Chương 9

BỘ ĐIỀU KHIỂN CHO PHÉP TRUY XUẤT LINH HOẠT BỘ NHỚ TĨNH

FLEXIBLE STATIC MEMORY CONTROLLER -FSMC

9.1 GIỚI THIỆU FSMC

FSMC là giao diện truyền dữ liệu 16 bit đồng bộ và bất đồng bộ giữa vi điều khiển với các loại bộ nhớ bên ngoài. FSMC có những đặc điểm sau:

- Giao tiếp truyền dữ liệu giữa AHB với các loại bộ nhớ ngoại có giao thức phù hợp.
- Đáp ứng được các yêu cầu về thời gian truy xuất của các loại bộ nhớ ngoại.
- Tất cả các bộ nhớ ngoại chia sẻ địa chỉ, dữ liệu và tín hiệu điều khiển. Do đó để chọn bộ nhớ cần truy xuất ta phải sử dụng thêm tín hiệu chọn chip(chip select) riêng biệt cho từng bộ nhớ. Tại một thời điểm khối FSMC chỉ có thể giao tiếp với một bộ nhớ và chỉ cho phép một truy xuất xảy ra.

9.2 NHỮNG ĐẶC ĐIỂM CHÍNH CỦA FSMC

FSMC của dòng ARM STM32F1X có những đặc điểm sau:

- Hỗ trợ giao tiếp với các loại bộ nhớ tĩnh.
  - Ram tĩnh (SRAM – static random access memory)
  - NOR Flash
  - PSRAM (4 bank bộ nhớ)
- Hai bank NAND Flash hỗ trợ ECC có thể kiểm tra tới 8 Kbyte dữ liệu.
- Hỗ trợ giao tiếp PC Card 16 bit.
- Hỗ trợ chế độ truy xuất burst cho phép đồng bộ các thiết bị(ví dụ giữa NOR Flash và PSRAM)
- Cho phép cài đặt chế độ 8 hoặc 16 bit dữ liệu
- Các tín hiệu điều khiển chip select được thiết kế độc lập cho từng bank bộ nhớ.
- Cho phép lập trình thời gian trễ cho phù hợp với nhiều loại bộ nhớ như là:
  - Thời gian ở trạng thái chờ (tối đa là 15)
  - Thời gian quay vòng của chu kỳ bus (tối đa 15)
  - Thời gian trễ trước khi cho phép xuất dữ liệu và ghi dữ liệu(tối đa 15)
  - Thời gian trễ cho việc đọc, ghi và giao thức truyền nhận.
- Cho phép truy xuất dữ liệu theo byte khi dùng PSRAM và SRAM
- Chuyển đổi từ bus 32 bit AHB sang 8 hoặc 16 bit nối tiếp để có thể truy xuất được các loại bộ nhớ ngoại 8 hoặc 16 bit.
- Hỗ trợ FIFO(First In First Out) dài 2 word mỗi word gồm 32 bit làm bộ đệm giúp giải phóng AHB khi truy xuất bộ nhớ tốc độ thấp.
- Hỗ trợ giao thức truyền bất đồng bộ.

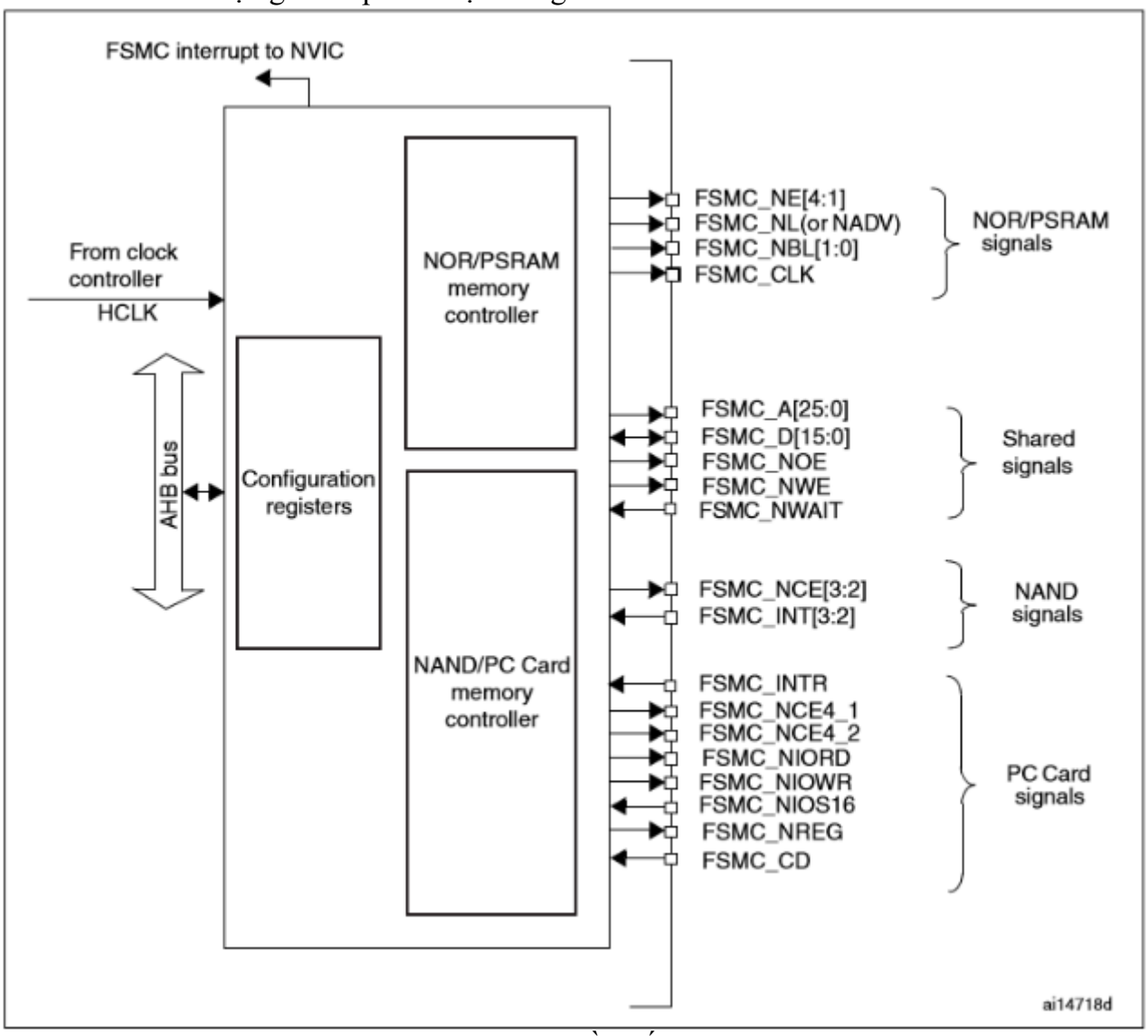
9.3 SƠ ĐỒ KHỐI VÀ CÁCH GIAO TIẾP VỚI BỘ NHỚ NGOẠI CỦA FSMC

9.3.1 Giới thiệu các khối chính của FSMC

FSMC có 4 khối chính:

- Giao diện AHB(bao gồm các thanh ghi cấu hình FSMC)
- Bộ điều khiển NOR Flash/PSRAM
- Bộ điều khiển NAND Flash/ PC Card

- Giao diện giao tiếp thiết bị bên ngoài.



Hình 9.1 Sơ đồ khối FSMC

9.3.2 Các tín hiệu FSMC sử dụng để giao tiếp với NOR Flash

Bảng 9.1 Các tín hiệu FSMC sử dụng để giao tiếp với NOR Flash

Tên các tín hiệu FSMC	I/O	Chức năng
CLK	O	Cấp xung clock cho chế độ truy xuất đồng bộ
A[25:0]	O	Bus địa chỉ(Address bus)
D[15:0]	I/O	Khi hoạt động ở chế độ không đa hợp thì đây là 16 đường dữ liệu. Khi hoạt động ở chế độ đa hợp thì đây là 16 đường đa hợp địa chỉ và dữ liệu.
NE[x]	O	Tín hiệu chọn chip(chip select), x=1..4
NOE	O	Cho phép ngõ ra(Output enable- Còn gọi là cho phép đọc)
NWE	O	Cho phép ghi(write enable)
NL(=NADV)	O	Cho phép chốt(Latch enable), tín hiệu này được dùng để báo là địa chỉ đã sẵn sàng đối với một số loại NOR Flash
NWAIT	I	Tín hiệu yêu cầu chờ từ NOR Flash

9.3.3 Các tín hiệu FSMC sử dụng để giao tiếp với PSRAM/SRAM

Bảng 9.2 Các tín hiệu FSMC sử dụng để giao tiếp với PSRAM/SRAM

Tên các Tín hiệu FSMC	I/O	Chức năng
CLK	O	Cấp xung clock cho chế độ truy xuất đồng bộ
A[25:0]	O	Bus địa chỉ(Address bus)
D[15:0]	I/O	Khi hoạt động ở chế độ không đa hợp thì đây là 16 đường dữ liệu. Khi hoạt động ở chế độ đa hợp thì đây là 16 đường đa hợp địa chỉ và dữ liệu.
NE[x]	O	Tín hiệu chọn chip(chip select), x=1..4
NOE	O	Cho phép ngõ ra(Output enable- Còn gọi là cho phép đọc)
NWE	O	Cho phép ghi(write enable)
NL(=NADV)	O	Cho phép chốt(Latch enable), tín hiệu này được dùng để báo là địa chỉ đã sẵn sàng đối với PSRAM
NWAIT	I	Tín hiệu yêu cầu chờ từ PSRAM
NBL[1]	O	Cho phép byte cao(Upper byte enable)
NBL[0]	O	Cho phép byte thấp(Lowed byte enable)

9.3.4 Các tín hiệu FSMC sử dụng để giao tiếp với NAND Flash 8 bit

Bảng 9.3 Các tín hiệu FSMC sử dụng để giao tiếp với NAND Flash 8 bit

Tên các tín hiệu FSMC	I/O	Chức năng
A[17]	O	Tín hiệu cho phép chốt địa chỉ ALE
A[16]	O	Tín hiệu cho phép chốt mã lệnh CLE
D[7:0]	I/O	8 bit đa hợp địa chỉ và dữ liệu
NCE[x]	O	Tín hiệu chọn chip(chip select), x=2,3
NOE(=NRE)	O	Cho phép ngõ ra(Còn gọi là cho phép đọc - NRE)
NWE	O	Cho phép ghi(write enable)
NWAIT/INT[3:2]	I	Tín hiệu báo FSMC biết NAND Flash sẵn sàng hay bận

9.3.5 Các tín hiệu FSMC sử dụng để giao tiếp với NAND Flash 16 bit

Bảng 9.4 Các tín hiệu FSMC sử dụng để giao tiếp với NAND Flash 16 bit

Tên các tín hiệu FSMC	I/O	Chức năng
A[17]	O	Chân cho phép chốt địa chỉ ALE
A[16]	O	Chân cho phép chốt mã lệnh CLE
D[15:0]	I/O	16 bit đa hợp địa chỉ và dữ liệu
NCE[x]	O	Chân chọn chip(chip select), x=2,3
NOE(=NRE)	O	Cho phép ngõ ra(Còn gọi là cho phép đọc - NRE)
NWE	O	Cho phép ghi(write enable)
NWAIT/INT[3:2]	I	Chân báo cho FSMC biết NAND Flash sẵn sàng hay bận

9.3.6 Các tín hiệu FSMC sử dụng để giao tiếp với PC Cart 16 bit

Bảng 9.5 Các tín hiệu FSMC sử dụng để giao tiếp với PC Cart 16 bit

Tên các tín hiệu FSMC	I/O	Chức năng
A[10:0]	O	Bus địa chỉ
NIOS16	I	Truyền dữ liệu tại I/O. Chân này phải được nối với GND
NIORD	O	Cho phép đọc tại I/O
NIOWR	O	Cho phép ghi tại I/O
NREG	O	Thanh ghi báo việc truy xuất đang xảy ra ở vùng Common hay vùng Attribute
D[15:0]	I/O	Đường dữ liệu hai chiều
NCE4_1	O	Tín hiệu chọn chip 1
NCE4_2	O	Tín hiệu chọn chip 2
NOE	O	Cho phép đọc tại vùng Common hay Attribute
NWE	O	Cho phép ghi tại vùng Common hay Attribute
NWAIT	I	Tín hiệu báo cho FSMC biết PC Cart bận hay sẵng sàng
INTR	I	Tín hiệu báo ngắt mà PC Cart gởi tới FSMC
CD	I	Tín hiệu báo sự hiện diện của PC Card

9.4 TỔ CHỨC ĐỊA CHỈ CỦA FSMC

FSMC chia vùng địa chỉ giao tiếp với các bộ nhớ ngoại ra làm 4 bank mỗi bank có kích thước 256Mbyte.

Bank 1 được dùng để giao tiếp tới 4 bộ nhớ NOR Flash hay PSRAM. Bank này được chia ra thành 4 bank nhỏ mỗi bank nhỏ có kích thước 64 Mbyte như sau:

- Bank 1 – NOR/PSRAM1 – Địa chỉ nền 6000 0000H – Được chọn bởi NE1
- Bank 1 – NOR/PSRAM2 – Địa chỉ nền 6400 0000H – Được chọn bởi NE2
- Bank 1 – NOR/PSRAM3– Địa chỉ nền 6800 0000H – Được chọn bởi NE3
- Bank 1 – NOR/PSRAM4– Địa chỉ nền 6C00 0000H – Được chọn bởi NE4

Bank 2 và 3 được dùng để giao tiếp với bộ nhớ NAND Flash(mỗi bộ nhớ dùng 1 bank)

Bank 4 được dùng để giao tiếp với PC Card

Chú ý:

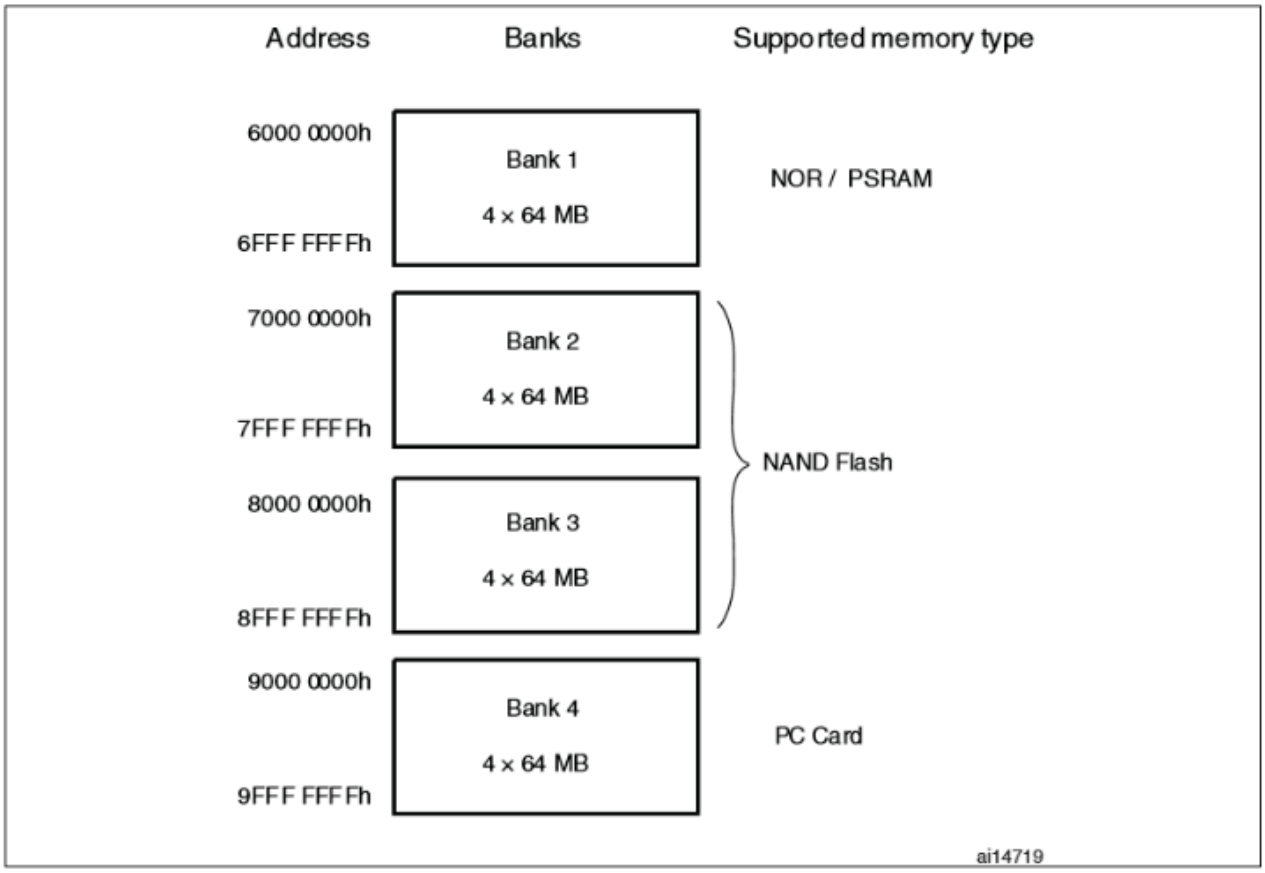
Khi sử dụng mỗi bank ta phải khai báo loại bộ nhớ mà ta đang sử dụng thông qua thanh ghi cấu hình.

Khi giao tiếp với bộ nhớ có bề rộng dữ liệu là 16 bit thì FSMC sẽ sử dụng các đường địa chỉ nội của AHB là HADDR[25:1] để tạo thành các đường địa chỉ để giao tiếp với bộ nhớ ngoại là FSMC_A[24:0].

Bất chấp bề rộng dữ liệu của bộ nhớ là 8 bit hay 16 bit thì khi kết nối ta luôn phải kết nối FSMC_A[0] với đường địa chỉ A[0] của bộ nhớ.

Bảng 9.6 Địa chỉ bộ nhớ ngoại

Bề rộng dữ liệu	Địa chỉ dữ liệu được cấp cho bộ nhớ
8 bit	HADDR[25:0]
16 bit	HADDR[25:1]>>1



Hình 9.2 Tổ chức địa chỉ của FSMC

Ví dụ 9.1: Một mạch giao tiếp STM32F103VET6 với màn hình LCD TFT 16 bit màu theo chuẩn FSMC như **hình 9.3**. SV hãy dựa vào hình và tính toán địa chỉ của vùng dữ liệu và mã lệnh của LCD.

Tính toán

Chân FSMC_NE4 được sử dụng để làm chân chip enable cho IC 74193 nên địa chỉ nền là 6C00 0000H

Để chân LCD_CS tích cực thì đường địa chỉ FSMC_A21= '1' và FSMC_A22= '0' nên địa chỉ offset là **01**00 0000 0000 0000 0000 0000B = 40 0000H

- **Giải thích:** Do bề rộng dữ liệu là 16 bit nên mặc dù phần cứng giao tiếp với FSMC_A22 và FSMC_A21 nhưng khi tính toán ta phải tính với A23 và A22 do các địa chỉ này bị dịch phải 1 bit theo **bảng 9.6** mà thành A22 và A21.

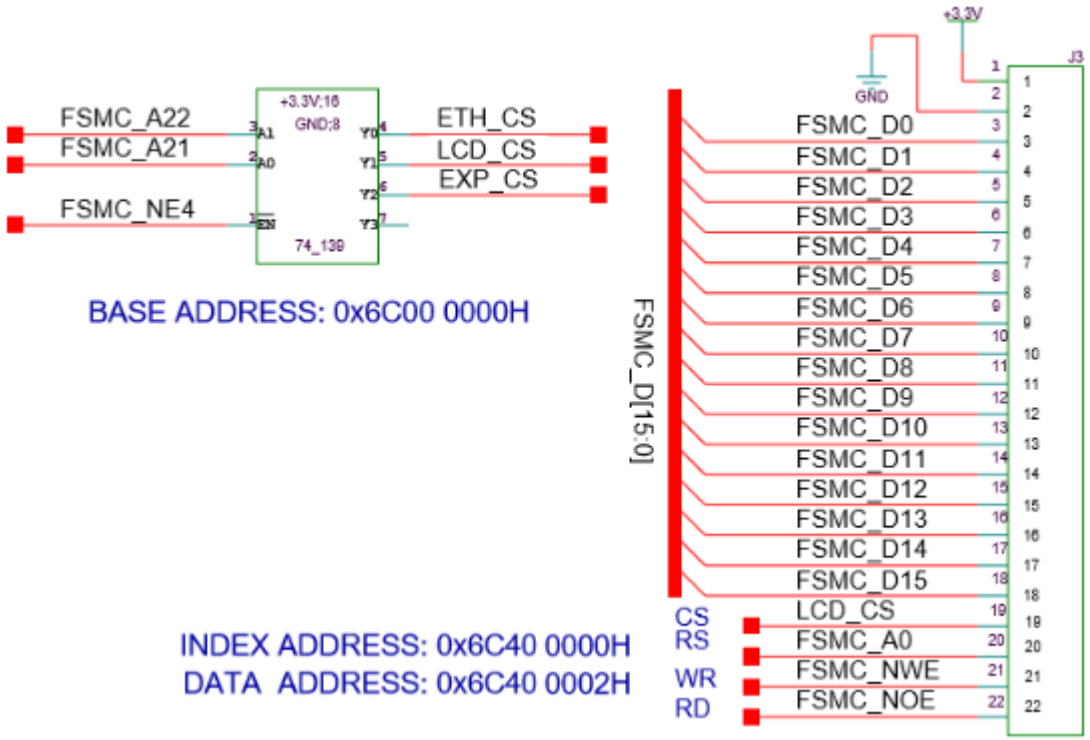
⇨ Vậy địa chỉ để truy xuất LCD TFT lúc này là 6C00 0000H + 40 0000H = 6C40 0000H

Chân RS của LCD(chân chọn thanh ghi mã lệnh – dữ liệu) được nối với FSMC_A0 nên:

- Để chọn thanh ghi mã lệnh (RS= '0') →FSMC_A0 = '0' → offset =0
- Để chọn thanh ghi dữ liệu (RS ='1') →FSMC_A0 = '1' → offset =2

⇨ Cuối cùng ta tính được:

- Địa chỉ thanh ghi mã lệnh là : 6C40 0000H +0 = 6C40 0000H
- Địa chỉ thanh ghi dữ liệu là : 6C40 0000H +2 = 6C40 0002H



Hình 9.3 Giao tiếp FSMC với LCD TFT

9.5 CÁC CHẾ ĐỘ GIAO TIẾP CƠ BẢN CỦA FSMC VỚI NOR FLASH/PSRAM

9.5.1 Chế độ giao tiếp bất đồng bộ và không đa hợp với bộ nhớ NOR Flash 16 bit

a. Các bước để cấu hình cho FSMC

Để điều khiển được bộ nhớ NOR Flash ta cần phải cài đặt các thông số sau cho FSMC

- Chọn bank cần sử dụng: Có 4 bank độc lập hỗ trợ giao tiếp với NOR Flash/PSRAM và mỗi bank có một chân chọn chip(chip select) riêng.
- Bật hoặc **tắt** chế độ đa hợp địa chỉ dữ liệu.
- Chọn loại bộ nhớ được sử dụng **: NOR Flash** hay SRAM hay PSRAM
- Chọn bề rộng của bus dữ liệu giao tiếp : 8 hoặc **16 bit**
- Cho phép hoặc tắt chế độ truy xuất burst để đồng bộ bộ nhớ NOR Flash
- Cấu hình cho tín hiệu chờ (Wait) : bật/ tắt, mức tích cực và thời gian trễ.
- Bật hoặc **tắt** chế độ mở rộng:
  - Nếu sử dụng chế độ mở rộng(set bit EXTMOD trong thanh ghi FSMC_BCRx) thì FSMC có thể hoạt động theo 4 chế độ A, B, C, D. Ta cũng có thể sử dụng kết hợp các chế độ này với nhau ví dụ như quá trình đọc ta cấu hình ở chế độ A nhưng quá trình ghi ta lại cấu hình ở chế độ B. Đối với chế độ mở rộng thời gian trễ cho quá trình đọc và ghi cũng được cấu hình độc lập với nhau thông qua 2 thanh ghi FSMC_BTR(cấu hình thời gian trễ cho quá trình đọc) và thanh ghi FSMC_BWTR(cấu hình thời gian trễ cho quá trình ghi).
  - Nếu không sử dụng chế độ mở rộng (xóa bit EXTMOD trong thanh ghi FSMC_BCRx) thì FSMC chỉ có thể hoạt động theo 1 trong 2 chế độ sau:
    Chế độ 1: Là chế độ mặc định khi ta chọn loại bộ nhớ giao tiếp là SRAM/PSRAM (chọn bằng cách gán các bit MTYP = 0x00 hoặc 0x01 trong thanh ghi FSMC_BCRx)
    Chế độ 2: Là chế độ mặc định khi ta chọn loại bộ nhớ giao tiếp là NOR Flash(gán các bit MTYP = 0x10 trong thanh ghi FSMC_BCRx).
- Khi không sử dụng chế độ mở rộng thì thời gian trễ cho quá trình đọc và ghi được cấu hình chung với nhau thông qua thanh ghi FSMC_BTR.

Bảng 9.7 Các chế độ giao tiếp bất đồng bộ của FSMC với NOR Flash/PSRAM

Chế độ bất đồng bộ	Loại bộ nhớ	
	SRAM/CRAM	NOR
Tắt chế độ mở rộng	Chế độ 1 (Mode 1)	Chế độ 2 (Mode 2)
Bật chế độ mở rộng	Chế độ A (Mode A)	Chế độ B,C,D (Mode B,C,D)

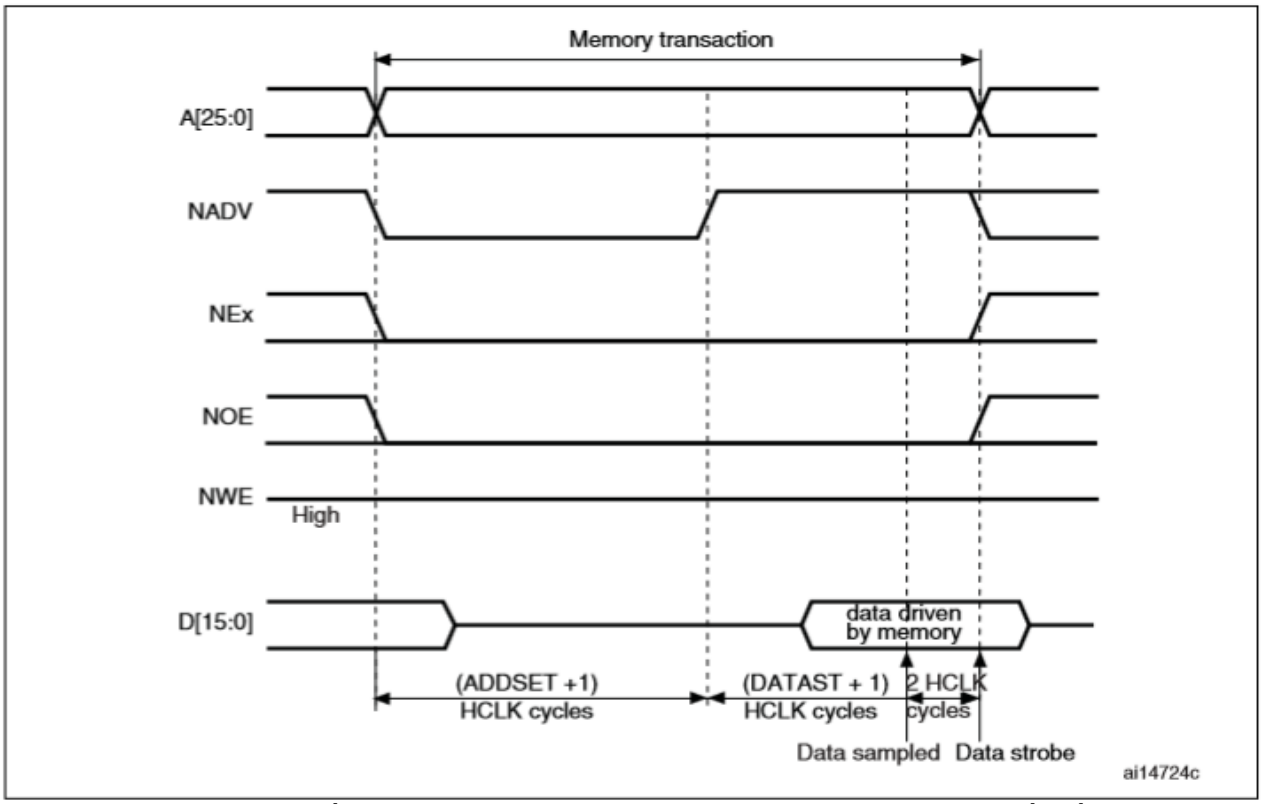
- Tính toán và cài đặt các thông số sau sao cho phù hợp với loại bộ nhớ được sử dụng
  - ADDSET : thời gian cài đặt địa chỉ (address setup time)
  - DATAST: thời gian cài đặt dữ liệu (data setup time)
  - ACCMOD: Chế độ truy xuất (access mode)
- Nếu sử dụng bộ nhớ đồng bộ thì ta phải tính toán và cài đặt thêm các thông số sau:
  - CLKDIV: hệ số chia xung clock (clock divide ratio)
  - DATLAT: độ trễ dữ liệu (data latency)

Chú ý: Ở một số loại NOR Flash quá trình đọc là đồng bộ nhưng quá trình ghi là bất đồng bộ. Vì vậy khi lập trình ta cần phải tính toán chính xác các thông số trên.

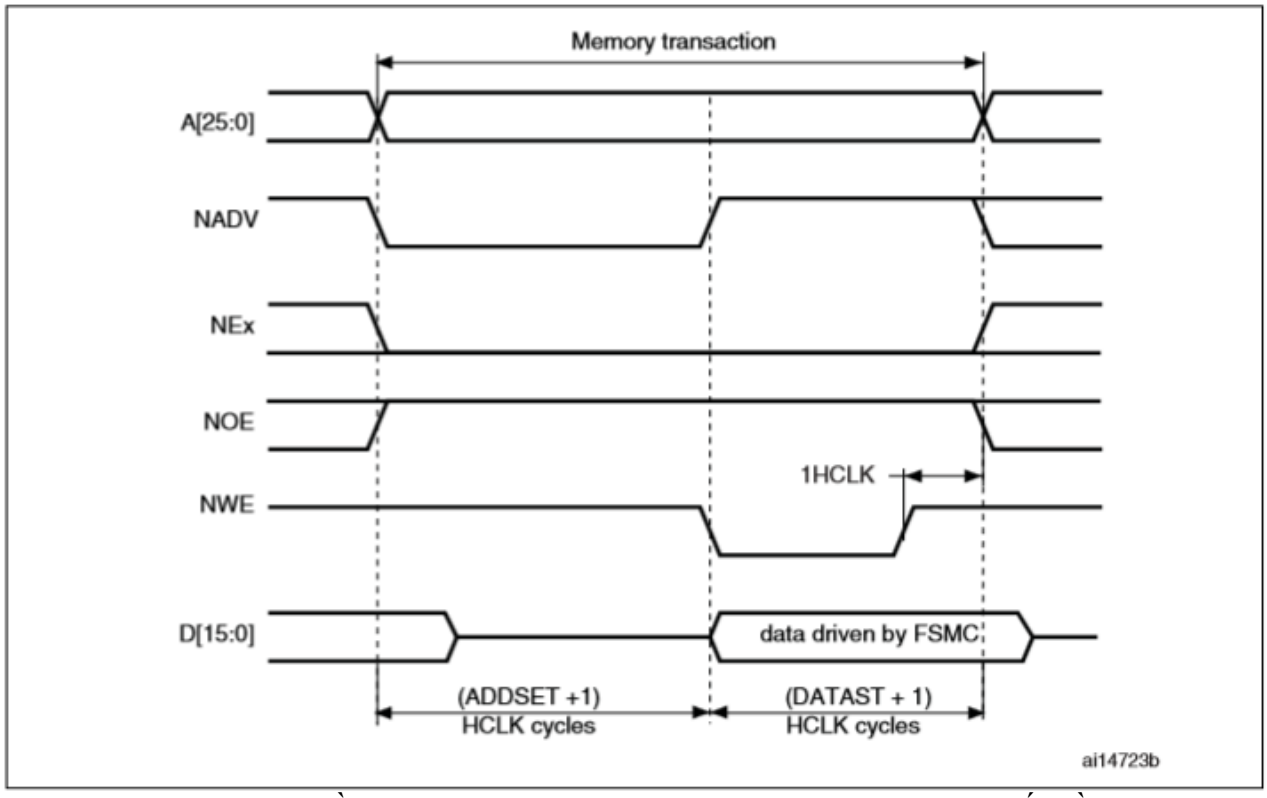
b. Tính toán thời gian trễ

Để tính toán được thời gian trễ ta cần dựa vào **hình 9.4** và **hình 9.5** đồng thời kết hợp với datasheet của ARM và loại bộ nhớ đang được sử dụng để tìm ra các thông số sau:

- Thời gian tối đa cho việc đọc và ghi.
- Những thông số thời gian trễ của FSMC.
- Những thông số thời gian trễ của bộ nhớ.



Hình 9.4 Giản đồ thời gian của quá trình đọc bộ nhớ NOR Flash bất đồng bộ



Hình 9.5 Giản đồ thời gian của quá trình ghi bộ nhớ NOR Flash bất đồng bộ

Dựa vào **hình 9.5** giản đồ thời gian của quá trình ghi ta có các công thức 1 và 2:

- ((ADDSET+1) + (DATAST +1)) x t_{HCLK} = max(t_{WC}) (1)
- DATAST x t_{HCLK} = t_{WP} (2)

Dựa vào **hình 9.4** giản đồ thời gian của quá trình đọc ta có công thức 3:

- DATAST ≥ (t_{AVQV} + $t_{su(Data\_NE)}$ + $t_{V(A\_NE)}$)/t_{HCLK} – ADDSET – 4 (3)

Để tính toán được 2 thông số ADDSET và DATAST ta phải dựa vào datasheet của loại bộ nhớ đang sử dụng ví dụ **bảng 9.8** là thông số về thời gian trễ của 2 loại bộ nhớ M29W128xx70 và S29GL128P90.

Bảng 9.8 Thời gian trễ của 2 loại bộ nhớ M29W128xx70 và S29GL128P90

Ký hiệu	Thông số	Giá trị		Đơn vị
		M29W128xx70	S29GL128P90	
t_{WC}	Khoảng thời gian từ khi địa chỉ sẵn sàng cho đến khi địa chỉ sẵn sàng cho lần ghi tiếp theo.	70	90	ns
t_{RC}	Khoảng thời gian từ khi địa chỉ sẵn sàng cho đến khi địa chỉ sẵn sàng cho lần đọc tiếp theo.	70	90	ns
t_{WP}	Khoảng thời gian chân cho phép ghi tích cực.	45	35	ns
t_{AVQV}	Khoảng thời gian từ khi địa chỉ sẵn sằng đến khi dữ liệu sẵn sàng.	70	90	ns

Bên cạnh đó ta cũng phải dựa vào các thông số cơ bản của dòng ARM đang sử dụng ví dụ đối với dòng STM32F10x khi hoạt động ở tần số 72MHz ta có:

Bảng 9.9 Các thông số cần chú ý của dòng STM32F10x khi lập trình FSMC

Ký hiệu	Giải thích	Giá trị	Đơn vị
HCLK	Tần số xung clock nội AHB	72	MHz
t_{HCLK}	Chu kỳ xung clock nội AHB	13.88	ns
$t_{su(Data\_NE)}+t_{V(A\_NE)}$	Thời gian từ khi có dữ liệu đến khi chân FSMC_NEx lên mức cao + Thời gian từ khi FSMC_NEx xuống mức thấp cho tới khi FSMC_A(địa chỉ sẵn sàng)	$2t_{HCLK}$ + 25	ns

Thế các giá trị về thời gian trễ trong **bảng 9.8 và 9.9** vào các công thức 1, 2, 3 ta được:

Đối với M29W128xx70

- ((ADDSET+1) + (DATAST +1)) x 13.88 = 70
- DATAST x 13.88 = 45

⇨ DATAST = 3.24 và ADDSET = 1

Nhưng DATAST phải thỏa (3)

- DATAST ≥ (t_{AVQV} + $t_{su(Data\_NE)}$ + $t_{V(A\_NE)}$)/t_{HCLK} – ADDSET – 4

⇨ DATAST ≥ (70 + 2 x 13.88 +25)/13.88 – 1 – 4

⇨ DATAST ≥ 3.84 (chọn bằng 4)

Vậy ta chọn DATAST = 4 và ADDSET = 1

Đối với S29GL128P90

⇨ Tính toán tương tự như trên ta chọn được DATAST = 5 và ADDSET = 2

c. Các lệnh thông dụng để cấu hình FSMC giao tiếp NOR Flash

Bảng 9.10 Các lệnh thông dụng để cấu hình NOR Flash / SRAM

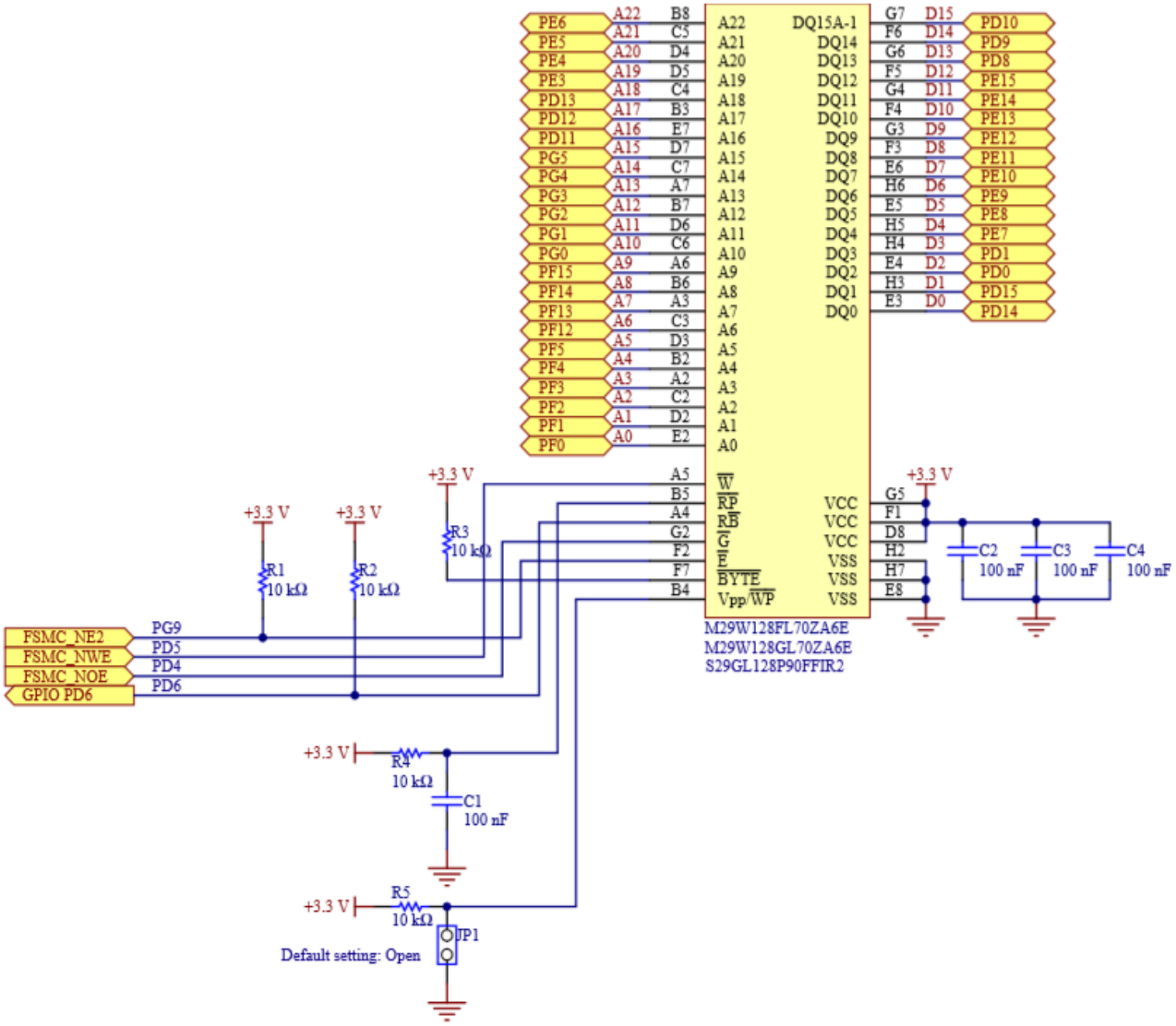
SỬ DỤNG THƯ VIỆN "stm32f10x_fsmc"	
Lệnh	
Thông số hay dùng	**Giải thích**
FSMC_NORSRAMInitTypeDef A; **FSMC_NORSRAMTimingInitTypeDef T;** (Khai báo biến **A** thuộc kiểu FSMC_NORSRAMInitTypeDef để lưu giá trị cấu hình) (Khai báo biến **T** thuộc kiểu FSMC_NORSRAMTimingInitTypeDef để lưu thời gian trễ)	
FSMC_NORSRAMDeInit(B); (Lệnh bỏ các cấu hình của 1 bank thuộc NOR Flash SRAM)	
B: FSMC_Bank1_NORSRAM1 FSMC_Bank1_NORSRAM2 FSMC_Bank1_NORSRAM3 FSMC_Bank1_NORSRAM4	**B:** Bank cần bỏ cấu hình NOR Flash SRAM bank 1 NOR Flash SRAM bank 2 NOR Flash SRAM bank 3 NOR Flash SRAM bank 4
T.FSMC_AddressSetupTime = C; (Lệnh cấu hình thông số thời gian cài đặt địa chỉ ADDSET)	
C: [0-15]	**C:** Số chu kỳ HCLK Phạm vi từ 0 đến 15

T.FSMC_AddressHoldTime = D; (Lệnh cấu hình thông số thời gian giữ địa chỉ ADDHOLD)	
D: [0-15]	**D:** Số chu kỳ HCLK Phạm vi từ 0 đến 15
T.FSMC_DataSetupTime = E; (Lệnh cấu hình thông số thời gian cài đặt giữ liệu DATAST)	
E: [0-255]	**D:** Số chu kỳ HCLK Phạm vi từ 0 đến 255
T.FSMC_BusTurnAroundDuration = F; (Lệnh cấu hình thời gian BusTurnAround- Chỉ sử dụng cho bộ nhớ NOR Flash đa hợp)	
F: [0-15]	**F:** Số chu kỳ HCLK Phạm vi từ 0 đến 15
T.FSMC_CLKDivision = G; (Lệnh cấu hình hệ số chia của bộ chia xung- Không dung cho bộ nhớ bất đồng bộ)	
G: [0-15]	**G:** Hệ số chia Phạm vi từ 0 đến 15
T.FSMC_DataLatency= H; (Lệnh cấu hình độ trễ - Chỉ dung cho NOR FLASH đồng bộ)	
H: [0-15]	**G:** Số chu kỳ xung cấp cho bộ nhớ Phạm vi từ 0 đến 15 **Chú ý:** Đối với các loại bộ nhớ khác NOR FLASH đồng bộ thì nên cấu hình là 0
T.FSMC_AccessMode = I; (Lệnh cấu hình chế độ truy xuất bất đồng bộ)	
I: FSMC_AccessMode_A FSMC_AccessMode_B FSMC_AccessMode_C FSMC_AccessMode_D	**I:** Chế độ truy xuất Chế đô A Chế độ B Chế độ C Chế độ D
A. FSMC_Bank = J; (Lệnh chọn bank NOR Flash SRAM được sử dụng)	
J: FSMC_Bank1_NORSRAM1 FSMC_Bank1_NORSRAM2 FSMC_Bank1_NORSRAM3 FSMC_Bank1_NORSRAM4	**J:** Bank cầu sử dụng NOR Flash SRAM bank 1 NOR Flash SRAM bank 2 NOR Flash SRAM bank 3 NOR Flash SRAM bank 4
A. FSMC_DataAddressMux = K; (Lệnh chọn chế độ đa hợp hoặc không đa hợp)	
K: FSMC_DataAddressMux_Disable FSMC_DataAddressMux_Enable	**K:** Đa hợp hoặc không Không đa hợp Đa hợp
A. FSMC_MemoryDataWidth = L; (Lệnh chọn bề rộng bus dữ liệu)	
L: FSMC_MemoryDataWidth_8b FSMC_MemoryDataWidth_16b	**L:** Bề rộng bus dữ liệu Giao tiếp 8 bit Giao tiếp 16 bit

A. FSMC_MemoryType = M; (Lệnh chọn loại bộ nhớ được sử dụng)	
M: FSMC_MemoryType_SRAM FSMC_MemoryType_PSRAM FSMC_MemoryType_NOR	**M:** Loai bộ nhớ SRAM PSRAM NOR Flash
A. FSMC_BurstAccessMode = N; (Lệnh cho phép hoặc cấm chế độ truy xuất Burst)	
N: FSMC_BurstAccessMode_Disable FSMC_BurstAccessMode_Enable	**N:** Cho phép hoặc cấm Cấm chế độ burst Cho phép chế độ burst
A. FSMC_WaitSignalPolarity = O; (Lệnh cấu hình mức tích cực cho tín hiệu Wait- chỉ dùng trong chế độ burst)	
O: FSMC_WaitSignalPolarity_Low FSMC_WaitSignalPolarity_High	**O:** Mức tích cực Tín hiệu Wait tích cực mức thấp Tín hiêu Wait tích cực mức cao
A. FSMC_WrapMode = P; (Lệnh cho phép hoặc cấm chế độ Wrapped burst- chỉ dùng trong chế độ burst)	
P: FSMC_WrapMode_Disable FSMC_WrapMode_Enable	**P:** Cho phép hoặc cấm Cấm Cho phép
A. FSMC_WaitSignalActive = Q; (Lệnh cấu hình tín hiệu wait - chỉ dùng trong chế độ burst)	
Q: FSMC_WaitSignalActive_BeforeWaitState FSMC_WaitSignalActive_DuringWaitState	**Q:** Thời điểm xuất tín hiệu wait Trước trạng thái chờ Trong suốt trạng thái chờ
A. FSMC_WaitSignal = R; (Lệnh cho phép hoặc cấm tín hiệu Wait- Chỉ dung trong chế độ burst)	
R: FSMC_WaitSignal_Disable FSMC_WaitSignal_Enable	**R:** Cho phép hoặc cấm Cấm Cho phép
A. FSMC_WriteOperation = S; (Lệnh cho phép hoặc cấm ghi)	
S: FSMC_WriteOperation_Disable FSMC_WriteOperation_Enable	**S:** Cho phép hoặc cấm Cấm Cho phép
A. FSMC_ExtendedMode = U; (Lệnh cho phép hoặc cấm chế độ mở rộng)	
U: FSMC_ExtendedMode_Disable FSMC_ExtendedMode_Enable	**U:** Cho phép hoặc cấm Cấm Cho phép
A. FSMC_WriteBurst = V; (Lệnh cho phép hoặc cấm chế độ ghi burst)	
V: FSMC_WriteBurst_Disable FSMC_WriteBurst_Enable	**V:** Cho phép hoặc cấm Cấm Cho phép

A. FSMC_ReadWriteTimingStruct =&T; (Lệnh cấu hình thông số **T** cho chế độ đọc và ghi)	
A.FSMC_WriteTimingStruct = &T; (Lệnh cấu hình thông số **T** cho chế độ ghi)	
FSMC_NORSRAMInit(&A); (Lệnh cấu hình các thông số lưu trong biến A cho FSMC)	
FSMC_NORSRAMCmd(W, X); (Lệnh lệnh cho phép hoặc cấm NOR Flash SRAM bank W)	
W:	**W:** Bank cầu sử dụng
FSMC_Bank1_NORSRAM1	NOR Flash SRAM bank 1
FSMC_Bank1_NORSRAM2	NOR Flash SRAM bank 2
FSMC_Bank1_NORSRAM3	NOR Flash SRAM bank 3
FSMC_Bank1_NORSRAM4	NOR Flash SRAM bank 4
X:	**X:** Cho phép hoặc cấm
DISABLE	Cấm
ENABLE	Cho phép

d. Sơ đồ nguyên lý giao tiếp FSMC với bộ nhớ NOR Flash bất đồng bộ và không đa hợp



Hình 9.6 Sơ đồ nguyên lý giao tiếp FSMC với NOR Flash bất đồng bộ và không đa hợp

e. Chương trình cấu hình giao tiếp FSMC với NOR Flash S29GL128P90

```
void CauhinhNORFlash(void)
{
  FSMC_NORSRAMInitTypeDef  FS;
  FSMC_NORSRAMTimingInitTypeDef  t;
  GPIO_InitTypeDef GPIO;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD|
                         RCC_APB2Periph_GPIOE|
                         RCC_APB2Periph_GPIOF|
                         RCC_APB2Periph_GPIOG, ENABLE);
  /*-- Cấu hình GPIO ----------------------------------------*/
  /*Cấu hình các đường dữ liệu */
  GPIO.GPIO_Pin=GPIO_Pin_0|GPIO_Pin_1| GPIO_Pin_8|GPIO_Pin_9 |
                        GPIO_Pin_10 | GPIO_Pin_14 |GPIO_Pin_15;
  GPIO.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOD, &GPIO);
  GPIO.GPIO_Pin=GPIO_Pin_7|GPIO_Pin_8| GPIO_Pin_9
                | GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_13 |
                 GPIO_Pin_14 | GPIO_Pin_15 | GPIO_Pin_10;
  GPIO_Init(GPIOE, &GPIO);
  /*Cấu hình các đường địa chỉ */
  GPIO.GPIO_Pin=GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2| GPIO_Pin_4 |
                        GPIO_Pin_5 | GPIO_Pin_12 | GPIO_Pin_14 |
                        GPIO_Pin_15 |GPIO_Pin_13 |GPIO_Pin_3;
  GPIO_Init(GPIOF, &GPIO);
  GPIO.GPIO_Pin =GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2
                 |GPIO_Pin_3 | GPIO_Pin_4 | GPIO_Pin_5;
  GPIO_Init(GPIOG, &GPIO);
  GPIO.GPIO_Pin=GPIO_Pin_11|GPIO_Pin_12| GPIO_Pin_13;
  GPIO_Init(GPIOD, &GPIO);
  GPIO.GPIO_Pin = GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5| GPIO_Pin_6;
  GPIO_Init(GPIOE, &GPIO);
  /*Cấu hình chân NOE và NWE */
  GPIO.GPIO_Pin = GPIO_Pin_4 | GPIO_Pin_5;
  GPIO_Init(GPIOD, &GPIO);
  /*Cấu hình chân NE2 */
  GPIO.GPIO_Pin = GPIO_Pin_9;
  GPIO_Init(GPIOG, &GPIO);
  /*Cấu hình chân PD6 nhận tín hiệu báo bận của NOR Flash */
  GPIO.GPIO_Pin = GPIO_Pin_6;
  GPIO.GPIO_Mode = GPIO_Mode_IN_FLOATING;
  GPIO_Init(GPIOD, &GPIO);
  /*-- Cấu hình FSMC ----------------------------------------*/
  t.FSMC_AddressSetupTime = 0x02;
  t.FSMC_AddressHoldTime = 0x00;
  t.FSMC_DataSetupTime = 0x05;
  t.FSMC_BusTurnAroundDuration = 0x00;
  t.FSMC_CLKDivision = 0x00;
  t.FSMC_DataLatency = 0x00;
```

```
  t.FSMC_AccessMode = FSMC_AccessMode_B;
  FS.FSMC_Bank = FSMC_Bank1_NORSRAM2;
  FS.FSMC_DataAddressMux = FSMC_DataAddressMux_Disable;
  FS.FSMC_MemoryType = FSMC_MemoryType_NOR;
  FS.FSMC_MemoryDataWidth = FSMC_MemoryDataWidth_16b;
  FS.FSMC_BurstAccessMode = FSMC_BurstAccessMode_Disable;
  FS.FSMC_AsynchronousWait = FSMC_AsynchronousWait_Disable;
  FS.FSMC_WaitSignalPolarity = FSMC_WaitSignalPolarity_Low;
  FS.FSMC_WrapMode = FSMC_WrapMode_Disable;
  FS.FSMC_WaitSignalActive=
                          FSMC_WaitSignalActive_BeforeWaitState;
  FS.FSMC_WriteOperation = FSMC_WriteOperation_Enable;
  FS.FSMC_WaitSignal = FSMC_WaitSignal_Disable;
  FS.FSMC_ExtendedMode = FSMC_ExtendedMode_Disable;
  FS.FSMC_WriteBurst = FSMC_WriteBurst_Disable;
  FS.FSMC_ReadWriteTimingStruct = &t;
  FS.FSMC_WriteTimingStruct = &t;
  FSMC_NORSRAMInit(&FS);
  /*Cho phép bank 1*/
  FSMC_NORSRAMCmd(FSMC_Bank1_NORSRAM2, ENABLE);
}
```

9.5.2 Chế độ giao tiếp bất đồng bộ và không đa hợp với bộ nhớ SRAM 16 bit

a. Các bước cấu hình cho FSMC

- Bật hoặc **tắt** chế độ đa hợp địa chỉ và dữ liệu.
- Lựa chọn loại bộ nhớ được sử dụng: NOR/**SRAM**/PSRAM
- Cấu hình bề rộng bus dữ liệu : 8 hoặc **16 bit**
- Bật hoặc **tắt** chế độ mở rộng.

b. Tính toán thời gian trễ

Dựa vào giản đồ thời gian **hình 9.7** của quá trình ghi ta có các **công thức 4 và 5**:

- $((ADDSET+1) + (DATAST+1)) \times t_{HCLK} = \max(t_{WC})$ (4)
- $DATAST \times t_{HCLK} = t_{PWE1}$ (5)

Dựa vào giản đồ thời gian **hình 9.8** của quá trình đọc ta có **công thức 6**:

- $DATAST \geq (t_{AA} + t_{su(Data\_NE)} + t_{V(A\_NE)})/t_{HCLK} - ADDSET - 4$ (6)

Để tính toán được 2 thông số ADDSET và DATAST ta phải dựa vào datasheet của loại bộ nhớ đang sử dụng ví dụ **bảng 9.11** là thông số về thời gian trễ SRAM **IS6WV51216BL.**

Bảng 9.11 Thời gian trễ của IS6WV51216BL

Ký hiệu	Giải thích	Giá trị	Đơn vị
t_{WC}	Thời gian của chu kỳ ghi(Write cycle time)	12	ns
t_{RC}	Thời gian của chu kỳ đọc(Read cycle time)	12	ns
T_{PWE1}	Độ rộng xung cho phép ghi (Write Enable Low to Write Enable High)	8	ns
t_{AA}	Thời gian truy xuất địa chỉ.(Address access time)	12	ns

- $((ADDSET+1) + (DATAST+1)) \times 13.88 = 12$
- $DATAST \times 13.88 = 8$

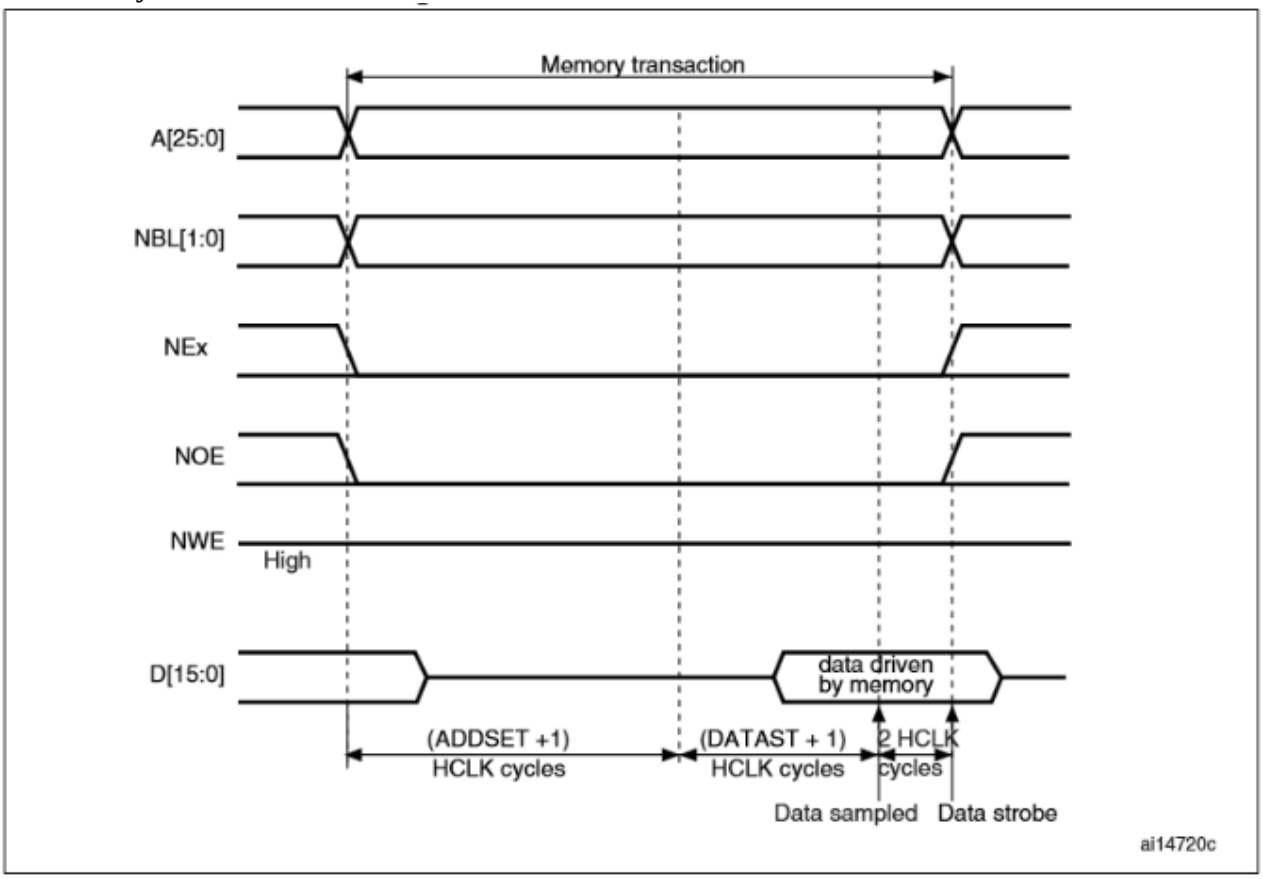
⇨ DATAST = 1 và ADDSET = 0

Nhưng DATAST phải thỏa (6)

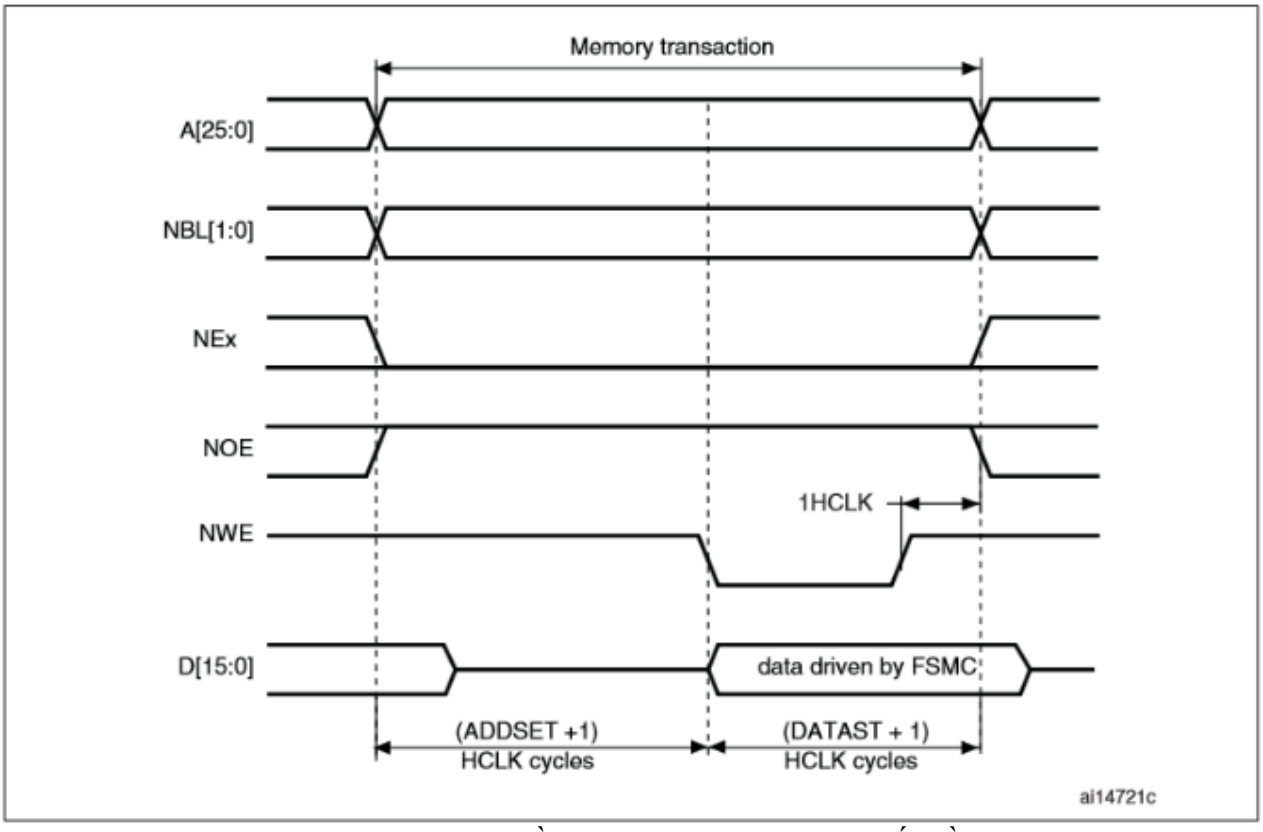
- DATAST ≥ (t_{AA} + $t_{su(Data\_NE)}$ + $t_{V(A\_NE)}$)/t_{HCLK} – ADDSET – 4

⇨ DATAST ≥ -0.334 (chọn bằng 1 vì DATAST còn phải thỏa (4) và (5))

Vậy : DATAST = 1 và ADDSET =0.

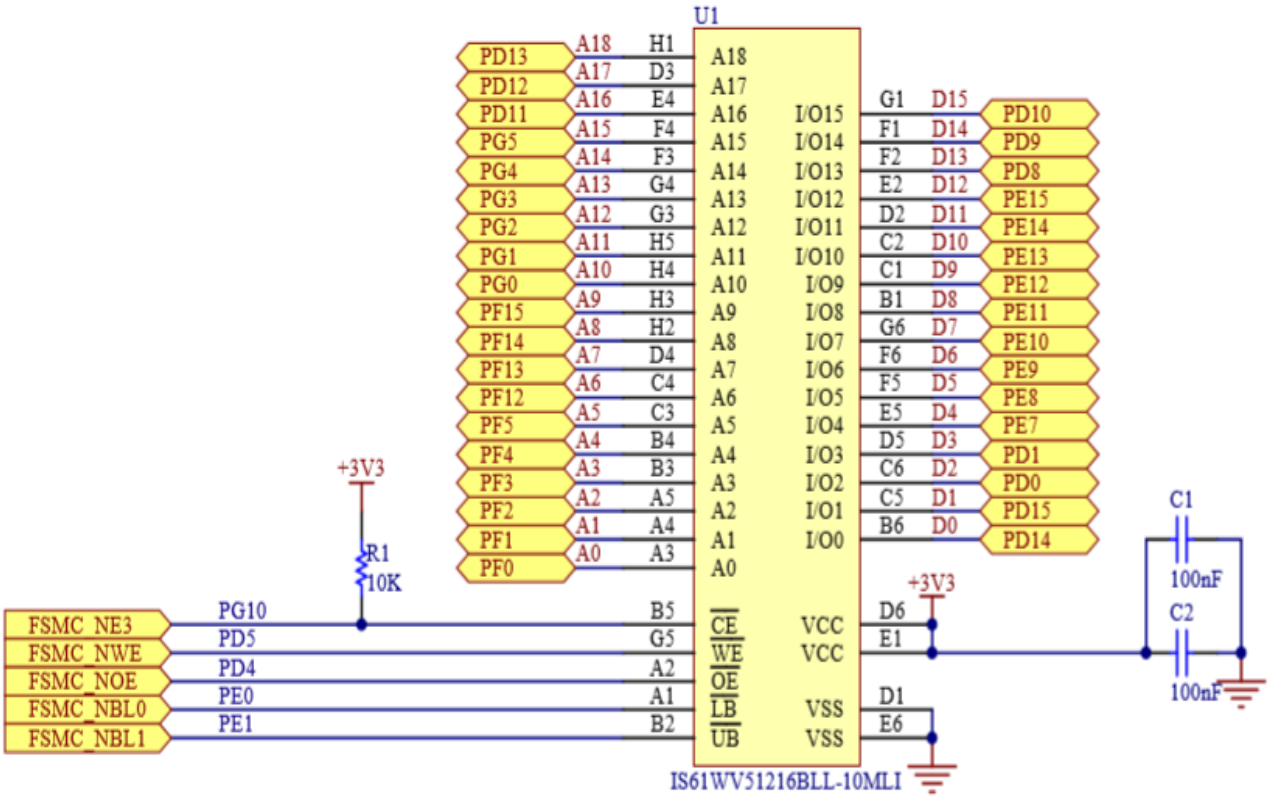


Hình 9.7 Giản đồ thời gian quá trình đọc bất đồng bộ SRAM



Hình 9.8 Giản đồ thời gian quá trình ghi bất đồng bộ SRAM

c. **Các lệnh thông dụng dùng để cấu hình SRAM (xem bảng 9.10)**
d. **Sơ đồ nguyên lý giao tiếp FSMC với bộ nhớ SRAM bất đồng bộ và không đa hợp**



Hình 9.9 Sơ đồ nguyên lý giao tiếp FSMC với SRAM bất đồng bộ và không đa hợp.

e. **Chương trình cấu hình giao tiếp FSMC với SRAM IS6WV51216BL**

```
void CauhinhSRAM(void)
{
  FSMC_NORSRAMInitTypeDef  FS;
  FSMC_NORSRAMTimingInitTypeDef  t;
  GPIO_InitTypeDef GPIO;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD|
                         RCC_APB2Periph_GPIOG|
                         RCC_APB2Periph_GPIOE|
                         RCC_APB2Periph_GPIOF, ENABLE);
/*-- Cấu hình GPIO ----------------------------------------*/
  /*Cấu hình các đường dữ liệu*/
  GPIO.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_8 |
           GPIO_Pin_10 | GPIO_Pin_14 | GPIO_Pin_15| GPIO_Pin_9;
  GPIO.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOD, &GPIO);
  GPIO.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);
  /*Cấu hình các đường địa chỉ*/
  GPIO.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 |GPIO_Pin_2 |
                  GPIO_Pin_3 |GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_12|
```

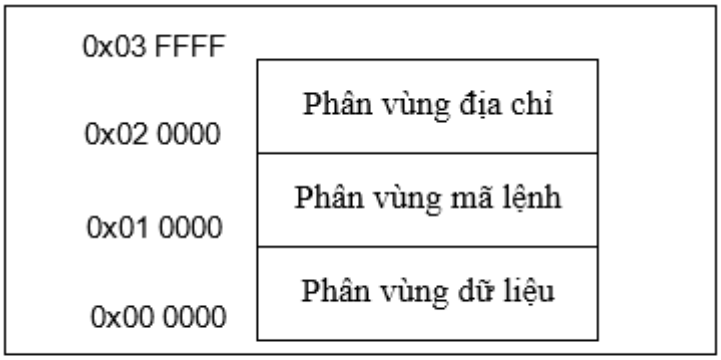
```
                    GPIO_Pin_13| GPIO_Pin_14 | GPIO_Pin_15;
  GPIO_Init(GPIOF, &GPIO);
  GPIO.GPIO_Pin =GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 |
                    GPIO_Pin_3 | GPIO_Pin_4 | GPIO_Pin_5;
  GPIO_Init(GPIOG, &GPIO);
  GPIO.GPIO_Pin = GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_13;
  GPIO_Init(GPIOD, &GPIO);
  /*Cấu hình chân NOE và NWE */
  GPIO.GPIO_Pin = GPIO_Pin_4 |GPIO_Pin_5;
  GPIO_Init(GPIOD, &GPIO);
  /* Cấu hình chân NE3 */
  GPIO.GPIO_Pin = GPIO_Pin_10;
  GPIO_Init(GPIOG, &GPIO);
  /*Cấu hình chân  NBL0, NBL1 */
  GPIO.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1;
  GPIO_Init(GPIOE, &GPIO);
/*-- Cấu hình FSMC ----------------------------------------*/
  t.FSMC_AddressSetupTime = 0;
  t.FSMC_AddressHoldTime = 0;
  t.FSMC_DataSetupTime = 1;
  t.FSMC_BusTurnAroundDuration = 0;
  t.FSMC_CLKDivision = 0;
  t.FSMC_DataLatency = 0;
  t.FSMC_AccessMode = FSMC_AccessMode_A;
  FS.FSMC_Bank = FSMC_Bank1_NORSRAM3;
  FS.FSMC_DataAddressMux = FSMC_DataAddressMux_Disable;
  FS.FSMC_MemoryType = FSMC_MemoryType_SRAM;
  FS.FSMC_MemoryDataWidth = FSMC_MemoryDataWidth_16b;
  FS.FSMC_BurstAccessMode = FSMC_BurstAccessMode_Disable;
  FS.FSMC_AsynchronousWait = FSMC_AsynchronousWait_Disable;
  FS.FSMC_WaitSignalPolarity = FSMC_WaitSignalPolarity_Low;
  FS.FSMC_WrapMode = FSMC_WrapMode_Disable;
  FS.FSMC_WaitSignalActive =
                          FSMC_WaitSignalActive_BeforeWaitState;
  FS.FSMC_WriteOperation = FSMC_WriteOperation_Enable;
  FS.FSMC_WaitSignal = FSMC_WaitSignal_Disable;
  FS.FSMC_ExtendedMode = FSMC_ExtendedMode_Disable;
  FS.FSMC_WriteBurst = FSMC_WriteBurst_Disable;
  FS.FSMC_ReadWriteTimingStruct = &t;
  FS.FSMC_WriteTimingStruct = &t;
  FSMC_NORSRAMInit(&FS);
  /*Cho phép bank */
  FSMC_NORSRAMCmd(FSMC_Bank1_NORSRAM3, ENABLE);
}
```

9.6 CÁC CHẾ ĐỘ GIAO TIẾP CƠ BẢN CỦA FSMC VỚI NAND FLASH

9.6.1 Giao tiếp với NAND Flash 8 bit

a. Đặc điểm của NAND Flash

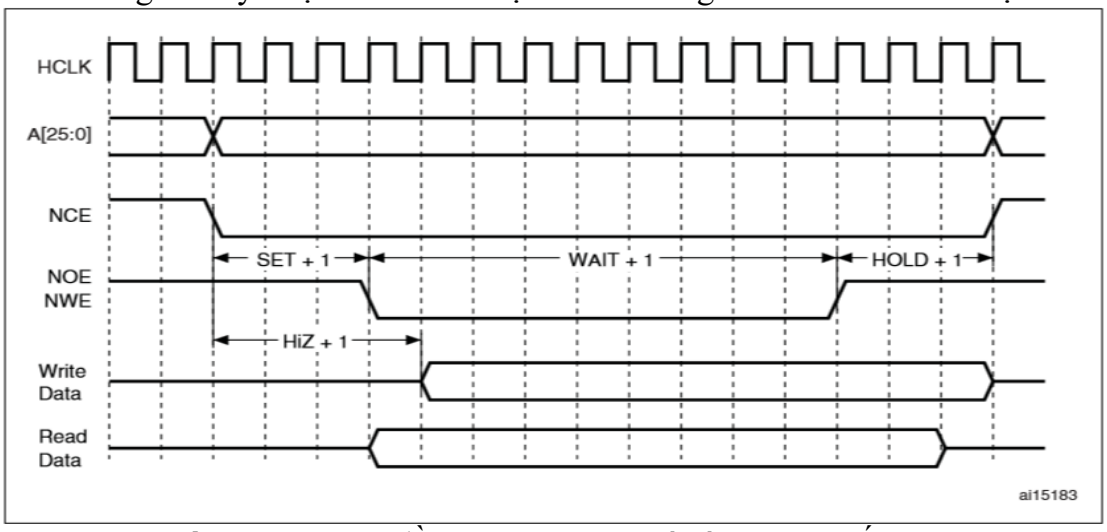
- Để đọc hoặc ghi bộ nhớ NAND Flash ta cần phải:
  - Gởi một lệnh tới NAND Flash
  - Gởi địa chỉ cần đọc hoặc ghi
  - Đọc hoặc ghi dữ liệu
- FSMC NAND bank được chia thành 3 phân vùng cho phép người đùng có thể dễ dàng lập trình: phân vùng dữ liệu, phân vùng địa chỉ và phân vùng mã lệnh.



Hình 9.10 Các phân vùng trên bank NAND FSMC

b. Các bước để cấu hình cho FSMC

- Cho phép hoặc **cấm** tín hiệu báo bận(Ready/Busy) hoạt động như là ngõ vào của FSMC.
- **Cho phép** hoặc cấm tín hiệu báo bận làm nguồn ngắt cho FSMC. Ngắt có thể được tạo ra bởi các trường hợp sau:
  - Có cạnh lên của tín hiệu Ready/Busy: bộ nhớ vừa hoàn thành tác vụ trước đó và đang sẵn sàng cho tác vụ tiếp theo.
  - Có cạnh xuống của tín hiệu Ready/Busy: bắt đầu thực hiện tác vụ mới.
  - Khi tín hiệu Ready/Busy ở mức cao: bộ nhớ đã sẵn sàng thực hiện tác vụ mới.
- Lựa chọn bề rộng của bus dữ liệu **8** hoặc 16 bit
- **Cho phép** hoặc cấm kiểm tra lỗi ECC
- Chọn kích thước của một trang ECC: có thể là 256, **512**, 1024, 2048 bytes
  FSMC cho phép người dùng lập trình thời gian trễ khác nhau trên những phân vùng khác nhau như common và attribute. Thời gian trễ bao gồm:
- **Setup time:** là thời gian tính theo đơn vị HCLK cần để thiết lập địa chỉ trước khi xác nhận lệnh. Đây là thời gian từ khi địa chỉ đã sẵn sàng có đến khi sảy ra hoạt động đọc hoặc ghi.
- **Wait time:** là thời gian tính theo đơn vị HCLK cần thiết để xác nhận lệnh. Thời gian này được tính từ khi tín hiệu NOE và NWE tích cực cho đến khi hết tích cực.
- **Hold time:** là thời gian tính theo đơn vị HCLK từ khi tín hiệu NOE và NWE hết tích cực cho đến hết chu kỳ.
- **Databus HIZ time:** là thời gian tính theo đơn vị HCLK mà bus dữ liệu ở trạng thái tổng trở cao. Thời gian này được tính từ khi địa chỉ sẵn sàng cho đến khi có dữ liệu.



Hình 9.11 Giản đồ thời gian của quá trình truy xuất NAND

c. Tính toán thời gian trễ

Từ **hình 9.11** ta có các công thức sau:

- ((SET+1) + (WAIT+1) + (HOLD+1)) x t_{HCLK} ≥ Write/Read access (7)
- Write/Read Enable Low to Write/Read Enable High = (WAIT+1) x t_{HCLK} (8)
- (HIZ +1) x t_{HCLK} ≥ Chip select setup time to Data setup. (9)
- (HOLD+1)x t_{HCLK} = Write Enable High to Chip Enable High (10)
- (WAIT+1) x t_{HCLK} ≥ (t_{REA} + $t_{su(D-NOE)}$) (11)

Ví dụ để tính toán thời gian trễ cho chip NAND512W3A2C ta phải tra trong datasheet bảng thông số thời gian trễ sau:

Bảng 9.12 Các thông số thời gian trễ của NAND512W3A2C cần để cấu hình FSMC

Ký hiệu	Giải thích	Giá trị	Đơn vị
t_{WC}	Thời gian của chu kỳ ghi (Write cycle time)	30	ns
t_{RC}	Thời gian của chu kỳ đọc (Read cycle time)	30	ns
t_{WP}	Độ rộng xung cho phép ghi (Write Enable Low to Write Enable High)	15	ns
t_{RP}	Độ rộng xung cho phép đọc (Read Enable Low to Read Enable High)	15	ns
t_{CH}	Thời gian từ khi tín hiệu cho phép ghi lên mức cao đến khi tín hiệu cho phép chip lên mức cao. (Write Enable High to Chip Enable High)	5	ns
t_{ALH}	Thời gian từ khi tín hiệu cho phép ghi lên mức cao đến khi tín hiệu cho phép chốt địa chỉ thay đổi. (Write Enable High to Address Latch Low/High)	5	ns
t_{CLH}	Thời gian từ khi tín hiệu cho phép ghi lên mức cao đến khi tín hiệu cho phép chốt lệnh thay đổi. (Write Enable High to Command Latch Low/High)	5	ns
t_{DS}	Thời gian từ khi có dữ liệu đến khi tín hiệu cho phép ghi lên mức cao (Data Vail to Write Enable High)	15	ns
t_{CS}	Thời gian từ khi tín hiệu cho phép chip xuống mức thấp đến khi tín hiệu cho phép ghi lên mức cao (Chip Enable Low to Write Enable High)	20	ns
t_{REA}	Thời gian từ khi tín hiệu cho phép đọc xuống mức thấp đến khi ngõ ra có dữ liệu. (Read Enable Low to Output Valid)	18	ns

Dựa vào **bảng 9.12** kết hợp với các công thức (7), (8), (9), (10) ta có:

- ((SET+1) + (WAIT+1) + (HOLD+1)) x t_{HCLK} ≥ max(t_{WC},t_{RC})
- (WAIT+1) x t_{HCLK} >= max(t_{WP},t_{RP})
- (HIZ +1) x t_{HCLK} ≥ t_{CS} - t_{DS}
- (HOLD+1)x t_{HCLK} ≥ max(t_{CH} , t_{CLH}, t_{ALH})

⇨ HOLD = 0, HIZ= 0, SET =0, WAIT = 0

Mặt khác WAIT phải thỏa công thức (11) với $t_{su(D-NOE)}$ =25 :

⇨ (WAIT+1) ≥ (18+25)/13.88 → WAIT ≥ 2.09 (Chọn bằng 3)

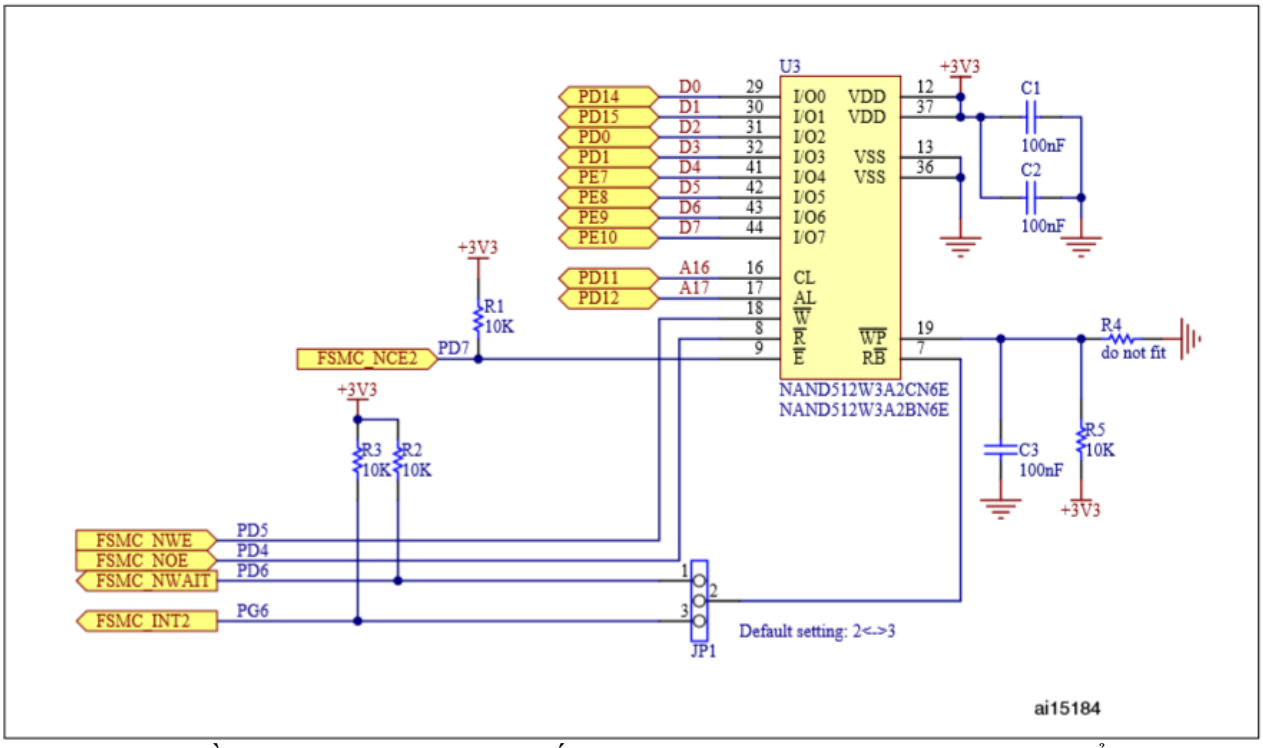
d. Các lệnh thông dụng để cấu hình NAND Flash

Bảng 9.13 Các lệnh thông dụng để cấu hình NAND Flash

SỬ DỤNG THƯ VIỆN "stm32f10x_fsmc"	
Lệnh	
Thông số hay dùng	**Giải thích**
FSMC_NANDInitTypeDef A; **FSMC_NAND_PCCARDTimingInitTypeDef T;** (Khai báo biến **A** thuộc kiểu FSMC_NANDInitTypeDef để lưu giá trị cấu hình) (Biến **T** thuộc kiểu FSMC_NAND_PCCARDTimingInitTypeDef để lưu thời gian trễ)	
NANDDeInit(B); (Lệnh bỏ các cấu hình của 1 bank thuộc NAND Flash)	
B: FSMC_Bank2_NAND FSMC_Bank3_NAND	**B:** Bank cần bỏ cấu hình NAND bank 2 NAND bank 3
T. FSMC_SetupTime= C; (Lệnh cấu hình thông số thời gian cài đặt địa chỉ ADDSET)	
C: [0-255]	**C:** Số chu kỳ HCLK Phạm vi từ 0 đến 255
T. FSMC_WaitSetupTime= D; (Lệnh cấu hình thông số thời gian chờ WAIT)	
D: [0-255]	**D:** Số chu kỳ HCLK Phạm vi từ 0 đến 255
T. FSMC_HoldSetupTime= E; (Lệnh cấu hình thông số thời gian giữ HOLD)	
E: [0-255]	**D:** Số chu kỳ HCLK Phạm vi từ 0 đến 255
T. FSMC_HiZSetupTime = F; (Lệnh cấu hình thông số thời gian tổng trở cao HIZ)	
F: [0-255]	**F:** Số chu kỳ HCLK Phạm vi từ 0 đến 255
A. FSMC_Bank= G; (Lệnh cấu hình bank NAND Flash được sử dụng)	
G: FSMC_Bank2_NAND FSMC_Bank3_NAND	**G:** Bank được sử dụng NAND bank 2 NAND bank 3
A. FSMC_Waitfeature= H; (Lệnh cho phép hoặc tắt Wait)	
H: FSMC_Waitfeature_Disable FSMC_Waitfeature_Enable	**H:** Cho phép hoặc tắt Tắt Cho phép
A. FSMC_MemoryDataWidth= I; (Lệnh cấu hình bề rộng bus dữ liệu)	
I: FSMC_MemoryDataWidth_8b FSMC_MemoryDataWidth_16b	**I:** bề rộng bus dữ liệu Giao tiếp 8 bit Giao tiếp 16 bit

A. FSMC_ECC= J; (Lệnh cho phép hoặc bắt bộ kiểm tra lỗi)	
J: FSMC_ECC_Disable FSMC_ECC_Enable	**J:** Cho phép hoặc tắt Tắt Cho phép
A. FSMC_ECCPageSize= K; (Lệnh cài đặt kích thước page EEC)	
K: FSMC_ECCPageSize_256Bytes FSMC_ECCPageSize_512Bytes FSMC_ECCPageSize_1024Bytes FSMC_ECCPageSize_2048Bytes FSMC_ECCPageSize_4096Bytes FSMC_ECCPageSize_8192Bytes	**K:** Kích thước page EEC 256 Byte 512 Byte 1024 Byte 2048 Byte 4096 Byte 8192 Byte
A. FSMC_TCLRSetupTime= L; (Lệnh cài đặt thời gian trễ từ khi CLE xuống mức thấp đến khi RE xuống mức thấp)	
L: [0-255]	**L:** Số chu kỳ HCLK Phạm vi từ 0 đến 255
A. FSMC_TARSetupTime= M; (Lệnh cài đặt thời gian trễ từ khi ALE xuống mức thấp đến khi RE xuống mức thấp)	
M: [0-255]	**M:** Số chu kỳ HCLK Phạm vi từ 0 đến 255
A. FSMC_CommonSpaceTimingStruct = &T; (Lệnh cài đặt thời gian trễ được lưu trong biến T cho vùng Common)	
A.FSMC_AttributeSpaceTimingStruct = &T; (Lệnh cài đặt thời gian trễ được lưu trong biến T cho vùng Attribute)	
FSMC_NANDInit(&A); (Lệnh cài đặt các thông số lưu trong biến A cho FSMC NAND)	
FSMC_NANDCmd(B, N); (Lệnh cho phép hoặc cấm bank NAND)	
B: FSMC_Bank2_NAND FSMC_Bank3_NAND **N:** DISABLE ENABLE	**B:** Bank cần bỏ cấu hình NAND bank 2 NAND bank 3 **N:** Cho phép hoặc cấm Cấm Cho phép

e. Sơ đồ nguyên lý giao tiếp FSMC với NAND Flash 8 bit



Hình 9.12 Sơ đồ nguyên lý mạch giao tiếp bộ nhớ NAND512W3A2C theo chuẩn FSMC

d. Chương trình cấu hình giao tiếp FSMC với NAND512W3A2C

```
void NAND_Init(void)
{
  GPIO_InitTypeDef GPIO;
  FSMC_NANDInitTypeDef FS;
  FSMC_NAND_PCCARDTimingInitTypeDef  t;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD |
                         RCC_APB2Periph_GPIOE |
                         RCC_APB2Periph_GPIOF |
                         RCC_APB2Periph_GPIOG, ENABLE);
  /*-- Cấu hình GPIO ----------------------------------------*/
  /*Cấu hình các chân CLE, ALE, D0->D3, NOE, NWE và NCE2  */
  GPIO.GPIO_Pin =  GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_14 |
                   GPIO_Pin_15 | GPIO_Pin_0 | GPIO_Pin_1 |
                   GPIO_Pin_4 | GPIO_Pin_5 |GPIO_Pin_7;
  GPIO.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_Init(GPIOD, &GPIO);
  /*Cấu hình các chân  D4->D7 */
  GPIO.GPIO_Pin =GPIO_Pin_7|GPIO_Pin_8|GPIO_Pin_9| GPIO_Pin_10;
  GPIO_Init(GPIOE, &GPIO);
  /* Cấu hình chân  NWAIT */
  GPIO.GPIO_Pin = GPIO_Pin_6;
  GPIO.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO.GPIO_Mode = GPIO_Mode_IPU;
  GPIO_Init(GPIOD, &GPIO);
  /*Cấu hình chân INT2 */
```

```
  GPIO.GPIO_Pin = GPIO_Pin_6;
  GPIO_Init(GPIOG, &GPIO);
  /*-- Cấu hình FSMC ----------------------------------------*/
  t.FSMC_SetupTime = 0x0;
  t.FSMC_WaitSetupTime = 0x3;
  t.FSMC_HoldSetupTime = 0x0;
  t.FSMC_HiZSetupTime = 0x0;
  FS.FSMC_Bank = FSMC_Bank2_NAND;
  FS.FSMC_Waitfeature = FSMC_Waitfeature_Enable;
  FS.FSMC_MemoryDataWidth = FSMC_MemoryDataWidth_8b;
  FS.FSMC_ECC = FSMC_ECC_Enable;
  FS.FSMC_ECCPageSize = FSMC_ECCPageSize_512Bytes;
  FS.FSMC_TCLRSetupTime = 0x00;
  FS.FSMC_TARSetupTime = 0x00;
  FS.FSMC_CommonSpaceTimingStruct = &t;
  FS.FSMC_AttributeSpaceTimingStruct = &t;
  FSMC_NANDInit(&FS);
  /*Cho phép bank2_NAND*/
  FSMC_NANDCmd(FSMC_Bank2_NAND, ENABLE);
}
```