

37 Flexible memory controller (FMC)

The Flexible memory controller (FMC) includes three memory controllers:

- The NOR/PSRAM memory controller
- The NAND/PC Card memory controller
- The Synchronous DRAM (SDRAM/Mobile LPDDR SDRAM) controller

This section applies to STM32F42xxx and STM32F43xxx only.

37.1 FMC main features

The FMC functional block makes the interface with synchronous and asynchronous static memories, SDRAM memories, and 16-bit PC memory cards. Its main purposes are:

- to translate AHB transactions into the appropriate external device protocol
- to meet the access time requirements of the external memory devices

All external memories share the addresses, data and control signals with the controller. Each external device is accessed by means of a unique Chip Select. The FMC performs only one access at a time to an external device.

The main features of the FMC controller are the following:

- Interface with static-memory mapped devices including:
 - Static random access memory (SRAM)
 - NOR Flash memory/OneNAND Flash memory
 - PSRAM (4 memory banks)
 - 16-bit PC Card compatible devices
 - Two banks of NAND Flash memory with ECC hardware to check up to 8 Kbytes of data
- Interface with synchronous DRAM (SDRAM/Mobile LPDDR SDRAM) memories
- Burst mode support for faster access to synchronous devices such as NOR Flash memory, PSRAM and SDRAM)
- Programmable continuous clock output for asynchronous and synchronous accesses
- 8-, 16- or 32-bit wide data bus
- Independent Chip Select control for each memory bank
- Independent configuration for each memory bank
- Write enable and byte lane select outputs for use with PSRAM, SRAM and SDRAM devices
- External asynchronous wait control
- Write Data FIFO with 16 x 32-bit depth
- Write Address FIFO with 16 x 30-bit depth
- Cacheable Read FIFO with 6 x 32-bit depth (6 x 14-bit address tag) for SDRAM controller.

The FMC embeds two Write FIFOs: a Write Data FIFO with a 16x33-bit depth and a Write Address FIFO with a 16x30-bit depth.

- The Write Data FIFO stores the AHB data to be written to the memory (up to 32 bits) plus one bit for the AHB transfer (burst or not sequential mode)
- The Write Address FIFO stores the AHB address (up to 28 bits) plus the AHB data size (up to 2 bits). When operating in burst mode, only the start address is stored except when crossing a page boundary (for PSRAM and SDRAM). In this case, the AHB burst is broken into two FIFO entries.

At startup the FMC pins must be configured by the user application. The FMC I/O pins which are not used by the application can be used for other purposes.

The FMC registers that define the external device type and associated characteristics are usually set at boot time and do not change until the next reset or power-up. However, the settings can be changed at any time.

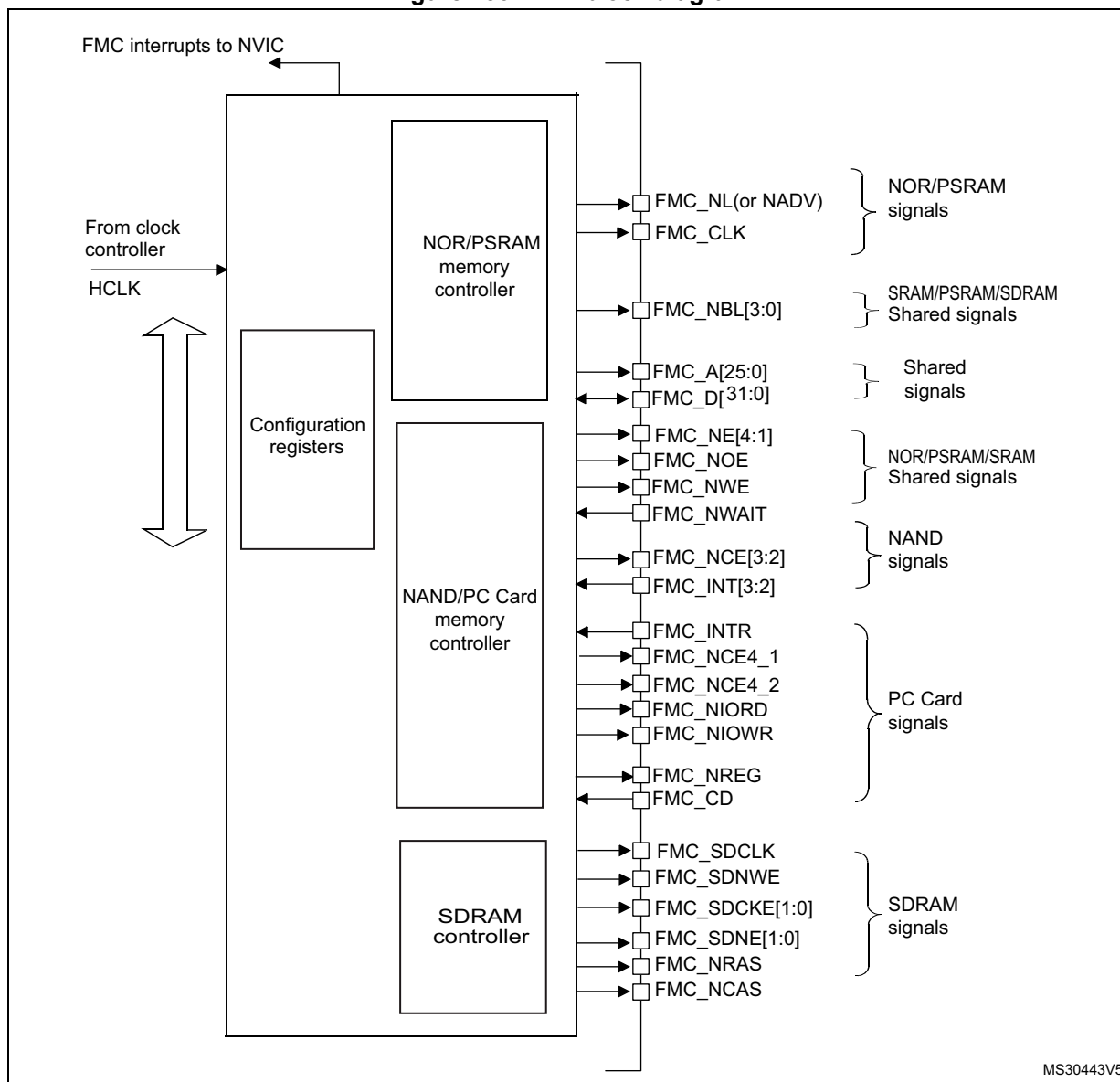
37.2 Block diagram

The FMC consists of five main blocks:

- The AHB interface (including the FMC configuration registers)
- The NOR Flash/PSRAM/SRAM controller
- The NAND Flash/PC Card controller
- The SDRAM controller
- The external device interface

The block diagram is shown in [Figure 456](#).

Figure 456. FMC block diagram



37.3 AHB interface

The AHB slave interface allows internal CPUs and other bus master peripherals to access the external memories.

AHB transactions are translated into the external device protocol. In particular, if the selected external memory is 16- or 8-bit wide, 32-bit wide transactions on the AHB are split into consecutive 16- or 8-bit accesses. The FMC Chip Select (FMC_NEx) does not toggle between consecutive accesses except when performing accesses in mode D with the extended mode enabled.

The FMC generates an AHB error in the following conditions:

- When reading or writing to an FMC bank (Bank 1 to 4) which is not enabled.
- When reading or writing to the NOR Flash bank while the FACCEN bit is reset in the FMC_BCRx register.
- When reading or writing to the PC Card banks while the FMC_CD input pin (Card Presence Detection) is low.
- When writing to a write protected SDRAM bank (WP bit set in the SDRAM_SDCRx register).
- When the SDRAM address range is violated (access to reserved address range)

The effect of an AHB error depends on the AHB master which has attempted the R/W access:

- If the access has been attempted by the Cortex®-M4 with FPU CPU, a hard fault interrupt is generated.
- If the access has been performed by a DMA controller, a DMA transfer error is generated and the corresponding DMA channel is automatically disabled.

The AHB clock (HCLK) is the reference clock for the FMC.

37.3.1 Supported memories and transactions

General transaction rules

The requested AHB transaction data size can be 8-, 16- or 32-bit wide whereas the accessed external device has a fixed data width. This may lead to inconsistent transfers.

Therefore, some simple transaction rules must be followed:

- AHB transaction size and memory data size are equal
There is no issue in this case.
- AHB transaction size is greater than the memory size:
In this case, the FMC splits the AHB transaction into smaller consecutive memory accesses to meet the external data width. The FMC Chip Select (FMC_NEx) does not toggle between the consecutive accesses.
- AHB transaction size is smaller than the memory size:
The transfer may or not be consistent depending on the type of external device:
 - Accesses to devices that have the byte select feature (SRAM, ROM, PSRAM, SDRAM)
In this case, the FMC allows read/write transactions and accesses the right data through its byte lanes BL[3:0].
byte to be written are addressed by NBL[3:0].
All memory byte are read (NBL[3:0] are driven low during read transaction) and the useless ones are discarded.
 - Accesses to devices that do not have the byte select feature (16-bit NOR and NAND Flash memories)
This situation occurs when a byte access is requested to a 16-bit wide Flash memory. Since the device cannot be accessed in byte mode (only 16-bit words can be read/written from/to the Flash memory), Write transactions and Read transactions are allowed (the controller reads the entire 16-bit memory word and uses only the required byte).

Configuration registers

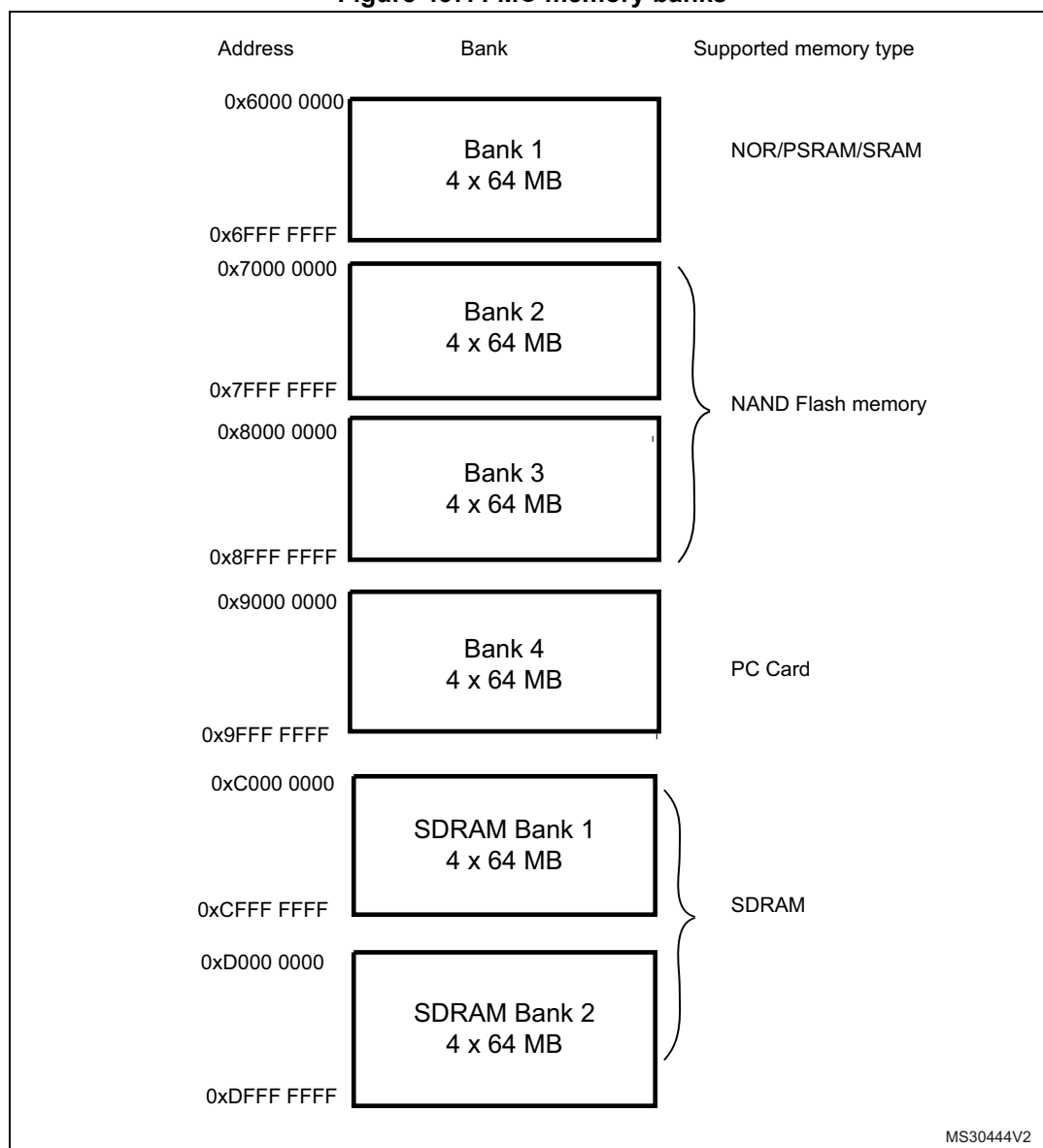
The FMC can be configured through a set of registers. Refer to [Section 37.5.6](#), for a detailed description of the NOR Flash/PSRAM controller registers. Refer to [Section 37.6.8](#), for a detailed description of the NAND Flash/PC Card registers and to [Section 37.7.5](#) for a detailed description of the SDRAM controller registers.

37.4 External device address mapping

From the FMC point of view, the external memory is divided into 6 fixed-size banks of 256 Mbyte each (see [Figure 457](#)):

- Bank 1 used to address up to 4 NOR Flash memory or PSRAM devices. This bank is split into 4 NOR/PSRAM subbanks with 4 dedicated Chip Selects, as follows:
 - Bank 1 - NOR/PSRAM 1
 - Bank 1 - NOR/PSRAM 2
 - Bank 1 - NOR/PSRAM 3
 - Bank 1 - NOR/PSRAM 4
- Banks 2 and 3 used to address NAND Flash memory devices (1 device per bank)
- Bank 4 used to address a PC Card
- Bank 5 and 6 used to address SDRAM devices (1 device per bank).

For each bank the type of memory to be used can be configured by the user application through the Configuration register.

Figure 457. FMC memory banks

37.4.1 NOR/PSRAM address mapping

HADDR[27:26] bits are used to select one of the four memory banks as shown in [Table 254](#).

Table 254. 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

1. HADDR are internal AHB address lines that are translated to external memory.

The HADDR[25:0] bits contain the external memory address. Since HADDR is a byte address whereas the memory is addressed at word level, the address actually issued to the memory varies according to the memory data width, as shown in the following table.

Table 255. NOR/PSRAM External memory address

Memory width ⁽¹⁾	Data address issued to the memory	Maximum memory capacity (bits)
8-bit	HADDR[25:0]	64 Mbyte x 8 = 512 Mbit
16-bit	HADDR[25:1] >> 1	64 Mbyte/2 x 16 = 512 Mbit
32-bit	HADDR[25:2] >> 2	64 Mbyte/4 x 32 = 512 Mbit

1. In case of a 16-bit external memory width, the FMC will internally use HADDR[25:1] to generate the address for external memory FMC_A[24:0]. In case of a 32-bit memory width, the FMC will internally use HADDR[25:2] to generate the external address.
Whatever the external memory width, FMC_A[0] should be connected to external memory address A[0].

Wrap support for NOR Flash/PSRAM

Wrap burst mode for synchronous memories is not supported. The memories must be configured in linear burst mode of undefined length.

37.4.2 NAND Flash memory/PC Card address mapping

In this case, three banks are available, each of them being divided into memory areas as indicated in [Table 256](#).

Table 256. NAND/PC Card memory mapping and timing registers

Start address	End address	FMC bank	Memory space	Timing register
0x9C00 0000	0x9FFF FFFF	Bank 4 - PC card	I/O	FMC_PIO4 (0xB0)
0x9800 0000	0x9BFF FFFF		Attribute	FMC_PATT4 (0xAC)
0x9000 0000	0x93FF FFFF		Common	FMC_PMEM4 (0xA8)
0x8800 0000	0x8BFF FFFF	Bank 3 - NAND Flash	Attribute	FMC_PATT3 (0x8C)
0x8000 0000	0x83FF FFFF		Common	FMC_PMEM3 (0x88)
0x7800 0000	0x7BFF FFFF	Bank 2- NAND Flash	Attribute	FMC_PATT2 (0x6C)
0x7000 0000	0x73FF FFFF		Common	FMC_PMEM2 (0x68)

For NAND Flash memory, the common and attribute memory spaces are subdivided into three sections (see in [Table 257](#) below) located in the lower 256 Kbytes:

- Data section (first 64 Kbytes in the common/attribute memory space)
- Command section (second 64 Kbytes in the common / attribute memory space)
- Address section (next 128 Kbytes in the common / attribute memory space)

Table 257. NAND bank selection

Section name	HADDR[17:16]	Address range
Address section	1X	0x020000-0x03FFFF
Command section	01	0x010000-0x01FFFF
Data section	00	0x000000-0x0FFFFF

The application software uses the 3 sections to access the NAND Flash memory:

- **To send a command to NAND Flash memory**, the software must write the command value to any memory location in the command section.
- **To specify the NAND Flash address that must be read or written**, the software must write the address value to any memory location in the address section. Since an address can be 4 or 5 byte long (depending on the actual memory size), several consecutive write operations to the address section are required to specify the full address.
- **To read or write data**, the software reads or writes the data from/to any memory location in the data section.

Since the NAND Flash memory automatically increments addresses, there is no need to increment the address of the data section to access consecutive memory locations.

37.4.3 SDRAM address mapping

The HADDR[28] bit (internal AHB address line 28) is used to select one of the two memory banks as indicated in [Table 258](#).

Table 258. SDRAM bank selection

HADDR[28]	Selected bank	Control register	Timing register
0	SDRAM Bank1	FMC_SDCR1	FMC_SDTR1
1	SDRAM Bank2	FMC_SDCR2	FMC_SDTR2

The following table shows SDRAM mapping for an 13-bit row ,a 11-bit column and 4 internal bank configurations.

Table 259. SDRAM address mapping

Memory width ⁽¹⁾	Internal bank	Row address	Column address ⁽²⁾	Maximum memory capacity (Mbyte)
8-bit	HADDR[25:24]	HADDR[23:11]	HADDR[10:0]	64 Mbyte: 4 x 8K x 2K
16-bit	HADDR[26:25]	HADDR[24:12]	HADDR[11:1]	128 Mbyte: 4 x 8K x 2K x 2
32-bit	HADDR[27:26]	HADDR[25:13]	HADDR[12:2]	256 Mbyte: 4 x 8K x 2K x 4

1. When interfacing with a 16-bit memory, the FMC internally uses the HADDR[11:1] internal AHB address lines to generate the external address. When interfacing with a 32-bit memory, the FMC internally uses HADDR[12:2] lines to generate the external address. Whatever the memory width, FMC_A[0] has to be connected to the external memory address A[0].
2. The AutoPrecharge is not supported. FMC_A[10] must be connected to the external memory address A[10] but it will be always driven 'low'.

The HADDR[27:0] bits are translated to external SDRAM address depending on the SDRAM controller configuration:

- Data size: 8, 16 or 32 bits
- Row size: 11, 12 or 13 bits
- Column size: 8, 9, 10 or 11 bits
- Number of internal banks: two or four internal banks

[Table 260](#) to [Table](#) shows the SDRAM address mapping versus the SDRAM controller configuration.

Table 260. SDRAM address mapping with 8-bit data bus width⁽¹⁾⁽²⁾

Row size configuration	HADDR(AHB Internal Address Lines)																															
	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				
11-bit row size configuration	Res.							Bank [1:0]		Row[10:0]										Column[7:0]												
	Res.						Bank [1:0]		Row[10:0]										Column[8:0]													
	Res.					Bank [1:0]		Row[10:0]										Column[9:0]														
	Res.				Bank [1:0]		Row[10:0]										Column[10:0]															
12-bit row size configuration	Res.						Bank [1:0]		Row[11:0]										Column[7:0]													
	Res.					Bank [1:0]		Row[11:0]										Column[8:0]														
	Res.				Bank [1:0]		Row[11:0]										Column[9:0]															
	Res.			Bank [1:0]		Row[11:0]										Column[10:0]																
13-bit row size configuration	Res.					Bank [1:0]		Row[12:0]										Column[7:0]														
	Res.				Bank [1:0]		Row[12:0]										Column[8:0]															
	Res.			Bank [1:0]		Row[12:0]										Column[9:0]																
	Res.		Bank [1:0]		Row[12:0]										Column[10:0]																	

1. BANK[1:0] are the Bank Address BA[1:0]. When only 2 internal banks are used, BA1 must always be set to '0'.
2. Access to Reserved (Res.) address range generates an AHB error.

Table 261. SDRAM address mapping with 16-bit data bus width⁽¹⁾⁽²⁾

Row size Configuration	HADDR(AHB address Lines)																										
	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
11-bit row size configuration	Res.						Bank [1:0]		Row[10:0]								Column[7:0]					BM0 ⁽³⁾					
	Res.					Bank [1:0]		Row[10:0]								Column[8:0]					BM0						
	Res.				Bank [1:0]		Row[10:0]								Column[9:0]					BM0							
	Res.			Bank [1:0]		Row[10:0]								Column[10:0]					BM0								
12-bit row size configuration	Res.						Bank [1:0]		Row[11:0]								Column[7:0]					BM0					
	Res.				Bank [1:0]		Row[11:0]								Column[8:0]					BM0							
	Res.			Bank [1:0]		Row[11:0]								Column[9:0]					BM0								
	Res.		Bank [1:0]		Row[11:0]								Column[10:0]					BM0									
13-bit row size configuration	Res.				Bank [1:0]		Row[12:0]								Column[7:0]					BM0							
	Res.			Bank [1:0]		Row[12:0]								Column[8:0]					BM0								
	Res.		Bank [1:0]		Row[12:0]								Column[9:0]					BM0									
	Re s.	Bank [1:0]	Row[12:0]								Column[10:0]					BM0											

1. BANK[1:0] are the Bank Address BA[1:0]. When only 2 internal banks are used, BA1 must always be set to '0'.

2. Access to Reserved space (Res.) generates an AHB error.

3. BM0: is the byte mask for 16-bit access.

Table 262. SDRAM address mapping with 32-bit data bus width⁽¹⁾⁽²⁾

Row size configuration	HADDR(AHB address Lines)																										
	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
11-bit row size configuration	Res.					Bank [1:0]		Row[10:0]										Column[7:0]					BM[1:0] ⁽³⁾				
	Res.				Bank [1:0]		Row[10:0]										Column[8:0]					BM[1:0]					
	Res.			Bank [1:0]		Row[10:0]										Column[9:0]					BM[1:0]						
	Res.		Bank [1:0]		Row[10:0]										Column[10:0]					BM[1:0]							
12-bit row size configuration	Res.				Bank [1:0]		Row[11:0]										Column[7:0]					BM[1:0]					
	Res.			Bank [1:0]		Row[11:0]										Column[8:0]					BM[1:0]						
	Res.		Bank [1:0]		Row[11:0]										Column[9:0]					BM[1:0]							
	Res.	Bank [1:0]		Row[11:0]										Column[10:0]					BM[1:0]								

Table 262. SDRAM address mapping with 32-bit data bus width⁽¹⁾⁽²⁾ (continued)

Row size configuration	HADDR(AHB address Lines)																											
	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
13-bit row size configuration	Res.		Bank [1:0]		Row[12:0]														Column[7:0]								BM[1:0]	
	Res.		Bank [1:0]		Row[12:0]														Column[8:0]								BM[1:0]	
	Res.	Bank [1:0]		Row[12:0]														Column[9:0]								BM[1:0]		
	Bank [1:0]		Row[12:0]														Column[10:0]								BM[1:0]			

1. BANK[1:0] are the Bank Address BA[1:0]. When only 2 internal banks are used, BA1 must always be set to '0'.
2. Access to Reserved space (Res.) generates an AHB error.
3. BM[1:0]: is the byte mask for 32-bit access.

37.5 NOR Flash/PSRAM controller

The FMC generates the appropriate signal timings to drive the following types of memories:

- Asynchronous SRAM and ROM
 - 8 bits
 - 16 bits
 - 32 bits
- PSRAM (Cellular RAM)
 - Asynchronous mode
 - Burst mode for synchronous accesses
 - Multiplexed or non-multiplexed
- NOR Flash memory
 - Asynchronous mode
 - Burst mode for synchronous accesses
 - Multiplexed or non-multiplexed

The FMC outputs a unique Chip Select signal, NE[4:1], per bank. All the other signals (addresses, data and control) are shared.

The FMC supports a wide range of devices through a programmable timings among which:

- 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 to support the widest variety of memories and timings
- Programmable continuous clock (FMC_CLK) output.

The FMC Clock (FMC_CLK) is a submultiple of the HCLK clock. It can be delivered to the selected external device either during synchronous accesses only or during asynchronous

and synchronous accesses depending on the CCKEN bit configuration in the FMC_BCR1 register:

- If the CCKEN bit is reset, the FMC generates the clock (CLK) only during synchronous accesses (Read/write transactions).
- If the CCKEN bit is set, the FMC generates a continuous clock during asynchronous and synchronous accesses. To generate the FMC_CLK continuous clock, Bank 1 must be configured in synchronous mode (see [Section 37.5.6: NOR/PSRAM controller registers](#)). Since the same clock is used for all synchronous memories, when a continuous output clock is generated and synchronous accesses are performed, the AHB data size has to be the same as the memory data width (MWID) otherwise the FMC_CLK frequency will be changed depending on AHB data transaction (refer to [Section 37.5.5: Synchronous transactions](#) for FMC_CLK divider ratio formula).

The size of each bank is fixed and equal to 64 Mbyte. Each bank is configured through dedicated registers (see [Section 37.5.6: NOR/PSRAM controller registers](#)).

The programmable memory parameters include access times (see [Table 263](#)) and support for wait management (for PSRAM and NOR Flash accessed in burst mode).

Table 263. Programmable NOR/PSRAM access parameters

Parameter	Function	Access mode	Unit	Min.	Max.
Address setup	Duration of the address setup phase	Asynchronous	AHB clock cycle (HCLK)	0	15
Address hold	Duration of the address hold phase	Asynchronous, muxed I/Os	AHB clock cycle (HCLK)	1	15
Data setup	Duration of the data setup phase	Asynchronous	AHB clock cycle (HCLK)	1	256
Bust turn	Duration of the bus turnaround phase	Asynchronous and synchronous read/write	AHB clock cycle (HCLK)	0	15
Clock divide ratio	Number of AHB clock cycles (HCLK) to build one memory clock cycle (CLK)	Synchronous	AHB clock cycle (HCLK)	2	16
Data latency	Number of clock cycles to issue to the memory before the first data of the burst	Synchronous	Memory clock cycle (CLK)	2	17

37.5.1 External memory interface signals

[Table 264](#), [Table 265](#) and [Table 266](#) list the signals that are typically used to interface with NOR Flash memory, SRAM and PSRAM.

Note: The prefix “N” identifies the signals which are active low.

NOR Flash memory, non-multiplexed I/Os**Table 264. Non-multiplexed I/O NOR Flash memory**

FMC signal name	I/O	Function
CLK	O	Clock (for synchronous access)
A[25:0]	O	Address bus
D[31:0]	I/O	Bidirectional data bus
NE[x]	O	Chip Select, x = 1..4
NOE	O	Output enable
NWE	O	Write enable
NL(=NADV)	O	Latch enable (this signal is called address valid, NADV, by some NOR Flash devices)
NWAIT	I	NOR Flash wait input signal to the FMC

The maximum capacity is 512 Mbits (26 address lines).

NOR Flash memory, 16-bit multiplexed I/Os**Table 265. 16-bit multiplexed I/O NOR Flash memory**

FMC signal name	I/O	Function
CLK	O	Clock (for synchronous access)
A[25:16]	O	Address bus
AD[15:0]	I/O	16-bit multiplexed, bidirectional address/data bus (the 16-bit address A[15:0] and data D[15:0] are multiplexed on the databus)
NE[x]	O	Chip Select, x = 1..4
NOE	O	Output enable
NWE	O	Write enable
NL(=NADV)	O	Latch enable (this signal is called address valid, NADV, by some NOR Flash devices)
NWAIT	I	NOR Flash wait input signal to the FMC

The maximum capacity is 512 Mbits.

PSRAM/SRAM, non-multiplexed I/Os**Table 266. Non-multiplexed I/Os PSRAM/SRAM**

FMC signal name	I/O	Function
CLK	O	Clock (only for PSRAM synchronous access)
A[25:0]	O	Address bus
D[31:0]	I/O	Data bidirectional bus

Table 266. Non-multiplexed I/Os PSRAM/SRAM (continued)

FMC signal name	I/O	Function
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 only for PSRAM input (memory signal name: NADV)
NWAIT	I	PSRAM wait input signal to the FMC
NBL[3]	O	Byte3 Upper byte enable (memory signal name: NUB)
NBL[2]	O	Byte2 Lowed byte enable (memory signal name: NLB)
NBL[1]	O	Byte1 Upper byte enable (memory signal name: NLB)
NBL[0]	O	Byte0 Lower byte enable (memory signal name: NLB)

The maximum capacity is 512 Mbits.

PSRAM, 16-bit multiplexed I/Os

Table 267. 16-Bit multiplexed I/O PSRAM

FMC signal name	I/O	Function
CLK	O	Clock (for synchronous access)
A[25:16]	O	Address bus
AD[15:0]	I/O	16-bit multiplexed, bidirectional address/data bus (the 16-bit address A[15:0] and data D[15:0] are multiplexed on the databus)
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 FMC
NBL[1]	O	Upper byte enable (memory signal name: NUB)
NBL[0]	O	Lowed byte enable (memory signal name: NLB)

The maximum capacity is 512 Mbits (26 address lines).

37.5.2 Supported memories and transactions

[Table 268](#) below shows an example of the supported devices, access modes and transactions when the memory data bus is 16-bit wide for NOR Flash memory, PSRAM and SRAM. The transactions not allowed (or not supported) by the FMC are shown in gray in this example.

Table 268. NOR Flash/PSRAM: Example of supported memories and transactions

Device	Mode	R/W	AHB data size	Memory data size	Allowed/not allowed	Comments
NOR Flash (muxed I/Os and nonmuxed I/Os)	Asynchronous	R	8	16	Y	
	Asynchronous	W	8	16	N	
	Asynchronous	R	16	16	Y	
	Asynchronous	W	16	16	Y	
	Asynchronous	R	32	16	Y	Split into 2 FMC accesses
	Asynchronous	W	32	16	Y	Split into 2 FMC accesses
	Asynchronous page	R	-	16	N	Mode is not supported
	Synchronous	R	8	16	N	
	Synchronous	R	16	16	Y	
	Synchronous	R	32	16	Y	
PSRAM (multiplexed I/Os and non-multiplexed I/Os)	Asynchronous	R	8	16	Y	
	Asynchronous	W	8	16	Y	Use of byte lanes NBL[1:0]
	Asynchronous	R	16	16	Y	
	Asynchronous	W	16	16	Y	
	Asynchronous	R	32	16	Y	Split into 2 FMC accesses
	Asynchronous	W	32	16	Y	Split into 2 FMC accesses
	Asynchronous page	R	-	16	N	Mode is not supported
	Synchronous	R	8	16	N	
	Synchronous	R	16	16	Y	
	Synchronous	R	32	16	Y	
	Synchronous	W	8	16	Y	Use of byte lanes NBL[1:0]
	Synchronous	W	16/32	16	Y	
SRAM and ROM	Asynchronous	R	8 / 16	16	Y	
	Asynchronous	W	8 / 16	16	Y	Use of byte lanes NBL[1:0]
	Asynchronous	R	32	16	Y	Split into 2 FMC accesses
	Asynchronous	W	32	16	Y	Split into 2 FMC accesses Use of byte lanes NBL[1:0]

37.5.3 General timing rules

Signals synchronization

- All controller output signals change on the rising edge of the internal clock (HCLK)
- In synchronous mode (read or write), all output signals change on the rising edge of HCLK. Whatever the CLKDIV value, all outputs change as follows:
 - NOEL/NWEL/ NEL/NADV L/ NADV H /NBLL/ Address valid outputs change on the falling edge of FMC_CLK clock.
 - NOEH/ NWEH / NEH/ NOEH/NBLH/ Address invalid outputs change on the rising edge of FMC_CLK clock.

37.5.4 NOR Flash/PSRAM controller asynchronous transactions

Asynchronous static memories (NOR Flash, PSRAM, SRAM)

- Signals are synchronized by the internal clock HCLK. This clock is not issued to the memory
- The FMC always samples the data before de-asserting the NOE signal. This guarantees that the memory data hold timing constraint is met (minimum Chip Enable high to data transition is usually 0 ns)
- If the extended mode is enabled (EXTMOD bit is set in the FMC_BCRx register), up to four extended modes (A, B, C and D) are available. It is possible to mix A, B, C and D modes for read and write operations. For example, read operation can be performed in mode A and write in mode B.
- If the extended mode is disabled (EXTMOD bit is reset in the FMC_BCRx register), the FMC can operate in Mode1 or Mode2 as follows:
 - Mode 1 is the default mode when SRAM/PSRAM memory type is selected (MTYP[1:0] = 0x0 or 0x01 in the FMC_BCRx register)
 - Mode 2 is the default mode when NOR memory type is selected (MTYP[1:0] = 0x10 in the FMC_BCRx register).

Mode 1 - SRAM/PSRAM (CRAM)

The next figures show the read and write transactions for the supported modes followed by the required configuration of FMC_BCRx, and FMC_BTRx/FMC_BWTRx registers.

Figure 458. Mode1 read access waveforms

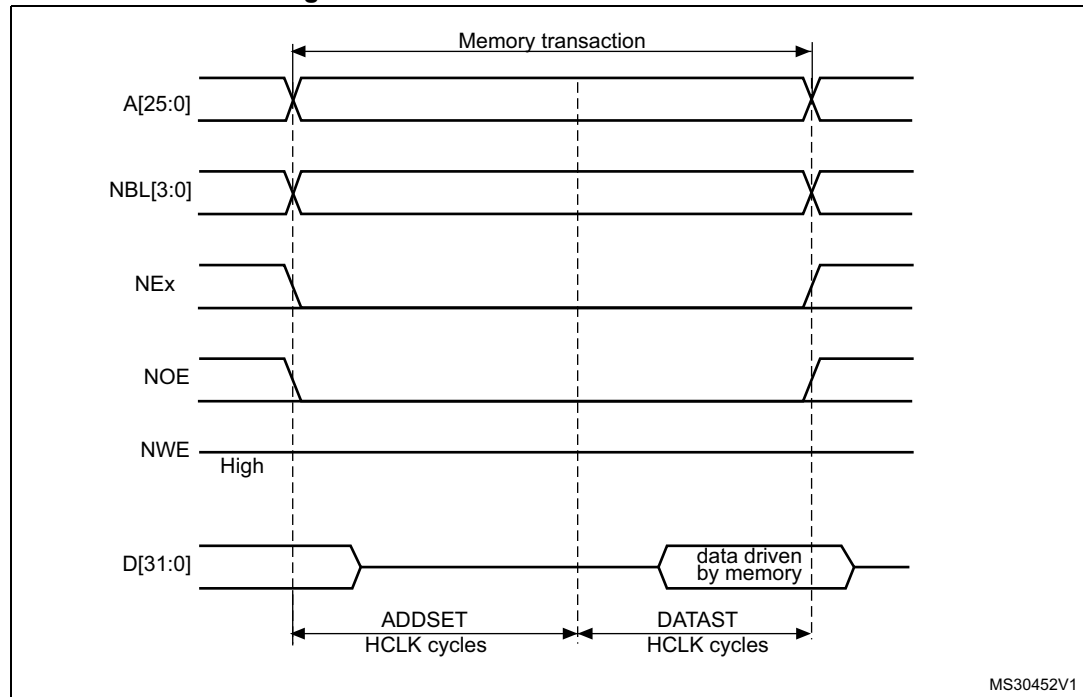
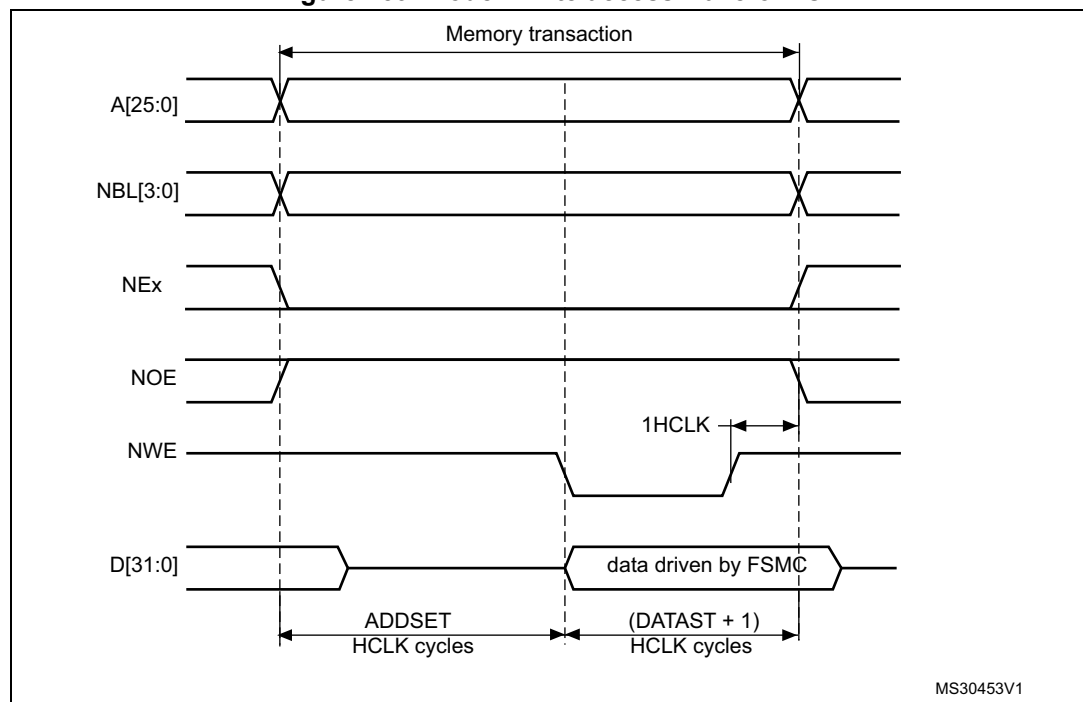


Figure 459. Mode1 write access waveforms



The one HCLK cycle at the end of the write transaction helps guarantee the address and data hold time after the NWE rising edge. Due to the presence of this HCLK cycle, the DATAST value must be greater than zero (DATAST > 0).

Table 269. FMC_BCRx bit fields

Bit number	Bit name	Value to set
31-21	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	0x0 (no effect in asynchronous mode)
18:16	CPSIZE	0x0 (no effect in asynchronous mode)
15	ASYNCAWAIT	Set to 1 if the memory supports this feature. Otherwise keep at 0.
14	EXTMOD	0x0
13	WAITEN	0x0 (no effect in asynchronous mode)
12	WREN	As needed
11	WAITCFG	Don't care
10	WRAPMOD	0x0
9	WAITPOL	Meaningful only if bit 15 is 1
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	Don't care
5-4	MWID	As needed
3-2	MTYP[1:0]	As needed, exclude 0x2 (NOR Flash memory)
1	MUXE	0x0
0	MBKEN	0x1

Table 270. FMC_BTRx bit fields

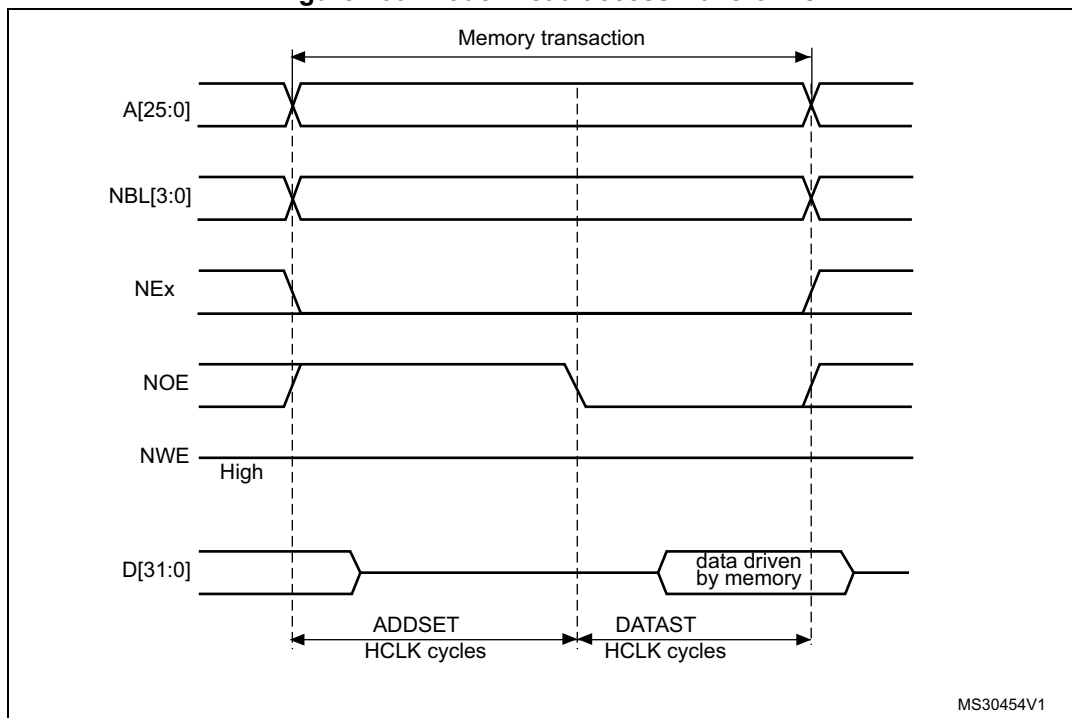
Bit number	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	Don't care
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST+1 HCLK cycles for write accesses, DATAST HCLK cycles for read accesses).

Table 270. FMC_BTRx bit fields (continued)

Bit number	Bit name	Value to set
7-4	ADDHLD	Don't care
3-0	ADDSET	Duration of the first access phase (ADDSET HCLK cycles). Minimum value for ADDSET is 0.

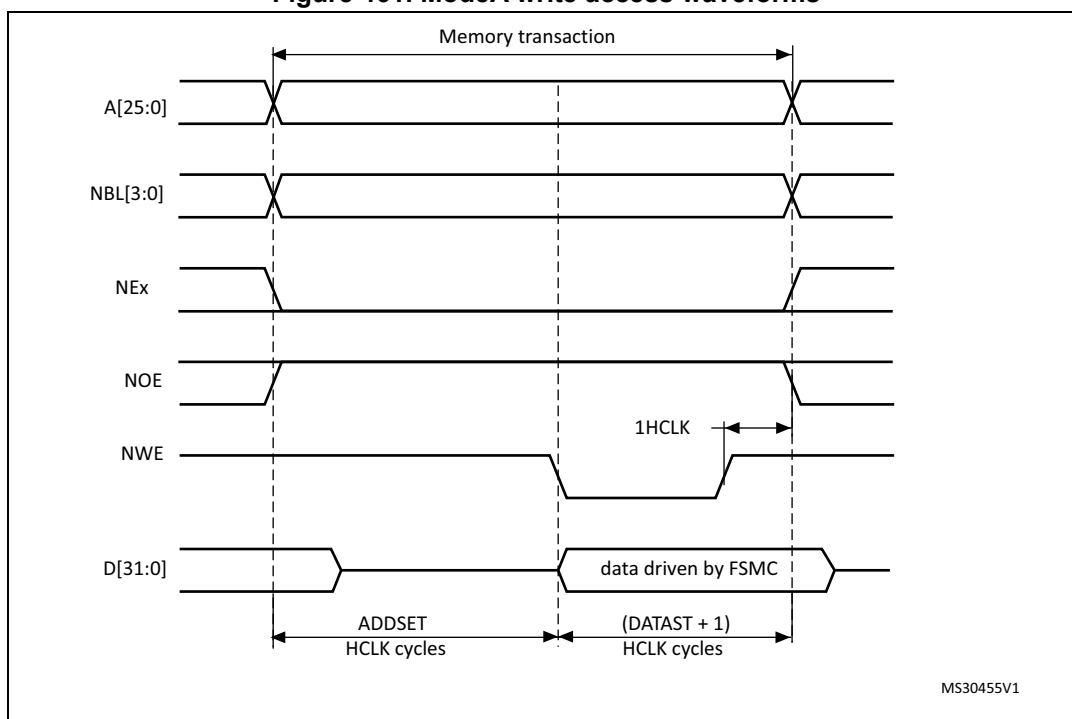
Mode A - SRAM/PSRAM (CRAM) OE toggling

Figure 460. ModeA read access waveforms



1. NBL[3:0] are driven low during the read access

Figure 461. ModeA write access waveforms



The differences compared with mode1 are the toggling of NOE and the independent read and write timings.

Table 271. FMC_BCRx bit fields

Bit number	Bit name	Value to set
31-21	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	0x0 (no effect in asynchronous mode)
18:16	CPSIZE	0x0 (no effect in asynchronous mode)
15	ASYNCWAIT	Set to 1 if the memory supports this feature. Otherwise keep at 0.
14	EXTMOD	0x1
13	WAITEN	0x0 (no effect in asynchronous mode)
12	WREN	As needed
11	WAITCFG	Don't care
10	WRAPMOD	0x0
9	WAITPOL	Meaningful only if bit 15 is 1
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	Don't care

Table 271. FMC_BCRx bit fields (continued)

Bit number	Bit name	Value to set
5-4	MWID	As needed
3-2	MTYP[1:0]	As needed, exclude 0x2 (NOR Flash memory)
1	MUXEN	0x0
0	MBKEN	0x1

Table 272. FMC_BTRx bit fields

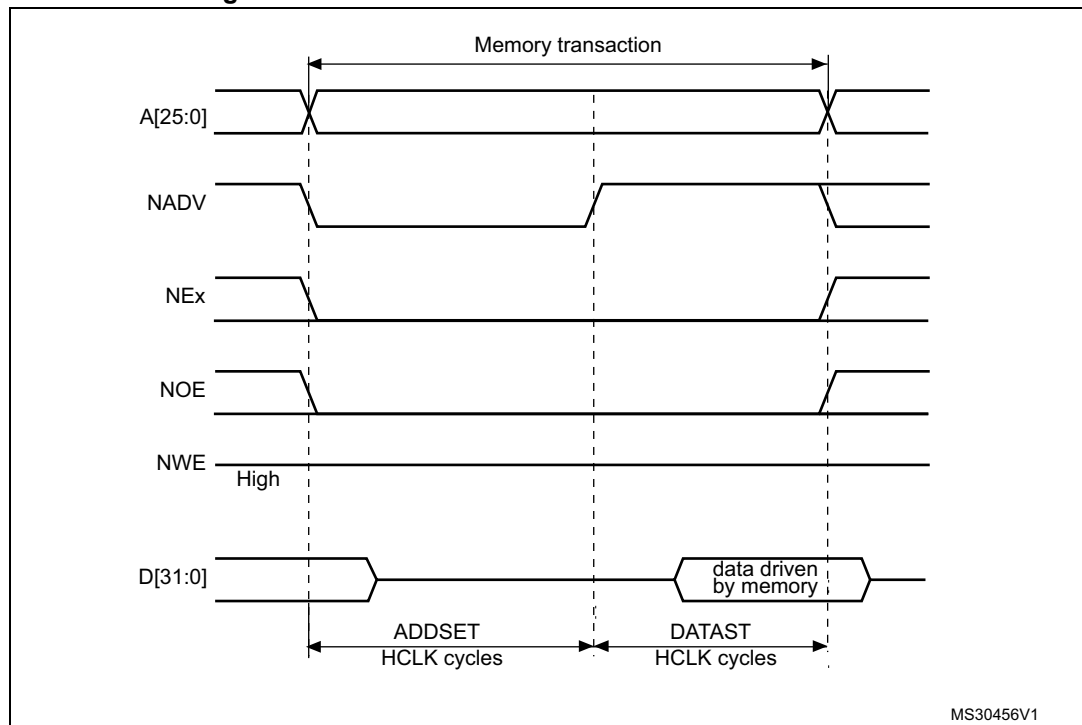
Bit number	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST HCLK cycles) for read accesses.
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Duration of the first access phase (ADDSET HCLK cycles) for read accesses. Minimum value for ADDSET is 0.

Table 273. FMC_BWTRx bit fields

Bit number	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST HCLK cycles) for write accesses.
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Duration of the first access phase (ADDSET HCLK cycles) for write accesses. Minimum value for ADDSET is 0.

Mode 2/B - NOR Flash

Figure 462. Mode2 and mode B read access waveforms



1. NBL[3:0] are driven low during the read access

Figure 463. Mode2 write access waveforms

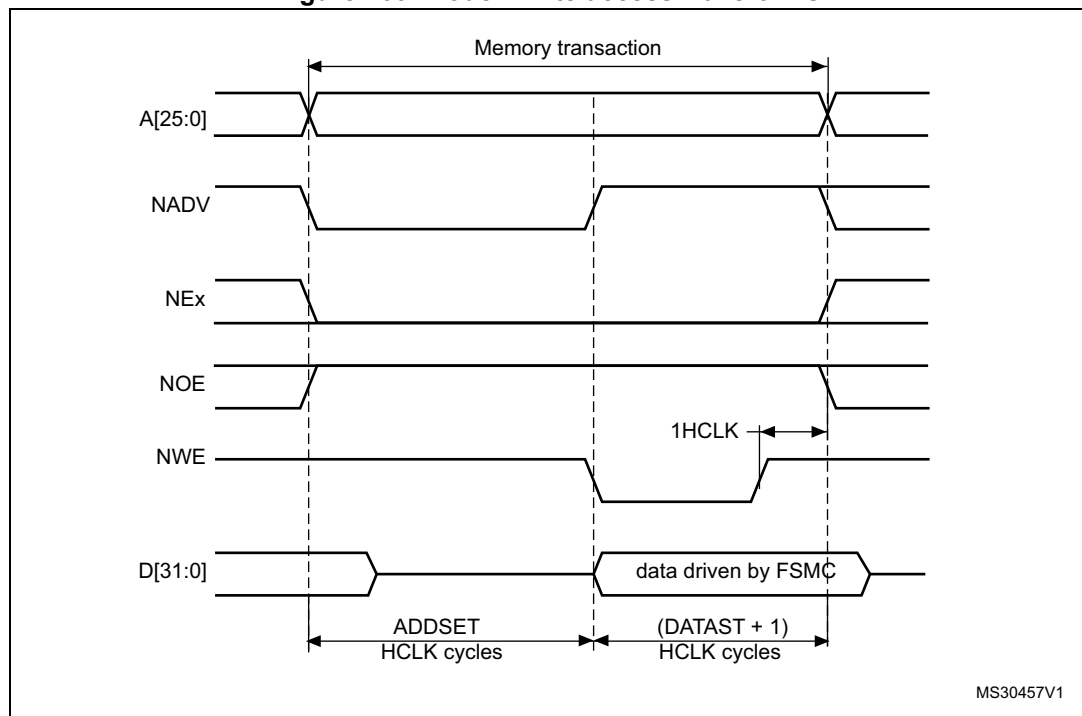
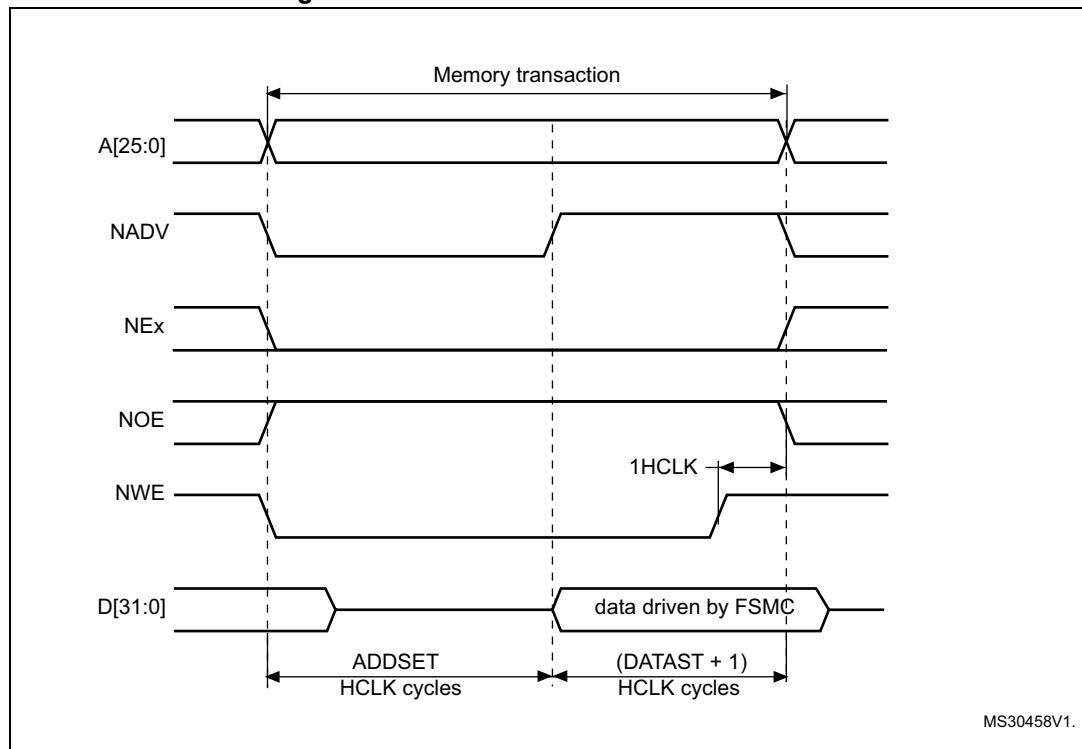


Figure 464. ModeB write access waveforms



The differences with mode1 are the toggling of NWE and the independent read and write timings when extended mode is set (Mode B).

Table 274. FMC_BCRx bit fields

Bit number	Bit name	Value to set
31-21	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	0x0 (no effect in asynchronous mode)
18:16	CPSIZE	0x0 (no effect in asynchronous mode)
15	ASYNCWAIT	Set to 1 if the memory supports this feature. Otherwise keep at 0.
14	EXTMOD	0x1 for mode B, 0x0 for mode 2
13	WAITEN	0x0 (no effect in asynchronous mode)
12	WREN	As needed
11	WAITCFG	Don't care
10	WRAPMOD	0x0
9	WAITPOL	Meaningful only if bit 15 is 1
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	0x1

Table 274. FMC_BCRx bit fields (continued)

Bit number	Bit name	Value to set
5-4	MWID	As needed
3-2	MTYP[1:0]	0x2 (NOR Flash memory)
1	MUXEN	0x0
0	MBKEN	0x1

Table 275. FMC_BTRx bit fields

Bit number	Bit name	Value to set
31-30	Reserved	0x0
29-28	ACCMOD	0x1 if extended mode is set
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the access second phase (DATAST HCLK cycles) for read accesses.
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Duration of the access first phase (ADDSET HCLK cycles) for read accesses. Minimum value for ADDSET is 0.

Table 276. FMC_BWTRx bit fields

Bit number	Bit name	Value to set
31-30	Reserved	0x0
29-28	ACCMOD	0x1 if extended mode is set
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the access second phase (DATAST HCLK cycles) for write accesses.
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Duration of the access first phase (ADDSET HCLK cycles) for write accesses. Minimum value for ADDSET is 0.

Note: The FMC_BWTRx register is valid only if the extended mode is set (mode B), otherwise its content is don't care.

Mode C - NOR Flash - OE toggling

Figure 465. ModeC read access waveforms

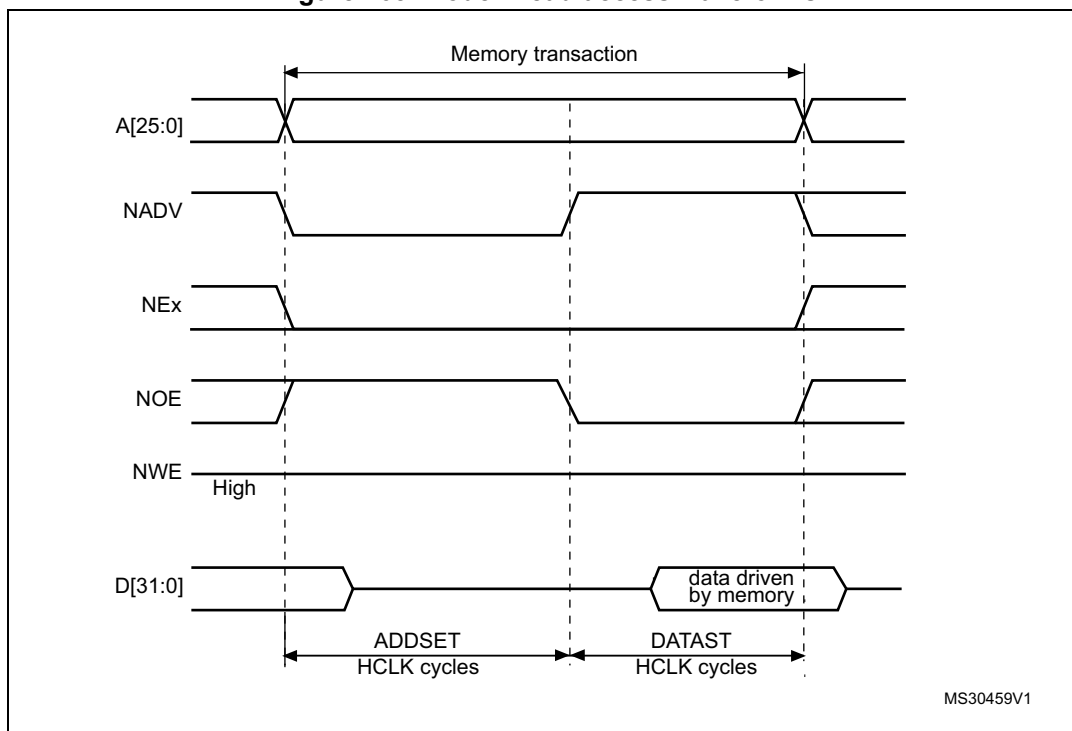
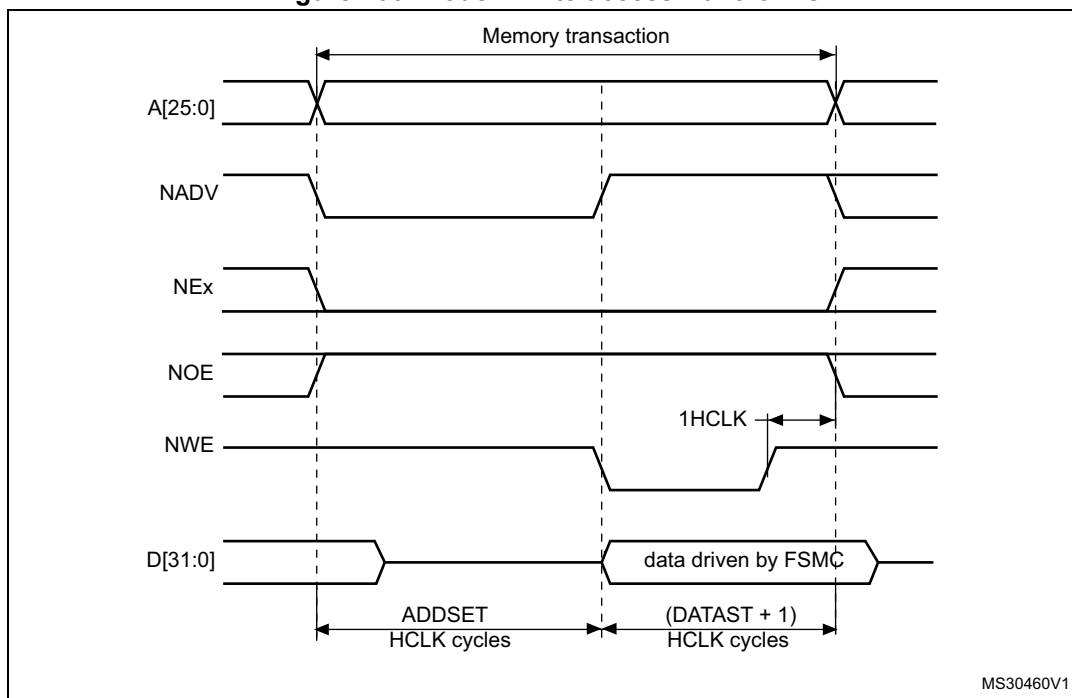


Figure 466. ModeC write access waveforms



The differences compared with mode1 are the toggling of NOE and the independent read and write timings.

Table 277. FMC_BCRx bit fields

Bit No.	Bit name	Value to set
31-21	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	0x0 (no effect in asynchronous mode)
18:16	Reserved	0x0 (no effect in asynchronous mode)
15	ASYNCAWAIT	Set to 1 if the memory supports this feature. Otherwise keep at 0.
14	EXTMOD	0x1
13	WAITEN	0x0 (no effect in asynchronous mode)
12	WREN	As needed
11	WAITCFG	Don't care
10	WRAPMOD	0x0
9	WAITPOL	Meaningful only if bit 15 is 1
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	0x1
5-4	MWID	As needed
3-2	MTYP[1:0]	0x02 (NOR Flash memory)
1	MUXEN	0x0
0	MBKEN	0x1

Table 278. FMC_BTRx bit fields

Bit No.	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	0x0
23-20	CLKDIV	0x0
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST HCLK cycles) for read accesses.
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Duration of the first access phase (ADDSET HCLK cycles) for read accesses. Minimum value for ADDSET is 0.

Table 279. FMC_BWTRx bit fields

Bit No.	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST HCLK cycles) for write accesses.
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Duration of the first access phase (ADDSET HCLK cycles) for write accesses. Minimum value for ADDSET is 0.

Mode D - asynchronous access with extended address

Figure 467. ModeD read access waveforms

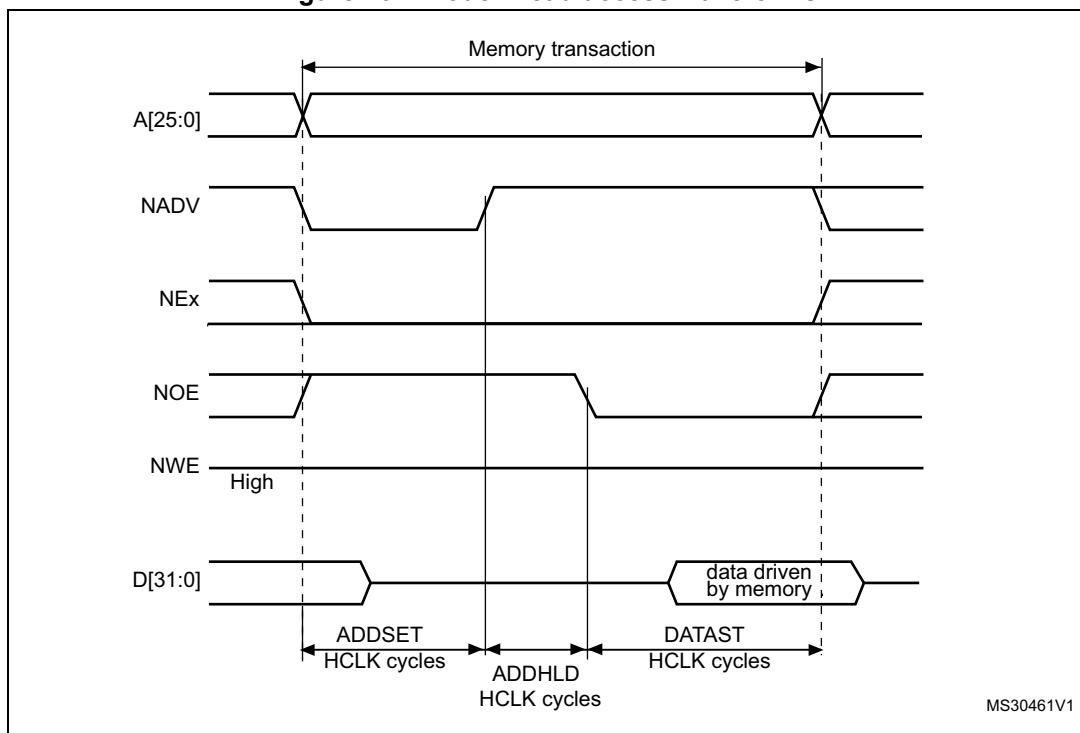
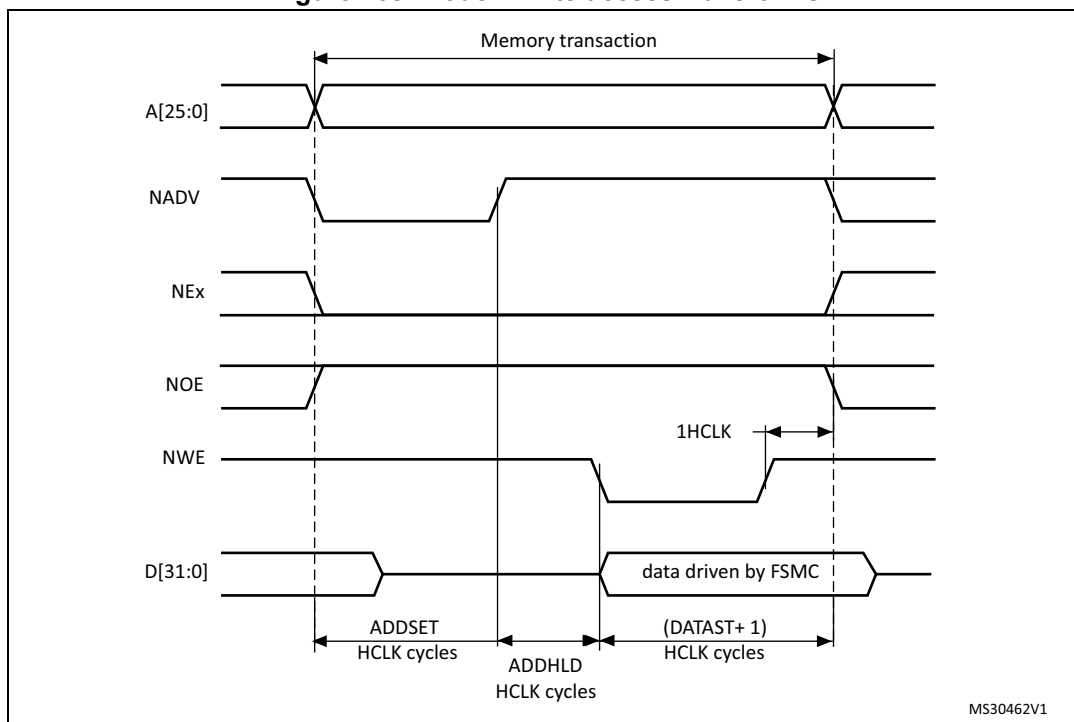


Figure 468. ModeD write access waveforms



The differences with mode1 are the toggling of NOE that goes on toggling after NADV changes and the independent read and write timings.

Table 280. FMC_BCRx bit fields

Bit No.	Bit name	Value to set
31-21	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	0x0 (no effect in asynchronous mode)
18:16	CPSIZE	0x0 (no effect in asynchronous mode)
15	ASYNCAWAIT	Set to 1 if the memory supports this feature. Otherwise keep at 0.
14	EXTMOD	0x1
13	WAITEN	0x0 (no effect in asynchronous mode)
12	WREN	As needed
11	WAITCFG	Don't care
10	WRAPMOD	0x0
9	WAITPOL	Meaningful only if bit 15 is 1
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	Set according to memory support
5-4	MWID	As needed

Table 280. FMC_BCRx bit fields (continued)

Bit No.	Bit name	Value to set
3-2	MTYP[1:0]	As needed
1	MUXEN	0x0
0	MBKEN	0x1

Table 281. FMC_BTRx bit fields

Bit No.	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST HCLK cycles) for read accesses.
7-4	ADDHLD	Duration of the middle phase of the read access (ADDHLD HCLK cycles)
3-0	ADDSET[3:0]	Duration of the first access phase (ADDSET HCLK cycles) for read accesses. Minimum value for ADDSET is 1.

Table 282. FMC_BWTRx bit fields

Bit No.	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST + 1 HCLK cycles) for write accesses.
7-4	ADDHLD	Duration of the middle phase of the write access (ADDHLD HCLK cycles)
3-0	ADDSET[3:0]	Duration of the first access phase (ADDSET HCLK cycles) for write accesses. Minimum value for ADDSET is 1.

Muxed mode - multiplexed asynchronous access to NOR Flash memory

Figure 469. Muxed read access waveforms

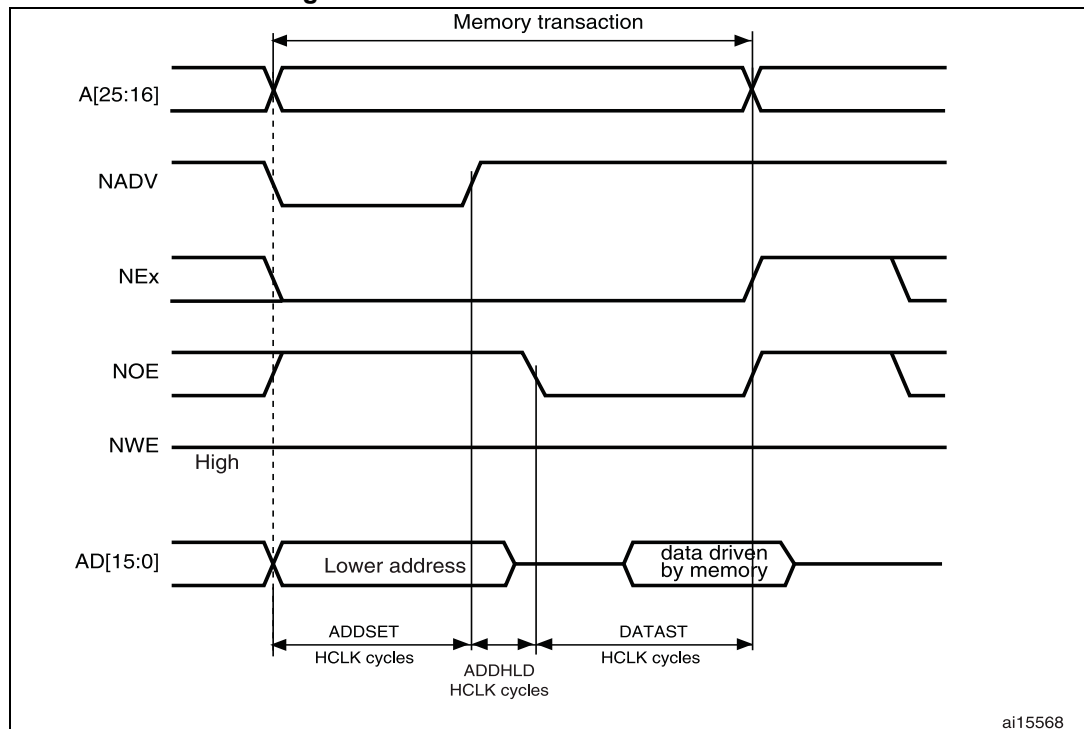
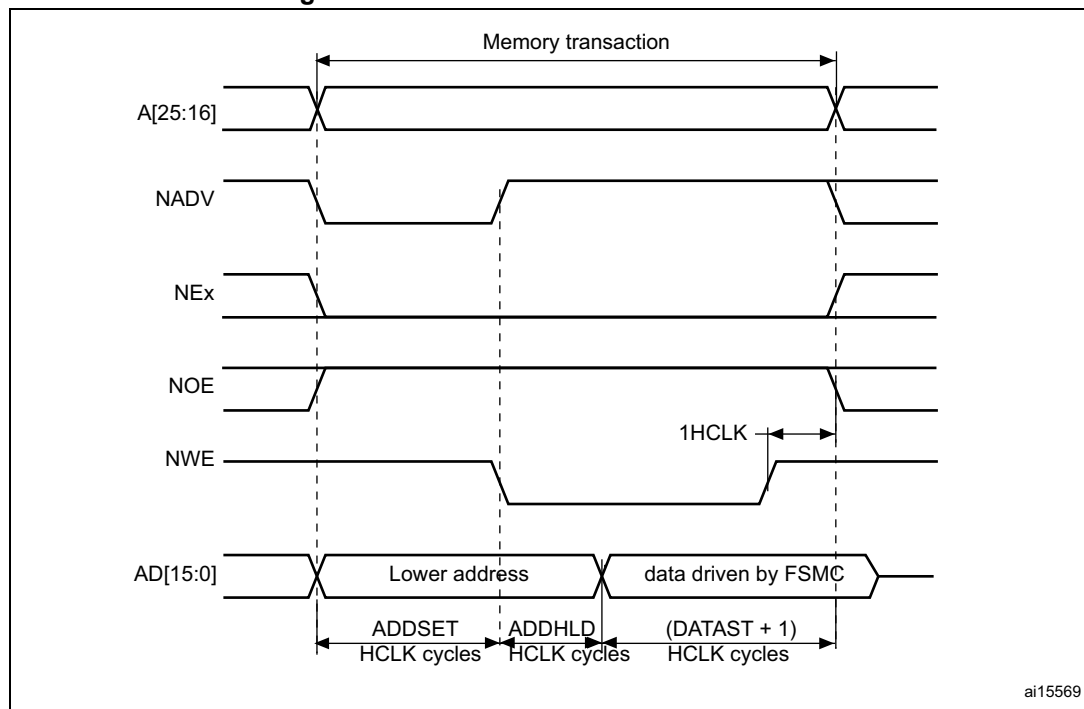


Figure 470. Muxed write access waveforms



The difference with mode D is the drive of the lower address byte(s) on the data bus.

Table 283. FMC_BCRx bit fields

Bit No.	Bit name	Value to set
31-21	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	0x0 (no effect in asynchronous mode)
18:16	CPSIZE	0x0 (no effect in asynchronous mode)
15	ASYNCWAIT	Set to 1 if the memory supports this feature. Otherwise keep at 0.
14	EXTMOD	0x0
13	WAITEN	0x0 (no effect in asynchronous mode)
12	WREN	As needed
11	WAITCFG	Don't care
10	WRAPMOD	0x0
9	WAITPOL	Meaningful only if bit 15 is 1
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	0x1
5-4	MWID	As needed
3-2	MTYP[1:0]	0x2 (NOR Flash memory)
1	MUXEN	0x1
0	MBKEN	0x1

Table 284. FMC_BTRx bit fields

Bit No.	Bit name	Value to set
31:30	Reserved	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	Don't care
23-20	CLKDIV	Don't care
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Duration of the second access phase (DATAST HCLK cycles for read accesses and DATAST+1 HCLK cycles for write accesses).
7-4	ADDHLD	Duration of the middle phase of the access (ADDHLD HCLK cycles).
3-0	ADDSET[3:0]	Duration of the first access phase (ADDSET HCLK cycles). Minimum value for ADDSET is 1.

WAIT management in asynchronous accesses

If the asynchronous memory asserts the WAIT signal to indicate that it is not yet ready to accept or to provide data, the ASYNCWAIT bit has to be set in FMC_BCRx register.

If the WAIT signal is active (high or low depending on the WAITPOL bit), the second access phase (Data setup phase), programmed by the DATAST bits, is extended until WAIT becomes inactive. Unlike the data setup phase, the first access phases (Address setup and Address hold phases), programmed by the ADDSET[3:0] and ADDHLD bits, are not WAIT sensitive and so they are not prolonged.

The data setup phase must be programmed so that WAIT can be detected 4 HCLK cycles before the end of the memory transaction. The following cases must be considered:

1. The memory asserts the WAIT signal aligned to NOE/NWE which toggles:

$$\text{DATAST} \geq (4 \times \text{HCLK}) + \text{max_wait_assertion_time}$$

2. The memory asserts the WAIT signal aligned to NEx (or NOE/NWE not toggling):
if

$$\text{max_wait_assertion_time} > \text{address_phase} + \text{hold_phase}$$

then:

$$\text{DATAST} \geq (4 \times \text{HCLK}) + (\text{max_wait_assertion_time} - \text{address_phase} - \text{hold_phase})$$

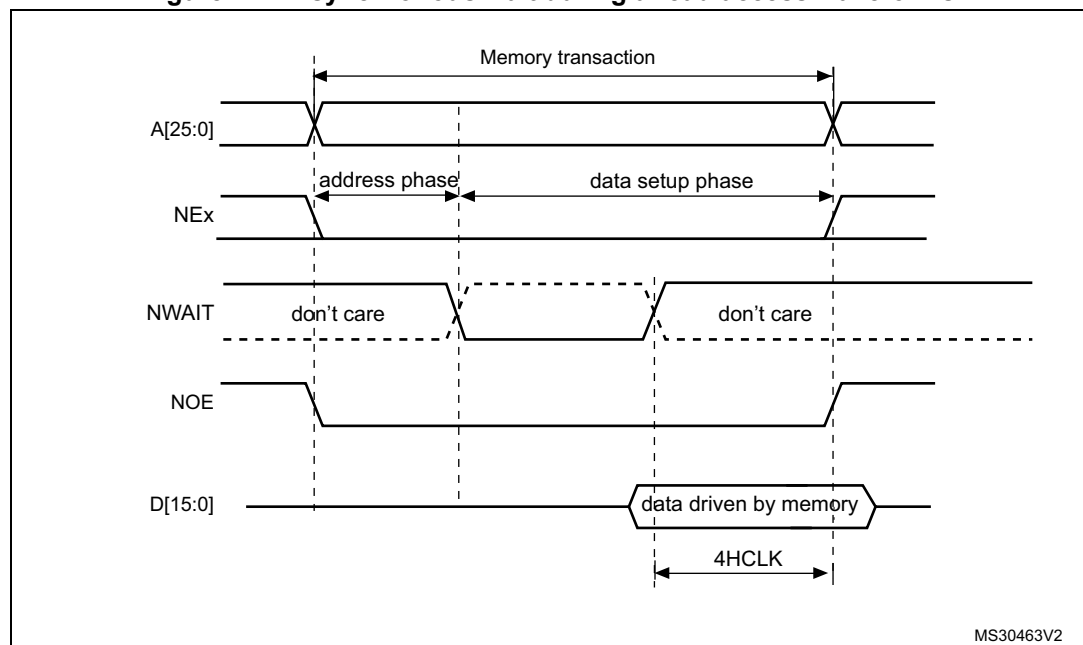
otherwise

$$\text{DATAST} \geq 4 \times \text{HCLK}$$

where max_wait_assertion_time is the maximum time taken by the memory to assert the WAIT signal once NEx/NOE/NWE is low.

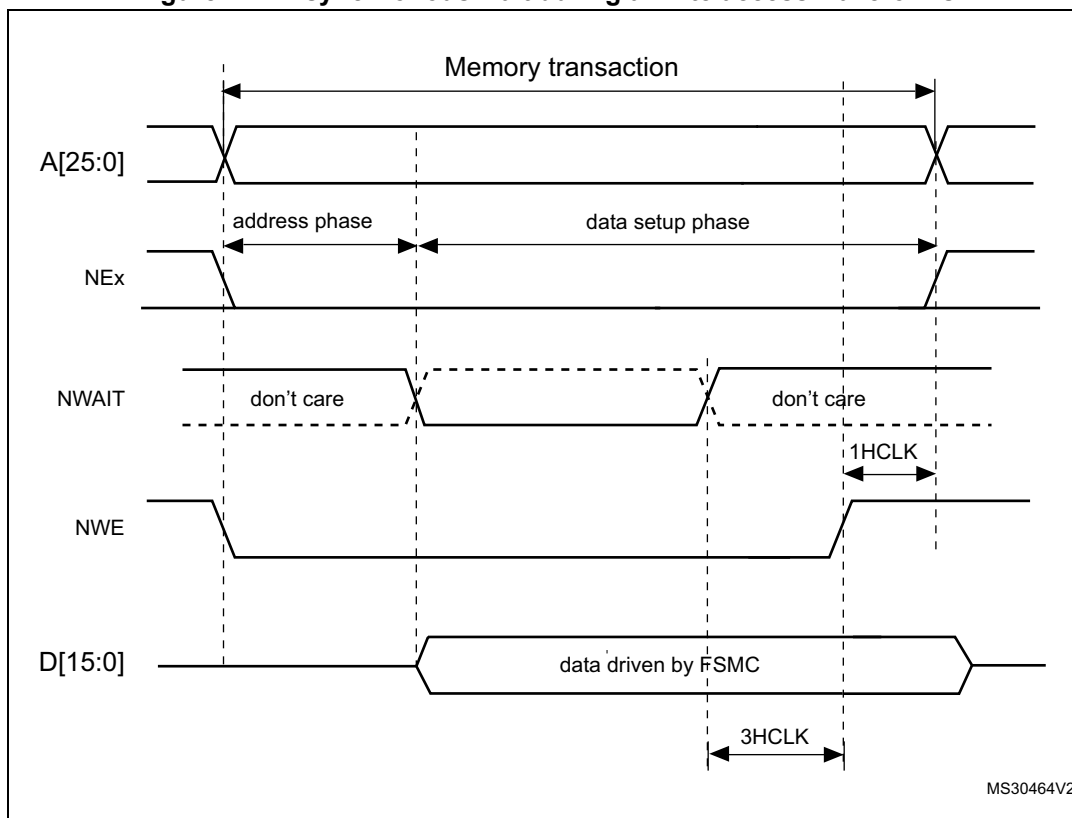
Figure 471 and Figure 472 show the number of HCLK clock cycles that are added to the memory access phase after WAIT is released by the asynchronous memory (independently of the above cases).

Figure 471. Asynchronous wait during a read access waveforms



1. NWAIT polarity depends on WAITPOL bit setting in FMC_BCRx register.

Figure 472. Asynchronous wait during a write access waveforms



1. NWAIT polarity depends on WAITPOL bit setting in FMC_BCRx register.

37.5.5 Synchronous transactions

The memory clock, FMC_CLK, is a submultiple of HCLK. It depends on the value of CLKDIV and the MWID/ AHB data size, following the formula given below:

$$\text{FMC_CLK divider ratio} = \max(\text{CLKDIV} + 1, \text{MWID}(\text{AHB data size}))$$

If MWID is 16 or 8 bits, the FMC_CLK divider ratio is always defined by the programmed CLKDIV value.

If MWID is 32 bits, the FMC_CLK divider ratio depends also on AHB data size.

Example:

- If CLKDIV=1, MWID=32 bits, AHB data size=8 bits, FMC_CLK=HCLK/4.
- If CLKDIV=1, MWID=16 bits, AHB data size=8 bits, FMC_CLK=HCLK/2.

NOR Flash memories specify a minimum time from NADV assertion to CLK high. To meet this constraint, the FMC does not issue the clock to the memory during the first internal clock cycle of the synchronous access (before NADV assertion). This guarantees that the rising edge of the memory clock occurs in the middle of the NADV low pulse.

Data latency versus NOR memory latency

The data latency is the number of cycles to wait before sampling the data. The DATLAT value must be consistent with the latency value specified in the NOR Flash configuration

register. The FMC does not include the clock cycle when NADV is low in the data latency count.

Caution: Some NOR Flash memories include the NADV Low cycle in the data latency count, so that the exact relation between the NOR Flash latency and the FMC DATLAT parameter can be either:

- NOR Flash latency = (DATLAT + 2) CLK clock cycles
- or NOR Flash latency = (DATLAT + 3) CLK clock cycles

Some recent memories assert NWAIT during the latency phase. In such cases DATLAT can be set to its minimum value. As a result, the FMC samples the data and waits long enough to evaluate if the data are valid. Thus the FMC detects when the memory exits latency and real data are processed.

Other memories do not assert NWAIT during latency. In this case the latency must be set correctly for both the FMC and the memory, otherwise invalid data are mistaken for good data, or valid data are lost in the initial phase of the memory access.

Single-burst transfer

When the selected bank is configured in burst mode for synchronous accesses, if for example an AHB single-burst transaction is requested on 16-bit memories, the FMC performs a burst transaction of length 1 (if the AHB transfer is 16 bits), or length 2 (if the AHB transfer is 32 bits) and de-assert the Chip Select signal when the last data is strobed.

Such transfers are not the most efficient in terms of cycles compared to asynchronous read operations. Nevertheless, a random asynchronous access would first require to re-program the memory access mode, which would altogether last longer.

Cross boundary page for Cellular RAM 1.5

Cellular RAM 1.5 does not allow burst access to cross the page boundary. The FMC controller allows to split automatically the burst access when the memory page size is reached by configuring the CPSIZE bits in the FMC_BCR1 register following the memory page size.

Wait management

For synchronous NOR Flash memories, NWAIT is evaluated after the programmed latency period, which corresponds to (DATLAT+2) CLK clock cycles.

If NWAIT is active (low level when WAITPOL = 0, high level when WAITPOL = 1), wait states are inserted until NWAIT is inactive (high level when WAITPOL = 0, low level when WAITPOL = 1).

When NWAIT is inactive, the data is considered valid either immediately (bit WAITCFG = 1) or on the next clock edge (bit WAITCFG = 0).

During wait-state insertion via the NWAIT signal, the controller continues to send clock pulses to the memory, keeping the Chip Select and output enable signals valid. It does not consider the data as valid.

In burst mode, there are two timing configurations for the NOR Flash NWAIT signal:

- The Flash memory asserts the NWAIT signal one data cycle before the wait state (default after reset).
- The Flash memory asserts the NWAIT signal during the wait state

The FMC supports both NOR Flash wait state configurations, for each Chip Select, thanks to the WAITCFG bit in the FMC_BCRx registers ($x = 0..3$).

Figure 473. Wait configuration waveforms

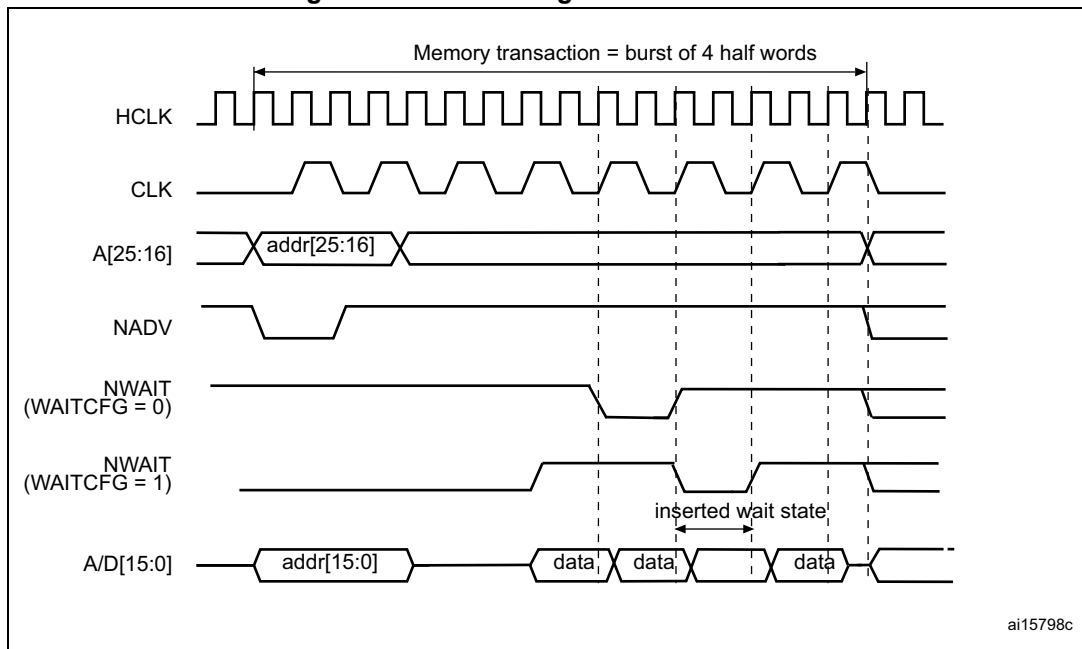
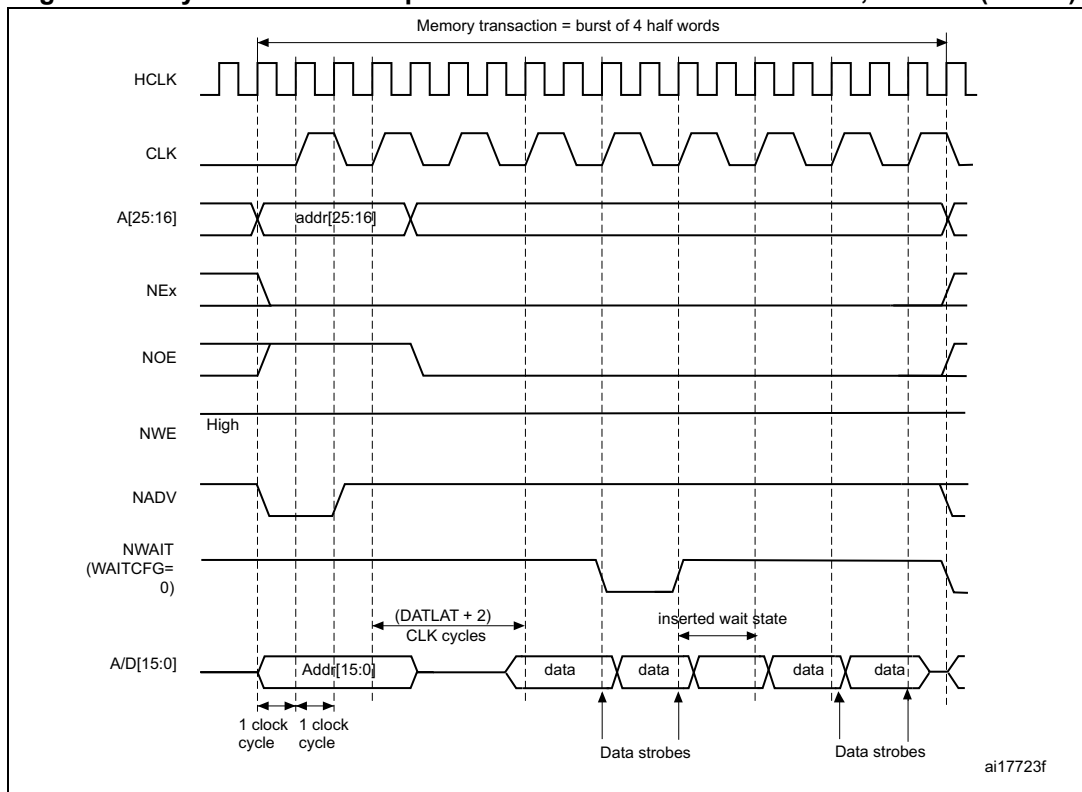


Figure 474. Synchronous multiplexed read mode waveforms - NOR, PSRAM (CRAM)



1. Byte lane outputs BL are not shown; for NOR access, they are held high, and, for PSRAM (CRAM) access,

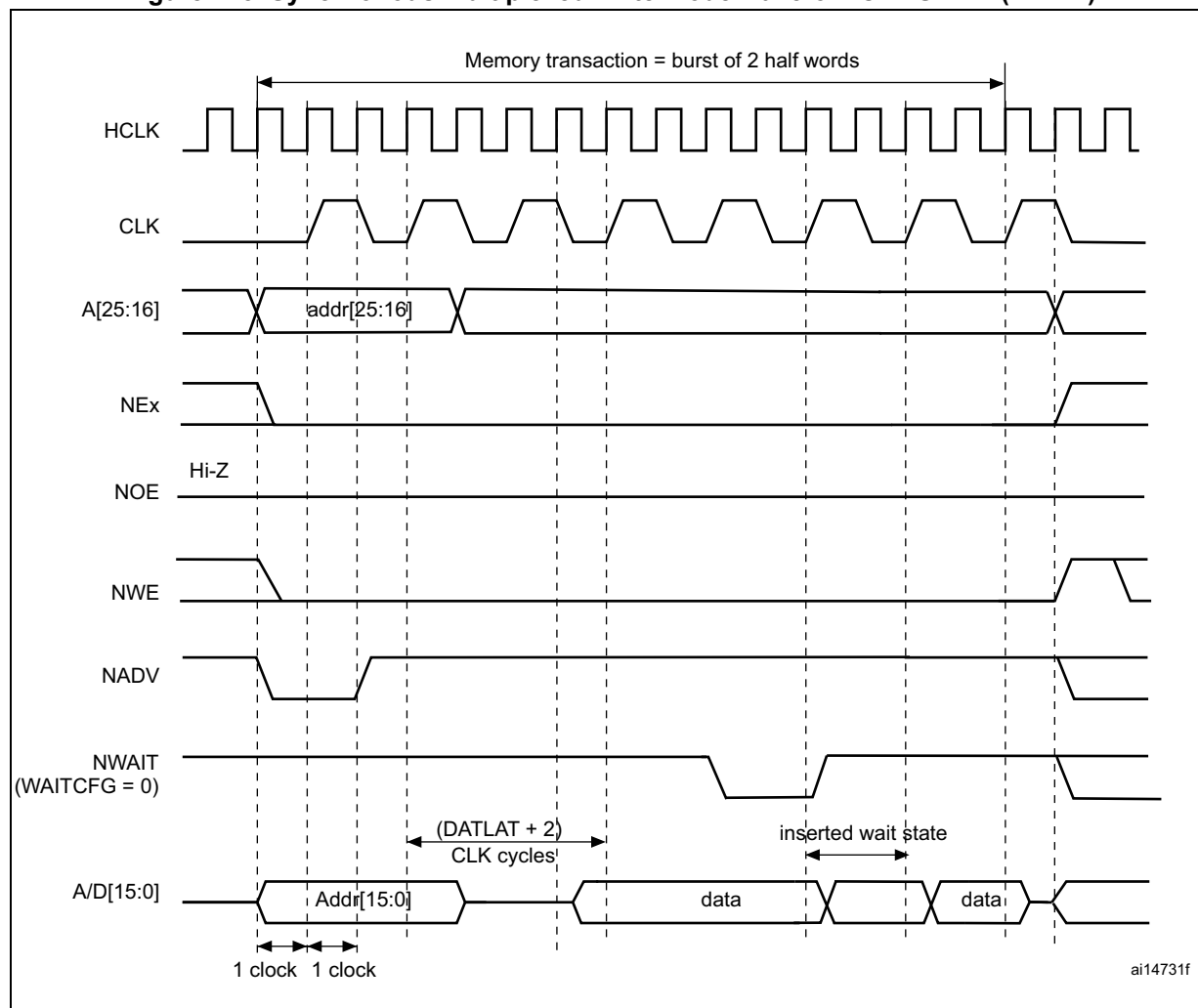
they are held low.

Table 285. FMC_BCRx bit fields

Bit No.	Bit name	Value to set
31-21	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	No effect on synchronous read
18-16	CPSIZE	As needed (0x1 for CRAM 1.5)
15	ASYNCAWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	to be set to 1 if the memory supports this feature, to be kept at 0 otherwise
12	WREN	no effect on synchronous read
11	WAITCFG	to be set according to memory
10	WRAPMOD	0x0
9	WAITPOL	to be set according to memory
8	BURSTEN	0x1
7	Reserved	0x1
6	FACCEN	Set according to memory support (NOR Flash memory)
5-4	MWID	As needed
3-2	MTYP[1:0]	0x1 or 0x2
1	MUXEN	As needed
0	MBKEN	0x1

Table 286. FMC_BTRx bit fields

Bit No.	Bit name	Value to set
31:30	Reserved	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	Data latency
27-24	DATLAT	Data latency
23-20	CLKDIV	0x0 to get CLK = HCLK (Not supported) 0x1 to get CLK = 2 × HCLK ..
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Don't care
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Don't care

Figure 475. Synchronous multiplexed write mode waveforms - PSRAM (CRAM)

1. The memory must issue NWAIT signal one cycle in advance, accordingly WAITCFG must be programmed to 0.
2. Byte Lane (NBL) outputs are not shown, they are held low while NEx is active.

Table 287. FMC_BCRx bit fields

Bit No.	Bit name	Value to set
31-20	Reserved	0x000
20	CCLKEN	As needed
19	CBURSTRW	0x1
18-16	CPSIZE	As needed (0x1 for CRAM 1.5)
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	to be set to 1 if the memory supports this feature, to be kept at 0 otherwise.
12	WREN	0x1

Table 287. FMC_BCRx bit fields (continued)

Bit No.	Bit name	Value to set
11	WAITCFG	0x0
10	WRAPMOD	0x0
9	WAITPOL	to be set according to memory
8	BURSTEN	no effect on synchronous write
7	Reserved	0x1
6	FACCEN	Set according to memory support
5-4	MWID	As needed
3-2	MTYP[1:0]	0x1
1	MUXEN	As needed
0	MBKEN	0x1

Table 288. FMC_BTRx bit fields

Bit No.	Bit name	Value to set
31-30	Reserved	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	Data latency
23-20	CLKDIV	0x0 to get CLK = HCLK (not supported) 0x1 to get CLK = 2 × HCLK
19-16	BUSTURN	Time between NEx high to NEx low (BUSTURN HCLK)
15-8	DATAST	Don't care
7-4	ADDHLD	Don't care
3-0	ADDSET[3:0]	Don't care

37.5.6 NOR/PSRAM controller registers

SRAM/NOR-Flash chip-select control registers 1..4 (FMC_BCR1..4)

Address offset: $8 * (x - 1)$, $x = 1...4$

Reset value: 0x0000 30DB for Bank1 and 0x0000 30D2 for Bank 2 to 4

This register contains the control information of each memory bank, used for SRAMs, PSRAM and NOR Flash memories.

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												CCLKEN	CPSIZE[2:0]	CBURSTRW	ASCYCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]			MTYP[1:0]		MUXEN	MBKEN
												rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31: 21 Reserved, must be kept at reset value

Bit 20 **CCLKEN**: Continuous Clock Enable.

This bit enables the FMC_CLK clock output to external memory devices.

0: The FMC_CLK is only generated during the synchronous memory access (read/write transaction). The FMC_CLK clock ratio is specified by the programmed CLKDIV value in the FMC_BCRx register (default after reset) .

1: The FMC_CLK is generated continuously during asynchronous and synchronous access. The FMC_CLK clock is activated when the CCLKEN is set.

Note: The CCLKEN bit of the FMC_BCR2..4 registers is don't care. It is only enabled through the FMC_BCR1 register. Bank 1 must be configured in synchronous mode to generate the FMC_CLK continuous clock.

Note: If CCLKEN bit is set, the FMC_CLK clock ratio is specified by CLKDIV value in the FMC_BTR1 register. CLKDIV in FMC_BWTR1 is don't care.

Note: If the synchronous mode is used and CCLKEN bit is set, the synchronous memories connected to other banks than Bank 1 are clocked by the same clock (the CLKDIV value in the FMC_BTR2..4 and FMC_BWTR2..4 registers for other banks has no effect.)

Bit 19 **CBURSTRW**: Write burst enable.

For PSRAM (CRAM) operating in burst mode, the bit enables synchronous accesses during write operations. The enable bit for synchronous read accesses is the BURSTEN bit in the FMC_BCRx register.

0: Write operations are always performed in asynchronous mode

1: Write operations are performed in synchronous mode.

Bits 18:16 **CPSIZE[2:0]**: CRAM page size.

These are used for Cellular RAM 1.5 which does not allow burst access to cross the address boundaries between pages. When these bits are configured, the FMC controller splits automatically the burst access when the memory page size is reached (refer to memory datasheet for page size).

000: No burst split when crossing page boundary (default after reset)

001: 128 bytes

010: 256 bytes

011: 512 bytes

100: 1024 bytes

Others: reserved

- Bit 15 **ASYNWAIT**: Wait signal during asynchronous transfers
 This bit enables/disables the FMC to use the wait signal even during an asynchronous protocol.
 0: NWAIT signal is not taken in to account when running an asynchronous protocol (default after reset)
 1: NWAIT signal is taken in to account when running an asynchronous protocol
- Bit 14 **EXTMOD**: Extended mode enable.
 This bit enables the FMC to program the write timings for non-multiplexed asynchronous accesses inside the FMC_BWTR register, thus resulting in different timings for read and write operations.
 0: values inside FMC_BWTR register are not taken into account (default after reset)
 1: values inside FMC_BWTR register are taken into account
Note: When the extended mode is disabled, the FMC can operate in Mode1 or Mode2 as follows:
 – Mode 1 is the default mode when the SRAM/PSRAM memory type is selected (MTYP[1:0] = 0x0 or 0x01)
 – Mode 2 is the default mode when the NOR memory type is selected (MTYP[1:0] = 0x10).
- Bit 13 **WAITEN**: Wait enable bit.
 This bit enables/disables wait-state insertion via the NWAIT signal when accessing the memory in synchronous mode.
 0: NWAIT signal is disabled (its level not taken into account, no wait state inserted after the programmed Flash latency period)
 1: NWAIT signal is enabled (its level is taken into account after the programmed latency period to insert wait states if asserted) (default after reset)
- Bit 12 **WREN**: Write enable bit.
 This bit indicates whether write operations are enabled/disabled in the bank by the FMC:
 0: Write operations are disabled in the bank by the FMC, an AHB error is reported,
 1: Write operations are enabled for the bank by the FMC (default after reset).
- Bit 11 **WAITCFG**: Wait timing configuration.
 The NWAIT signal indicates whether the data from the memory are valid or if a wait state must be inserted when accessing the memory in synchronous mode. This configuration bit determines if NWAIT is asserted by the memory one clock cycle before the wait state or during the wait state:
 0: NWAIT signal is active one data cycle before wait state (default after reset),
 1: NWAIT signal is active during wait state (not used for PSRAM).
- Bit 10 **WRAPMOD**: Wrapped burst mode support.
 Defines whether the controller will or not split an AHB burst wrap access into two linear accesses. Valid only when accessing memories in burst mode
 0: Direct wrapped burst is not enabled (default after reset),
 1: Direct wrapped burst is enabled.
Note: This bit has no effect as the CPU and DMA cannot generate wrapping burst transfers.
- Bit 9 **WAITPOL**: Wait signal polarity bit.
 Defines the polarity of the wait signal from memory used for either in synchronous or asynchronous mode:
 0: NWAIT active low (default after reset),
 1: NWAIT active high.
- Bit 8 **BURSTEN**: Burst enable bit.
 This bit enables/disables synchronous accesses during read operations. It is valid only for synchronous memories operating in burst mode:
 0: Burst mode disabled (default after reset). Read accesses are performed in asynchronous mode.
 1: Burst mode enable. Read accesses are performed in synchronous mode.
- Bit 7 Reserved, must be kept at reset value

- Bit 6 **FACCEN**: Flash access enable
 Enables NOR Flash memory access operations.
 0: Corresponding NOR Flash memory access is disabled
 1: Corresponding NOR Flash memory access is enabled (default after reset)
- Bits 5:4 **MWID[1:0]**: Memory data bus width.
 Defines the external memory device width, valid for all type of memories.
 00: 8 bits,
 01: 16 bits (default after reset),
 10: 32 bits,
 11: reserved, do not use.
- Bits 3:2 **MTYP[1:0]**: Memory type.
 Defines the type of external memory attached to the corresponding memory bank:
 00: SRAM (default after reset for Bank 2...4)
 01: PSRAM (CRAM)
 10: NOR Flash/OneNAND Flash (default after reset for Bank 1)
 11: reserved
- Bit 1 **MUXEN**: Address/data multiplexing enable bit.
 When this bit is set, the address and data values are multiplexed on the data bus, valid only with NOR and PSRAM memories:
 0: Address/Data nonmultiplexed
 1: Address/Data multiplexed on databus (default after reset)
- 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

SRAM/NOR-Flash chip-select timing registers 1..4 (FMC_BTR1..4)

Address offset: $0x04 + 8 * (x - 1)$, $x = 1..4$

Reset value: 0x0FFF FFFF

Reset value: 0x0FFF FFFF

FMC_BTRx bits are written by software to add a delay at the end of a read /write transaction. This delay allows matching the minimum time between consecutive transactions (t_{EHEL} from NEx high to FMC_NEx low) and the maximum time required by the memory to free the data bus after a read access (t_{EHQZ}).

This register contains the control information of each memory bank, used for SRAMs, PSRAM and NOR Flash memories. If the EXTMOD bit is set in the FMC_BCRx register, then this register is partitioned for write and read access, that is, 2 registers are available: one to configure read accesses (this register) and one to configure write accesses (FMC_BWTRx registers).

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[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATA[7:0]							ADDHLD[3:0]				ADDSET[3:0]				
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:30 Reserved, must be kept at reset value

Bits 29:28 **ACCMOD[1:0]**: Access mode

Specifies the asynchronous access modes as shown in the timing diagrams. These bits are taken into account only when the EXTMOD bit in the FMC_BCRx register is 1.

00: access mode A

01: access mode B

10: access mode C

11: access mode D

Bits 27:24 **DATLAT[3:0]**: Data latency for synchronous memory (see note below bit description table)

For synchronous accesses with read/write burst mode enabled (BURSTEN / CBURSTRW bits set), this field defines the number of memory clock cycles (+2) to issue to the memory before reading/writing the first data. This timing parameter is not expressed in HCLK periods, but in FMC_CLK periods. For asynchronous accesses, this value is don't care.

0000: Data latency of 2 CLK clock cycles for first burst access

1111: Data latency of 17 CLK clock cycles for first burst access (default value after reset)

Bits 23:20 **CLKDIV[3:0]**: Clock divide ratio (for FMC_CLK signal)

Defines the period of FMC_CLK clock output signal, expressed in number of HCLK cycles:

0000: Reserved

0001: FMC_CLK period = 2 × HCLK periods

0010: FMC_CLK period = 3 × HCLK periods

1111: FMC_CLK period = 16 × HCLK periods (default value after reset)

In asynchronous NOR Flash, SRAM or PSRAM accesses, this value is don't care.

Note: Refer to section 37.5.5: Synchronous transactions for FMC_CLK divider ratio formula)

Bits 19:16 **BUSTURN[3:0]**: Bus turnaround phase duration

These bits are written by software to add a delay at the end of a write-to-read (and read-to-write) transaction. This delay allows to match the minimum time between consecutive transactions (t_{EHEL} from NEx high to NEx low) and the maximum time needed by the memory to free the data bus after a read access (t_{EHQZ}). The programmed bus turnaround delay is inserted between an asynchronous read (muxed or mode D) or write transaction and any other asynchronous /synchronous read or write to or from a static bank. The bank can be the same or different in case of read, in case of write the bank can be different except for muxed or mode D.

In some cases, whatever the programmed BUSTURN values, the bus turnaround delay is fixed as follows:

- The bus turnaround delay is not inserted between two consecutive asynchronous write transfers to the same static memory bank except for modes muxed and D.
- There is a bus turnaround delay of 1 FMC clock cycle between:
 - Two consecutive asynchronous read transfers to the same static memory bank except for modes muxed and D.
 - An asynchronous read to an asynchronous or synchronous write to any static bank or dynamic bank except for modes muxed and D.
 - An asynchronous (modes 1, 2, A, B or C) read and a read from another static bank.
- There is a bus turnaround delay of 2 FMC clock cycle between:
 - Two consecutive synchronous writes (burst or single) to the same bank.
 - A synchronous write (burst or single) access and an asynchronous write or read transfer to or from static memory bank (the bank can be the same or different for the case of read).
 - Two consecutive synchronous reads (burst or single) followed by any synchronous/asynchronous read or write from/to another static memory bank.
- There is a bus turnaround delay of 3 FMC clock cycle between:
 - Two consecutive synchronous writes (burst or single) to different static bank.
 - A synchronous write (burst or single) access and a synchronous read from the same or a different bank.

0000: BUSTURN phase duration = 0 HCLK clock cycle added

...

1111: BUSTURN phase duration = 15 x HCLK clock cycles added (default value after reset)

Bits 15:8 **DATAST[7:0]**: Data-phase duration

These bits are written by software to define the duration of the data phase (refer to [Figure 458](#) to [Figure 470](#)), used in asynchronous accesses:

0000 0000: Reserved

0000 0001: DATAST phase duration = 1 × HCLK clock cycles

0000 0010: DATAST phase duration = 2 × HCLK clock cycles

...

1111 1111: DATAST phase duration = 255 × HCLK clock cycles (default value after reset)

For each memory type and access mode data-phase duration, please refer to the respective figure ([Figure 458](#) to [Figure 470](#)).

Example: Mode1, write access, DATAST=1: Data-phase duration= DATAST+1 = 2 HCLK clock cycles.

Note: In synchronous accesses, this value is don't care.

Bits 7:4 **ADDHLD[3:0]**: Address-hold phase duration

These bits are written by software to define the duration of the *address hold* phase (refer to [Figure 467](#) to [Figure 470](#)), used in mode D or multiplexed accesses:

0000: Reserved

0001: ADDHLD phase duration = 1 × HCLK clock cycle

0010: ADDHLD phase duration = 2 × HCLK clock cycle

...

1111: ADDHLD phase duration = 15 × HCLK clock cycles (default value after reset)

For each access mode address-hold phase duration, please refer to the respective figure ([Figure 467](#) to [Figure 470](#)).

Note: In synchronous accesses, this value is not used, the address hold phase is always 1 memory clock period duration.

Bits 3:0 **ADDSET[3:0]**: Address setup phase duration

These bits are written by software to define the duration of the *address setup* phase (refer to [Figure 458](#) to [Figure 470](#)), used in SRAMs, ROMs and asynchronous NOR Flash and PSRAM accesses:

0000: ADDSET phase duration = 0 × HCLK clock cycle

...

1111: ADDSET phase duration = 15 × HCLK clock cycles (default value after reset)

For each access mode address setup phase duration, please refer to the respective figure (refer to [Figure 458](#) to [Figure 470](#)).

Note: In synchronous accesses, this value is don't care.

In Muxed mode or Mode D, the minimum value for ADDSET is 1.

Note: PSRAMs (CRAMs) have a variable latency due to internal refresh. Therefore these memories issue the NWAIT signal during the whole latency phase to prolong the latency as needed.

With PSRAMs (CRAMs) the filled DATLAT must be set to 0, so that the FMC exits its latency phase soon and starts sampling NWAIT from memory, then starts to read or write when the memory is ready.

This method can be used also with the latest generation of synchronous Flash memories that issue the NWAIT signal, unlike older Flash memories (check the datasheet of the specific Flash memory being used).

SRAM/NOR-Flash write timing registers 1..4 (FMC_BWTR1..4)

Address offset: $0x104 + 8 * (x - 1)$, $x = 1...4$

Reset value: 0x0FFF FFFF

This register contains the control information of each memory bank. It is used for SRAMs, PSRAMs and NOR Flash memories. When the EXTMOD bit is set in the FMC_BCRx register, then this register is active for write access.

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[1:0]		Reserved											BUSTURN[3:0]				DATA[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
rw		rw										rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw				

Bits 15:8 **DATAST[3:0]**: Data-phase duration.

These bits are written by software to define the duration of the data phase (refer to [Figure 458](#) to [Figure 470](#)), used in asynchronous SRAM, PSRAM and NOR Flash memory accesses:

0000 0000: Reserved

0000 0001: DATAST phase duration = 1 × HCLK clock cycles

0000 0010: DATAST phase duration = 2 × HCLK clock cycles

...

1111 1111: DATAST phase duration = 255 × HCLK clock cycles (default value after reset)

Bits 7:4 **ADDHLD[3:0]**: Address-hold phase duration.

These bits are written by software to define the duration of the *address hold* phase (refer to [Figure 467](#) to [Figure 470](#)), used in asynchronous multiplexed accesses:

0000: Reserved

0001: ADDHLD phase duration = 1 × HCLK clock cycle

0010: ADDHLD phase duration = 2 × HCLK clock cycle

...

1111: ADDHLD phase duration = 15 × HCLK clock cycles (default value after reset)

Note: In synchronous NOR Flash accesses, this value is not used, the address hold phase is always 1 Flash clock period duration.

Bits 3:0 **ADDSET[3:0]**: Address setup phase duration.

These bits are written by software to define the duration of the *address setup* phase in HCLK cycles (refer to [Figure 467](#) to [Figure 470](#)), used in asynchronous accesses:

0000: ADDSET phase duration = 0 × HCLK clock cycle

...

1111: ADDSET phase duration = 15 × HCLK clock cycles (default value after reset)

Note: In synchronous NOR Flash and PSRAM accesses, this value is not used, the address setup phase is always 1 Flash clock period duration. In muxed mode, the minimum ADDSET value is 1.

37.6 NAND Flash/PC Card controller

The FMC generates the appropriate signal timings to drive the following types of device:

- 8- and 16-bit NAND Flash memories
- 16-bit PC Card compatible devices

The NAND Flash/PC Card controller can control three external banks, Bank 2, 3 and 4:

- Bank 2 and Bank 3 support NAND Flash devices
- Bank 4 supports PC Card devices.

Each bank is configured through dedicated registers ([Section 37.6.8](#)). The programmable memory parameters include access timings (shown in [Table 289](#)) and ECC configuration.

Table 289. Programmable NAND Flash/PC Card access parameters

Parameter	Function	Access mode	Unit	Min.	Max.
Memory setup time	Number of clock cycles (HCLK) required to set up the address before the command assertion	Read/Write	AHB clock cycle (HCLK)	1	256
Memory wait	Minimum duration (in HCLK clock cycles) of the command assertion	Read/Write	AHB clock cycle (HCLK)	2	255
Memory hold	Number of clock cycles (HCLK) during which the address must be held (as well as the data if a write access is performed) after the command de-assertion	Read/Write	AHB clock cycle (HCLK)	1	254
Memory databus high-Z	Number of clock cycles (HCLK) during which the data bus is kept in high-Z state after a write access has started	Write	AHB clock cycle (HCLK)	1	255

37.6.1 External memory interface signals

The following tables list the signals that are typically used to interface NAND Flash memory and PC Card.

Note: The prefix "N" identifies the signals which are active low.

8-bit NAND Flash memory

Table 290. 8-bit NAND Flash

FMC signal name	I/O	Function
A[17]	O	NAND Flash address latch enable (ALE) signal
A[16]	O	NAND Flash command latch enable (CLE) signal
D[7:0]	I/O	8-bit multiplexed, bidirectional address/data bus
NCE[x]	O	Chip Select, x = 2, 3
NOE(= NRE)	O	Output enable (memory signal name: read enable, NRE)
NWE	O	Write enable
NWAIT/INT[3:2]	I	NAND Flash ready/busy input signal to the FMC

Theoretically, there is no capacity limitation as the FMC can manage as many address cycles as needed.

16-bit NAND Flash memory**Table 291. 16-bit NAND Flash**

FMC signal name	I/O	Function
A[17]	O	NAND Flash address latch enable (ALE) signal
A[16]	O	NAND Flash command latch enable (CLE) signal
D[15:0]	I/O	16-bit multiplexed, bidirectional address/data bus
NCE[x]	O	Chip Select, x = 2, 3
NOE(= NRE)	O	Output enable (memory signal name: read enable, NRE)
NWE	O	Write enable
NWAIT/INT[3:2]	I	NAND Flash ready/busy input signal to the FMC

Theoretically, there is no capacity limitation as the FMC can manage as many address cycles as needed.

Table 292. 16-bit PC Card

FMC signal name	I/O	Function
A[10:0]	O	Address bus
NIORD	O	Output enable for I/O space
NIOWR	O	Write enable for I/O space
NREG	O	Register signal indicating if access is in Common or Attribute space
D[15:0]	I/O	Bidirectional databus
NCE4_1	O	Chip Select 1
NCE4_2	O	Chip Select 2 (indicates if access is 16-bit or 8-bit)
NOE	O	Output enable in Common and in Attribute space
NWE	O	Write enable in Common and in Attribute space
NWAIT	I	PC Card wait input signal to the FMC (memory signal name IORDY)
INTR	I	PC Card interrupt to the FMC (only for PC Cards that can generate an interrupt)
CD	I	PC Card presence detection. Active high. If an access is performed to the PC Card banks while CD is low, an AHB error is generated. Refer to Section 37.3: AHB interface

37.6.2 NAND Flash / PC Card supported memories and transactions

Table 293 shows the supported devices, access modes and transactions. Transactions not allowed (or not supported) by the NAND Flash / PC Card controller are shown in gray.

Table 293. Supported memories and transactions

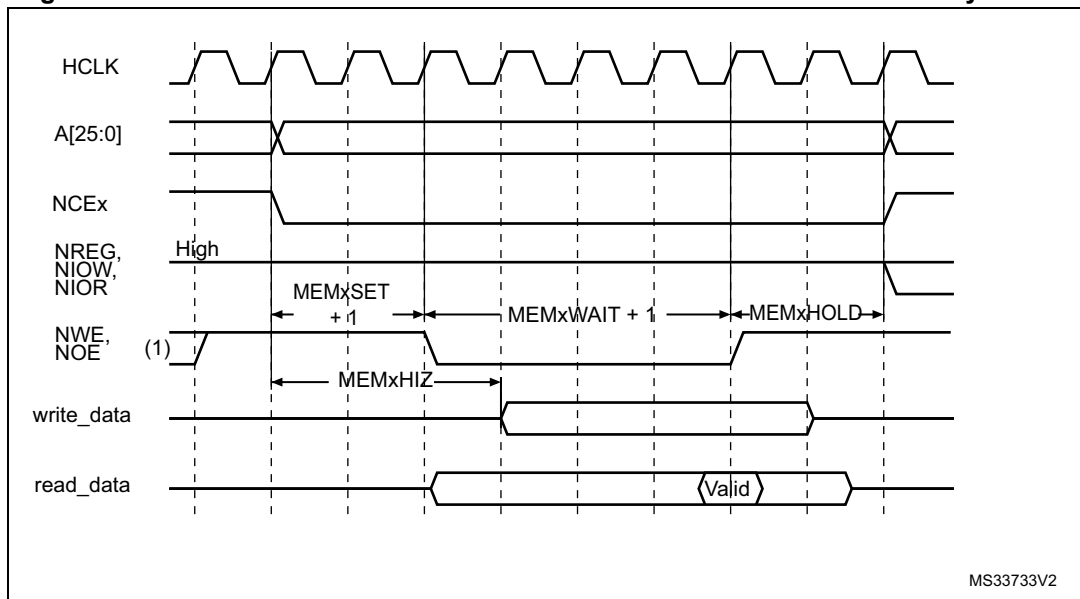
Device	Mode	R/W	AHB data size	Memory data size	Allowed/ not allowed	Comments
NAND 8-bit	Asynchronous	R	8	8	Y	-
	Asynchronous	W	8	8	Y	-
	Asynchronous	R	16	8	Y	Split into 2 FMC accesses
	Asynchronous	W	16	8	Y	Split into 2 FMC accesses
	Asynchronous	R	32	8	Y	Split into 4 FMC accesses
	Asynchronous	W	32	8	Y	Split into 4 FMC accesses
NAND 16-bit	Asynchronous	R	8	16	Y	-
	Asynchronous	W	8	16	N	-
	Asynchronous	R	16	16	Y	-
	Asynchronous	W	16	16	Y	-
	Asynchronous	R	32	16	Y	Split into 2 FMC accesses
	Asynchronous	W	32	16	Y	Split into 2 FMC accesses

37.6.3 Timing diagrams for NAND Flash memory and PC Card

Each PC Card/CompactFlash and NAND Flash memory bank is managed through a set of registers:

- Control register: FMC_PCRx
- Interrupt status register: FMC_SRx
- ECC register: FMC_ECCRx
- Timing register for Common memory space: FMC_PMEMx
- Timing register for Attribute memory space: FMC_PATTx
- Timing register for I/O space: FMC_PIOx

Each timing configuration register contains three parameters used to define number of HCLK cycles for the three phases of any PC Card/CompactFlash or NAND Flash access, plus one parameter that defines the timing for starting driving the data bus when a write access is performed. [Figure 476](#) shows the timing parameter definitions for common memory accesses, knowing that Attribute and I/O (only for PC Card) memory space access timings are similar.

Figure 476. NAND Flash/PC Card controller waveforms for common memory access

1. NOE remains high (inactive) during write accesses. NWE remains high (inactive) during read accesses.
2. For write accesses, the hold phase delay is $(\text{MEMHOLD}) \times \text{HCLK cycles}$, while it is $(\text{MEMHOLD} + 2) \times \text{HCLK cycles}$ for read accesses.

37.6.4 NAND Flash operations

The command latch enable (CLE) and address latch enable (ALE) signals of the NAND Flash memory device are driven by address signals from the FMC controller. This means that to send a command or an address to the NAND Flash memory, the CPU has to perform a write to a specific address in its memory space.

A typical page read operation from the NAND Flash device requires the following steps:

3. Program and enable the corresponding memory bank by configuring the FMC_PCRx and FMC_PMEMx (and for some devices, FMC_PATTx, see [Section 37.6.5: NAND Flash prewait functionality](#)) registers according to the characteristics of the NAND Flash memory (PWID bits for the data bus width of the NAND Flash, PTYP = 1, PWAITEN = 0 or 1 as needed, see [Section 37.4.2: NAND Flash memory/PC Card address mapping](#) for timing configuration).
4. The CPU performs a byte write to the common memory space, with data byte equal to one Flash command byte (for example 0x00 for Samsung NAND Flash devices). The LE input of the NAND Flash memory is active during the write strobe (low pulse on NWE), thus the written byte is interpreted as a command by the NAND Flash memory. Once the command is latched by the memory device, it does not need to be written again for the following page read operations.
5. The CPU can send the start address (STARTAD) for a read operation by writing four byte (or three for smaller capacity devices), STARTAD[7:0], STARTAD[16:9], STARTAD[24:17] and finally STARTAD[25] (for 64 Mb x 8 bit NAND Flash memories) in the common memory or attribute space. The ALE input of the NAND Flash device is active during the write strobe (low pulse on NWE), thus the written byte are interpreted as the start address for read operations. Using the attribute memory space makes it possible to use a different timing configuration of the FMC, which can be used to

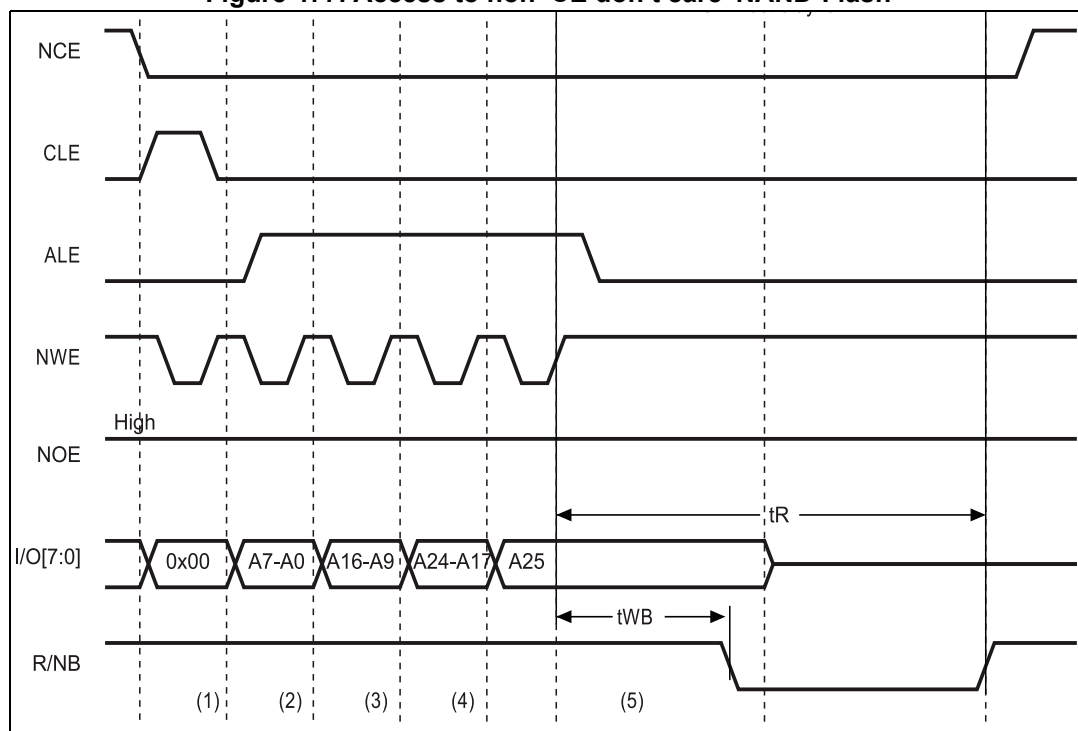
implement the prewait functionality needed by some NAND Flash memories (see details in [Section 37.6.5: NAND Flash prewait functionality](#)).

6. The controller waits for the NAND Flash memory to be ready (R/NB signal high), before starting a new access to the same or another memory bank. While waiting, the controller holds the NCE signal active (low).
7. The CPU can then perform byte read operations from the common memory space to read the NAND Flash page (data field + Spare field) byte by byte.
8. The next NAND Flash page can be read without any CPU command or address write operation. This can be done in three different ways:
 - by simply performing the operation described in step 5
 - a new random address can be accessed by restarting the operation at step 3
 - a new command can be sent to the NAND Flash device by restarting at step 2

37.6.5 NAND Flash prewait functionality

Some NAND Flash devices require that, after writing the last part of the address, the controller waits for the R/NB signal to go low. (see [Figure 457](#)).

Figure 477. Access to non 'CE don't care' NAND-Flash



1. CPU wrote byte 0x00 at address 0x7001 0000.
2. CPU wrote byte A7~A0 at address 0x7002 0000.
3. CPU wrote byte A16~A9 at address 0x7002 0000.
4. CPU wrote byte A24~A17 at address 0x7002 0000.
5. CPU wrote byte A25 at address 0x7802 0000: FMC performs a write access using FMC_PATT2 timing definition, where $ATTHOLD \geq 7$ (providing that $(7+1) \times HCLK = 112 \text{ ns} > t_{WB} \text{ max}$). This guarantees that NCE remains low until R/NB goes low and high again (only requested for NAND Flash memories where NCE is not don't care).

When this functionality is required, it can be ensured by programming the MEMHOLD value to meet the t_{WB} timing. However CPU read accesses to the NAND Flash memory has a hold delay of $(MEMHOLD + 2) \times HCLK$ cycles, while CPU write accesses have a hold delay of $(MEMHOLD) \times HCLK$ cycles.

To cope with this timing constraint, the attribute memory space can be used by programming its timing register with an ATTHOLD value that meets the t_{WB} timing, and by keeping the MEMHOLD value at its minimum value. The CPU must then use the common memory space for all NAND Flash read and write accesses, except when writing the last address byte to the NAND Flash device, where the CPU must write to the attribute memory space.

37.6.6 Computation of the error correction code (ECC) in NAND Flash memory

The FMC PC Card controller includes two error correction code computation hardware blocks, one per memory bank. They reduce the host CPU workload when processing the ECC by software.

These two ECC blocks are identical and associated with Bank 2 and Bank 3. As a consequence, no hardware ECC computation is available for memories connected to Bank 4.

The ECC algorithm implemented in the FMC can perform 1-bit error correction and 2-bit error detection per 256, 512, 1 024, 2 048, 4 096 or 8 192 byte read or written from/to the NAND Flash memory. It is based on the Hamming coding algorithm and consists in calculating the row and column parity.

The ECC modules monitor the NAND Flash data bus and read/write signals (NCE and NWE) each time the NAND Flash memory bank is active.

The ECC operates as follows:

- When accessing NAND Flash memory bank 2 or bank 3, the data present on the D[15:0] bus is latched and used for ECC computation.
- When accessing any other address in NAND Flash memory, the ECC logic is idle, and does not perform any operation. As a result, write operations to define commands or addresses to the NAND Flash memory are not taken into account for ECC computation.

Once the desired number of byte has been read/written from/to the NAND Flash memory by the host CPU, the FMC_ECCR2/3 registers must be read to retrieve the computed value. Once read, they should be cleared by resetting the ECCEN bit to '0'. To compute a new data block, the ECCEN bit must be set to one in the FMC_PCR2/3 registers.

To perform an ECC computation:

1. Enable the ECCEN bit in the FMC_PCR2/3 register.
2. Write data to the NAND Flash memory page. While the NAND page is written, the ECC block computes the ECC value.
3. Read the ECC value available in the FMC_ECCR2/3 register and store it in a variable.
4. Clear the ECCEN bit and then enable it in the FMC_PCR2/3 register before reading back the written data from the NAND page. While the NAND page is read, the ECC block computes the ECC value.
5. Read the new ECC value available in the FMC_ECCR2/3 register.
6. If the two ECC values are the same, no correction is required, otherwise there is an ECC error and the software correction routine returns information on whether the error can be corrected or not.

37.6.7 PC Card/CompactFlash operations

Address spaces and memory accesses

The FMC supports CompactFlash devices and PC Cards in Memory mode and I/O mode (True IDE mode is not supported).

The CompactFlash and PC Cards are made of 3 memory spaces:

- Common Memory space
- Attribute space
- I/O Memory space

The nCE2 and nCE1 pins (FMC_NCE4_2 and FMC_NCE4_1 respectively) select the card and indicate whether a byte or a word operation is being performed: nCE2 accesses the odd byte on D15-8 and nCE1 accesses the even byte on D7-0 if A0=0 or the odd byte on D7-0 if A0=1. The full word is accessed on D15-0 if both nCE2 and nCE1 are low.

The memory space is selected by asserting low nOE for read accesses or nWE for write accesses, combined with the low assertion of nCE2/nCE1 and nREG.

- If pin nREG=1 during the memory access, the common memory space is selected
- If pin nREG=0 during the memory access, the attribute memory space is selected

The I/O space is selected by asserting nIORD space for read accesses or nIOWR for write accesses [instead of nOE/nWE for memory space], combined with nCE2/nCE1. Note that nREG must also be asserted low when accessing I/O space.

Three type of accesses are allowed for a 16-bit PC Card:

- Accesses to Common Memory space for data storage can be either 8-bit accesses at even addresses or 16-bit AHB accesses.
Note that 8-bit accesses at odd addresses are not supported and nCE2 will not be driven low. A 32-bit AHB request is translated into two 16-bit memory accesses.
- Accesses to Attribute Memory space where the PC Card stores configuration information are limited to 8-bit AHB accesses at even addresses.

Note that a 16-bit AHB access will be converted into a single 8-bit memory transfer: nCE1 will be asserted low, nCE2 will be asserted high and only the even byte on D7-D0 will be valid. Instead a 32-bit AHB access will be converted into two 8-bit memory

transfers at even addresses: nCE1 will be asserted low, NCE2 will be asserted high and only the even byte will be valid.

- Accesses to I/O space can be either 8-bit or 16 bit AHB accesses.

Table 294. 16-bit PC-Card signals and access type

nCE2	nCE1	nREG	nOE/nWE	nIORD	A10	A9	A7-1	A0	Space	Access type	Allowed/not Allowed
1	0	1	0	1	X	X	X-X	X	Common Memory Space	Read/Write byte on D7-D0	YES
0	1	1	0	1	X	X	X-X	X		Read/Write byte on D15-D8	Not supported
0	0	1	0	1	X	X	X-X	0		Read/Write word on D15-D0	YES
X	0	0	0	1	0	1	X-X	0	Attribute Space	Read or Write Configuration Registers	YES
X	0	0	0	1	0	0	X-X	0		Read or Write CIS (Card Information Structure)	YES
1	0	0	0	1	X	X	X-X	1	Attribute Space	Invalid Read or Write (odd address)	YES
0	1	0	0	1	X	X	X-X	x		Invalid Read or Write (odd address)	YES
1	0	0	1	0	X	X	X-X	0	I/O space	Read Even Byte on D7-0	YES
1	0	0	1	0	X	X	X-X	1		Read Odd Byte on D7-0	YES
1	0	0	1	0	X	X	X-X	0		Write Even Byte on D7-0	YES
1	0	0	1	0	X	X	X-X	1		Write Odd Byte on D7-0	YES
0	0	0	1	0	X	X	X-X	0		Read Word on D15-0	YES
0	0	0	1	0	X	X	X-X	0		Write word on D15-0	YES
0	1	0	1	0	X	X	X-X	X		Read Odd Byte on D15-8	Not supported
0	1	0	1	0	X	X	X-X	X		Write Odd Byte on D15-8	Not supported

FMC Bank 4 gives access to those 3 memory spaces as described in [Section 37.4.2: NAND Flash memory/PC Card address mapping](#) and [Table 256: NAND/PC Card memory mapping and timing registers](#).

Wait feature

The CompactFlash or PC Card may request the FMC to extend the length of the access phase programmed by MEMWAITx/ATTWAITx/IOWAITx bits, asserting the nWAIT signal after nOE/nWE or nIORD/nIOWR activation if the wait feature is enabled through the PWAITEN bit in the FMC_PCRx register. To detect correctly the nWAIT assertion, the MEMWAITx/ATTWAITx/IOWAITx bits must be programmed as follows:

$$xxWAITx \geq 4 + \frac{\text{max_wait_assertion_time}}{HCLK}$$

where max_wait_assertion_time is the maximum time taken by nWAIT to go low once nOE/nWE or nIORD/nIOWR is low.

After WAIT de-assertion, the FMC extends the WAIT phase for 4 HCLK clock cycles.

37.6.8 NAND Flash/PC Card controller registers

PC Card/NAND Flash control registers 2..4 (FMC_PCR2..4)

Address offset: $0x40 + 0x20 * (x - 1)$, $x = 2..4$

Reset value: 0x0000 0018

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												ECCPS[2:0]			TAR[3:0]				TCLR[3:0]				Reserved		ECCEN		PWID[1:0]		PTYP		PBKEN		PWA/ITEN		Reserved	
																									r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		

Bits 31:20 Reserved, must be kept at reset value

Bits 19:17 **ECCPS[2:0]**: ECC page size.

Defines the page size for the extended ECC:

000: 256 byte
 001: 512 byte
 010: 1024 byte
 011: 2048 byte
 100: 4096 byte
 101: 8192 byte

Bits 16:13 **TAR[3:0]**: ALE to RE delay.

Sets time from ALE low to RE low in number of AHB clock cycles (HCLK).

Time is: $t_{ar} = (TAR + SET + 2) \times THCLK$ where THCLK is the HCLK clock period

0000: 1 HCLK cycle (default)
 1111: 16 HCLK cycles

Note: SET is MEMSET or ATTSET according to the addressed space.

Bits 12:9 **TCLR[3:0]**: CLE to RE delay.

Sets time from CLE low to RE low in number of AHB clock cycles (HCLK).

Time is: $t_{clr} = (TCLR + SET + 2) \times THCLK$ where THCLK is the HCLK clock period

0000: 1 HCLK cycle (default)
 1111: 16 HCLK cycles

Note: SET is MEMSET or ATTSET according to the addressed space.

Bits 8:7 Reserved, must be kept at reset value

Bit 6 **ECCEN**: ECC computation logic enable bit

0: ECC logic is disabled and reset (default after reset),
 1: ECC logic is enabled.

Bits 5:4 **PWID[1:0]**: Data bus width.

Defines the external memory device width.

00: 8 bits
 01: 16 bits (default after reset). This value is mandatory for PC Cards.
 10: reserved, do not use
 11: reserved, do not use

Bit 3 **PTYP**: Memory type.

Defines the type of device attached to the corresponding memory bank:

- 0: PC Card, CompactFlash, CF+ or PCMCIA
- 1: NAND Flash (default after reset)

Bit 2 **PBKEN**: PC Card/NAND Flash memory bank enable bit.

Enables the memory bank. Accessing a disabled memory bank causes an ERROR on AHB bus

- 0: Corresponding memory bank is disabled (default after reset)
- 1: Corresponding memory bank is enabled

Bit 1 **PWAITEN**: Wait feature enable bit.

Enables the Wait feature for the PC Card/NAND Flash memory bank:

- 0: disabled
- 1: enabled

Note: For a PC Card, when the wait feature is enabled, the MEMWAITx/ATTWAITx/IOWAITx bits must be programmed to a value as follows:

$$xxWAITx \geq 4 + \frac{\text{max_wait_assertion_time}}{HCLK}$$

Where max_wait_assertion_time is the maximum time taken by NWAIT to go low once nOE/nWE or nIORD/nIOWR is low.

Bit 0 Reserved.

FIFO status and interrupt register 2..4 (FMC_SR2..4)

Address offset: $0x44 + 0x20 * (x-1)$, $x = 2..4$

Reset value: 0x0000 0040

This register contains information about the FIFO status and interrupt. The FMC features a FIFO that is used when writing to memories to transfer up to 16 words of data from the AHB.

This is used to quickly write to the FIFO and free the AHB for transactions to peripherals other than the FMC, while the FMC is draining its FIFO into the memory. One of these register bits indicates the status of the FIFO, for ECC purposes.

The ECC is calculated while the data are written to the memory. To read the correct ECC, the software must consequently wait until the FIFO is empty.

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																								FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS	
																								r	rW	rW	rW	rW	rW	rW	

Bits 31:7 Reserved, must be kept at reset value

Bit 6 **FEMPT**: FIFO empty.

Read-only bit that provides the status of the FIFO

- 0: FIFO not empty
- 1: FIFO empty

Bit 5 **IFEN**: Interrupt falling edge detection enable bit

- 0: Interrupt falling edge detection request disabled
- 1: Interrupt falling edge detection request enabled

- Bit 4 **ILEN**: Interrupt high-level detection enable bit
 0: Interrupt high-level detection request disabled
 1: Interrupt high-level detection request enabled
- Bit 3 **IREN**: Interrupt rising edge detection enable bit
 0: Interrupt rising edge detection request disabled
 1: Interrupt rising edge detection request enabled
- Bit 2 **IFS**: Interrupt falling edge status
 The flag is set by hardware and reset by software.
 0: No interrupt falling edge occurred
 1: Interrupt falling edge occurred

Note: This bit is set by programming it to 1 by software.

- Bit 1 **ILS**: Interrupt high-level status
 The flag is set by hardware and reset by software.
 0: No Interrupt high-level occurred
 1: Interrupt high-level occurred
- Bit 0 **IRS**: Interrupt rising edge status
 The flag is set by hardware and reset by software.
 0: No interrupt rising edge occurred
 1: Interrupt rising edge occurred

Note: This bit is set by programming it to 1 by software.

Common memory space timing register 2..4 (FMC_PMEM2..4)

Address offset: Address: $0x48 + 0x20 * (x - 1)$, $x = 2..4$

Reset value: 0xFCFC FCFC

Each FMC_PMEMx ($x = 2..4$) read/write register contains the timing information for PC Card or NAND Flash memory bank x. This information is used to access either the common memory space of the 16-bit PC Card/CompactFlash, or the NAND Flash for command, address write access and data read/write access.

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
MEMHIZ[7:0]								MEMHOLD[7:0]								MEMWAIT[7:0]								MEMSET[7:0]							
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	rw	rw

Bits 31:24 **MEMHIZ[7:0]**: Common memory x data bus Hi-Z time

Defines the number of HCLK clock cycles during which the data bus is kept Hi-Z after the start of a PC Card/NAND Flash write access to common memory space on socket x. This is only valid for write transactions:

- 0000 0000: 1 HCLK cycle
- 1111 1110: 255 HCLK cycles
- 1111 1111: Reserved.

Bits 23:16 **MEMHOLD[7:0]**: Common memory x hold time

For NAND Flash read accesses to the common memory space, these bits define the number of (HCLK+2) clock cycles during which the address is held after the command is deasserted (NWE, NOE).

For NAND Flash write accesses to the common memory space, these bits define the number of HCLK clock cycles during which the data are held after the command is deasserted (NWE, NOE).

0000 0000: reserved

0000 0001: 1 HCLK cycle for write accesses, 3 HCLK cycles for read accesses

1111 1110: 254 HCLK cycle for write accesses, 256 HCLK cycles for read accesses

1111 1111: Reserved.

Bits 15:8 **MEMWAIT[7:0]**: Common memory x wait time

Defines the minimum number of HCLK (+1) clock cycles to assert the command (NWE, NOE), for PC Card/NAND Flash read or write access to common memory space on socket x. The duration of command assertion is extended if the wait signal (NWAIT) is active (low) at the end of the programmed value of HCLK:

0000 0000: reserved

0000 0001: 2 HCLK cycles (+ wait cycle introduced by deasserting NWAIT)

1111 1110: 255 HCLK cycles (+ wait cycle introduced by deasserting NWAIT)

1111 1111: Reserved

Bits 7:0 **MEMSET[7:0]**: Common memory x setup time

Defines the number of HCLK (+1) clock cycles to set up the address before the command assertion (NWE, NOE), for PC Card/NAND Flash read or write access to common memory space on socket x:

0000 0000: 1 HCLK cycle

1111 1110: 255 HCLK cycles

1111 1111: Reserved.

Attribute memory space timing registers 2..4 (FMC_PATT2..4)

Address offset: $0x4C + 0x20 * (x - 1)$, $x = 2..4$

Reset value: 0xFCFC FCFC

Each FMC_PATTx ($x = 2..4$) read/write register contains the timing information for PC Card/CompactFlash or NAND Flash memory bank x. It is used for 8-bit accesses to the attribute memory space of the PC Card/CompactFlash or to access the NAND Flash for the last address write access if the timing must differ from that of previous accesses (for Ready/Busy management, refer to [Section 37.6.5: NAND Flash prewait functionality](#)).

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
ATTHIZ[7:0]								ATTHOLD[7:0]								ATTWAIT[7:0]								ATTSET[7:0]							
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	rW	

Bits 31:24 **ATTHIZ[7:0]**: Attribute memory x data bus Hi-Z time

Defines the number of HCLK clock cycles during which the data bus is kept in Hi-Z after the start of a PC CARD/NAND Flash write access to attribute memory space on socket x. Only valid for write transaction:

0000 0000: 0 HCLK cycle

1111 1110: 255 HCLK cycles

1111 1111: Reserved.

Bits 23:16 ATTHOLD[7:0]: Attribute memory x hold time

For PC Card/NAND Flash read accesses to attribute memory space on socket x, these bits define the number of HCLK clock cycles (HCLK +2) clock cycles during which the address is held after the command is deasserted (NWE, NOE).

For PC Card/NAND Flash write accesses to attribute memory space on socket x, these bits define the number of HCLK clock cycles during which the data are held after the command is deasserted (NWE, NOE).

0000 0000: Reserved

0000 0001: 1 HCLK cycle for write access, 3 HCLK cycles for read accesses

1111 1110: 254 HCLK cycle for write access, 256 HCLK cycles for read accesses

1111 1111: Reserved.

Bits 15:8 ATWAIT[7:0]: Attribute memory x wait time

Defines the minimum number of HCLK (+1) clock cycles to assert the command (NWE, NOE), for PC Card/NAND Flash read or write access to attribute memory space on socket x. The duration for command assertion is extended if the wait signal (NWAIT) is active (low) at the end of the programmed value of HCLK:

0000 0000: reserved

0000 0001: 2 HCLK cycles (+ wait cycle introduced by deassertion of NWAIT)

1111 1110: 255 HCLK cycles (+ wait cycle introduced by the card deasserting NWAIT)

1111 1111: Reserved

Bits 7:0 ATTSET[7:0]: Attribute memory x setup time

Defines the number of HCLK (+1) clock cycles to set up address before the command assertion (NWE, NOE), for PC CARD/NAND Flash read or write access to attribute memory space on socket x:

0000 0000: 1 HCLK cycle

1111 1110: 255 HCLK cycles

1111 1111: Reserved

I/O space timing register 4 (FMC_PIO4)

Address offset: 0xB0

Reset value: 0xFCFCFCFC

The FMC_PIO4 read/write registers contain the timing information used to access the I/O space of the 16-bit PC Card/CompactFlash.

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
IOHIZ[7:0]								IOHOLD[7:0]								IOWAIT[7:0]								IOSET[7:0]							
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	rW	rW

Bits 31:24 IOHIZ[7:0]: I/O x data bus Hi-Z time

Defines the number of HCLK clock cycles during which the data bus is kept in Hi-Z after the start of a PC Card write access to I/O space on socket x. Only valid for write transaction:

0000 0000: 0 HCLK cycle

1111 1111: 255 HCLK cycles

Bits 23:16 IOHOLD[7:0]: I/O x hold time

Defines the number of HCLK clock cycles during which the address is held (and data for write access) after the command deassertion (NWE, NOE), for PC Card read or write access to I/O space on socket x:

0000 0000: reserved

0000 0001: 1 HCLK cycle

1111 1111: 255 HCLK cycles

Bits 15:8 IOWAIT[7:0]: I/O x wait time

Defines the minimum number of HCLK (+1) clock cycles to assert the command (SMNWE, SMNOE), for PC Card read or write access to I/O space on socket x. The duration for command assertion is extended if the wait signal (NWAIT) is active (low) at the end of the programmed value of HCLK:

0000 0000: reserved, do not use this value

0000 0001: 2 HCLK cycles (+ wait cycle introduced by deassertion of NWAIT)

1111 1111: 256 HCLK cycles (+ wait cycle introduced by the Card deasserting NWAIT)

Bits 7:0 IOSET[7:0]: I/O x setup time

Defines the number of HCLK (+1) clock cycles to set up the address before the command assertion (NWE, NOE), for PC Card read or write access to I/O space on socket x:

0000 0000: 1 HCLK cycle

1111 1111: 256 HCLK cycles

ECC result registers 2/3 (FMC_ECCR2/3)

Address offset: $0x54 + 0x20 * (x - 1)$, $x = 2$ or 3

Reset value: 0x0000 0000

These registers contain the current error correction code value computed by the ECC computation modules of the FMC controller (one module per NAND Flash memory bank). When the CPU reads the data from a NAND Flash memory page at the correct address (refer to [Section 37.6.6: Computation of the error correction code \(ECC\) in NAND Flash memory](#)), the data read/written from/to the NAND Flash memory are processed automatically by the ECC computation module. When X byte have been read (according to the ECCPS field in the FMC_PCRx registers), the CPU must read the computed ECC value from the FMC_ECCx registers. It then verifies if these computed parity data are the same as the parity value recorded in the spare area, to determine whether a page is valid, and, to correct it otherwise. The FMC_ECCRx registers should be cleared after being read by setting the ECCEN bit to '0'. To compute a new data block, the ECCEN bit must be set to '1'.

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
ECC[31:0]																															
r																															

Bits 31:0 **ECC[31:0]**: ECC result

This field contains the value computed by the ECC computation logic. [Table 295](#) describes the contents of these bit fields.

Table 295. ECC result relevant bits

ECCPS[2:0]	Page size in byte	ECC bits
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

37.7 SDRAM controller

37.7.1 SDRAM controller main features

The main features of the SDRAM controller are the following:

- Two SDRAM banks with independent configuration
- 8-bit, 16-bit, 32-bit data bus width
- 13-bits Address Row, 11-bits Address Column, 4 internal banks: 4x16Mx32bit (256 MB), 4x16Mx16bit (128 MB), 4x16Mx8bit (64 MB)
- Word, half-word, byte access
- SDRAM clock can be HCLK/2 or HCLK/3
- Automatic row and bank boundary management
- Multibank ping-pong access
- Programmable timing parameters
- Automatic Refresh operation with programmable Refresh rate
- Self-refresh mode
- Power-down mode
- SDRAM power-up initialization by software
- CAS latency of 1,2,3
- Cacheable Read FIFO with depth of 6 lines x32-bit (6 x14-bit address tag)

37.7.2 SDRAM External memory interface signals

At startup, the SDRAM I/O pins used to interface the FMC SDRAM controller with the external SDRAM devices must be configured by the user application. The SDRAM controller I/O pins which are not used by the application, can be used for other purposes.

Table 296. SDRAM signals

SDRAM signal	I/O type	Description	Alternate function
SDCLK	O	SDRAM clock	
SDCKE[1:0]	O	SDCKE0: SDRAM Bank 1 Clock Enable SDCKE1: SDRAM Bank 2 Clock Enable	
SDNE[1:0]	O	SDNE0: SDRAM Bank 1 Chip Enable SDNE1: SDRAM Bank 2 Chip Enable	
A[12:0]	O	Address	FMC_A[12:0]
D[31:0]	I/O	Bidirectional data bus	FMC_D[31:0]
BA[1:0]	O	Bank Address	FMC_A[15:14]
NRAS	O	Row Address Strobe	
NCAS	O	Column Address Strobe	
SDNWE	O	Write Enable	
NBL[3:0]	O	Output Byte Mask for write accesses (memory signal name: DQM[3:0])	FMC_NBL[3:0]

37.7.3 SDRAM controller functional description

All SDRAM controller outputs (signals, address and data) change on the falling edge of the memory clock (FMC_SDCLK).

SDRAM initialization

The initialization sequence is managed by software. If the two banks are used, the initialization sequence must be generated simultaneously to Bank 1 and Bank 2 by setting the Target Bank bits CTB1 and CTB2 in the FMC_SDCMR register:

1. Program the memory device features into the FMC_SDCRx register. The SDRAM clock frequency, RBURST and RPIPE must be programmed in the FMC_SDCR1 register.
2. Program the memory device timing into the FMC_SDTRx register. The TRP and TRC timings must be programmed in the FMC_SDTR1 register.
3. Set MODE bits to '001' and configure the Target Bank bits (CTB1 and/or CTB2) in the FMC_SDCMR register to start delivering the clock to the memory (SDCKE is driven high).
4. Wait during the prescribed delay period. Typical delay is around 100 μ s (refer to the SDRAM datasheet for the required delay after power-up).
5. Set MODE bits to '010' and configure the Target Bank bits (CTB1 and/or CTB2) in the FMC_SDCMR register to issue a "Precharge All" command.
6. Set MODE bits to '011', and configure the Target Bank bits (CTB1 and/or CTB2) as well as the number of consecutive Auto-refresh commands (NRFS) in the FMC_SDCMR register. Refer to the SDRAM datasheet for the number of Auto-refresh commands that should be issued. Typical number is 8.
7. Configure the MRD field according to your SDRAM device, set the MODE bits to '100', and configure the Target Bank bits (CTB1 and/or CTB2) in the FMC_SDCMR register to issue a "Load Mode Register" command in order to program the SDRAM. In particular:
 - a) The CAS latency must be selected following configured value in FMC_SDCR1/2 registers
 - b) The Burst Length (BL) of 1 must be selected by configuring the M[2:0] bits to 000 in the mode register (refer to the SDRAM datasheet). If the Mode Register is not the same for both SDRAM banks, this step has to be repeated twice, once for each bank, and the Target Bank bits set accordingly.
8. Program the refresh rate in the FMC_SDRTR register
The refresh rate corresponds to the delay between refresh cycles. Its value must be adapted to SDRAM devices.
9. For mobile SDRAM devices, to program the extended mode register it should be done once the SDRAM device is initialized: First, a dummy read access should be performed while BA1=1 and BA=0 (refer to SDRAM address mapping section for BA[1:0] address mapping) in order to select the extended mode register instead of Load mode register and then program the needed value.

At this stage the SDRAM device is ready to accept commands. If a system reset occurs during an ongoing SDRAM access, the data bus might still be driven by the SDRAM device. Therefore the SDRAM device must be first reinitialized after reset before issuing any new access by the NOR Flash/PSRAM/SRAM or NAND Flash/PC Card controller.

Note: *If two SDRAM devices are connected to the FMC, all the accesses performed at the same time to both devices by the Command Mode register (Load Mode Register and Self-refresh*

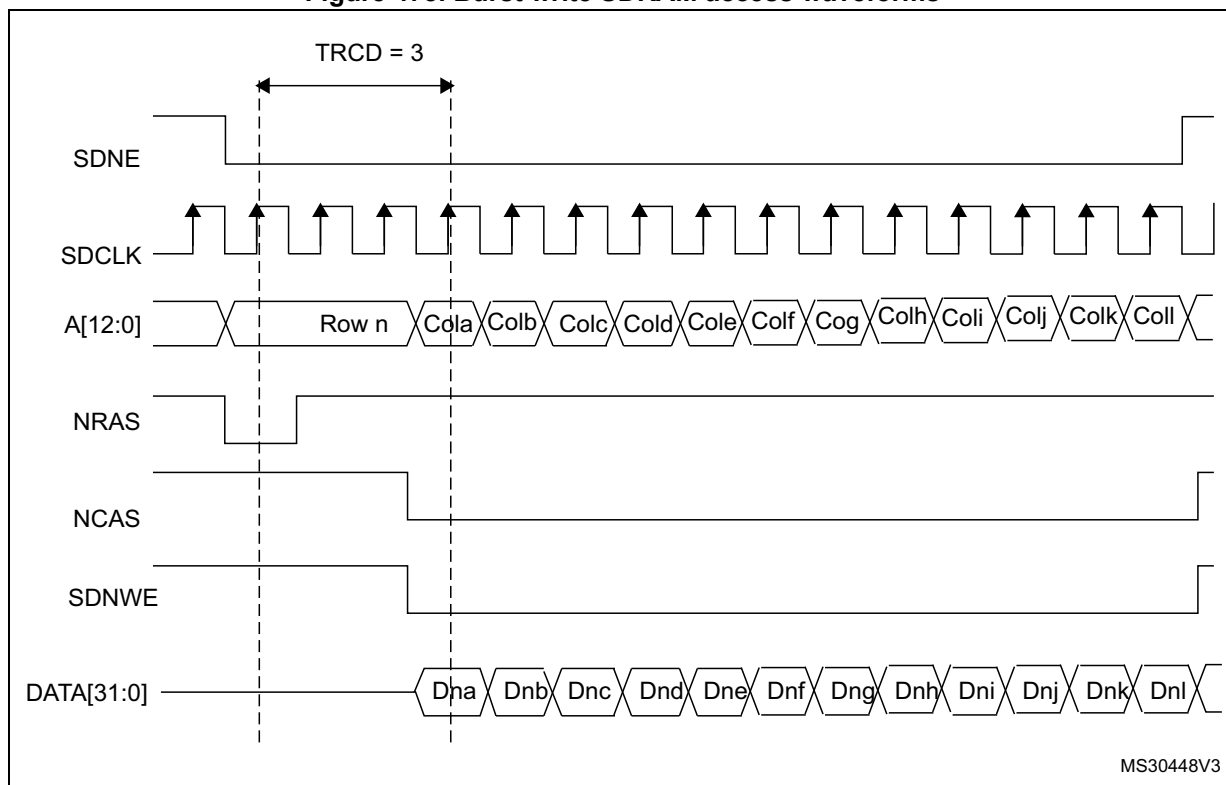
commands) are issued using the timing parameters configured for SDRAM Bank 1 (TMRD, TRAS and TXSR timings) in the FMC_SDTR1 register.

SDRAM controller write cycle

The SDRAM controller accepts single and burst write requests and translates them into single memory accesses. In both cases, the SDRAM controller keeps track of the active row for each bank to be able to perform consecutive write accesses to different banks (Multibank ping-pong access).

Before performing any write access, the SDRAM bank write protection must be disabled by clearing the WP bit in the FMC_SDCRx register.

Figure 478. Burst write SDRAM access waveforms



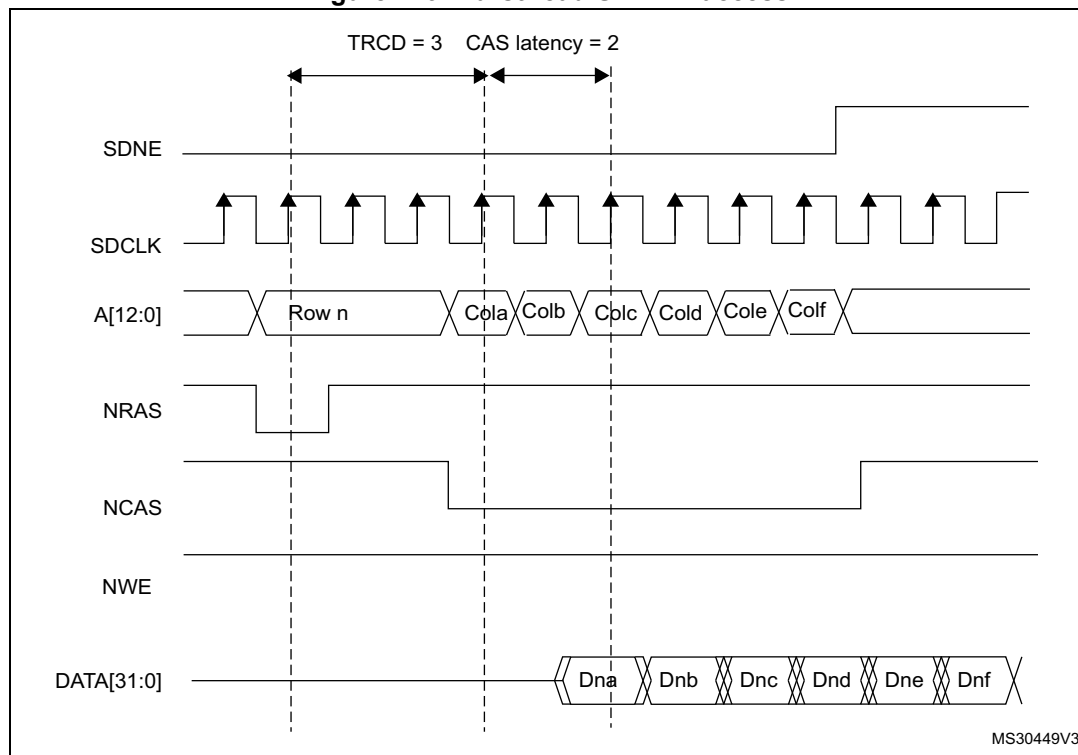
The SDRAM controller always checks the next access.

- If the next access is in the same row or in another active row, the write operation is carried out,
- if the next access targets another row (not active), the SDRAM controller generates a precharge command, activates the new row and initiates a write command.

SDRAM controller read cycle

The SDRAM controller accepts single and burst read requests and translates them into single memory accesses. In both cases, the SDRAM controller keeps track of the active row in each bank to be able to perform consecutive read accesses in different banks (Multibank ping-pong access).

Figure 479. Burst read SDRAM access



The FMC SDRAM controller features a Cacheable read FIFO (6 lines x 32 bits). It is used to store data read in advance during the CAS latency period and during the RPIPE delay. The following formula is applied:

$$\text{Number of anticipated data} = \text{CAS latency} + 1 + (\text{RPIPE delay}) / 2$$

The RBURST bit must be set in the FMC_SDCR1 register to anticipate the next read access.

Example:

- CAS latency = 3, RPIPE delay = 0: 4 data (not committed) are stored in the FIFO.
- CAS latency = 3, RPIPE delay = 2: 5 data (not committed) are stored in the FIFO.

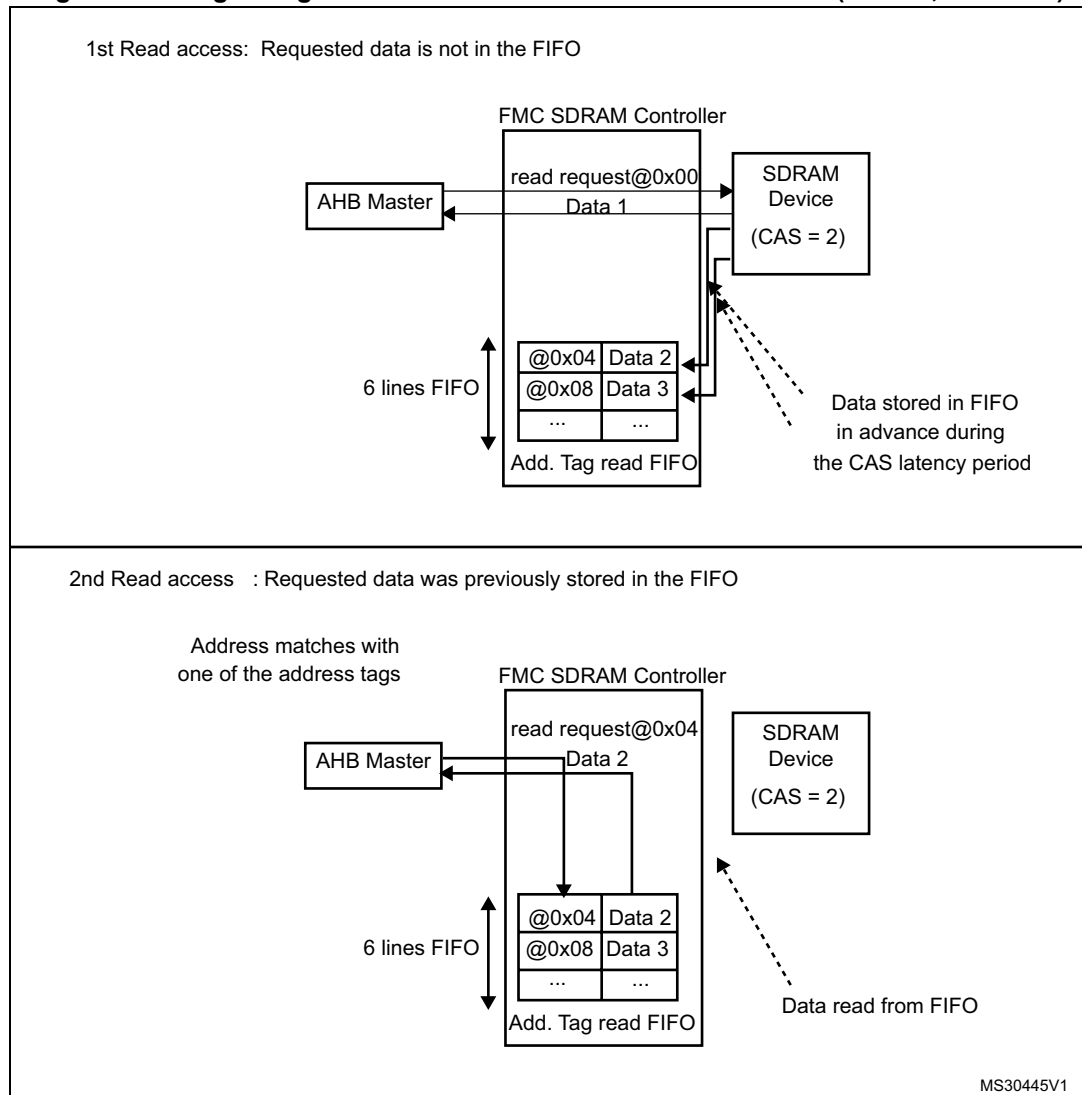
The read FIFO features a 14-bit address tag to each line to identify its content: 11 bits for the column address, 2 bits to select the internal bank and the active row, and 1 bit to select the SDRAM device

When the end of the row is reached in advance during an AHB burst read, the data read in advance (not committed) are not stored in the read FIFO. For single read access, data are correctly stored in the FIFO.

Each time a read request occurs, the SDRAM controller checks:

- If the address matches one of the address tags, data are directly read from the FIFO and the corresponding address tag/ line content is cleared and the remaining data in the FIFO are compacted to avoid empty lines.
- Otherwise, a new read command is issued to the memory and the FIFO is updated with new data. If the FIFO is full, the older data are lost.

Figure 480. Logic diagram of Read access with RBURST bit set (CAS=2, RPIPE=0)



During a write access or a Precharge command, the read FIFO is flushed and ready to be filled with new data.

After the first read request, if the current access was not performed to a row boundary, the SDRAM controller anticipates the next read access during the CAS latency period and the RPIPE delay (if configured). This is done by incrementing the memory address. The following condition must be met:

- RBURST control bit should be set to '1' in the FMC_SDCR1 register.

The address management depends on the next AHB request:

- Next AHB request is sequential (AHB Burst)
In this case, the SDRAM controller increments the address.
- Next AHB request is not sequential
 - If the new read request targets the same row or another active row, the new address is passed to the memory and the master is stalled for the CAS latency period, waiting for the new data from memory.
 - If the new read request does not target an active row, the SDRAM controller generates a Precharge command, activates the new row, and initiates a read command.

If the RURST is reset, the read FIFO is not used.

Row and bank boundary management

When a read or write access crosses a row boundary, if the next read or write access is sequential and the current access was performed to a row boundary, the SDRAM controller executes the following operations:

1. Precharge of the active row,
2. Activation of the new row
3. Start of a read/write command.

At a row boundary, the automatic activation of the next row is supported for all columns and data bus width configurations.

If necessary, the SDRAM controller inserts additional clock cycles between the following commands:

- Between Precharge and Active commands to match TRP parameter (only if the next access is in a different row in the same bank),
- Between Active and Read commands to match the TRCD parameter.

These parameters are defined into the FMC_SDTRx register.

Refer to [Figure 481](#) and [Figure 482](#) for read and burst write access crossing a row boundary.

Figure 481. Read access crossing row boundary

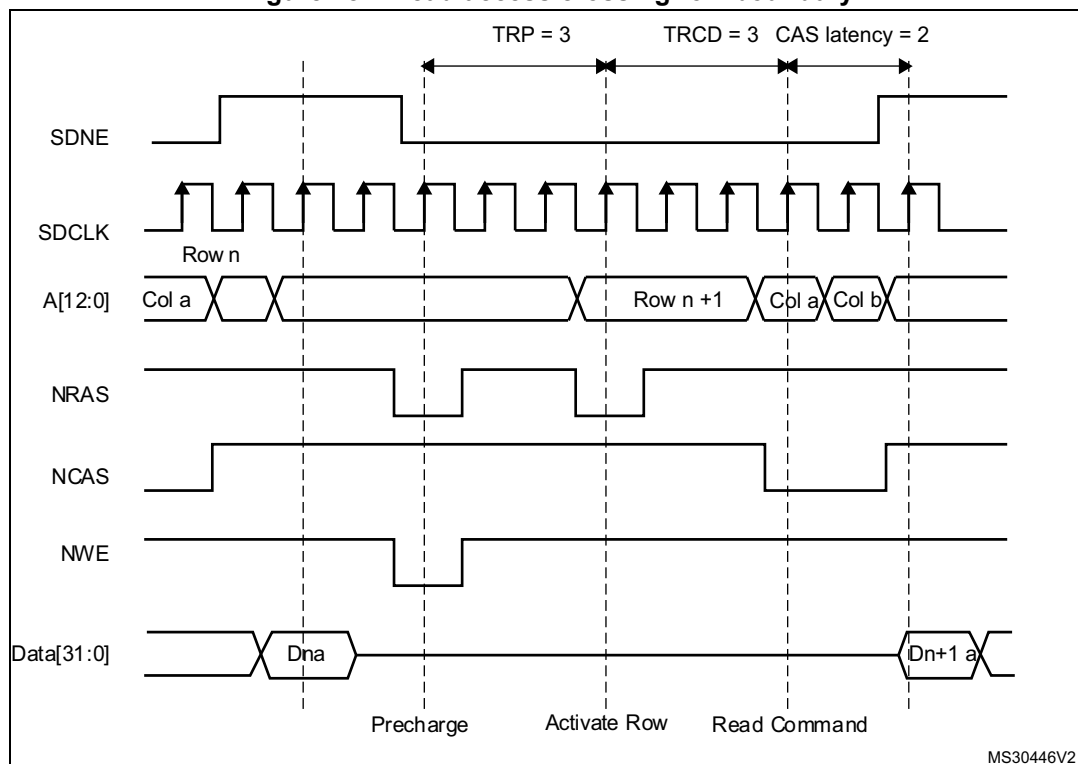
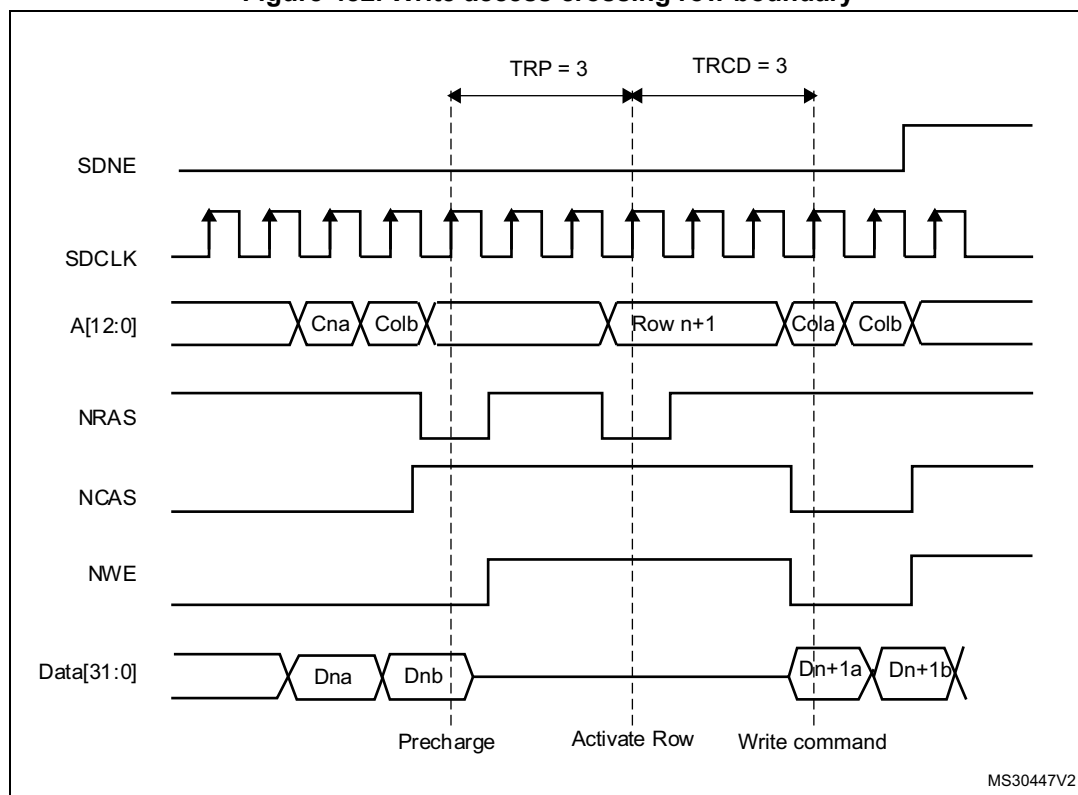


Figure 482. Write access crossing row boundary



If the next access is sequential and the current access crosses a bank boundary, the SDRAM controller activates the first row in the next bank and initiates a new read/write command. Two cases are possible:

- If the current bank is not the last one, the active row in the new bank must be precharged. At a bank boundary, the automatic activation of the next row is supported for all rows/columns and data bus width configuration.
- For 13-bit row address, 11-bit column address, 4 internal banks and bus width 32-bit SDRAM memories, if the current bank is the last one and the selected SDRAM device is connected to Bank 1, the SDRAM controller continues to read/write from the second SDRAM device (assuming it has been initialized):
 - a) The SDRAM controller activates the first row (after precharging the active row, if there is already an active row in the first internal bank, and initiates a new read/write command.
 - b) If the first row is already activated, the SDRAM controller just initiates a read/write command.

Note: *At bank boundary, if the current bank is the last one, the automatic activation of the next row is supported only when addressing 13-bit rows, 11-bit columns, 4 internal banks and 32-bit data bus SDRAM devices. Otherwise, the SDRAM address range is violated and an AHB error is generated.*

SDRAM controller refresh cycle

The Auto-refresh command is used to refresh the SDRAM device content. The SDRAM controller periodically issues auto-refresh commands. An internal counter is loaded with the COUNT value in the register FMC_SDRTR. This value defines the number of memory clock cycles between the refresh cycles (refresh rate). When this counter reaches zero, an internal pulse is generated.

If a memory access is ongoing, the auto-refresh request is delayed. However, if the memory access and the auto-refresh requests are generated simultaneously, the auto-refresh request takes precedence.

If the memory access occurs during an auto-refresh operation, the request is buffered and processed when the auto-refresh is complete.

If a new auto-refresh request occurs while the previous one was not served, the RE (Refresh Error) bit is set in the Status register. An Interrupt is generated if it has been enabled (REIE = '1').

If SDRAM lines are not in idle state (not all row are closed), the SDRAM controller generates a PALL (Precharge ALL) command before the auto-refresh.

If the Auto-refresh command is generated by the FMC_SDCMR Command Mode register (Mode bits = '011'), a PALL command (Mode bits = '010') must be issued first.

37.7.4 Low power modes

Two low power modes are available:

- Self-refresh mode
The auto-refresh cycles are performed by the SDRAM device itself to retain data without external clocking.
- Power-down mode
The auto-refresh cycles are performed by the SDRAM controller.

Self-refresh mode

This mode is selected by setting the MODE bits to '101' and by configuring the Target Bank bits (CTB1 and/or CTB2) in the FMC_SDCMR register.

The SDRAM clock stops running after a TRAS delay and the internal refresh timer stops counting only if one of the following conditions is met:

- A Self-refresh command is issued to both devices
- One of the devices is not activated (SDRAM bank is not initialized).

Before entering Self-Refresh mode, the SDRAM controller automatically issues a PALL command.

If the Write data FIFO is not empty, all data are sent to the memory before activating the Self-refresh mode and the BUSY status flag remains set.

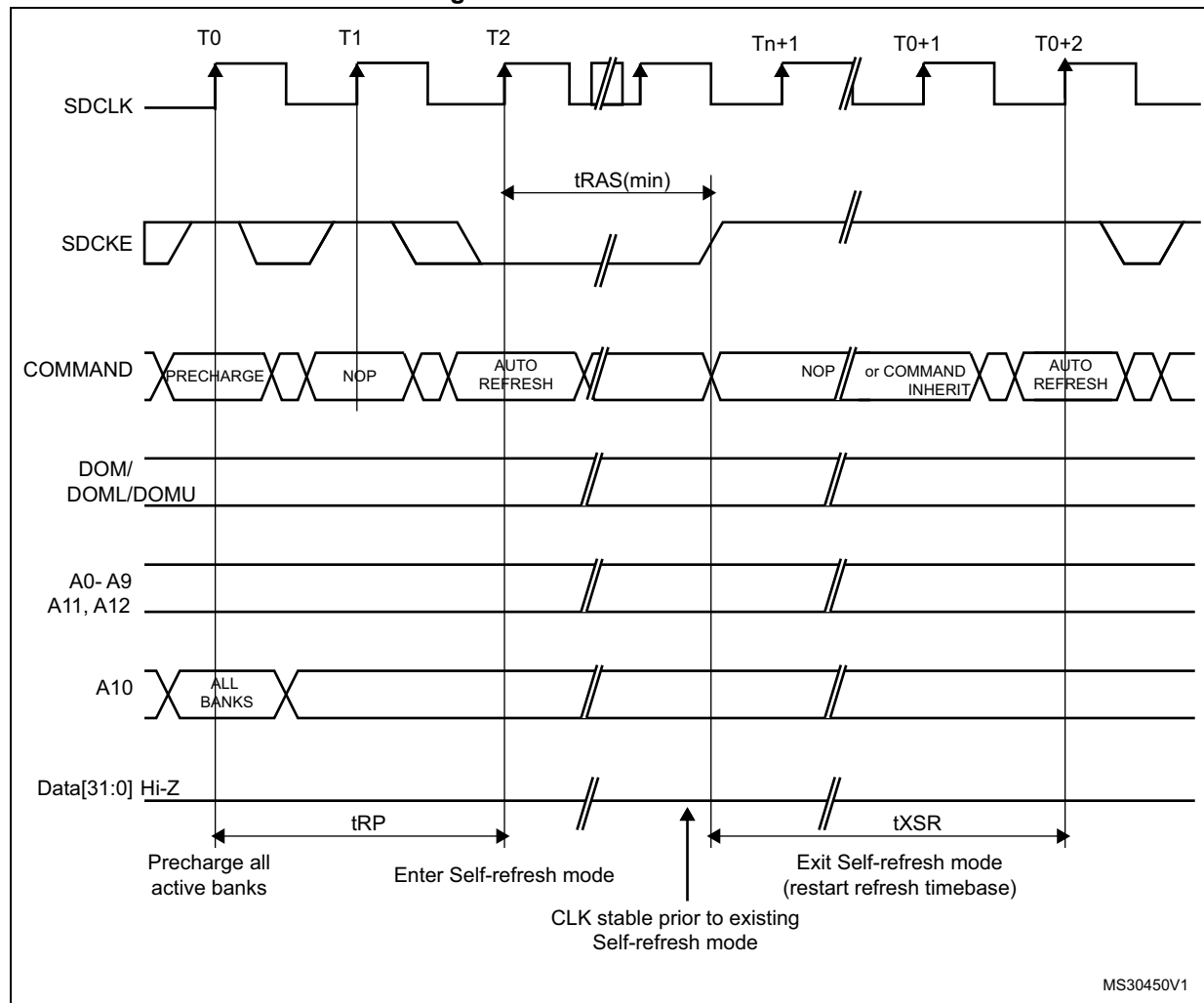
In Self-refresh mode, all SDRAM device inputs become don't care except for SDCKE which remains low.

The SDRAM device must remain in Self-refresh mode for a minimum period of time of TRAS and can remain in Self-refresh mode for an indefinite period beyond that. To guarantee this minimum period, the BUSY status flag remains high after the Self-refresh activation during a TRAS delay.

As soon as an SDRAM device is selected, the SDRAM controller generates a sequence of commands to exit from Self-refresh mode. After the memory access, the selected device remains in Normal mode.

To exit from Self-refresh, the MODE bits must be set to '000' (Normal mode) and the Target Bank bits (CTB1 and/or CTB2) must be configured in the FMC_SDCMR register.

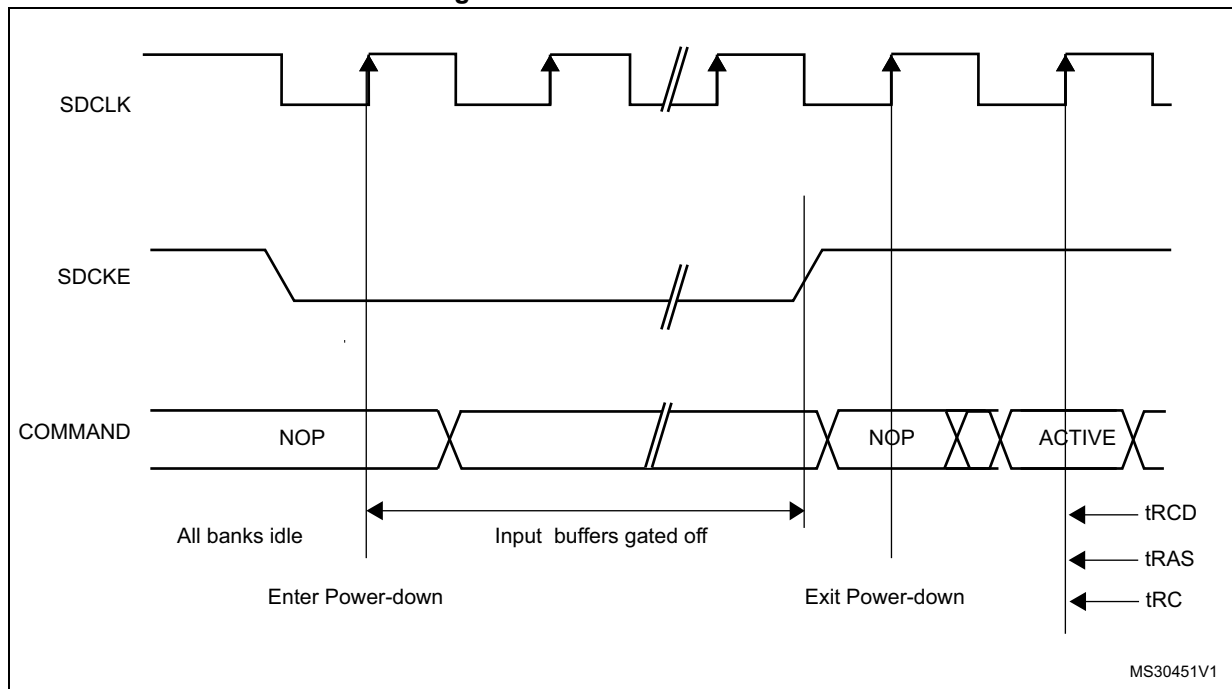
Figure 483. Self-refresh mode



Power-down mode

This mode is selected by setting the MODE bits to '110' and by configuring the Target Bank bits (CTB1 and/or CTB2) in the FMC_SDCMR register.

Figure 484. Power-down mode



If the Write data FIFO is not empty, all data are sent to the memory before activating the Power-down mode.

As soon as an SDRAM device is selected, the SDRAM controller exits from the Power-down mode. After the memory access, the selected SDRAM device remains in Normal mode.

During Power-down mode, all SDRAM device input and output buffers are deactivated except for the SDCKE which remains low.

The SDRAM device cannot remain in Power-down mode longer than the refresh period and cannot perform the Auto-refresh cycles by itself. Therefore, the SDRAM controller carries out the refresh operation by executing the operations below:

1. Exit from Power-down mode and drive the SDCKE high
2. Generate the PALL command only if a row was active during Power-down mode
3. Generate the auto-refresh command
4. Drive SDCKE low again to return to Power-down mode.

To exit from Power-down mode, the MODE bits must be set to '000' (Normal mode) and the Target Bank bits (CTB1 and/or CTB2) must be configured in the FMC_SDCMR register.

37.7.5 SDRAM controller registers

SDRAM Control registers 1,2 (FMC_SDCR1,2)

Address offset: $0x140 + 4 * (x - 1)$, $x = 1, 2$

Reset value: 0x0000 02D0

This register contains the control parameters for each SDRAM memory bank

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																	RPIPE[1:0]		RBURST	SDCLK[1:0]		WP	CAS[1:0]		NB	MWID[1:0]		NR[1:0]		NC[1:0]	
																	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:15 Reserved, must be kept at reset value

Bits 14:13 **RPIPE[1:0]**: Read pipe

These bits define the delay, in HCLK clock cycles, for reading data after CAS latency.

- 00: No HCLK clock cycle delay
- 01: One HCLK clock cycle delay
- 10: Two HCLK clock cycle delay
- 11: reserved, do not use

Note: The corresponding bits in the FMC_SDCR2 register are read only.

Bit 12 **RBURST**: Burst read

This bit enables burst read mode. The SDRAM controller anticipates the next read commands during the CAS latency and stores data in the Read FIFO.

- 0: single read requests are not managed as bursts
- 1: single read requests are always managed as bursts

Note: The corresponding bit in the FMC_SDCR2 register is don't care.

Bits 11:10 **SDCLK[1:0]**: SDRAM clock configuration

These bits define the SDRAM clock period for both SDRAM banks and allow disabling the clock before changing the frequency. In this case the SDRAM must be re-initialized.

- 00: SDCLK clock disabled
- 01: Reserved
- 10: SDCLK period = 2 x HCLK periods
- 11: SDCLK period = 3 x HCLK periods

Note: The corresponding bits in the FMC_SDCR2 register are read only.

Bit 9 **WP**: Write protection

This bit enables write mode access to the SDRAM bank.

- 0: Write accesses allowed
- 1: Write accesses ignored

Bits 8:7 **CAS[1:0]**: CAS Latency

This bits sets the SDRAM CAS latency in number of memory clock cycles

- 00: reserved, do not use.
- 01: 1 cycle
- 10: 2 cycles
- 11: 3 cycles

Bit 6 **NB**: Number of internal banks

This bit sets the number of internal banks.

0: Two internal Banks

1: Four internal Banks

Bits 5:4 **MWID[1:0]**: Memory data bus width.

These bits define the memory device width.

00: 8 bits

01: 16 bits

10: 32 bits

11: reserved, do not use.

Bits 3:2 **NR[1:0]**: Number of row address bits

These bits define the number of bits of a row address.

00: 11 bit

01: 12 bits

10: 13 bits

11: reserved, do not use.

Bits 1:0 **NC[1:0]**: Number of column address bits

These bits define the number of bits of a column address.

00: 8 bits

01: 9 bits

10: 10 bits

11: 11 bits.

Note: Before modifying the *RBURST* or *RPIPE* settings or disabling the *SDCLK* clock, the user must first send a *PALL* command to make sure ongoing operations are complete.

SDRAM Timing registers 1,2 (FMC_SDTR1,2)

Address offset: $0x148 + 4 * (x - 1)$, $x = 1,2$

Reset value: 0x0FFF FFFF

This register contains the timing parameters of each SDRAM bank

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				TRCD[3:0]				TRP[3:0]				TWR[3:0]				TRC[3:0]				TRAS[3:0]				TXSR[3:0]				TMRD[3:0]			
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	rW	rW

Bits 31:28 Reserved, must be kept at reset value

Bits 27:24 **TRCD[3:0]**: Row to column delay

These bits define the delay between the Activate command and a Read/Write command in number of memory clock cycles.

0000: 1 cycle.

0001: 2 cycles

....

1111: 16 cycles

Bits 23:20 **TRP[3:0]**: Row precharge delay

These bits define the delay between a Precharge command and another command in number of memory clock cycles. The TRP timing is only configured in the FMC_SDTR1 register. If two SDRAM devices are used, the TRP must be programmed with the timing of the slowest device.

0000: 1 cycle

0001: 2 cycles

....

1111: 16 cycles

Note: The corresponding bits in the FMC_SDTR2 register are don't care.

Bits 19:16 **TWR[3:0]**: Recovery delay

These bits define the delay between a Write and a Precharge command in number of memory clock cycles.

0000: 1 cycle

0001: 2 cycles

....

1111: 16 cycles

Note: TWR must be programmed to match the write recovery time (t_{WR}) defined in the SDRAM datasheet, and to guarantee that:

$$TWR \geq TRAS - TRCD \text{ and } TWR \geq TRC - TRCD - TRP$$

Example: TRAS= 4 cycles, TRCD= 2 cycles. So, TWR >= 2 cycles. TWR must be programmed to 0x1.

If two SDRAM devices are used, the FMC_SDTR1 and FMC_SDTR2 must be programmed with the same TWR timing corresponding to the slowest SDRAM device.

Bits 15:12 **TRC[3:0]**: Row cycle delay

These bits define the delay between the Refresh command and the Activate command, as well as the delay between two consecutive Refresh commands. It is expressed in number of memory clock cycles. The TRC timing is only configured in the FMC_SDTR1 register. If two SDRAM devices are used, the TRC must be programmed with the timings of the slowest device.

0000: 1 cycle

0001: 2 cycles

....

1111: 16 cycles

Note: TRC must match the TRC and TRFC (Auto Refresh period) timings defined in the SDRAM device datasheet.

Note: The corresponding bits in the FMC_SDTR2 register are don't care.

Bits 11:8 **TRAS[3:0]**: Self refresh time

These bits define the minimum Self-refresh period in number of memory clock cycles.

0000: 1 cycle

0001: 2 cycles

....

1111: 16 cycles

Bits 7:4 **TXSR[3:0]**: Exit Self-refresh delay

These bits define the delay from releasing the Self-refresh command to issuing the Activate command in number of memory clock cycles.

0000: 1 cycle

0001: 2 cycles

....

1111: 16 cycles

Bits 3:0 **TMRD[3:0]**: Load Mode Register to Active

These bits define the delay between a Load Mode Register command and an Active or Refresh command in number of memory clock cycles.

0000: 1 cycle

0001: 2 cycles

....

1111: 16 cycles

Note: *If two SDRAM devices are connected, all the accesses performed simultaneously to both devices by the Command Mode register (Load Mode Register and Self-refresh commands) are issued using the timing parameters configured for Bank 1 (TMRD, TRAS and TXSR timings) in the FMC_SDTR1 register.*

The TRP and TRC timings are only configured in the FMC_SDTR1 register. If two SDRAM devices are used, the TRP and TRC timings must be programmed with the timings of the slowest device.

SDRAM Command Mode register (FMC_SDCMR)

Address offset: 0x150

Reset value: 0x0000 0000

This register contains the command issued when the SDRAM device is accessed. This register is used to initialize the SDRAM device, and to activate the Self-refresh and the Power-down modes. As soon as the MODE field is written, the command will be issued only to one or to both SDRAM banks according to CTB1 and CTB2 command bits. This register is the same for both SDRAM banks.

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										MRD[11:0]										NRFS[3:0]				CTB1	CTB2	MODE[2:0]									
										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 31:22 Reserved, must be kept at reset value

Bits 21:9 **MRD[11:0]**: Mode Register definition

This 13-bit field defines the SDRAM Mode Register content. The Mode Register is programmed using the Load Mode Register command.

Bits 8:5 **NRFS[3:0]**: Number of Auto-refresh

These bits define the number of consecutive Auto-refresh commands issued when MODE = '011'.

0000: 1 Auto-refresh cycle

0001: 2 Auto-refresh cycles

....

1110: 15 Auto-refresh cycles

1111: Reserved

Bit 4 **CTB1**: Command Target Bank 1

This bit indicates whether the command will be issued to SDRAM Bank 1 or not.

0: Command not issued to SDRAM Bank 1

1: Command issued to SDRAM Bank 1

Bit 3 **CTB2**: Command Target Bank 2

This bit indicates whether the command will be issued to SDRAM Bank 2 or not.

- 0: Command not issued to SDRAM Bank 2
- 1: Command issued to SDRAM Bank 2

Bits 2:0 **MODE[2:0]**: Command mode

These bits define the command issued to the SDRAM device.

- 000: Normal Mode
- 001: Clock Configuration Enable
- 010: PALL ("All Bank Precharge") command
- 011: Auto-refresh command
- 100: Load Mode Register
- 101: Self-refresh command
- 110: Power-down command
- 111: Reserved

Note: When a command is issued, at least one Command Target Bank bit (CBT1 or CBT2) must be set. If both banks are used, the commands must be issued to the two banks at the same time by setting the CBT1 and CBT2 bits.

SDRAM Refresh Timer register (FMC_SDRTR)

Address offset: 0x154

Reset value: 0x0000 0000

This register sets the refresh rate in number of SDCLK clock cycles between the refresh cycles by configuring the Refresh Timer Count value.

$$\text{Refresh rate} = (\text{SDRAM refresh rate} \times \text{SDRAM clock frequency}) - 20$$

$$\text{SDRAM refresh rate} = \text{SDRAM refresh period} / \text{Number of rows}$$

Example

$$\text{SDRAM refresh rate} = 64 \text{ ms} / (8196 \text{ rows}) = 7.81 \mu\text{s}$$

where 64 ms is the SDRAM refresh period.

$$7.81 \mu\text{s} \times 60 \text{ MHz} = 468.6$$

The refresh rate must be increased by 20 SDRAM clock cycles (as in the above example) to obtain a safe margin if an internal refresh request occurs when a read request has been accepted. It corresponds to a COUNT value of '0000111000000' (448).

This 13-bit field is loaded into a timer which is decremented using the SDRAM clock. This timer generates a refresh pulse when zero is reached. The COUNT value must be set at least to 41 SDRAM clock cycles.

As soon as the FMC_SDRTR register is programmed, the timer starts counting. If the value programmed in the register is '0', no refresh is carried out. This register must not be reprogrammed after the initialization procedure to avoid modifying the refresh rate.

Each time a refresh pulse is generated, this 13-bit COUNT field is reloaded into the counter.

If a memory access is in progress, the Auto-refresh request is delayed. However, if the memory access and Auto-refresh requests are generated simultaneously, the Auto-refresh takes precedence. If the memory access occurs during a refresh operation, the request is buffered to be processed when the refresh is complete.

This register common to SDRAM bank 1 and bank 2.

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																	REIE	COUNT[12:0]													CRE
																	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Bits 31: 15 Reserved, must be kept at reset value

Bit 14 **REIE**: RES Interrupt Enable

0: Interrupt is disabled

1: An Interrupt is generated if RE = 1

Bits 13:1 **COUNT[12:0]**: Refresh Timer Count

This 13-bit field defines the refresh rate of the SDRAM device. It is expressed in number of memory clock cycles. It must be set at least to 41 SDRAM clock cycles (0x29).

$COUNT = (SDRAM \text{ refresh rate} \times SDRAM \text{ clock frequency}) - 20$

$SDRAM \text{ refresh rate} = SDRAM \text{ refresh period} / \text{Number of rows}$

Bit 0 **CRE**: Clear Refresh error flag

This bit is used to clear the Refresh Error Flag (RE) in the Status Register.

0: no effect

1: Refresh Error flag is cleared

Note: *The programmed COUNT value must not be equal to the sum of the following timings: $TWR + TRP + TRC + TRCD + 4$ memory clock cycles.*

SDRAM Status register (FMC_SDSR)

Address offset: 0x158

Reset value: 0x0000 0000

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																											BUSY	MODES2[1:0]		MODES1[1:0]		RE
																											r	r	r	r	r	

Bits 31:5 Reserved, must be kept at reset value

Bit 5 **BUSY**: Busy status

This bit defines the status of the SDRAM controller after a Command Mode request

0: SDRAM Controller is ready to accept a new request

1; SDRAM Controller is not ready to accept a new request

Bits 4:3 **MODES2[1:0]**: Status Mode for Bank 2

This bit defines the Status Mode of SDRAM Bank 2.

00: Normal Mode

01: Self-refresh mode

10: Power-down mode

Bits 2:1 **MODES1[1:0]**: Status Mode for Bank 1

This bit defines the Status Mode of SDRAM Bank 1.

00: Normal Mode

01: Self-refresh mode

10: Power-down mode

Bit 0 **RE**: Refresh error flag

0: No refresh error has been detected

1: A refresh error has been detected

An interrupt is generated if REIE = 1 and RE = 1

37.8 FMC register map

The following table summarizes the FMC registers.

Table 297. FMC register map

Offset	Register	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
0x00	FMC_BCR1	Reserved												CCLKEN	CBURSTRW	CPSIZE[2:0]	ASYNCPWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]	MTYP[1:0]	MUXEN	MBKEN			
0x08	FMC_BCR2	Reserved												CBURSTRW	CPSIZE[2:0]	ASYNCPWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]	MTYP[1:0]	MUXEN	MBKEN				
0x10	FMC_BCR3	Reserved												CBURSTRW	CPSIZE[2:0]	ASYNCPWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]	MTYP[1:0]	MUXEN	MBKEN				
0x18	FMC_BCR4	Reserved												CBURSTRW	CPSIZE[2:0]	ASYNCPWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]	MTYP[1:0]	MUXEN	MBKEN				
0x04	FMC_BTR1	Res.	ACCMOD[1:0]				DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]				ADDHLD[3:0]		ADDSET[3:0]														
0x0C	FMC_BTR2	Res.	ACCMOD[1:0]				DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]				ADDHLD[3:0]		ADDSET[3:0]														
0x14	FMC_BTR3	Res.	ACCMOD[1:0]				DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]				ADDHLD[3:0]		ADDSET[3:0]														
0x1C	FMC_BTR4	Res.	ACCMOD[1:0]				DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]				ADDHLD[3:0]		ADDSET[3:0]														
0x104	FMC_BWTR1	Res.	ACCMOD[1:0]				Res.				BUSTURN[3:0]		DATAST[7:0]				ADDHLD[3:0]		ADDSET[3:0]														

Table 297. FMC register map (continued)

Offset	Register	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
0x10C	FMC_BWTR2	Res.		ACCMOD[1:0]		Res.					BUSTURN[3:0]				DATAST[7:0]					ADDHLD[3:0]				ADDSET[3:0]									
0x104	FMC_BWTR3	Res.				Res.					BUSTURN[3:0]				DATAST[7:0]					ADDHLD[3:0]				ADDSET[3:0]									
0x10C	FMC_BWTR4	Res.				Res.					BUSTURN[3:0]				DATAST[7:0]					ADDHLD[3:0]				ADDSET[3:0]									
0x60	FMC_PCR2	Reserved												ECCPS[2:0]			TAR[3:0]		TCLR[3:0]			Res.	ECCEN	PWID[1:0]		PTYP		PBKEN	PWAITEN	Reserved			
0x80	FMC_PCR3	Reserved												ECCPS[2:0]			TAR[3:0]		TCLR[3:0]			Res.	ECCEN	PWID[1:0]		PTYP		PBKEN	PWAITEN	Reserved			
0xA0	FMC_PCR4	Reserved												ECCPS[2:0]			TAR[3:0]		TCLR[3:0]			Res.	ECCEN	PWID[1:0]		PTYP		PBKEN	PWAITEN	Reserved			
0x64	FMC_SR2	Reserved																								FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS	
0x84	FMC_SR3	Reserved																								FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS	
0xA4	FMC_SR4	Reserved																								FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS	
0x68	FMC_PMEM2	MEMHIZ[7:0]						MEMHOLD[7:0]						MEMWAIT[7:0]						MEMSET[7:0]													
0x88	FMC_PMEM3	MEMHIZ[7:0]						MEMHOLD[7:0]						MEMWAIT[7:0]						MEMSET[7:0]													
0xA8	FMC_PMEM4	MEMHIZ[7:0]						MEMHOLD[7:0]						MEMWAIT[7:0]						MEMSET[7:0]													
0x6C	FMC_PATT2	ATTHIZ[7:0]						ATTHOLD[7:0]						ATTWAIT[7:0]						ATTSET[7:0]													
0x8C	FMC_PATT3	ATTHIZ[7:0]						ATTHOLD[7:0]						ATTWAIT[7:0]						ATTSET[7:0]													
0xAC	FMC_PATT4	ATTHIZ[7:0]						ATTHOLD[7:0]						ATTWAIT[7:0]						ATTSET[7:0]													
0xB0	FMC_PIO4	IOHIZ[7:0]						IOHOLD[7:0]						IOWAIT[7:0]						IOSET[7:0]													
0x74	FMC_ECCR2	ECC[31:0]																															
0x94	FMC_ECCR3	ECC[31:0]																															
0x140	FMC_SDCR_1	Reserved																RPIPE[1:0]	RBURST	CLK[1:0]	WP	CAS[1:0]	NB	MWID[1:0]				NR[1:0]		NC[1:0]			
0x144	FMC_SDCR_2	Reserved																		CLK[1:0]	WP	CAS[1:0]	NB	MWID[1:0]				NR[1:0]		NC[1:0]			
0x148	FMC_SDTR1	Reserved				TRCD[3:0]			TRP[3:0]			TWR[3:0]			TRC[3:0]			TRAS[3:0]			TXSR[3:0]			TMRD[3:0]									
0x14C	FMC_SDTR2	Reserved				TRCD[3:0]			TRP[3:0]			TWR[3:0]			TRC[3:0]			TRAS[3:0]			TXSR[3:0]			TMRD[3:0]									
0x150	FMC_SDCMR	Reserved										MRD[12:0]						NRFS[3:0]				CTB1		CTB2		MODE[2:0]							

Table 297. FMC register map (continued)

Offset	Register	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
0x154	FMC_SDRTR	Reserved															REIE	COUNT[12:0]															CRE
0x158	FMC_SDSR	Reserved																										BUSY	MODES2[1:0]		MODES1[1:0]		RE

