
78     } > mprjram AT > flash
79
80 }

```

data (.data 和 .bss)	mprj (0x30000000)
code	mprjram (0x38000000)

6. Observe SDRAM access conflicts with SDRAM

refresh (reduce the refresh period)

SDRAM access conflicts 跟 refresh 是處理 SDRAM 的重要任務，當系統同時嘗試訪問一個 SDRAM 區域而該區域正要進行刷新時，可能會發生存取衝突，可能會導致效能降低，那麼降低 refresh period 可以有效的增加 SDRAM access time，這樣的效果可以使 access conflict 的情況降低，使 SDRAM 的效能增加。

7. Others

```

ubuntu@ubuntu2004:~/Desktop/correct/soc_lab0-main/soc_lab0-main/lab-sdram/testbench/counter_la_mm$ source run_clean
ubuntu@ubuntu2004:~/Desktop/correct/soc_lab0-main/soc_lab0-main/lab-sdram/testbench/counter_la_mm$ source run_sim
Reading counter_la_mm.hex
counter_la_mm.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_mm.vcd opened for output.
LA Test 1 started
counter_la_mm_tb.uut.mprj.mprj.user_bram : at time 1562988000 ERROR: Bank is not Activated for Read
counter_la_mm_tb.uut.mprj.mprj.user_bram : at time 2126763000 ERROR: Bank is not Activated for Read
counter_la_mm_tb.uut.mprj.mprj.user_bram : at time 2371163000 ERROR: Bank is not Activated for Read
counter_la_mm_tb.uut.mprj.mprj.user_bram : at time 2615563000 ERROR: Bank is not Activated for Read
counter_la_mm_tb.uut.mprj.mprj.user_bram : at time 2859963000 ERROR: Bank is not Activated for Read
counter_la_mm_tb.uut.mprj.mprj.user_bram : at time 4157163000 ERROR: Bank is not Activated for Read
counter_la_mm_tb.uut.mprj.mprj.user_bram : at time 5516963000 ERROR: Bank is not Activated for Read
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x003e
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0044
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x004a
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0050

```

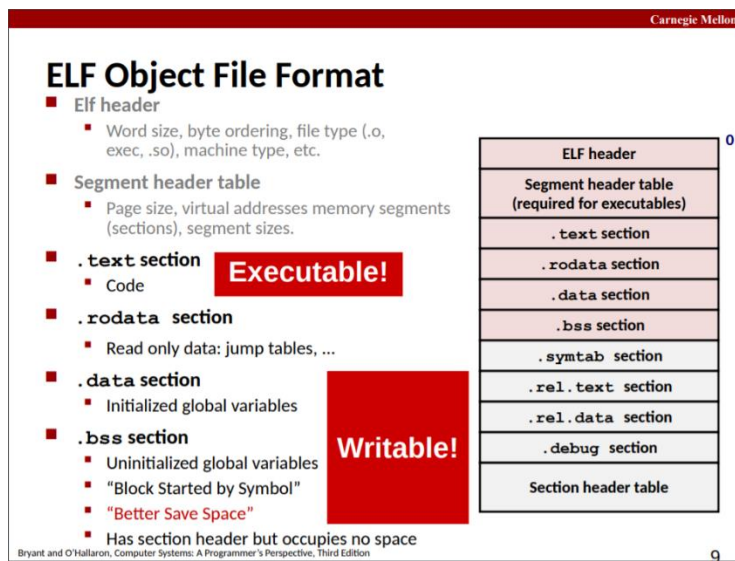
完成矩陣相乘 matmul 功能。

8. What else do you observe

Lab 中對於 sdram 更為熟悉，包括讀寫方式、狀態。還有學習對於 linker file 中的些許概念。

在將每個 *.c 檔產生出對應的 *.o 後，最後使用像是 ld 這樣的 linker，以這些產生的 object files 作為輸入，生成出一個可以載入和運行的 executable。

object files 是透過特定的格式來組織的，每個作業系統的 object files 也不盡相同，這裡我們將以 Linux 上的 ELF 格式作為主要的討論對象。



上圖中展示了一個基本的 ELF layout。在最開始的內容是 ELF header，ELF header 包含其生成的機器類型(x86-64)、byte ordering、object file 類型(relocatable / executable / shared)，section 的大小和數量等等資訊。其後則有不同的 section，典型的 relocatable object file 通常會包含：

- .text: 被編譯後的程式 binary
- .rodata: 只可讀的 data
- .data: 已初始化(不為 0) 的 global 或者 static variable
- .bss: 未初始化或者初始為 0 的 global 或者 static variable