MIAT_STM32
內部與外部SRAM存取控制實驗

浯陽科技有限公司

WU-YANG

Technology Co., Ltd.

Declared Version

Training Only

Declare

Document Number	
Document Version	***1.00***
Release Date	
Document Title	MIAT_STM32 內部與外部SRAM存取控制實驗
Exercise Time	■
Platform	■ ***MIAT_STM32.V2*** ■ ***MIAT IOB.V1***
Peripheral	■
Author	■ ***WU-YANG Technology Co., Ltd.***

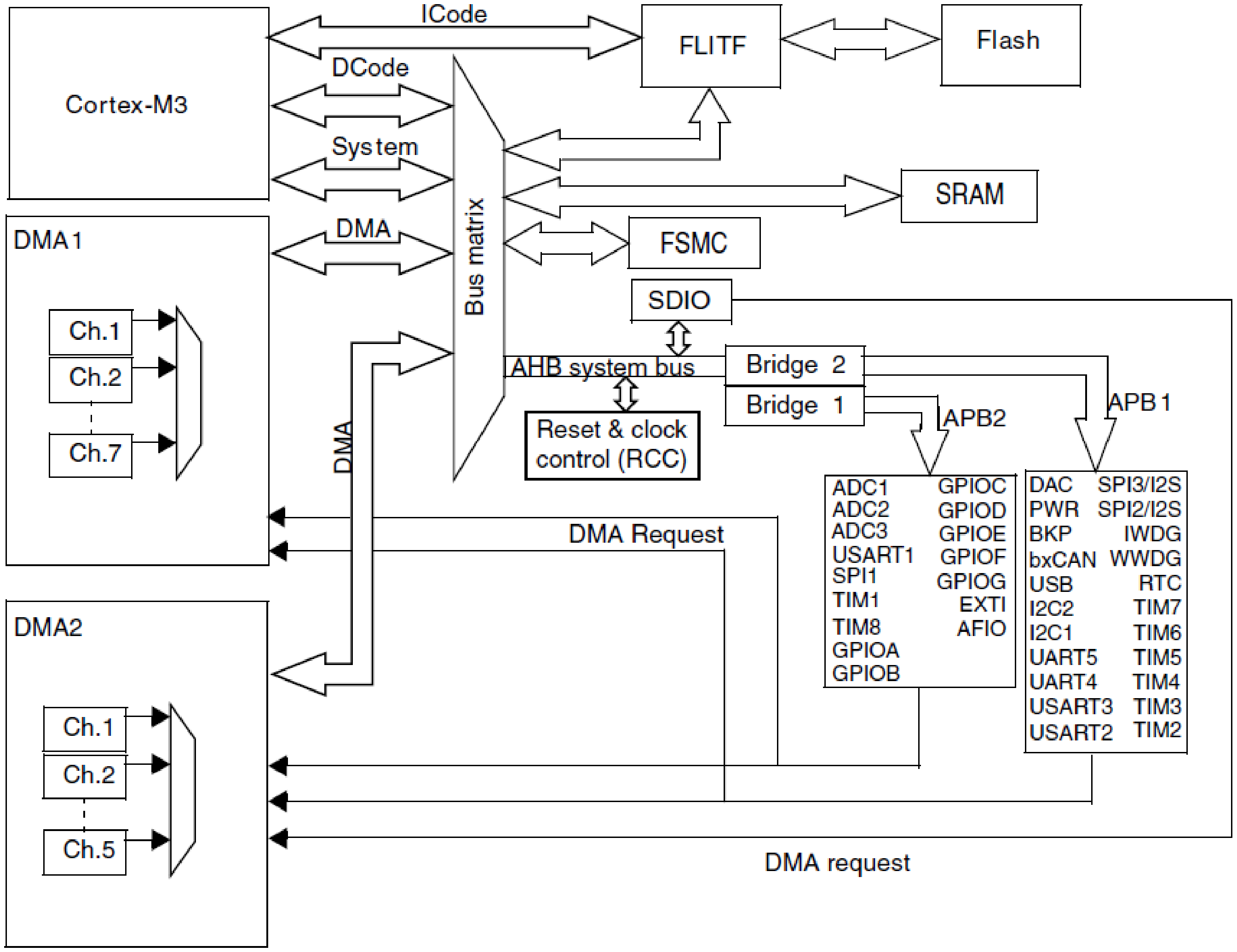
實驗目的(一)

- □ 使用MIAT_STM32實驗板透過Flexible static memory controller (FSMC)控制內部與外部SRAM進行存取控制實驗，並利用LED確認存取是否正常。

實驗原理

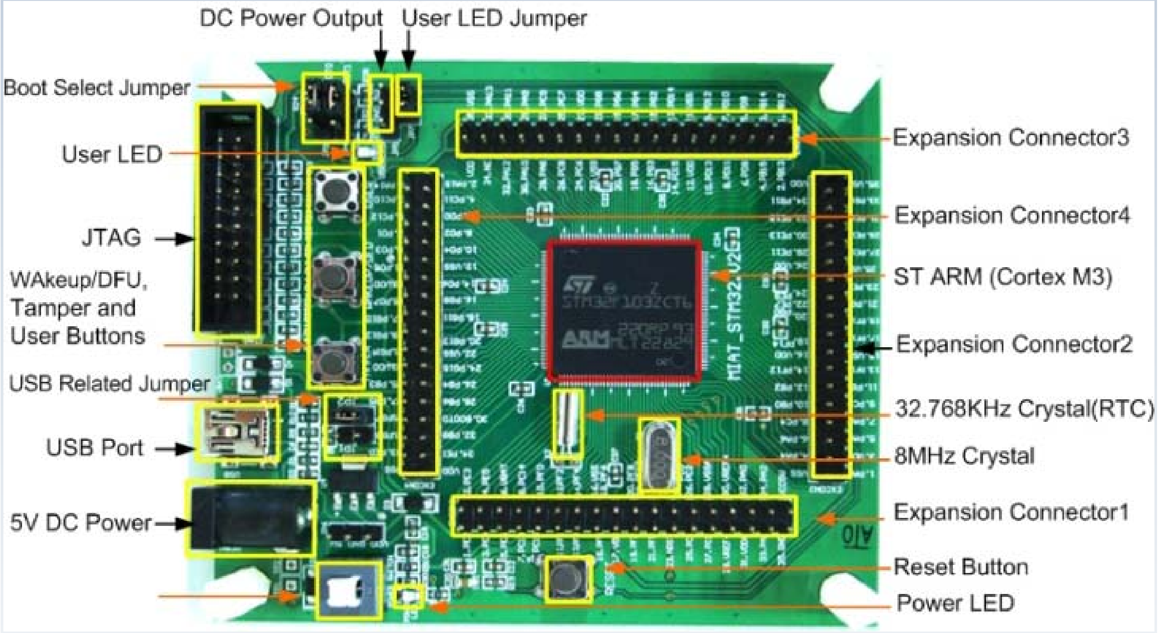
- System architecture
- Embedded SRAM
  - Features
  - Memory map
  - RVMDK環境設定
- External SRAM
  - IS61LV25616AL
  - FSMC (flexible static memory controller)
- Development Flow
- ARM Configure

System architecture

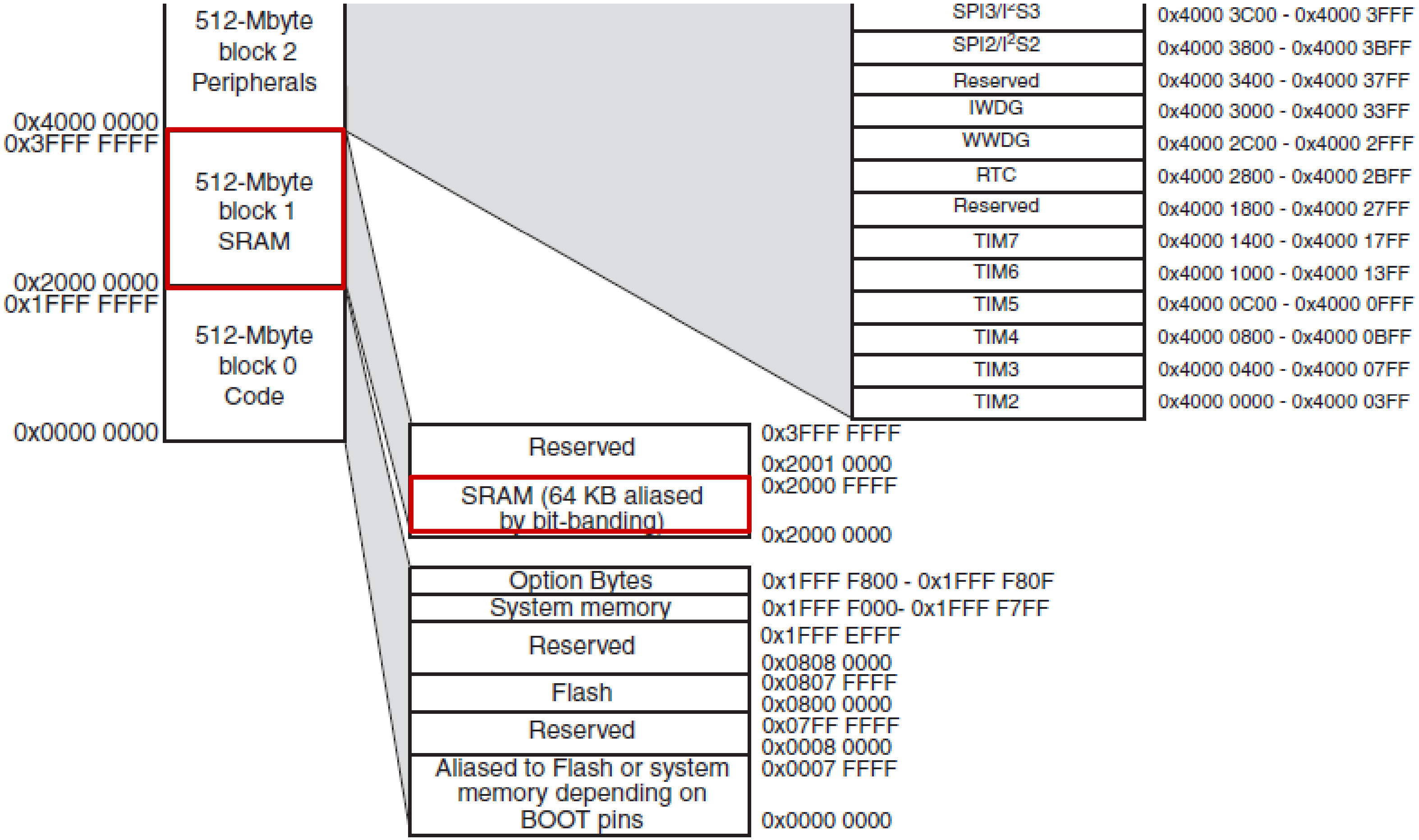


Embedded SRAM

- Features
  - 48Kbytes of embedded SRAM
  - accessed (read/write) at CPU clock speed with 0 wait states



Embedded SRAM Memory map

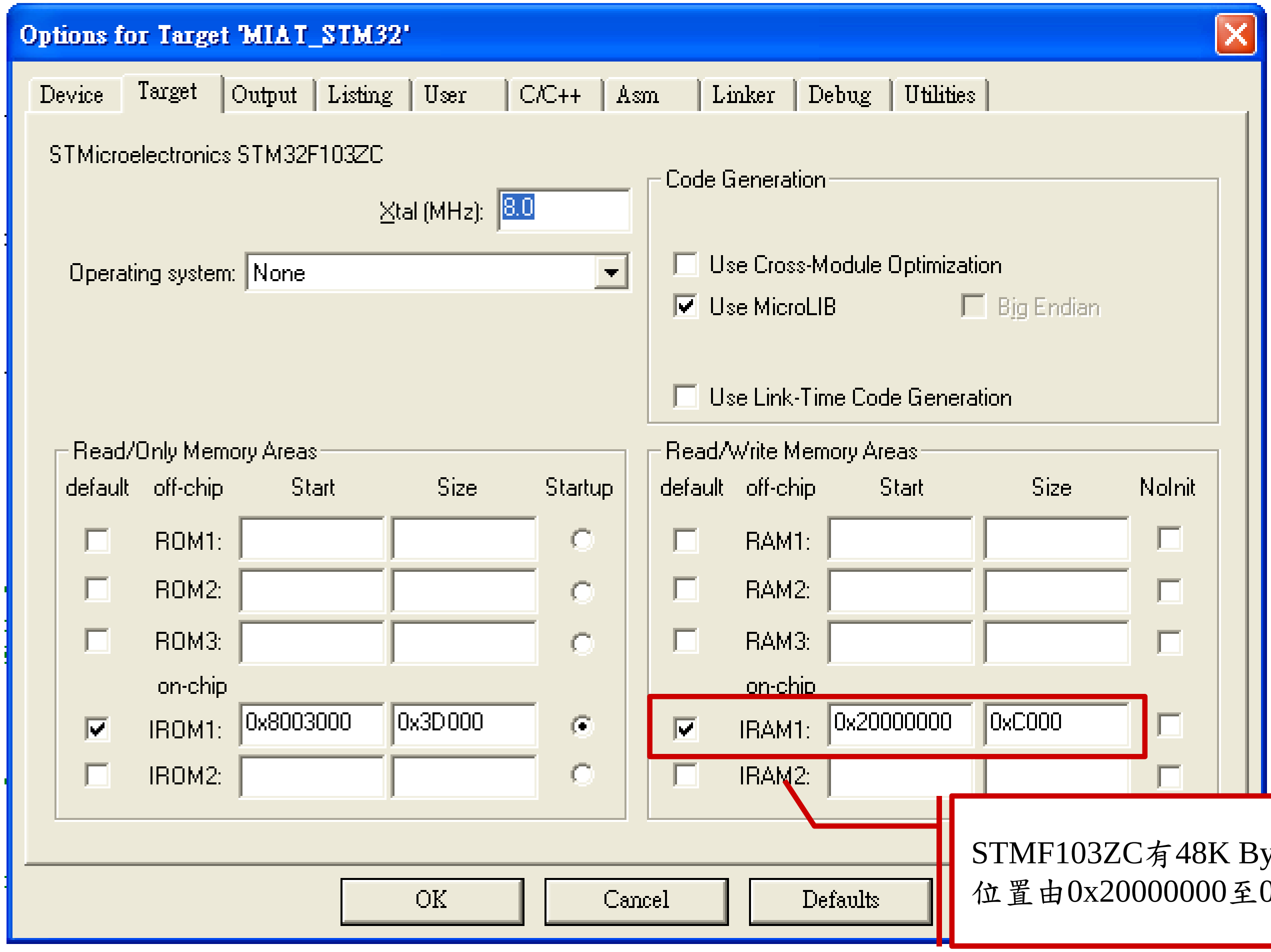


	512-Mbyte block 2 Peripherals
0x4000 0000 0x3FFF FFFF	512-Mbyte block 1 SRAM
0x2000 0000 0x1FFF FFFF	512-Mbyte block 0 Code
0x0000 0000	

SPI3/I^{2}S3	0x4000 3C00 - 0x4000 3FFF
SPI2/I^{2}S2	0x4000 3800 - 0x4000 3BFF
Reserved	0x4000 3400 - 0x4000 37FF
IWDG	0x4000 3000 - 0x4000 33FF
WWDG	0x4000 2C00 - 0x4000 2FFF
RTC	0x4000 2800 - 0x4000 2BFF
Reserved	0x4000 1800 - 0x4000 27FF
TIM7	0x4000 1400 - 0x4000 17FF
TIM6	0x4000 1000 - 0x4000 13FF
TIM5	0x4000 0C00 - 0x4000 0FFF
TIM4	0x4000 0800 - 0x4000 0BFF
TIM3	0x4000 0400 - 0x4000 07FF
TIM2	0x4000 0000 - 0x4000 03FF

Reserved	0x3FFF FFFF 0x2001 0000
SRAM (64 KB aliased by bit-banding)	0x2000 FFFF 0x2000 0000
Option Bytes	0x1FFF F800 - 0x1FFF F80F
System memory	0x1FFF F000- 0x1FFF F7FF
Reserved	0x1FFF EFFF 0x0808 0000
Flash	0x0807 FFFF 0x0800 0000
Reserved	0x07FF FFFF 0x0008 0000
Aliased to Flash or system memory depending on BOOT pins	0x0007 FFFF 0x0000 0000

RVMDK環境設定



Options for Target 'MIAT_STM32'
Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities
OK
Cancel
Defaults

STMicroelectronics STM32F103ZC

Xtal (MHz): 8.0

Operating system: None

Code Generation

- ☐ Use Cross-Module Optimization
- ☑ Use MicroLIB ☐ Big Endian
- ☐ Use Link-Time Code Generation

Read/Only Memory Areas

default	off-chip	Start	Size	Startup
☐	ROM1:			○
☐	ROM2:			○
☐	ROM3:			○
	on-chip			
☑	IROM1:	0x8003000	0x3D000	●
☐	IROM2:			○

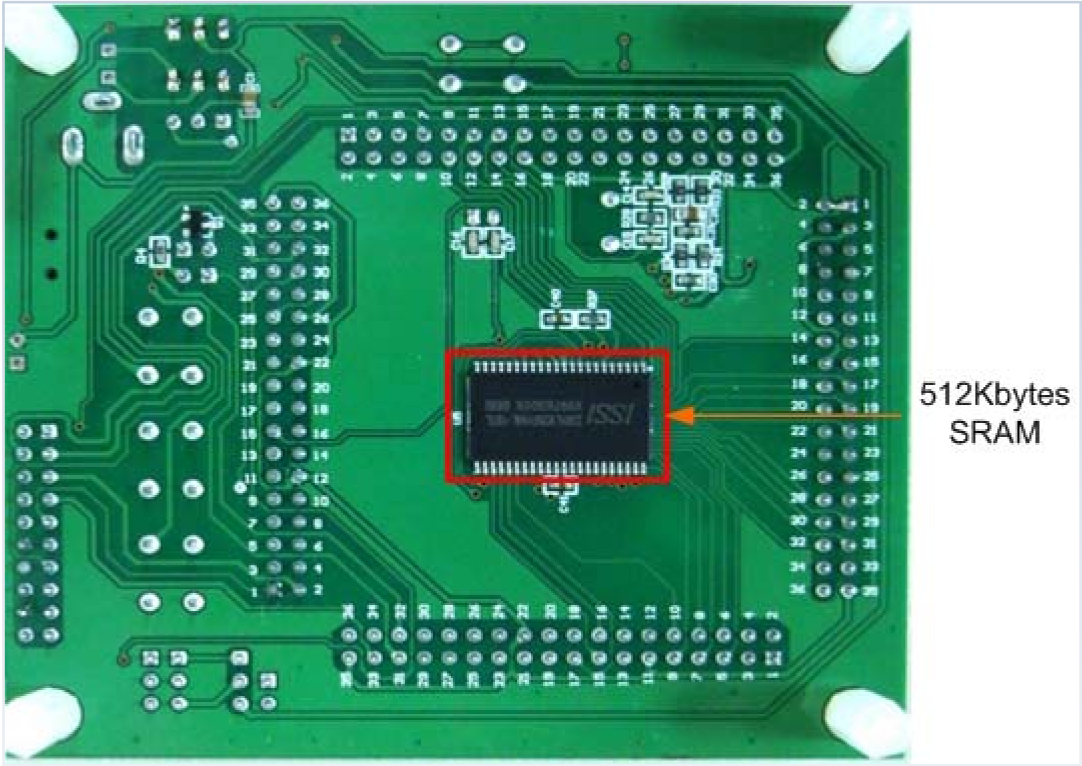
Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
☐	RAM1:			☐
☐	RAM2:			☐
☐	RAM3:			☐
	on-chip			
☑	IRAM1:	0x20000000	0xC000	☐
☐	IRAM2:			☐

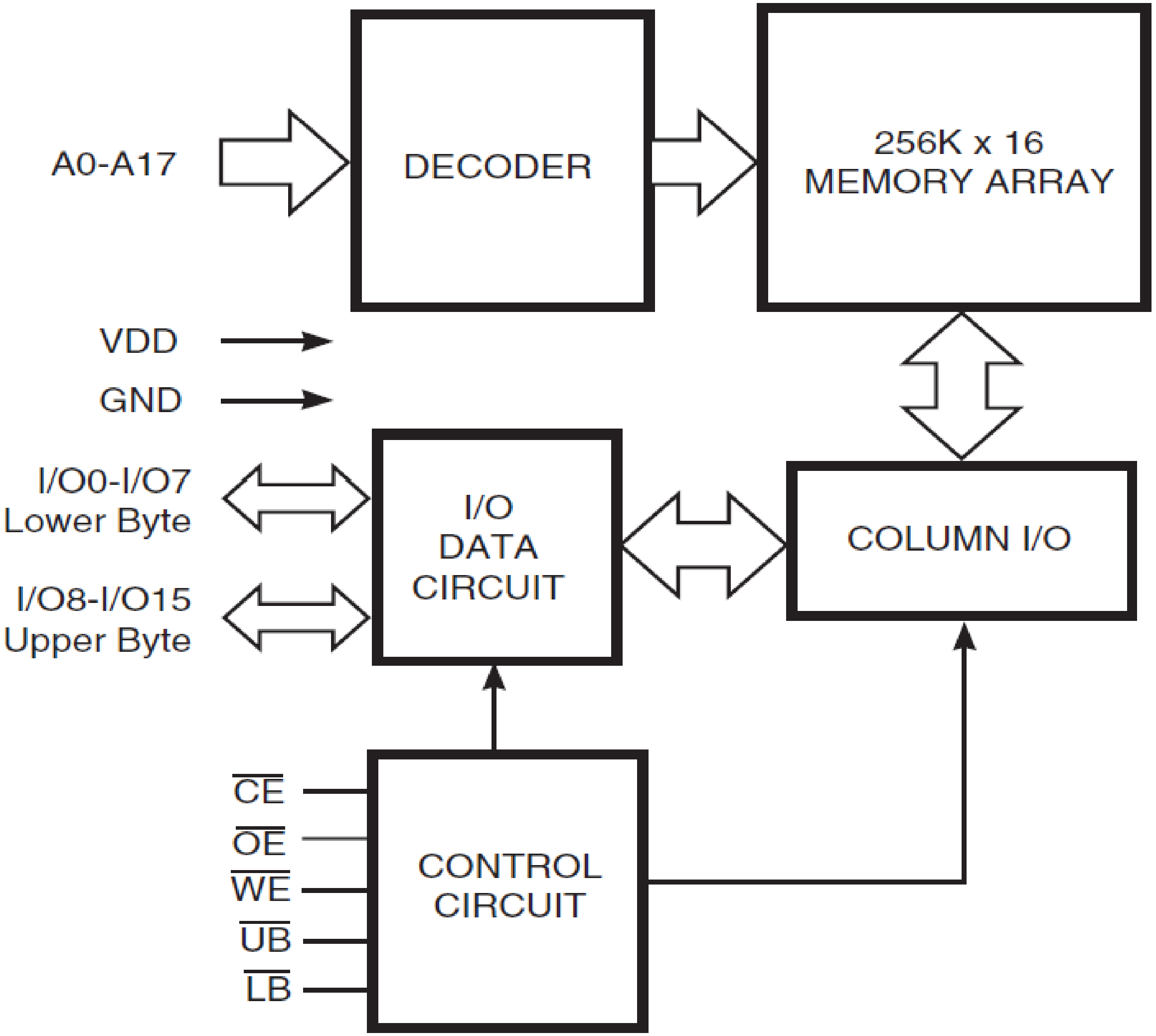
STMF103ZC有48K Bytes的SRAM
位置由0x20000000至0x2000C000

External SRAM (IS61LV25616AL)

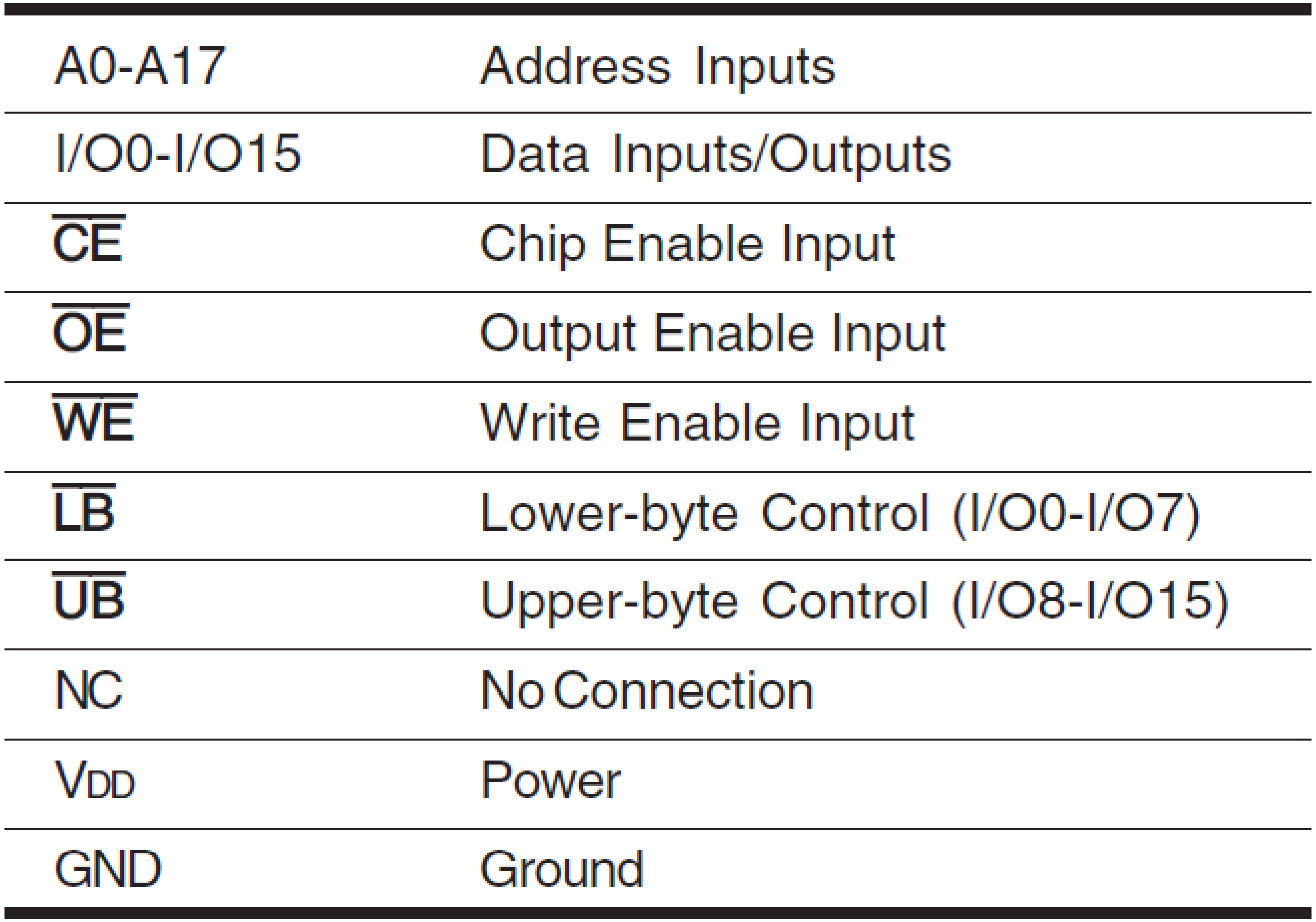
- □ Features
  - High-speed access time 10 ns
  - 256K x 16



FUNCTIONAL BLOCK DIAGRAM

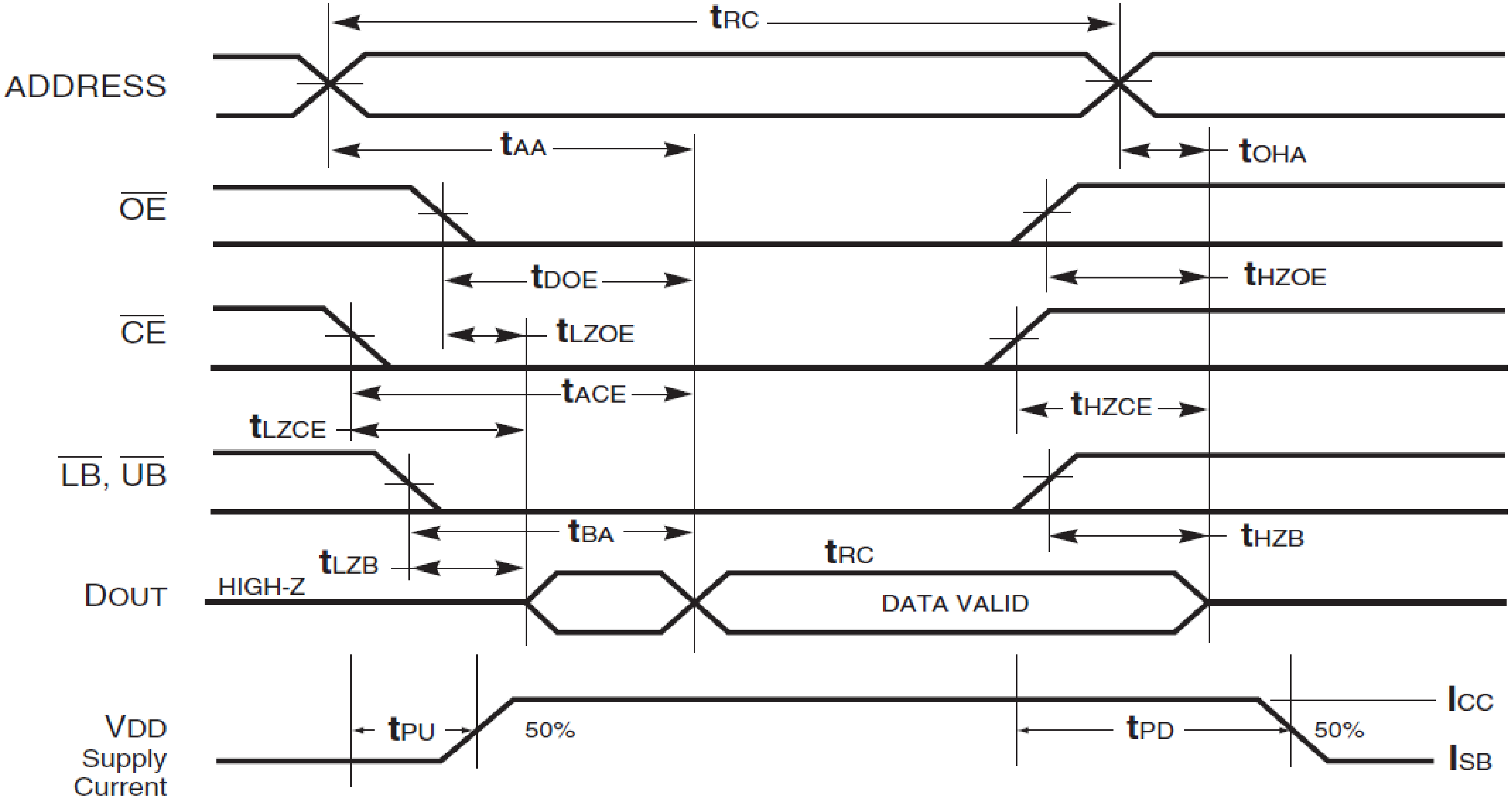


PIN DESCRIPTIONS

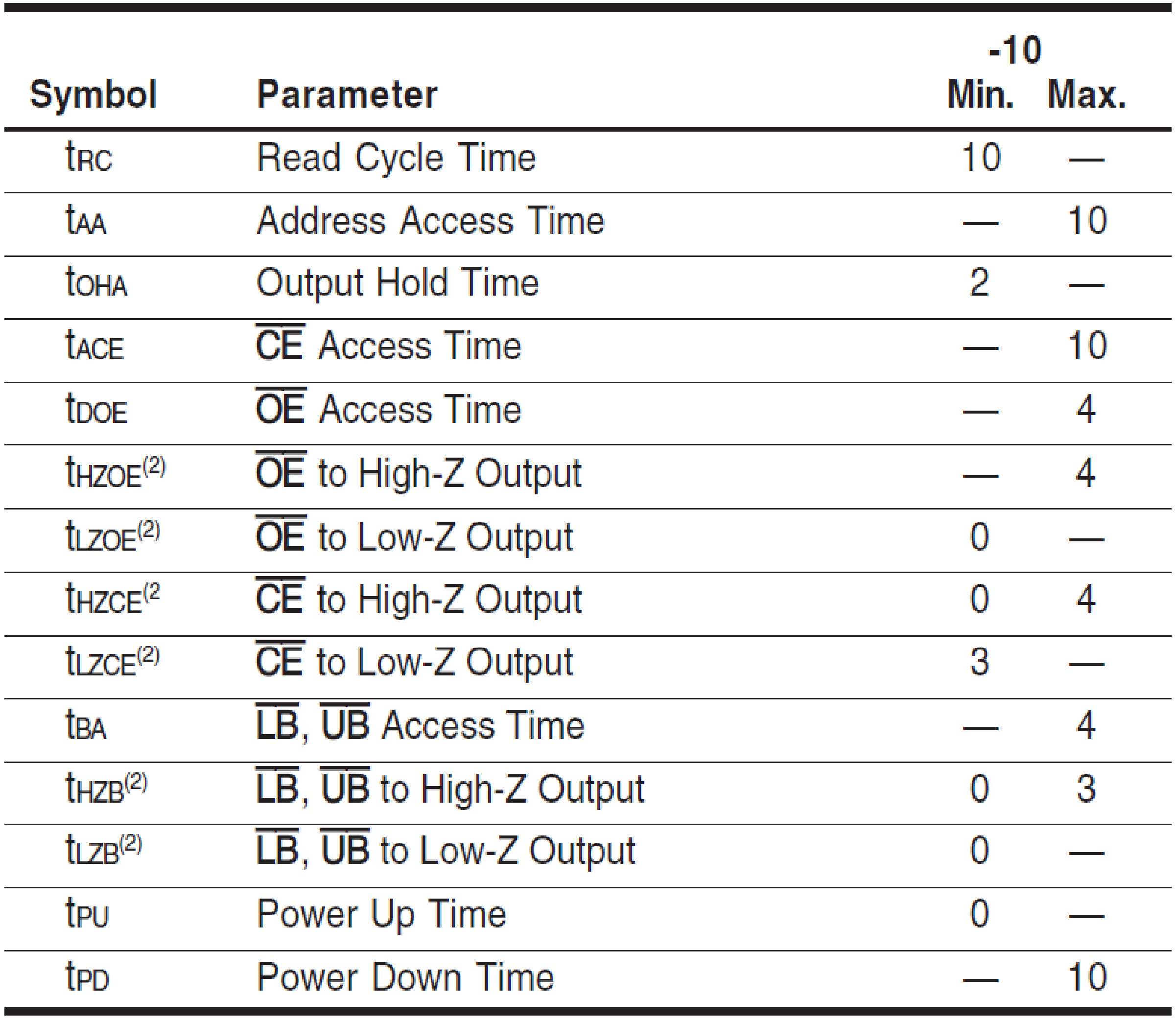


A0-A17	Address Inputs
I/O0-I/O15	Data Inputs/Outputs
$\overline{CE}$	Chip Enable Input
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
$\overline{LB}$	Lower-byte Control (I/O0-I/O7)
$\overline{UB}$	Upper-byte Control (I/O8-I/O15)
NC	No Connection
V_{DD}	Power
GND	Ground

READ CYCLE

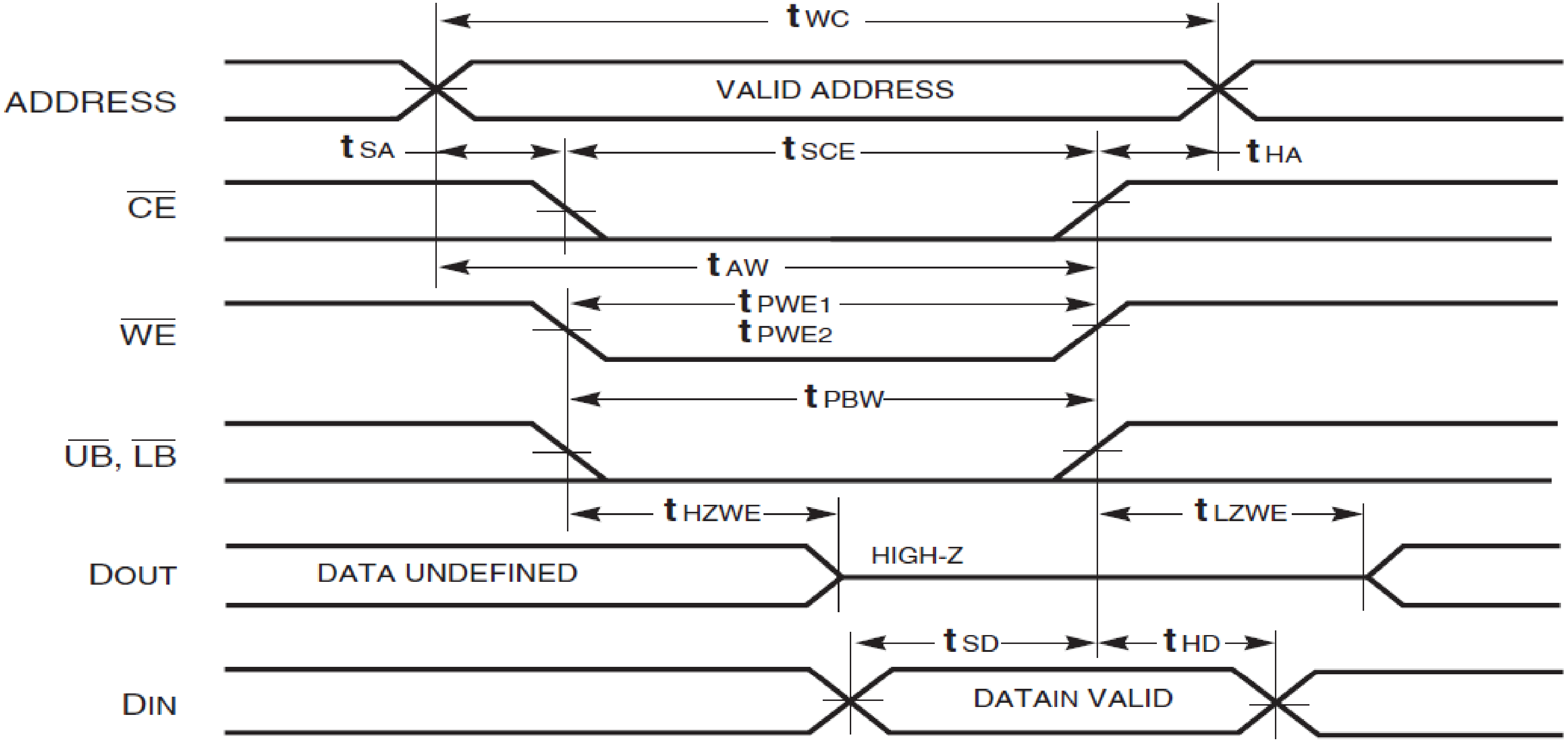

tRC
ADDRESS
tAA
tOHA
$\overline{OE}$
tDOE
tHZOE
$\overline{CE}$
tLZOE
tACE
tHZCE
tLZCE
$\overline{LB}$, $\overline{UB}$
tBA
tHZB
tLZB
tRC
DOUT
HIGH-Z
DATA VALID
VDD
Supply
Current
tPU
50%
tPD
50%
ICC
ISB

READ CYCLE SWITCHING CHARACTERISTICS



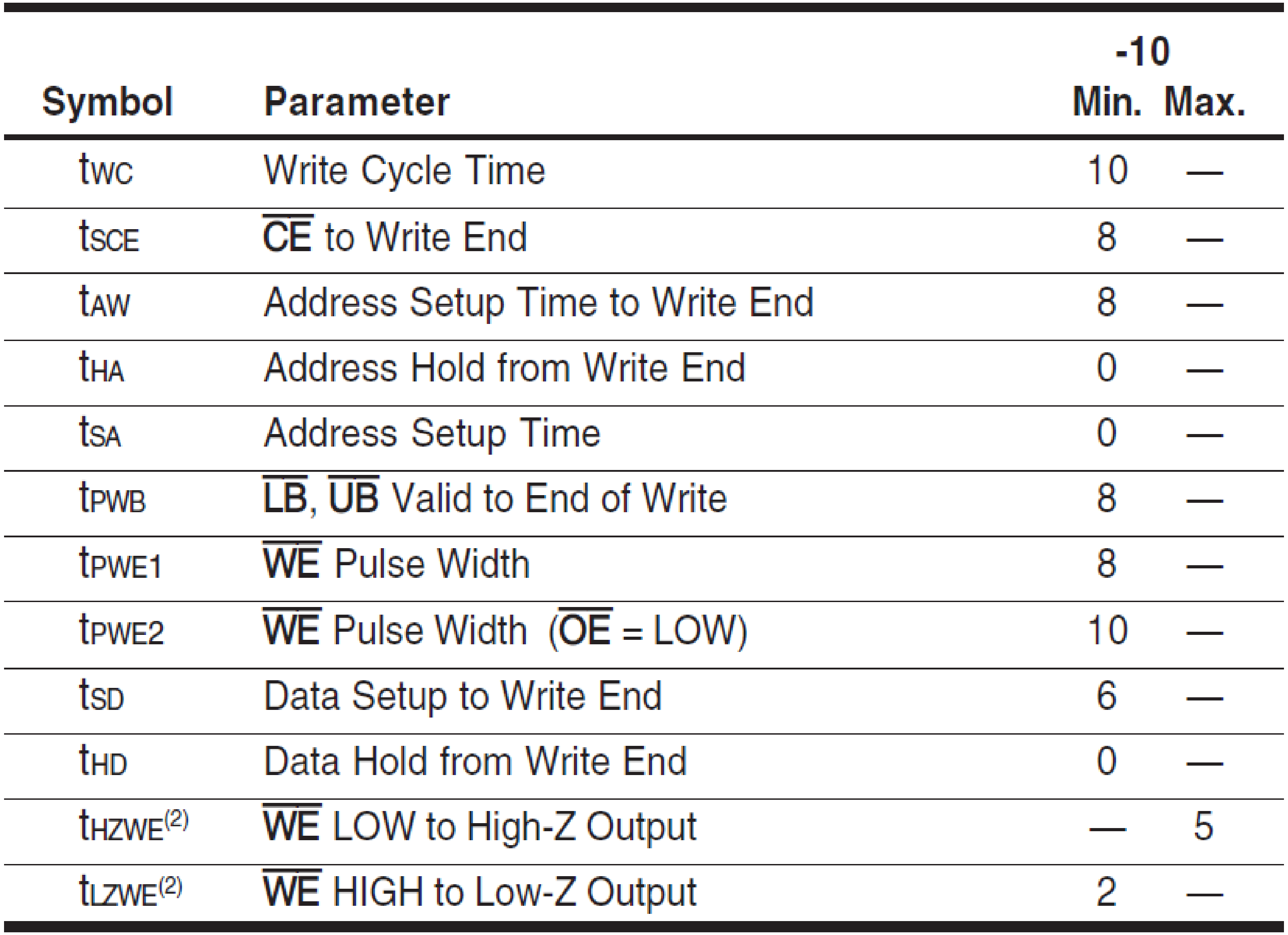
		-10	
Symbol	Parameter	Min.	Max.
t_{RC}	Read Cycle Time	10	—
t_{AA}	Address Access Time	—	10
t_{OHA}	Output Hold Time	2	—
t_{ACE}	$\overline{CE}$ Access Time	—	10
t_{DOE}	$\overline{OE}$ Access Time	—	4
t_{HZOE}[2]	$\overline{OE}$ to High-Z Output	—	4
t_{LZOE}[2]	$\overline{OE}$ to Low-Z Output	0	—
t_{HZCE}[2	$\overline{CE}$ to High-Z Output	0	4
t_{LZCE}[2]	$\overline{CE}$ to Low-Z Output	3	—
t_{BA}	$\overline{LB}$, $\overline{UB}$ Access Time	—	4
t_{HZB}[2]	$\overline{LB}$, $\overline{UB}$ to High-Z Output	0	3
t_{LZB}[2]	$\overline{LB}$, $\overline{UB}$ to Low-Z Output	0	—
t_{PU}	Power Up Time	0	—
t_{PD}	Power Down Time	—	10

WRITE CYCLE



t WC
ADDRESS
VALID ADDRESS
t SA
t SCE
t HA
CE
t AW
WE
t PWE1
t PWE2
t PBW
UB, LB
t HZWE
t LZWE
DOUT
DATA UNDEFINED
HIGH-Z
t SD
t HD
DIN
DATAIN VALID

WRITE CYCLE SWITCHING CHARACTERISTICS



Symbol	Parameter	-10	
		Min.	Max.
t_{WC}	Write Cycle Time	10	—
t_{SCE}	$\overline{CE}$ to Write End	8	—
t_{AW}	Address Setup Time to Write End	8	—
t_{HA}	Address Hold from Write End	0	—
t_{SA}	Address Setup Time	0	—
t_{PWB}	$\overline{LB}$, $\overline{UB}$ Valid to End of Write	8	—
t_{PWE1}	$\overline{WE}$ Pulse Width	8	—
t_{PWE2}	$\overline{WE}$ Pulse Width ($\overline{OE}$ = LOW)	10	—
t_{SD}	Data Setup to Write End	6	—
t_{HD}	Data Hold from Write End	0	—
t_{HZWE}(2)	$\overline{WE}$ LOW to High-Z Output	—	5
t_{LZWE}(2)	$\overline{WE}$ HIGH to Low-Z Output	2	—

FSMC

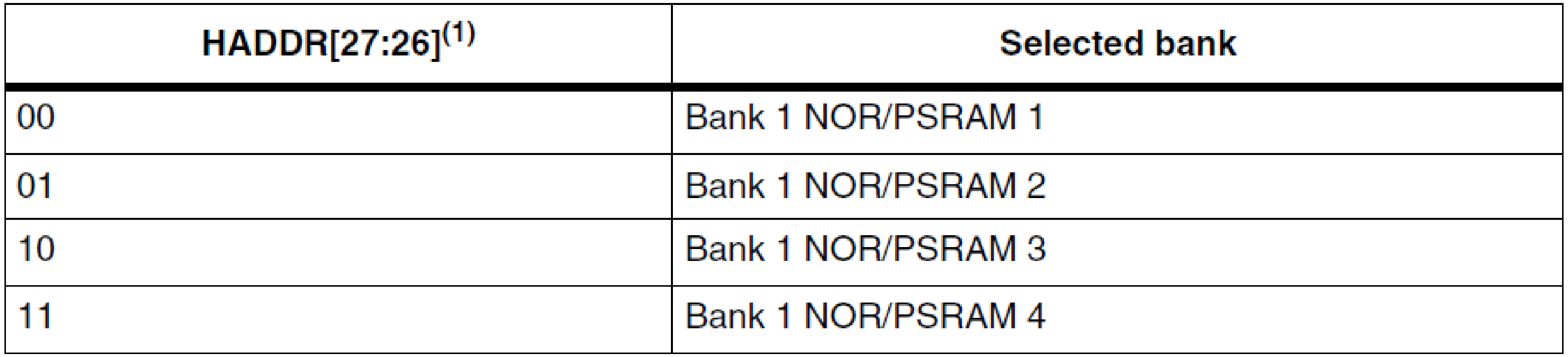
- Features
  - Interfaces with static memory-mapped devices including:
    - Static random access memory (SRAM)
    - Read-only memory (ROM)
    - NOR Flash memory
    - PSRAM (4 memory banks)8- or 16-bit wide databus
  - 8- or 16-bit wide databus
  - Independent chip select control for each memory bank
  - Independent configuration for each memory bank

FSMC

- Programmable timings to support a wide range of devices, in particular:
  - □ Programmable wait states (up to 15)
  - □ Programmable bus turnaround cycles (up to 15)
  - □ Programmable output enable and write enable delays (up to 15)
  - □ Independent read and write timings and protocol, so as to support the widest variety of memories and timings

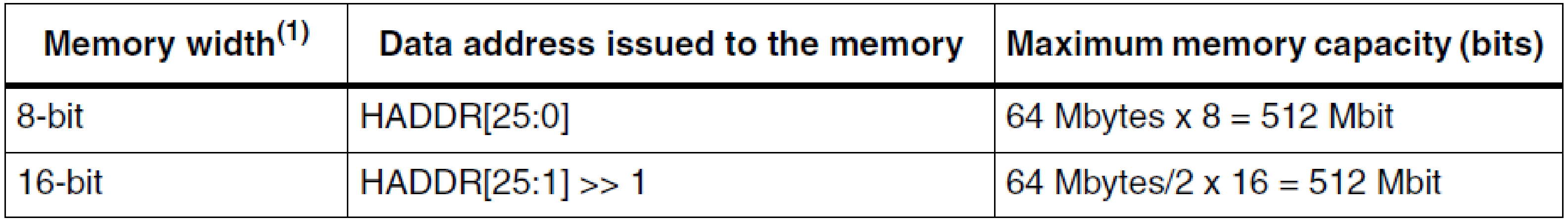
NOR/PSRAM address mapping

- NOR/PSRAM bank selection



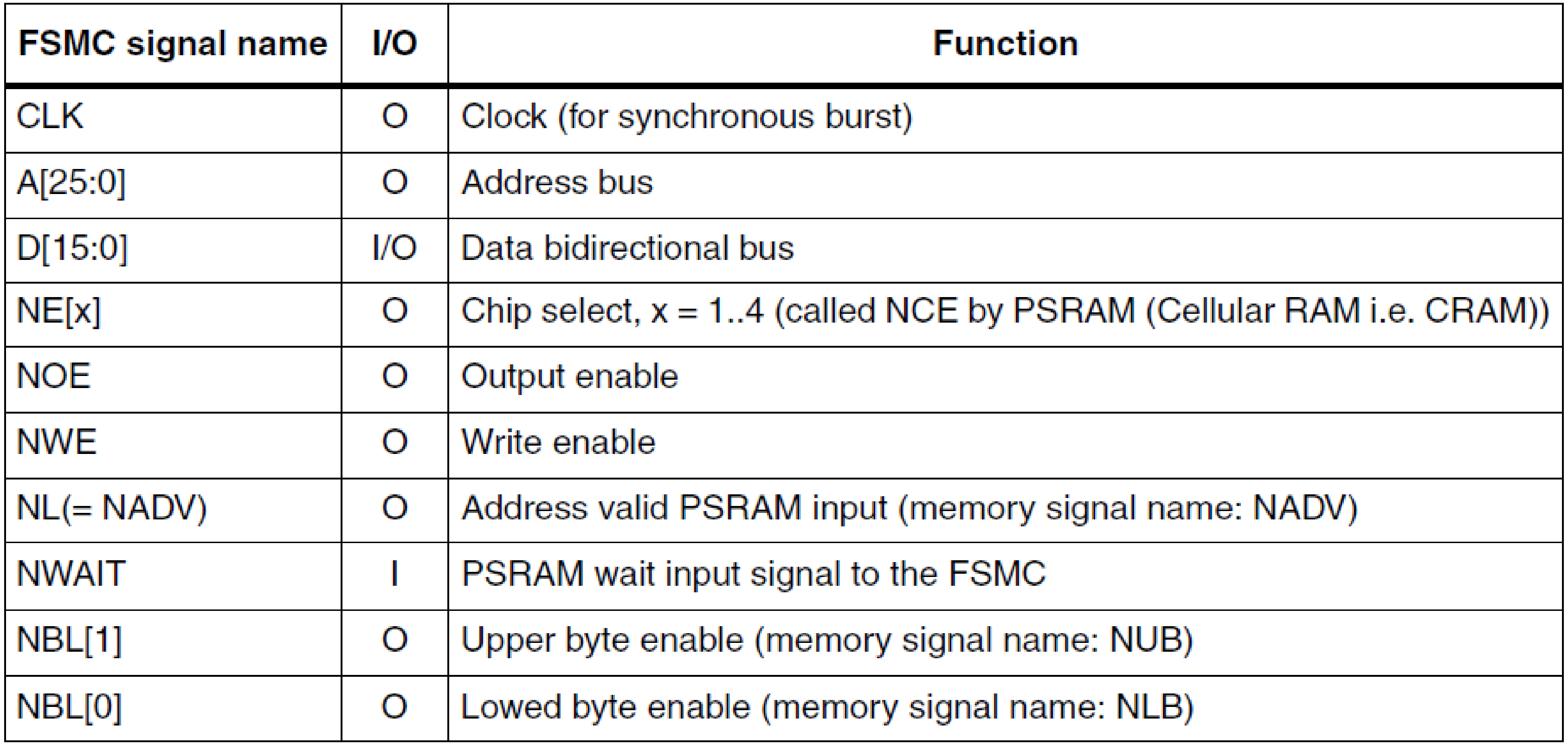
HADDR[27:26][1]	Selected bank
00	Bank 1 NOR/PSRAM 1
01	Bank 1 NOR/PSRAM 2
10	Bank 1 NOR/PSRAM 3
11	Bank 1 NOR/PSRAM 4

- External memory address



Memory width[1]	Data address issued to the memory	Maximum memory capacity (bits)
8-bit	HADDR[25:0]	64 Mbytes x 8 = 512 Mbit
16-bit	HADDR[25:1] >> 1	64 Mbytes/2 x 16 = 512 Mbit

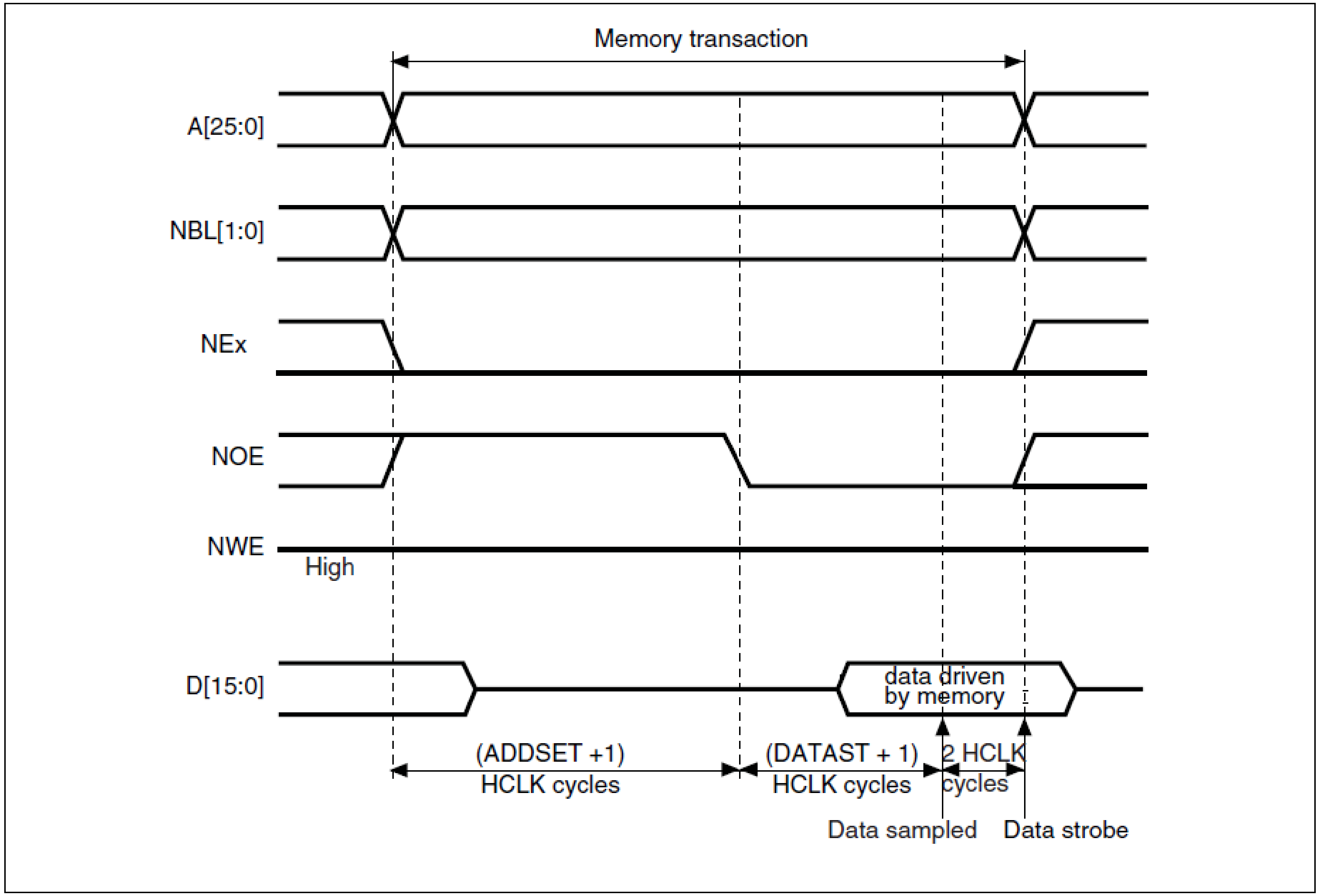
External memory interface signals



FSMC signal name	I/O	Function
CLK	O	Clock (for synchronous burst)
A[25:0]	O	Address bus
D[15:0]	I/O	Data bidirectional bus
NE[x]	O	Chip select, x = 1..4 (called NCE by PSRAM (Cellular RAM i.e. CRAM))
NOE	O	Output enable
NWE	O	Write enable
NL(= NADV)	O	Address valid PSRAM input (memory signal name: NADV)
NWAIT	I	PSRAM wait input signal to the FSMC
NBL[1]	O	Upper byte enable (memory signal name: NUB)
NBL[0]	O	Lowed byte enable (memory signal name: NLB)

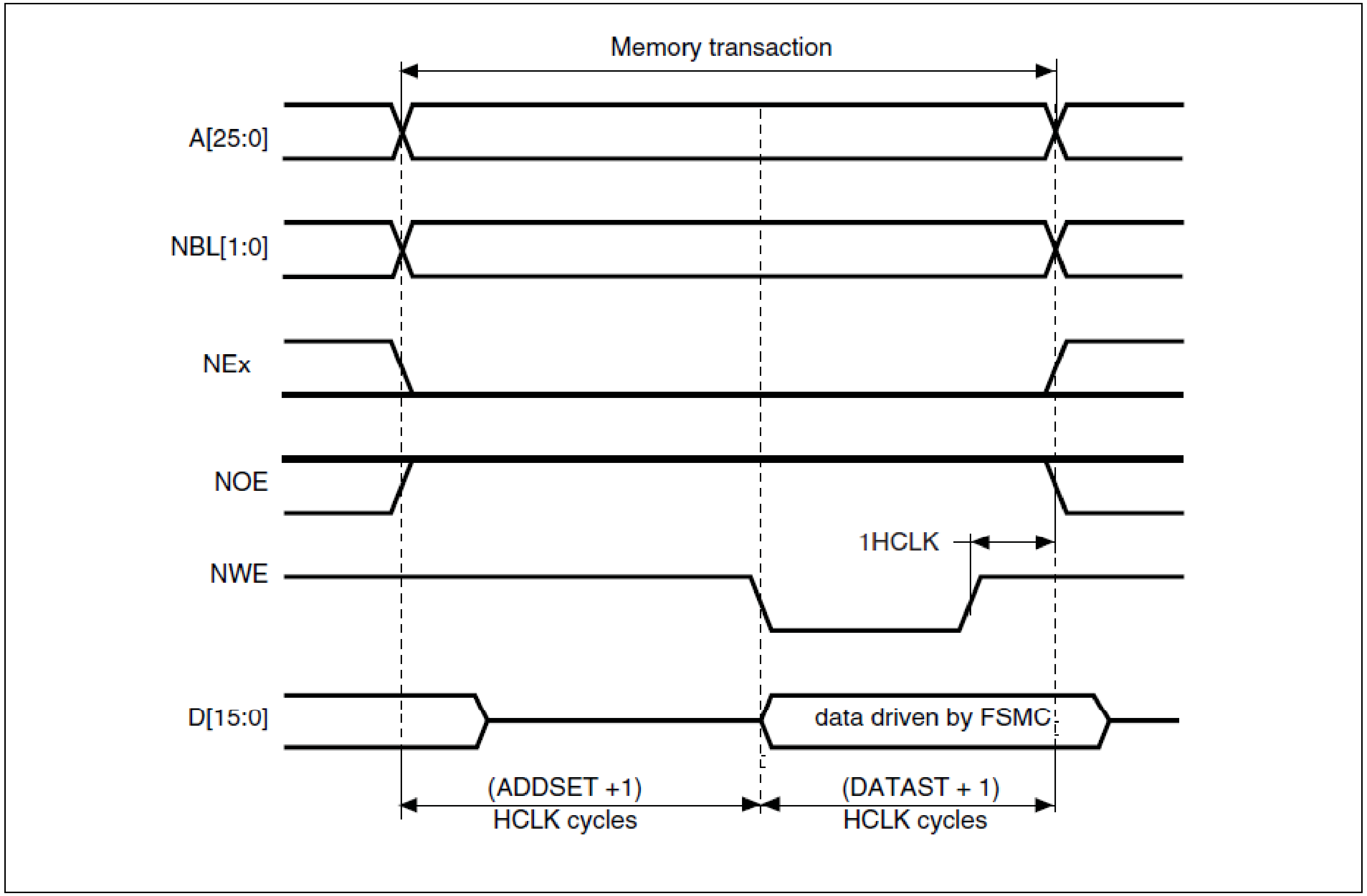
NOR Flash/PSRAM controller timing diagrams

- read accesses

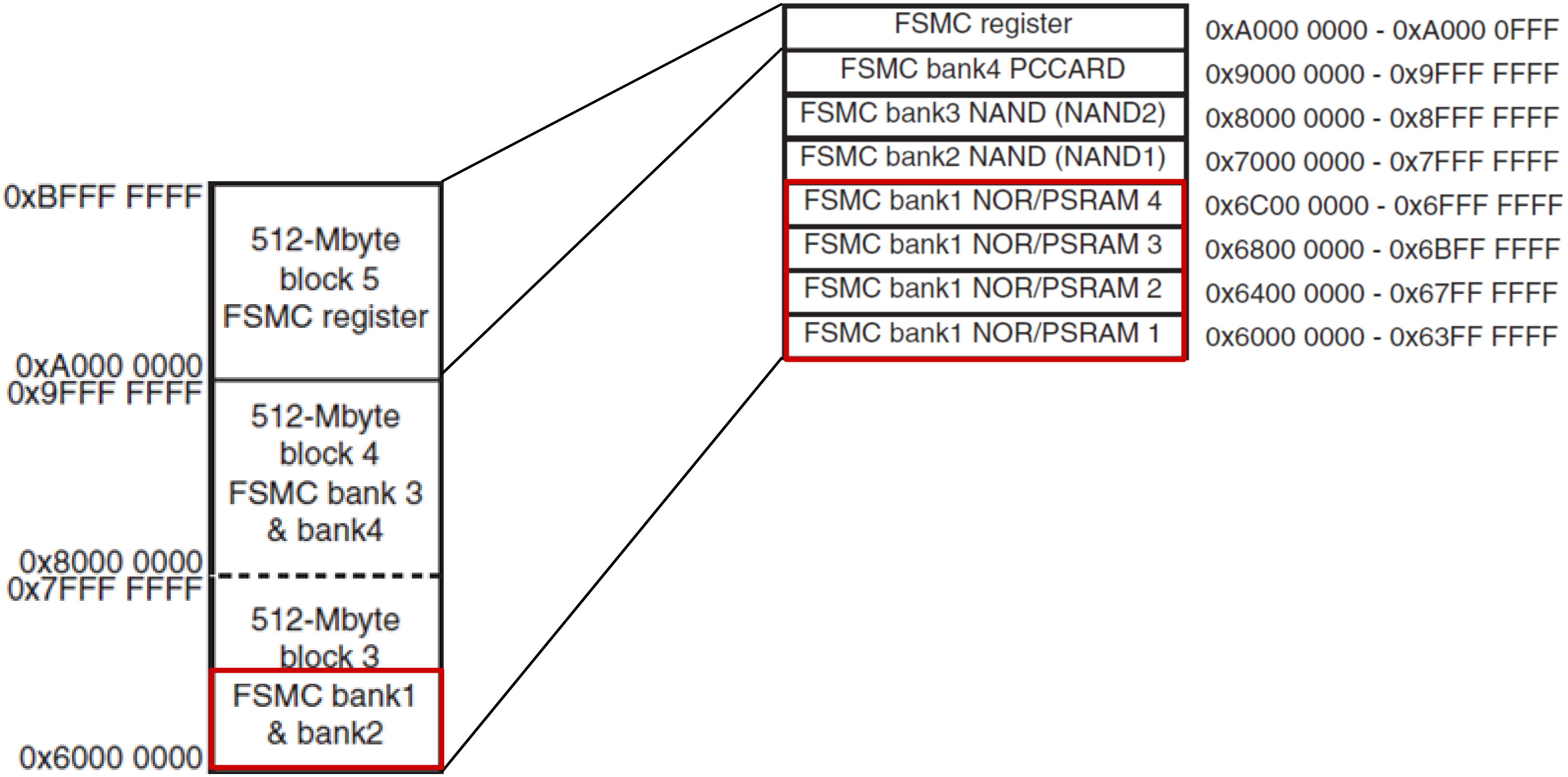


NOR Flash/PSRAM controller timing diagrams

- write accesses



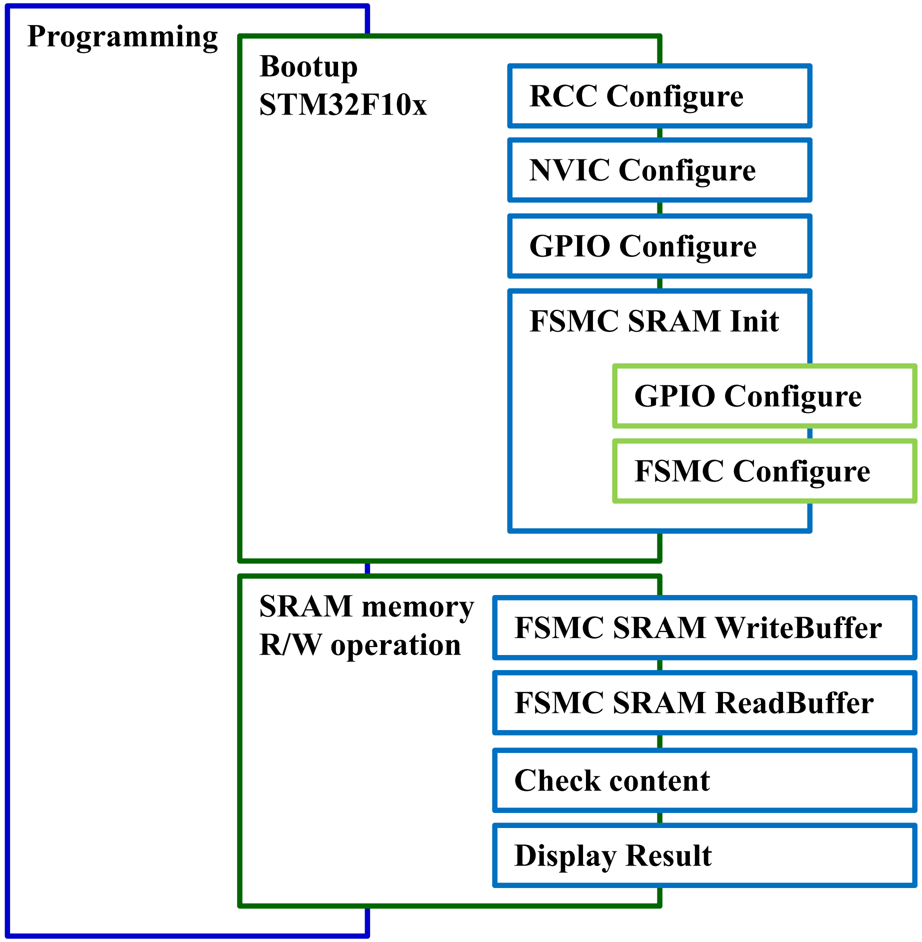
FSMC Memory map



FSMC register	0xA000 0000 - 0xA000 0FFF
FSMC bank4 PCCARD	0x9000 0000 - 0x9FFF FFFF
FSMC bank3 NAND (NAND2)	0x8000 0000 - 0x8FFF FFFF
FSMC bank2 NAND (NAND1)	0x7000 0000 - 0x7FFF FFFF
FSMC bank1 NOR/PSRAM 4	0x6C00 0000 - 0x6FFF FFFF
FSMC bank1 NOR/PSRAM 3	0x6800 0000 - 0x6BFF FFFF
FSMC bank1 NOR/PSRAM 2	0x6400 0000 - 0x67FF FFFF
FSMC bank1 NOR/PSRAM 1	0x6000 0000 - 0x63FF FFFF

Development Flow

Embedded Software Side



Bootup STM32F10x

```
int main(void)
{
#ifdef DEBUG
  debug();
#endif

  /* System Clocks Configuration */
  RCC_Configuration();

  /* NVIC Configuration */
  NVIC_Configuration();

  /* GPIO Configuration */
  GPIO_Configuration();

/* Write/read to/from FSMC SRAM memory */

  /* Configure FSMC Bank1 NOR/SRAM1 */
  FSMC_SRAM_Init();


……………………………………………………………………..
```

Configure FSMC IO

GPIO FwLib Functions List

Function name	Description
RCC_APB2PeriphClock Cmd	Enables or disables the High Speed APB (APB2) peripheral clock.
GPIO_Init	Initializes the GPIOx peripheral according to the specified parameters in the GPIO_InitStruct.
RCC_AHBPeriphClock Cmd	Enables or disables the AHB peripheral clock.

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD | RCC_APB2Periph_GPIOG |
RCC_APB2Periph_GPIOE | RCC_APB2Periph_GPIOF, ENABLE);

/*-- GPIO Configuration ------------------------------------------------*/
  /* SRAM Data lines configuration */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_8 |
GPIO_Pin_9 | GPIO_Pin_10 | GPIO_Pin_14 | GPIO_Pin_15;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOD, &GPIO_InitStructure);
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9 |
GPIO_Pin_10 | GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_13 | GPIO_Pin_14 |
GPIO_Pin_15;
  GPIO_Init(GPIOE, &GPIO_InitStructure);
  /* SRAM Address lines configuration */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 |
GPIO_Pin_3 | GPIO_Pin_4 | GPIO_Pin_5 | GPIO_Pin_12 | GPIO_Pin_13 |
GPIO_Pin_14 | GPIO_Pin_15;
  GPIO_Init(GPIOF, &GPIO_InitStructure);
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 |
GPIO_Pin_3 | GPIO_Pin_4 | GPIO_Pin_5;
  GPIO_Init(GPIOG, &GPIO_InitStructure);
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_13;
  GPIO_Init(GPIOD, &GPIO_InitStructure);
  /* NOE and NWE configuration */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4 |GPIO_Pin_5;
  GPIO_Init(GPIOD, &GPIO_InitStructure);
  /* NE1configuration */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7;
  GPIO_Init(GPIOD, &GPIO_InitStructure);
  /* NBL0, NBL1 configuration */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1;
  GPIO_Init(GPIOE, &GPIO_InitStructure);
  /* Enable the FSMC Clock */
  RCC_AHBPeriphClockCmd(RCC_AHBPeriph_FSMC, ENABLE);
```

Configure FSMC

FSMC FwLib Functions List

Function name	Description
RCC_AHBPeriphClockCmd	Enables or disables the AHB peripheral clock.
FSMC_NORSRAMInit	Initializes the FSMC NOR memory bank according to the parameters specified in FSMC_NORInitStruct.
FSMC_NORSRAMCm	Enables or disables the NOR/SRAM memory bank1.

```
/* Enable the FSMC Clock */
  RCC_AHBPeriphClockCmd(RCC_AHBPeriph_FSMC, ENABLE);

/*-- FSMC Configuration --*/
  p.FSMC_AddressSetupTime = 0;
  p.FSMC_AddressHoldTime = 0;
  p.FSMC_DataSetupTime = 2;
  p.FSMC_BusTurnAroundDuration = 0;
  p.FSMC_CLKDivision = 0;
  p.FSMC_DataLatency = 0;
  p.FSMC_AccessMode = FSMC_AccessMode_A;
  FSMC_NORSRAMInitStructure.FSMC_Bank = FSMC_Bank1_NORSRAM1;
  FSMC_NORSRAMInitStructure.FSMC_DataAddressMux =
FSMC_DataAddressMux_Disable;
  FSMC_NORSRAMInitStructure.FSMC_MemoryType = FSMC_MemoryType_SRAM;
  FSMC_NORSRAMInitStructure.FSMC_MemoryDataWidth = FSMC_MemoryDataWidth_16b;
  FSMC_NORSRAMInitStructure.FSMC_BurstAccessMode =
FSMC_BurstAccessMode_Disable;
  FSMC_NORSRAMInitStructure.FSMC_WaitSignalPolarity =
FSMC_WaitSignalPolarity_Low;
  FSMC_NORSRAMInitStructure.FSMC_WrapMode = FSMC_WrapMode_Disable;
  FSMC_NORSRAMInitStructure.FSMC_WaitSignalActive =
FSMC_WaitSignalActive_BeforeWaitState;
  FSMC_NORSRAMInitStructure.FSMC_WriteOperation =
FSMC_WriteOperation_Enable;
  FSMC_NORSRAMInitStructure.FSMC_WaitSignal = FSMC_WaitSignal_Disable;
  FSMC_NORSRAMInitStructure.FSMC_ExtendedMode = FSMC_ExtendedMode_Disable;
  FSMC_NORSRAMInitStructure.FSMC_AsyncWait = FSMC_AsyncWait_Disable;
  FSMC_NORSRAMInitStructure.FSMC_WriteBurst = FSMC_WriteBurst_Disable;
  FSMC_NORSRAMInitStructure.FSMC_ReadWriteTimingStruct = &p;
  FSMC_NORSRAMInitStructure.FSMC_WriteTimingStruct = &p;
  FSMC_NORSRAMInit(&FSMC_NORSRAMInitStructure);

  /* Enable FSMC Bank1_SRAM Bank */
  FSMC_NORSRAMCmd(FSMC_Bank1_NORSRAM1, ENABLE);
```

硬體電路配置

Mapping Table

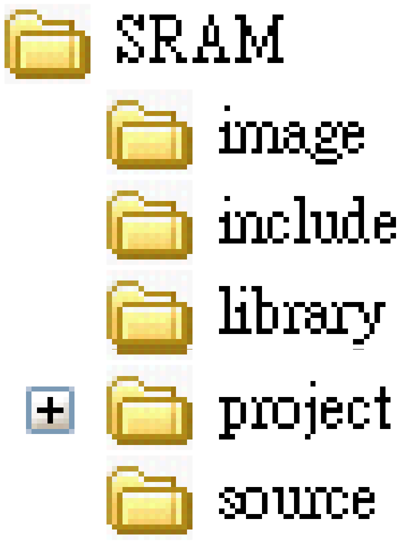
Num.	SRAM	STM32	Num.	SRAM	STM32	Num.	SRAM	STM32
1	A0	FSMC_A0	14	A13	FSMC_A13	27	I/O8	FSMC_D8
2	A1	FSMC_A1	15	A14	FSMC_A14	28	I/O9	FSMC_D9
3	A2	FSMC_A2	16	A15	FSMC_A15	29	I/O10	FSMC_D10
4	A3	FSMC_A3	17	A16	FSMC_A16	30	I/O11	FSMC_D11
5	A4	FSMC_A4	18	A17	FSMC_A17	31	I/O12	FSMC_D12
6	A5	FSMC_A5	19	I/O0	FSMC_D0	32	I/O13	FSMC_D13
7	A6	FSMC_A6	20	I/O1	FSMC_D1	33	I/O14	FSMC_D14
8	A7	FSMC_A7	21	I/O2	FSMC_D2	34	I/O15	FSMC_D15
9	A8	FSMC_A8	22	I/O3	FSMC_D3	35	CE	FSMC_nNE1
10	A9	FSMC_A9	23	I/O4	FSMC_D4	36	OE	FSMC_nOE
11	A10	FSMC_A10	24	I/O5	FSMC_D5	37	WE	FSMC_nWE
12	A11	FSMC_A11	25	I/O6	FSMC_D6	38	UB	FSMC_NBL1
13	A12	FSMC_A12	26	I/O7	FSMC_D7	39	LB	FSMC_NBL0

實驗步驟

- □ 範例目錄架構
- □ 範例說明
- □ 預設定義說明
- □ 燒錄MIAT_STM32

範例目錄架構

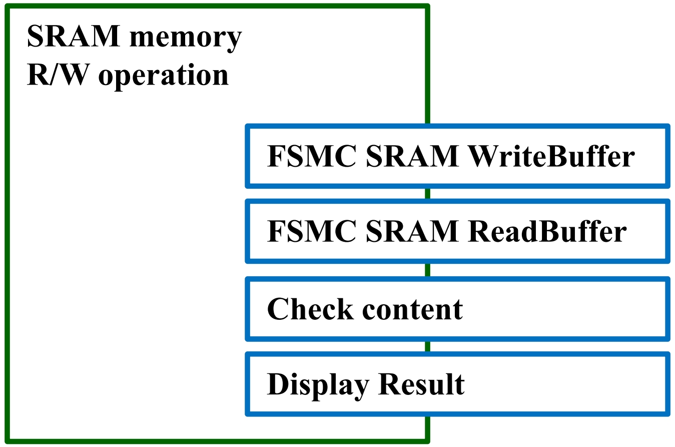
- 範例目錄
  - 測試映像檔
  - 含括檔
  - 函式庫
  - 專案檔
  - 原始碼



```
SRAM
    image
    include
    library
  + project
    source
```

範例說明

Embedded Software Side

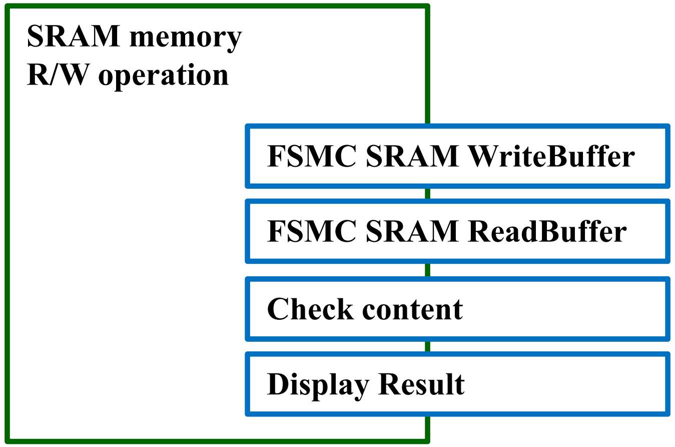


SRAM memory R/W operation

```
  /* Write data to FSMC SRAM memory */
  /* Fill the buffer to send */
  Fill_Buffer(TxBuffer, BUFFER_SIZE, 0x3212);
  FSMC_SRAM_WriteBuffer(TxBuffer, WRITE_READ_ADDR, BUFFER_SIZE);


  /* Read data from FSMC SRAM memory */
  FSMC_SRAM_ReadBuffer(RxBuffer, WRITE_READ_ADDR, BUFFER_SIZE);

  /* Read back SRAM memory and check content correctness */
  for (Index = 0x00; (Index < BUFFER_SIZE) && (WriteReadStatus == 0);
Index++)
  {
    if (RxBuffer[Index] != TxBuffer[Index])
    {
      WriteReadStatus = Index + 1;
    }
  }
```

範例說明

Embedded Software Side



Display Result

```
while (1)
 {
   if (WriteReadStatus == 0)
   {  /* OK Turn on USERLED */
     GPIO_SetBits(GPIOF, GPIO_Pin_11);
   }
   else
   { /* KO Turn off USERLED */
     GPIO_ResetBits(GPIOF, GPIO_Pin_11);
     /* Insert delay */
     Delay(0xAFFFF);
     /* Turn on USERLED */
     GPIO_SetBits(GPIOF, GPIO_Pin_11);
     /* Insert delay */
     Delay(0xAFFFF);
   }
 }
```

如果寫入與讀取Buffer內容相同，外部記憶體使用正常，USERLED紅燈恆亮

如果寫入與讀取Buffer內容不同，外部記憶體使用異常，USERLED紅燈閃爍

預設定義說明

- #define Bank1_SRAM1_ADDR ((u32)0x60000000)
  - 定義SRAM起始點
- #define BUFFER_SIZE 0x400
  - 定義測試資料大小
  - 資料大小必需小於0xC000
- #define WRITE_READ_ADDR 0x8000
  - 定義SRAM寫入起始點
  - 寫入起始點WRITE_READ_ADDR + BUFFER_SIZE 必需小於0x6000C000

燒錄MIAT_STM32

- □ Rebuilder all target files產生HEX
- □ DFU File Manager轉換HEX產生DFU
- □ DfuSe Demonstration燒錄DFU
- □ Leave DFU mode

內部與外部SRAM存取控制實驗

實驗一

WU-YANG

Technology Co., Ltd.

實驗一練習

- 練習:
  - 修改寫入位置測試是否正常
  - 修改寫入資料大小測試是否正常
  - 取消FSMC_SRAM_Init測試是否正常

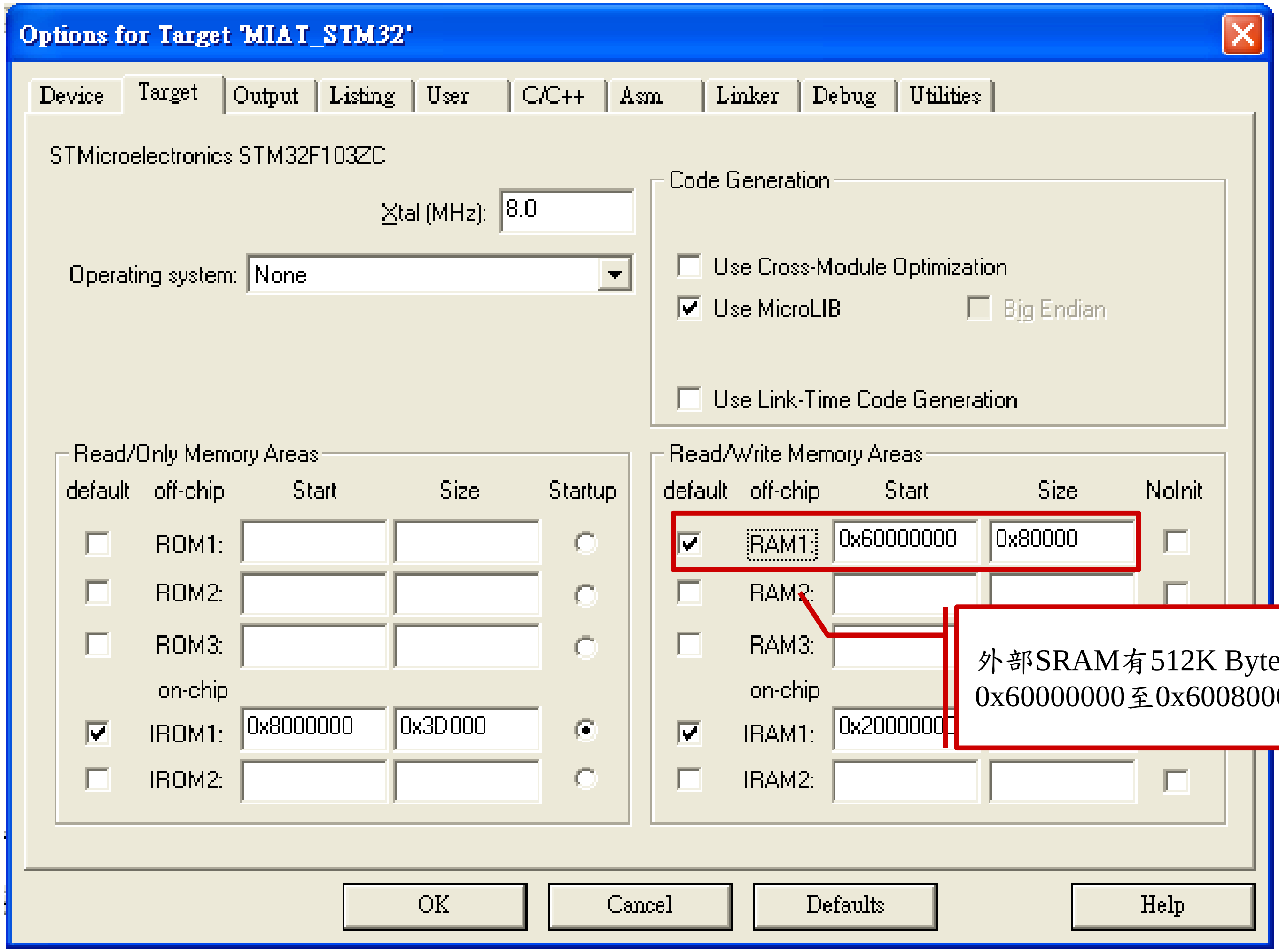
實驗目的(二)

- 使用MIAT_STM32實驗板透過FSMC控制外部SRAM並設定為data memory進行存取控制實驗，同樣利用LED確認存取是否正常。

實驗原理

- □ External SRAM
- □ RVMDK環境設定
- □ Development Flow
- □ ARM Configure
- □ Startup Code

RVMDK環境設定



STMicroelectronics STM32F103ZC

Xtal (MHz): 8.0

Operating system: None

Code Generation

☐ Use Cross-Module Optimization

☑ Use MicroLIB ☐ Big Endian

☐ Use Link-Time Code Generation

Read/Only Memory Areas

default	off-chip	Start	Size	Startup
☐	ROM1:			○
☐	ROM2:			○
☐	ROM3:			○
	on-chip			
☑	IROM1:	0x8000000	0x3D000	●
☐	IROM2:			○

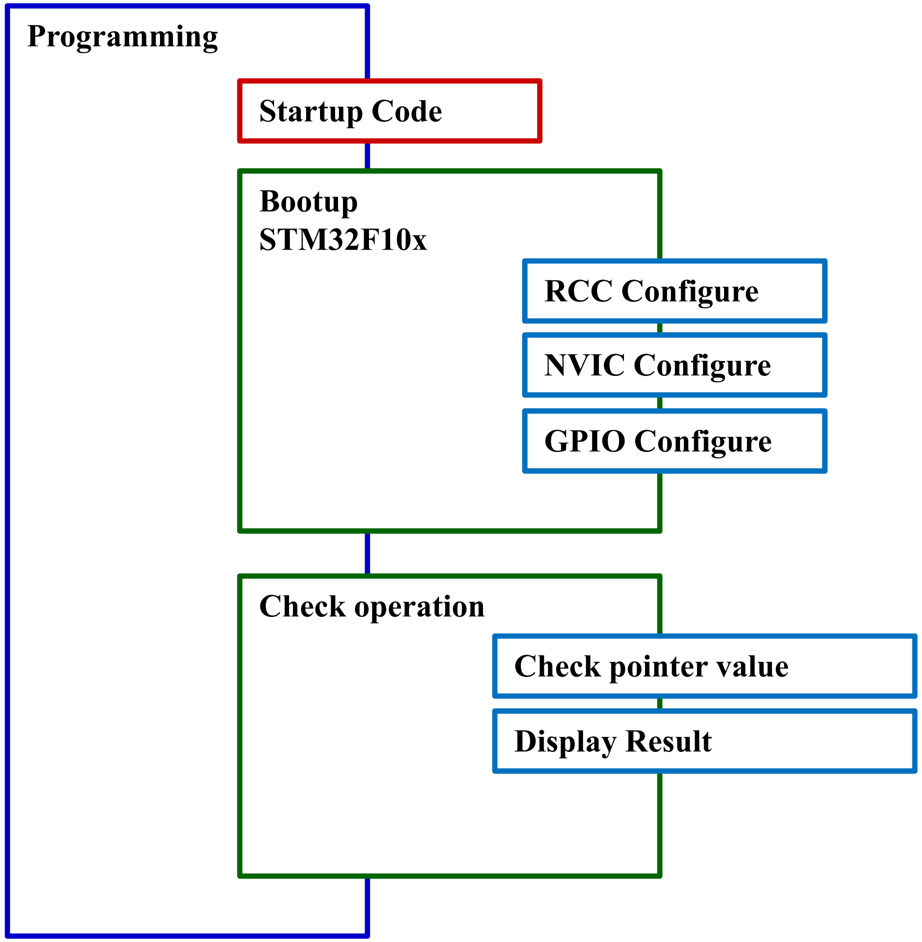
Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
☑	RAM1:	0x60000000	0x80000	☐
☐	RAM2:			☐
☐	RAM3:			
	on-chip			
☑	IRAM1:	0x2000000		
☐	IRAM2:			☐

外部SRAM有512K Bytes位置由0x60000000至0x60080000

Development Flow

Embedded Software Side

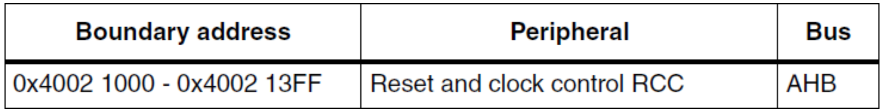


Bootup STM32F10x

```
int main(void)
{
#ifdef DEBUG
  debug();
#endif

  /* System Clocks Configuration */
  RCC_Configuration();

  /* NVIC Configuration */
  NVIC_Configuration();

  /* GPIO Configuration */
  GPIO_Configuration();

 for (Index = 0; Index <1024 ; Index++)
  {
    Tab[Index] =Index;
  }
  TabAddr = (u32)Tab; /* should be 0x600000xx */

  /* Get main stack pointer value */
  MSPValue = __MRS_MSP(); /* should be 0x2000xxxx */
…………………………………………………………………..
```

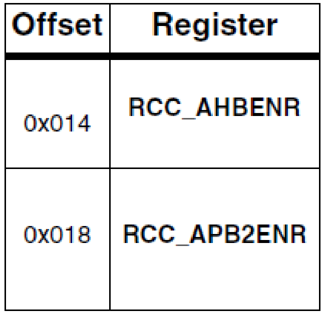
Startup Code

Register boundary addresses



Boundary address	Peripheral	Bus
0x4002 1000 - 0x4002 13FF	Reset and clock control RCC	AHB

RCC register map



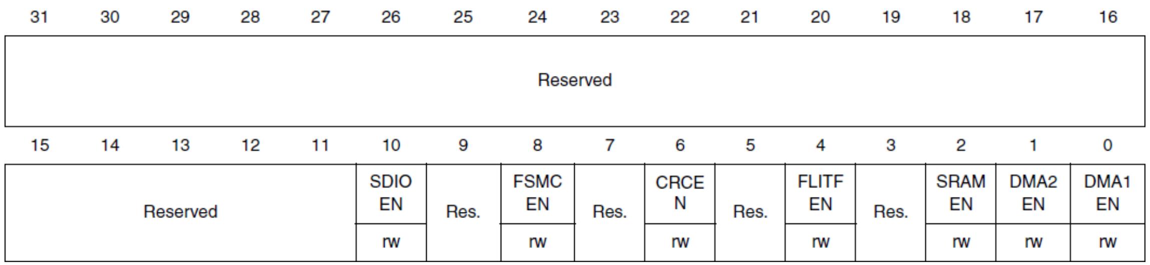
Offset	Register
0x014	RCC_AHBENR
0x018	RCC_APB2ENR

RCC Configure

```
; Enable FSMC clock

LDR R0,= 0x00000114

LDR R1,= 0x40021014

STR R0,[R1]

; Enable GPIOD, GPIOE, GPIOF and GPIOG clocks

LDR R0,= 0x000001E0

LDR R1,= 0x40021018

STR R0,[R1]
```

RCC Register

AHB Peripheral Clock enable register (RCC_AHBENR)

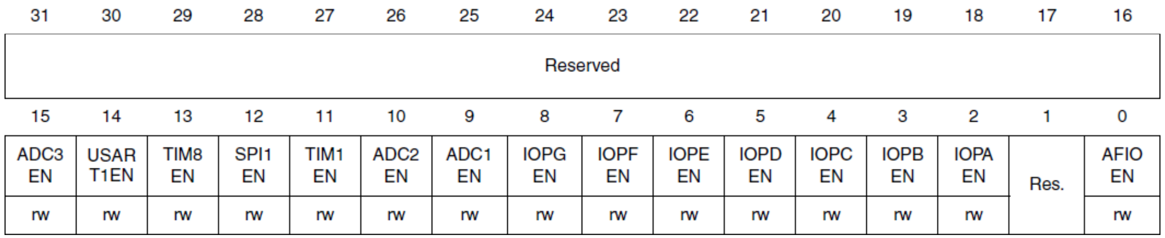


31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SDIO EN	Res.	FSMC EN	Res.	CRCE N	Res.	FLITF EN	Res.	SRAM EN	DMA2 EN	DMA1 EN
					rw		rw		rw		rw		rw	rw	rw

Bit 8 **FSMCEN: FSMC clock enable**
Set and cleared by software.
0: FSMC clock disabled
1: FSMC clock enabled

APB2 peripheral clock enable register (RCC_APB2ENR)



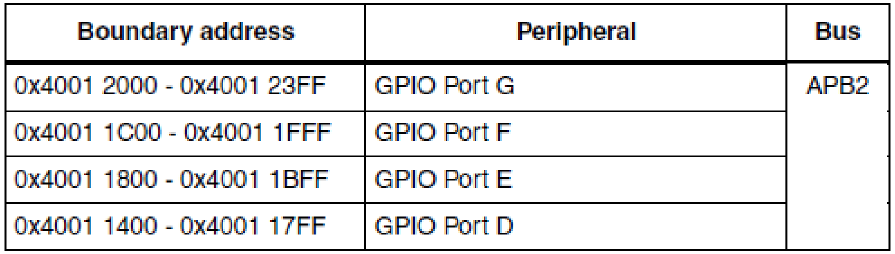
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 EN	USAR T1EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	IOPG EN	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bit 8/7/6/5 **IOPGEN: I/O port G/F/E/D clock enable**
Set and cleared by software.
0: I/O port G/F/E/D clock disabled
1: I/O port G/F/E/D clock enabled

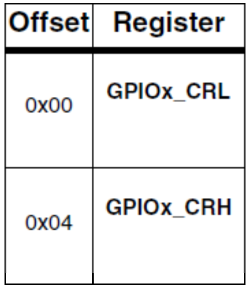
Startup Code

Register boundary addresses



Boundary address	Peripheral	Bus
0x4001 2000 - 0x4001 23FF	GPIO Port G	APB2
0x4001 1C00 - 0x4001 1FFF	GPIO Port F	
0x4001 1800 - 0x4001 1BFF	GPIO Port E	
0x4001 1400 - 0x4001 17FF	GPIO Port D	

GPIO register map



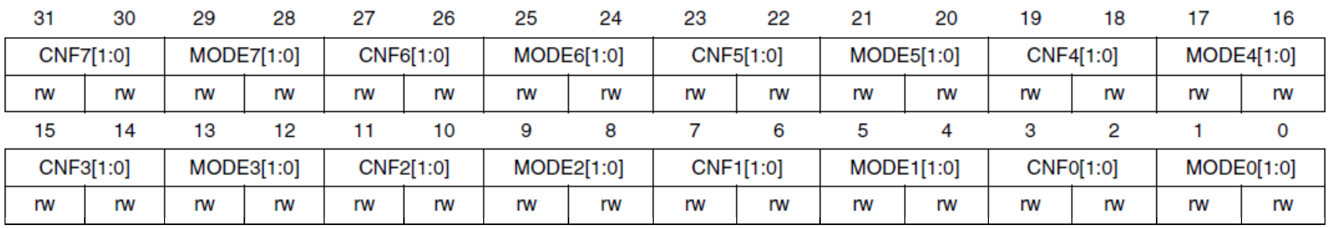
Offset	Register
0x00	GPIOx_CRL
0x04	GPIOx_CRH

GPIO Configure

```
; SRAM Data lines, NOE and NWE configuration
; SRAM Address lines configuration
; NOE, NEW, NE1, NBL0, NBL1 configuration
LDR R0,= 0xB4BB44BB
LDR R1,= 0x40011400
STR R0,[R1]
LDR R0,= 0xBBBBBBBB
LDR R1,= 0x40011404
STR R0,[R1]
LDR R0,= 0xB44444BB
LDR R1,= 0x40011800
STR R0,[R1]
LDR R0,= 0xBBBBBBBB
LDR R1,= 0x40011804
STR R0,[R1]
LDR R0,= 0x44BBBBBB
LDR R1,= 0x40011C00
STR R0,[R1]
LDR R0,= 0xBBBB4444
LDR R1,= 0x40011C04
STR R0,[R1]
LDR R0,= 0x44BBBBBB
LDR R1,= 0x40012000
STR R0,[R1]
```

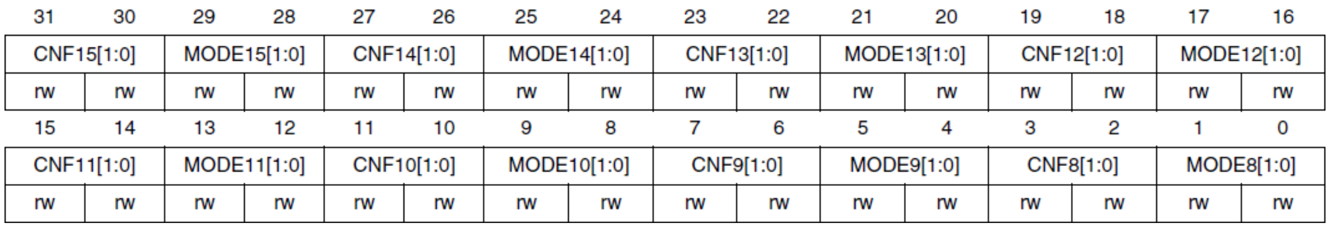
GPIO Register

Port configuration register low



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Port configuration register high



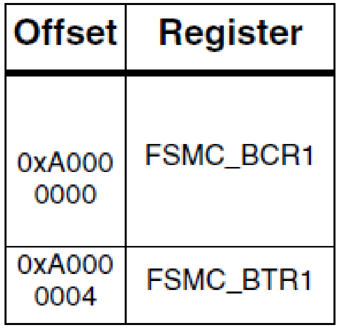
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30, 27:26,23:22, 19:18, 15:14,11:10, 7:6, 3:2
CNFy[1:0]: Port x configuration bits (y= 0 .. 15)
In input mode (MODE[1:0]=00):
00: Analog input mode
01: Floating input (reset state)
10: Input with pull-up / pull-down
11: Reserved
In output mode (MODE[1:0] >00):
00: General purpose output push-pull
01: General purpose output Open-drain
10: Alternate function output Push-pull
11: Alternate function output Open-drain

Bits 29:28, 25:24,
21:20, 17:16, 13:12,
9:8, 5:4, 1:0
MODEy[1:0]: Port x mode bits (y= 0 .. 15)
00: Input mode (reset state)
01: Output mode, max speed 10 MHz.
10: Output mode, max speed 2 MHz.
11: Output mode, max speed 50 MHz.

Startup Code

FSMCregister map



Offset	Register
0xA000 0000	FSMC_BCR1
0xA000 0004	FSMC_BTR1

FSMC Configure

```
; FSMC Configuration
; Enable FSMC Bank1_SRAM Bank


LDR R0,= 0x00001011

LDR R1,= 0xA0000000

STR R0,[R1]


LDR R0,= 0x00000200

LDR R1,= 0xA0000004

STR R0,[R1]
```

FSMC Register

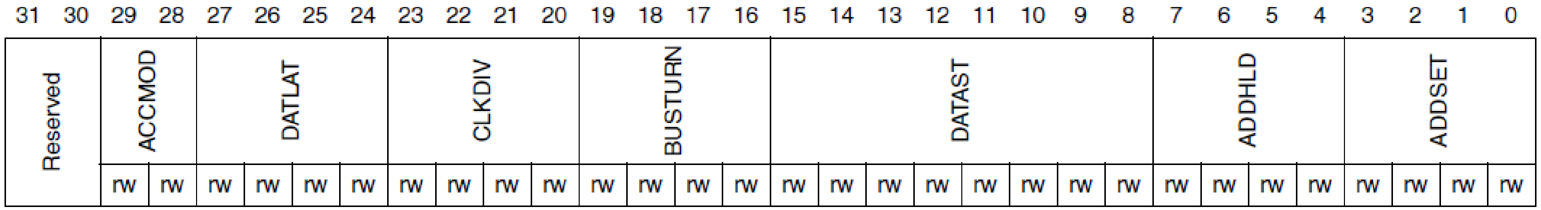
SRAM/NOR-Flash chip-select control registers 1 (FSMC_BCR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CBURSTRW	Reserved				EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID		MTYP		MUXEN	MBKEN
												rw					rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

Bit 12 **WREN: Write enable bit.**
This bit indicates whether write operations are enabled/disabled in the bank by the FSMC:
0: Write operations are disabled in the bank by the FSMC, an AHB error is reported,
1: Write operations are enabled for the bank by the FSMC (default after reset).
Bits 5:4 **MWID: Memory databus width.**
Defines the external memory device width, valid for all type of memories.
00: 8 bits,
01: 16 bits (default after reset),
10: reserved, do not use,
11: reserved, do not use.
Bit 0 **MBKEN: Memory bank enable bit.**
Enables the memory bank. After reset Bank1 is enabled, all others are disabled. Accessing a disabled bank causes an ERROR on AHB bus.
0: Corresponding memory bank is disabled
1: Corresponding memory bank is enabled

FSMC Register

SRAM/NOR-Flash chip-select timing registers 1 (FSMC_BTR1)



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ACCMOD		DATLAT				CLKDIV				BUSTURN				DATAST								ADDHLD				ADDSET			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

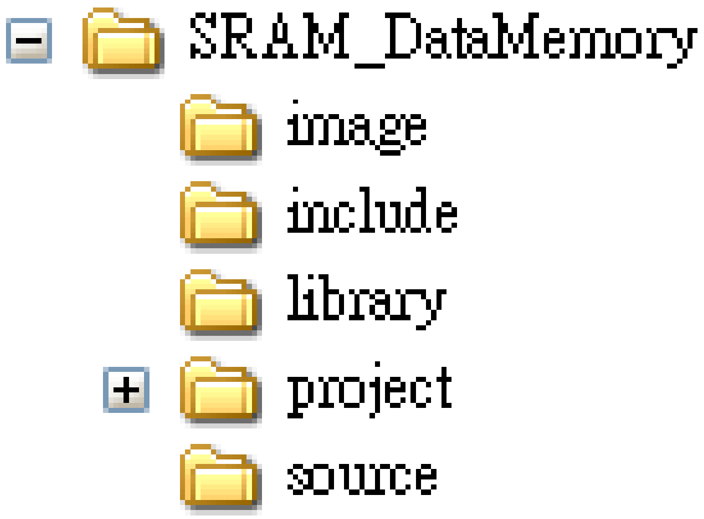
Bits 15:8 **DATAST: Data-phase duration**
0000 0000: Reserved
0000 0001: DATAST phase duration = 2 × HCLK clock cycles
0000 0010: DATAST phase duration = 3 × HCLK clock cycles
...
1111 1111: DATAST phase duration = 256 × HCLK clock cycles (default value after reset)
Bits 7:4 **ADDHLD: Address-hold phase duration**
0000: Reserved
0001: ADDHLD phase duration = 2 × HCLK clock cycle
0010: ADDHLD phase duration = 3 × HCLK clock cycle
...
1111: ADDHLD phase duration = 16 × HCLK clock cycles (default value after reset)
Bits 3:0 **ADDSET: Address setup phase duration**
These bits are written by software to define the duration of the *address setup phase (refer to Figure 162 to Figure 172), used in SRAMs, ROMs and asynchronous NOR Flash:*
0000: ADDSET phase duration = 1 × HCLK clock cycle
...
1111: ADDSET phase duration = 16 × HCLK clock cycles (default value after reset)

實驗步驟

- □ 範例目錄架構
- □ 範例說明
- □ 預設定義說明

範例目錄架構

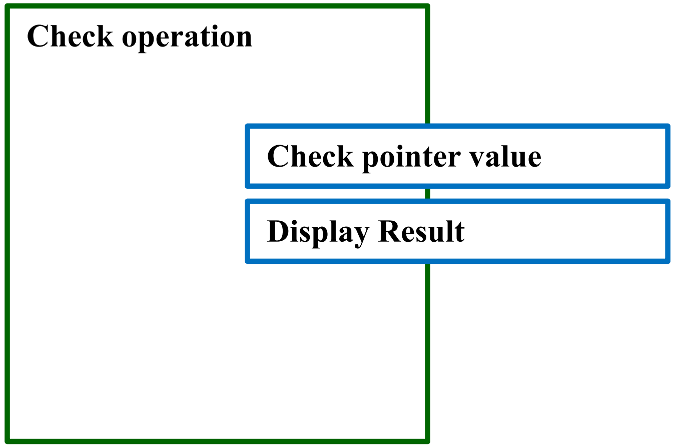
- 範例目錄
  - 測試映像檔
  - 含括檔
  - 函式庫
  - 專案檔
  - 原始碼



```
⊟ SRAM_DataMemory
     image
     include
     library
  ⊞  project
     source
```

範例說明

Embedded Software Side



Check operation

Check pointer value

Display Result

Display Result

```
 /* Infinite loop */
 while (1)
 {
   if (((TabAddr&0xFFFFFF00) == 0x60000000) &&
((MSPValue&0xFFFF0000) == 0x20000000))
   {  /* OK Turn on USERLED */
     GPIO_SetBits(GPIOF, GPIO_Pin_11);
   }
   else
   { /* KO Turn off USERLED */
     GPIO_ResetBits(GPIOF, GPIO_Pin_11);
     /* Insert delay */
     Delay(0xAFFFF);
     /* Turn on USERLED */
     GPIO_SetBits(GPIOF, GPIO_Pin_11);
     /* Insert delay */
     Delay(0xAFFFF);
   }
```

如果Tab位置在0x600000??且Stack pointer位置在0x2000????，外部記憶體使用正常，USERLED紅燈恆亮

如果Tab位置不在0x600000??或Stack pointer位置不在0x2000????，外部記憶體使用異常，USERLED紅燈閃爍

預設定義說明

- DATA_IN_ExtSRAM EQU 1
  - External SRAM Configuration
    - 0=> DISABLE
    - 1=> ENABLE
- u32 Tab[1024]
  - 宣告於外部SRAM的記憶體

data memory存取控制實驗

實驗二

WU-YANG

Technology Co., Ltd.

實驗二練習

- 練習:
  - 取消RVMDK環境外部SRAM設定測試是否正常
  - 修改DATA_IN_ExtSRAM EQU 0測試是否正常
  - 修改外部SRAM記憶體Tab變數大小測試是否正常

Q & A