# Topic 3

# Memory Management and Access

*Department of Electronics*

*Academic Year 14/15*

*(ver 25-10-2014)*

# Index

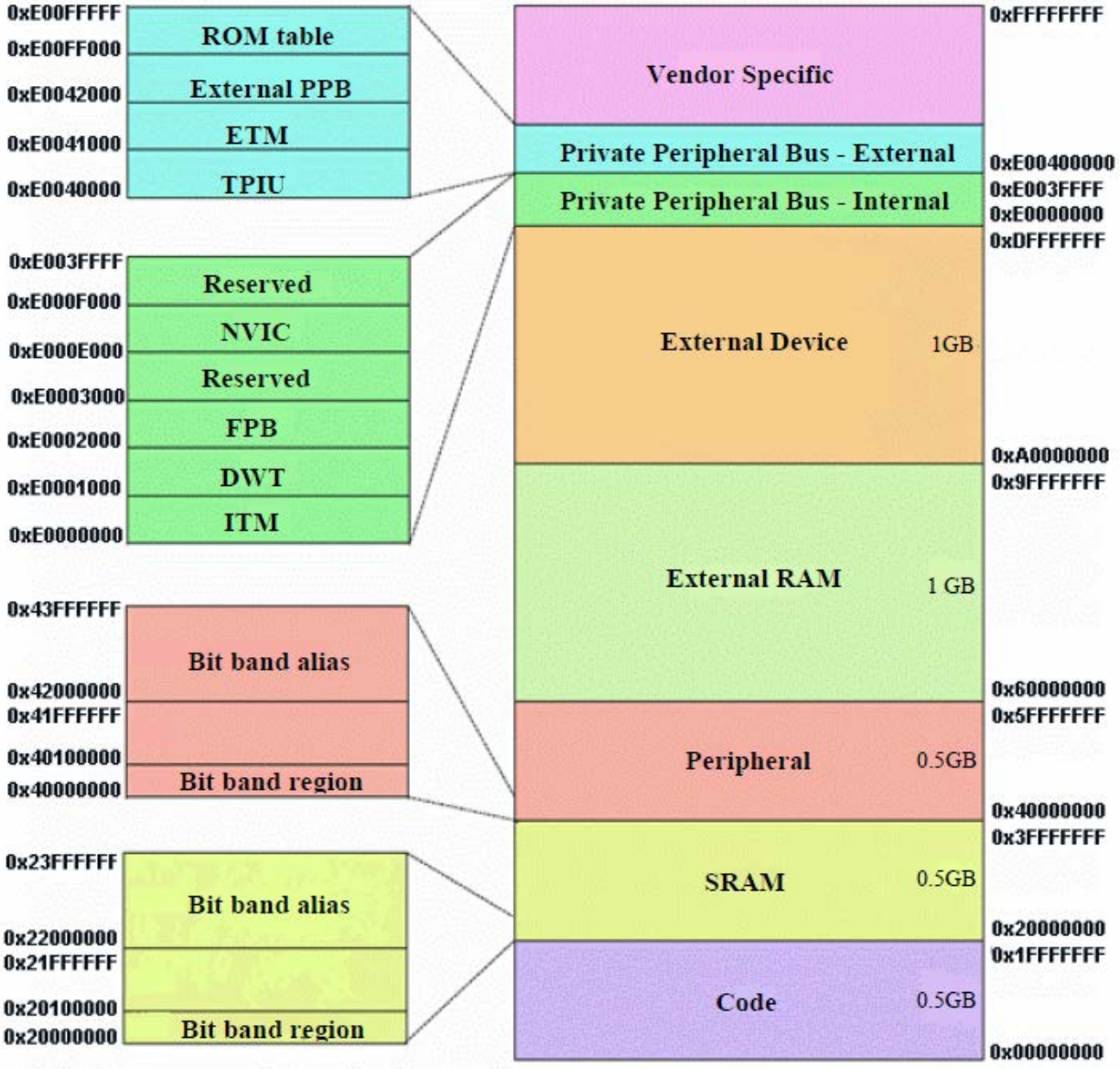
# Index

# 3.6.1. Cortex-M3 Memory Map: The Map

- The Cortex-M3 has a 4Gbyte predefined memory map
  - Built-in peripherals, NVIC, debugging units, … can be accessed by simple memory access instructions
- Memory space is divided
  - C-M3 design has an internal bus infrastructure optimized for this memory usage
- Portability is ensured also among different manufacturers
  - Though different unused parts in the memory maps change among the different devices:

    for example, LPC1850 has more external memory map than LPC1788

# 3.6.1. Cortex-M3 Memory Map: LPC17xx Map



# 3.6.1. Cortex-M3 Memory Map: LPC17xx Map

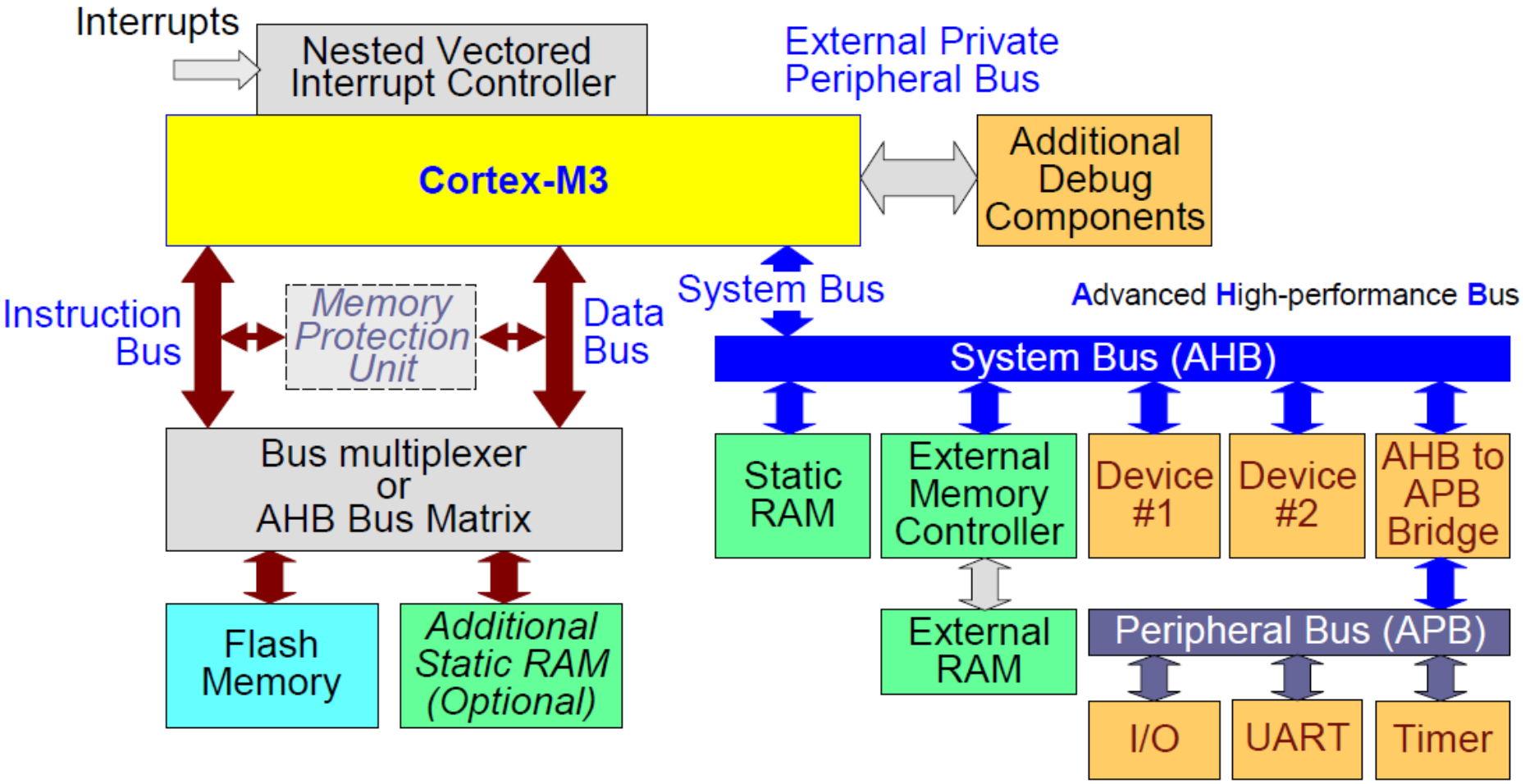
**Table 3. LPC17xx memory usage and details**

| Address range | General Use | Address range details and description | |
|---|---|---|---|
| 0x0000 0000 to 0x1FFF FFFF | On-chip non-volatile memory | 0x0000 0000 - 0x0007 FFFF | For devices with 512 kB of flash memory. |
| | | 0x0000 0000 - 0x0003 FFFF | For devices with 256 kB of flash memory. |
| | | 0x0000 0000 - 0x0001 FFFF | For devices with 128 kB of flash memory. |
| | | 0x0000 0000 - 0x0000 FFFF | For devices with 64 kB of flash memory. |
| | | 0x0000 0000 - 0x0000 7FFF | For devices with 32 kB of flash memory. |
| | On-chip SRAM | 0x1000 0000 - 0x1000 7FFF | For devices with 32 kB of local SRAM. |
| | | 0x1000 0000 - 0x1000 3FFF | For devices with 16 kB of local SRAM. |
| | | 0x1000 0000 - 0x1000 1FFF | For devices with 8 kB of local SRAM. |
| | Boot ROM | 0x1FFF 0000 - 0x1FFF 1FFF | 8 kB Boot ROM with flash services. |
| 0x2000 0000 to 0x3FFF FFFF | On-chip SRAM (typically used for peripheral data) | 0x2007 C000 - 0x2007 FFFF | AHB SRAM - bank 0 (16 kB), present on devices with 32 kB or 64 kB of total SRAM. |
| | | 0x2008 0000 - 0x2008 3FFF | AHB SRAM - bank 1 (16 kB), present on devices with 64 kB of total SRAM. |
| | GPIO | 0x2009 C000 - 0x2009 FFFF | GPIO. |
| 0x4000 0000 to 0x5FFF FFFF | APB Peripherals | 0x4000 0000 - 0x4007 FFFF | APB0 Peripherals, up to 32 peripheral blocks, 16 kB each. |
| | | 0x4008 0000 - 0x400F FFFF | APB1 Peripherals, up to 32 peripheral blocks, 16 kB each. |
| | AHB peripherals | 0x5000 0000 - 0x501F FFFF | DMA Controller, Ethernet interface, and USB interface. |
| 0xE000 0000 to 0xE00F FFFF | Cortex-M3 Private Peripheral Bus | 0xE000 0000 - 0xE00F FFFF | Cortex-M3 related functions, includes the NVIC and System Tick Timer. |

# 3.6.2. Cortex-M3 Memory Map: Bus Interfaces

- Cortex M3 has a Harvard architecture, and thus, different bus interfaces
  - Code Memory Buses, I-Code y D-Code: 32 bit buses used to access code memory region, optimized for the fetch operation
  - System Bus (AHB): 32 bit bus used to access data memory region, where RAM, peripherals and some external devices are located. The EMC used to access to external devices are connected to this bus
  - Private Peripheral Bus, PPB: 32 bit bus used to access to the part of the System Level region not accessed through the System Bus, such as debugging units. It has two versions, internal and external (APB)
  - DAP Bus: 32-bit bus specifically used to attach debug interface blocks such as JTAG-DP or SW-DP

# 3.6.2. Cortex-M3 Memory Map: Bus Interfaces

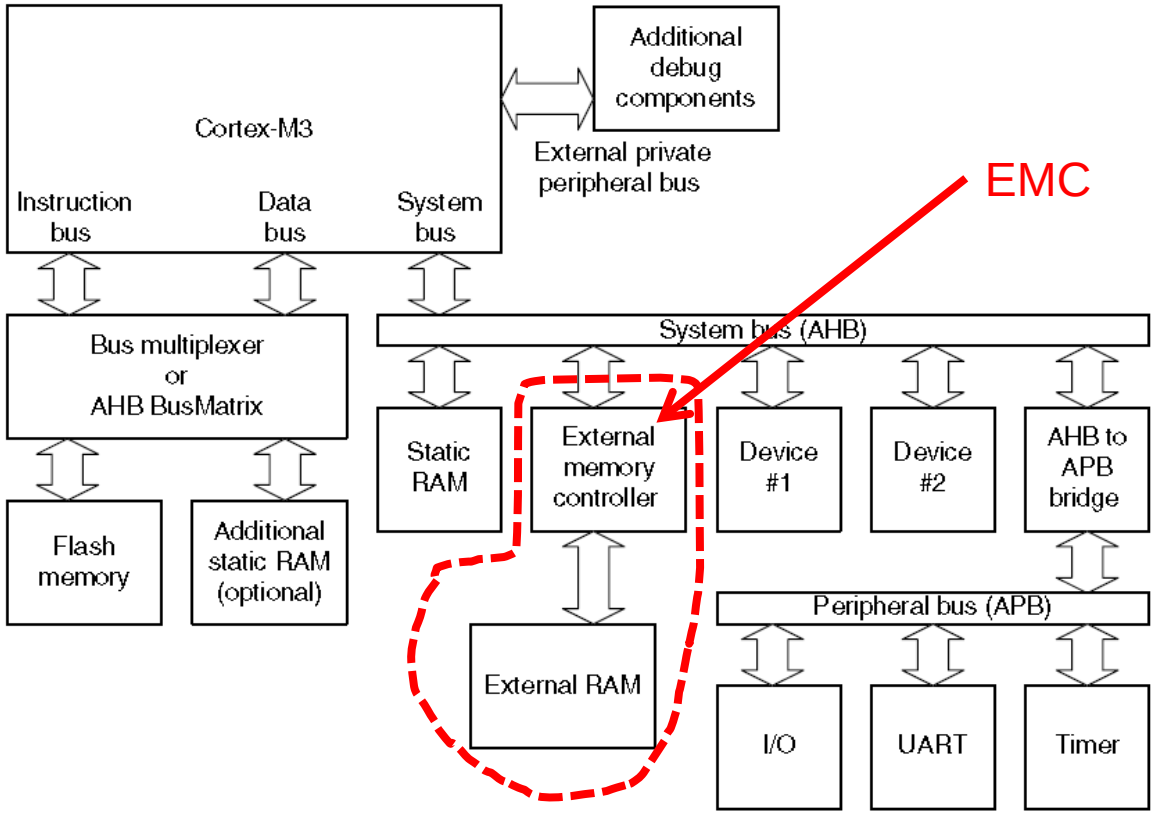


# 3.6.2. Cortex-M3 Memory Map: Bus Interfaces

| Memory Map | Interface |
|---|---|
| Code | Instruction fetches are performed over the ICode bus. Data accesses are performed over the DCode bus. |
| SRAM | Instruction fetches and data accesses are performed over the system bus. |
| SRAM_bitband | Alias region. Data accesses are aliases. Instruction accesses are not aliases. |
| Peripheral | Instruction fetches and data accesses are performed over the system bus. |
| Periph_bitband | Alias region. Data accesses are aliases. Instruction accesses are not aliases. |
| External RAM | Instruction fetches and data accesses are performed over the system bus. |
| External Device | Instruction fetches and data accesses are performed over the system bus. |
| Private Peripheral Bus | Accesses to:<br>• *Instrumentation Trace Macrocell* (ITM)<br>• *Nested Vectored Interrupt Controller* (NVIC)<br>• *Flashpatch and Breakpoint* (FPB)<br>• *Data Watchpoint and Trace* (DWT)<br>• *Memory Protection Unit* (MPU)<br>are performed to the processor internal *Private Peripheral Bus* (PPB).<br>Accesses to:<br>• *Trace Point Interface Unit* (TPIU)<br>• *Embedded Trace Macrocell* (ETM)<br>• System areas of the PPB memory map<br>are performed over the external PPB interface.<br>This memory region is *Execute Never* (XN), and so instruction fetches are prohibited. An MPU, if present, cannot change this. |
| System | System segment for vendor system peripherals. This memory region is XN, and so instruction fetches are prohibited. An MPU, if present, cannot change this. |

# 3.7.1. The EMC: General characteristics

- What is a External Memory Controller (EMC)?
  - EMC is a unit normally included in microcontrollers, designed to manage external implementation and access to the memory map



# 3.7.1. The EMC: General characteristics

- What are the normal tasks of an EMC?
  - Buffering Address, Data and Control busses
  - Implementing Bank enable signals (BLs), ordering data when misalignments or access to different size data
  - Implementing CS signals to enable external sections in the map, allowing software configuration of these sections size and position
  - Dynamic RAM management (refresh and demultiplexing busses)
  - Programming wait cycles and timings, such as setup, strobe, and hold times
- Some examples:
  - Cortex M3: LP18xx (NXP), LPC24xx (NXP), MCB2400 (NI), STM32XX (STM)
  - Other devices: µC MCF52xx ColdFire (Freescale), DSP TMS320C6000 (TI)
  - Stand-alone EMCs: 24815 (Freescale), STM32F10xxx (STM), OPB-EMC (Xilinx)

# 3.7.2. The EMC: Memory map and connections

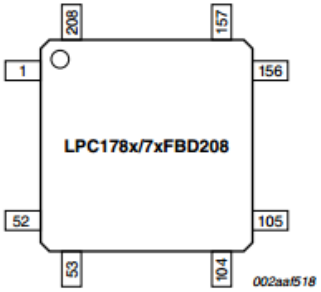
- In Cortex-M3 processor, included for example in LPC177x/178x

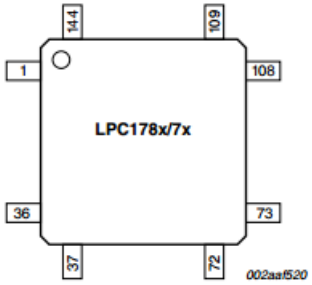
Table 6. LPC178x/177x memory usage and details

| Address range | General Use | Address range details and description | |
|---|---|---|---|
| 0x0000 0000 to 0x1FFF FFFF | On-chip non-volatile memory | 0x0000 0000 - 0x0007 FFFF | For devices with 512 kB of flash memory. |
| | | 0x0000 0000 - 0x0003 FFFF | For devices with 256 kB of flash memory. |
| | | 0x0000 0000 - 0x0001 FFFF | For devices with 128 kB of flash memory. |
| | | 0x0000 0000 - 0x0000 FFFF | For devices with 64 kB of flash memory. |
| | On-chip SRAM | 0x1000 0000 - 0x1000 FFFF | For devices with 64 kB of local SRAM. |
| | | 0x1000 0000 - 0x1000 7FFF | For devices with 32 kB of local SRAM. |
| | | 0x1000 0000 - 0x1000 3FFF | For devices with 16 kB of local SRAM. |
| | Boot ROM | 0x1FFF 0000 - 0x1FFF 1FFF | 8 kB Boot ROM with flash services. |
| 0x2000 0000 to 0x3FFF FFFF | On-chip SRAM (typically used for peripheral data) | 0x2000 0000 - 0x2000 1FFF | Peripheral RAM - bank 0 (first 8 kB) |
| | | 0x2002 0000 - 0x2000 3FFF | Peripheral RAM - bank 0 (second 8 kB) |
| | | 0x2000 4000 - 0x2000 7FFF | Peripheral RAM - bank 1 (16 kB) |
| | AHB peripherals | 0x2008 0000 - 0x200B FFFF | See Figure 6 for details |
| 0x4000 0000 to 0x7FFF FFFF | APB Peripherals | 0x4000 0000 - 0x4007 FFFF | APB0 Peripherals, up to 32 peripheral blocks of 16 kB each. |
| | | 0x4008 0000 - 0x400F FFFF | APB1 Peripherals, up to 32 peripheral blocks of 16 kB each. |
| 0x8000 0000 to 0xDFFF FFFF | Off-chip Memory via the External Memory Controller | Four static memory chip selects: | |
| | | 0x8000 0000 - 0x83FF FFFF | Static memory chip select 0 (up to 64 MB) |
| | | 0x9000 0000 - 0x93FF FFFF | Static memory chip select 1 (up to 64 MB) |
| | | 0x9800 0000 - 0x9BFF FFFF | Static memory chip select 2 (up to 64 MB) |
| | | 0x9C00 0000 - 0x9FFF FFFF | Static memory chip select 3 (up to 64 MB) |
| | | Four dynamic memory chip selects: | |
| | | 0xA000 0000 - 0xAFFF FFFF | Dynamic memory chip select 0 (up to 256MB) |
| | | 0xB000 0000 - 0xBFFF FFFF | Dynamic memory chip select 1 (up to 256MB) |
| | | 0xC000 0000 - 0xCFFF FFFF | Dynamic memory chip select 2 (up to 256MB) |
| | | 0xD000 0000 - 0xDFFF FFFF | Dynamic memory chip select 3 (up to 256MB) |
| 0xE000 0000 to 0xE00F FFFF | Cortex-M3 Private Peripheral Bus | 0xE000 0000 - 0xE00F FFFF | Cortex-M3 related functions, includes the NVIC and System Tick Timer. |

# 3.7.2. The EMC: Memory map and connections

- LPC178x:



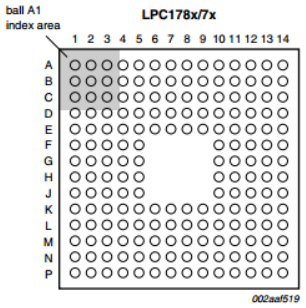




Table 109. EMC configuration

| Device package | Data bus widths supported | Pins available | Dynamic memory configuration registers[1][2] | Static memory configuration registers[1][3] | External memory connections |
|---|---|---|---|---|---|
| 144-pin | 8-bit | EMC_A[15:0] | | EMCStaticConfig1/0 | Section 9.14.3 |
| | | EMC_D[7:0] | | EMCStaticWaitWen1/0 | |
| | | $\overline{\text{EMC\_OE}}$ | | EMCStaticWaitOen1/0 | |
| | | $\overline{\text{EMC\_WE}}$ | | EMCStaticWaitRd1/0 | |
| | | $\overline{\text{EMC\_CS1:0}}$ | | EMCStaticWaitPage1/0 | |
| | | | | EMCStaticWaitWr1/0 | |
| | | | | EMCStaticWaitTurn1/0 | |
| 180-pin | 16-bit, 8-bit | EMC_A[19:0] | EMCDynamicConfig1/0 | EMCStaticConfig1/0 | Section 9.14.2 |
| | | EMC_D[15:0] | EMCDynamicRasCas1/0 | EMCStaticWaitWen1/0 | Section 9.14.3 |
| | | $\overline{\text{EMC\_OE}}$ | | EMCStaticWaitOen1/0 | |
| | | $\overline{\text{EMC\_WE}}$ | | EMCStaticWaitRd1/0 | |
| | | $\overline{\text{EMC\_BLS1:0}}$ | | EMCStaticWaitPage1/0 | |
| | | $\overline{\text{EMC\_CS1:0}}$ | | EMCStaticWaitWr1/0 | |
| | | $\overline{\text{EMC\_DYCS1:0}}$ | | EMCStaticWaitTurn1/0 | |
| | | $\overline{\text{EMC\_CAS}}$ | | | |
| | | $\overline{\text{EMC\_RAS}}$ | | | |
| | | EMC_CLK1:0 | | | |
| | | EMC_CKE1:0 | | | |
| | | EMC_DQM1:0 | | | |
| 208-pin | 32-bit, 16-bit, 8-bit | EMC_A[25:0] | EMCDynamicConfig3/2/1/0 | EMCStaticConfig3/2/1/0 | Section 9.14.1 |
| | | EMC_D[31:0] | EMCDynamicRasCas3/2/1/0 | EMCStaticWaitWen3/2/1/0 | Section 9.14.2 |
| | | $\overline{\text{EMC\_OE}}$ | | EMCStaticWaitOen3/2/1/0 | Section 9.14.3 |
| | | $\overline{\text{EMC\_WE}}$ | | EMCStaticWaitRd3/2/1/0 | |
| | | $\overline{\text{EMC\_BLS3:0}}$ | | EMCStaticWaitPage3/2/1/0 | |
| | | $\overline{\text{EMC\_CS3:0}}$ | | EMCStaticWaitWr3/2/1/0 | |
| | | $\overline{\text{EMC\_DYCS3:0}}$ | | EMCStaticWaitTurn3/2/1/0 | |
| | | $\overline{\text{EMC\_CAS}}$ | | | |
| | | $\overline{\text{EMC\_RAS}}$ | | | |
| | | EMC_CLK1:0 | | | |
| | | EMC_CKE3:0 | | | |
| | | EMC_DQM3:0 | | | |

# 3.7.2. The EMC: Memory map and connections

- LPC177x/178x: External memory map + Pin Signals

| | | | |
|---|---|---|---|
| 0x8000 0000 to 0xDFFF FFFF | Off-chip Memory via the External Memory Controller | Four static memory chip selects: | |
| | | 0x8000 0000 - 0x83FF FFFF | Static memory chip select 0 (up to 64 MB) |
| | | 0x9000 0000 - 0x93FF FFFF | Static memory chip select 1 (up to 64 MB) |
| | | 0x9800 0000 - 0x9BFF FFFF | Static memory chip select 2 (up to 64 MB) |
| | | 0x9C00 0000 - 0x9FFF FFFF | Static memory chip select 3 (up to 64 MB) |
| | | Four dynamic memory chip selects: | |
| | | 0xA000 0000 - 0xAFFF FFFF | Dynamic memory chip select 0 (up to 256MB) |
| | | 0xB000 0000 - 0xBFFF FFFF | Dynamic memory chip select 1 (up to 256MB) |
| | | 0xC000 0000 - 0xCFFF FFFF | Dynamic memory chip select 2 (up to 256MB) |
| | | 0xD000 0000 - 0xDFFF FFFF | Dynamic memory chip select 3 (up to 256MB) |

**Table 7. External memory controller pin configuration**

| Part | Data bus pins | Address bus pins | Control pins | |
|---|---|---|---|---|
| | | | SRAM | SDRAM |
| LPC1788FBD208 | EMC_D[31:0] | EMC_A[25:0] | EMC_BLS[3:0], EMC_CS[3:0], EMC_OE, EMC_WE | EMC_RAS, EMC_CAS, EMC_DYCS[3:0], EMC_CLK[1:0], EMC_CKE[3:0], EMC_DQM[3:0] |

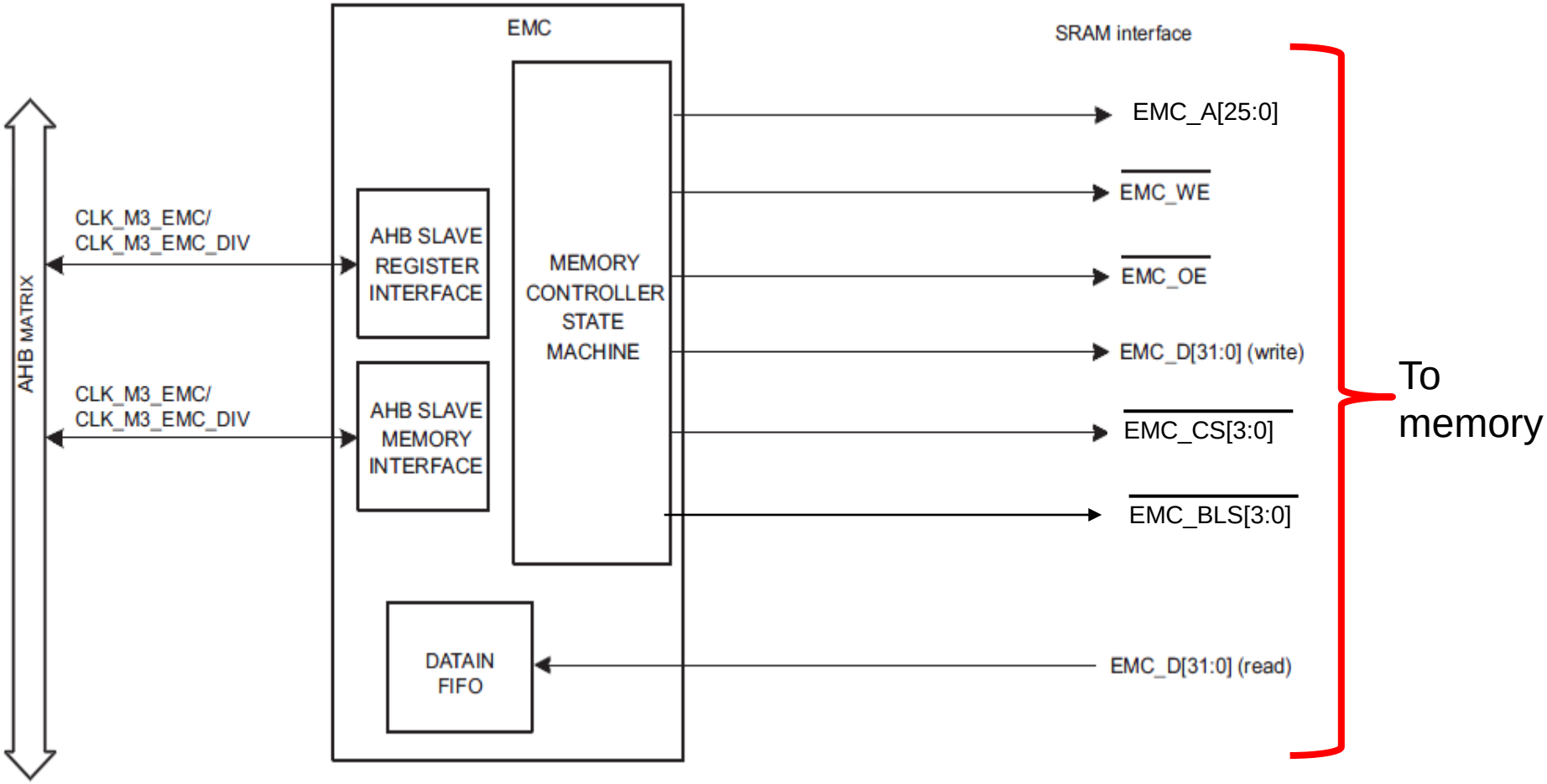
# 3.7.2. The EMC: Memory map and connections

- LPC177x/178x: Pin Signals

| Pin function | Direction | Description |
|---|---|---|
| EMC_A[25:0] | O | Address bus |
| EMC_D[31:0] | I/O | Data bus |
| $\overline{\text{EMC\_BLS[3:0]}}$ | O | Byte lane select |
| $\overline{\text{EMC\_CS[3:0]}}$ | O | Static RAM memory bank select |
| $\overline{\text{EMC\_OE}}$ | O | Output enable |
| $\overline{\text{EMC\_WE}}$ | O | Write enable |
| $\overline{\text{EMC\_CKEOUT[3:0]}}$ | O | SDRAM clock enable signals |
| EMC_CLK[3:0]; EMC_CLK01; EMC_CLK23 | O | SDRAM clock signals |
| $\overline{\text{EMC\_DQMOUT[3:0]}}$ | O | Data mask output to SDRAM memory banks |
| $\overline{\text{EMC\_DYCS[3:0]}}$ | O | SDRAM memory bank select |
| $\overline{\text{EMC\_CAS}}$ | O | Column address strobe |
| $\overline{\text{EMC\_RAS}}$ | O | Row address strobe |

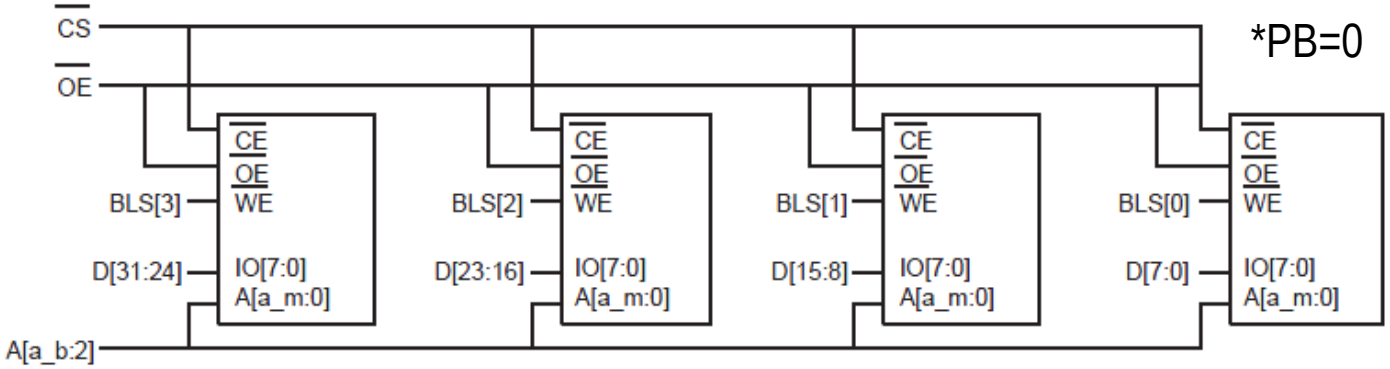
# 3.7.2. The EMC: Memory map and connections

- LPC178x: SRAM interface

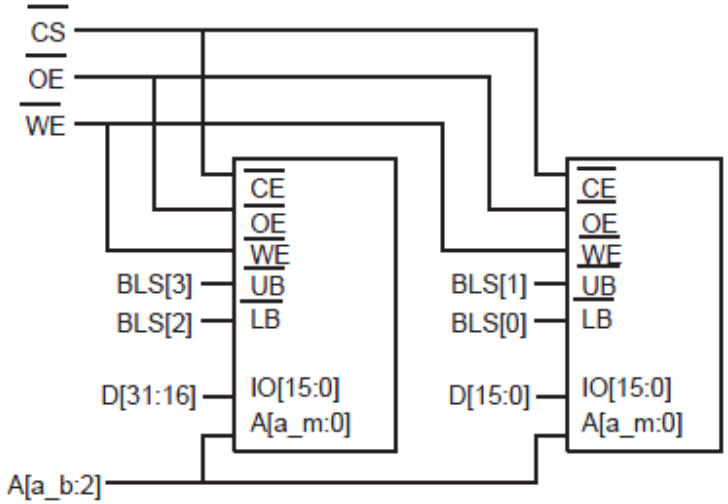


# 3.7.2. The EMC: Memory map and connections

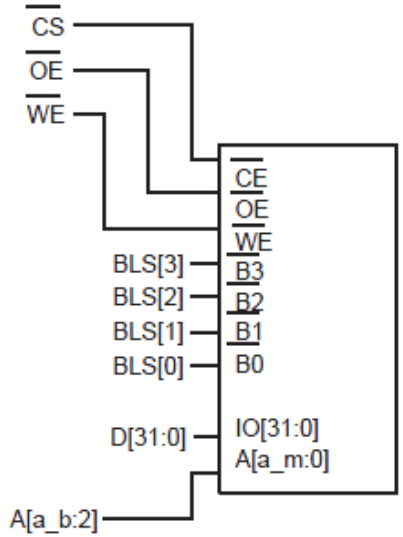
- LPC178x: SRAM connection




a. 32 bit wide memory bank interfaced to four 8 bit memory chips



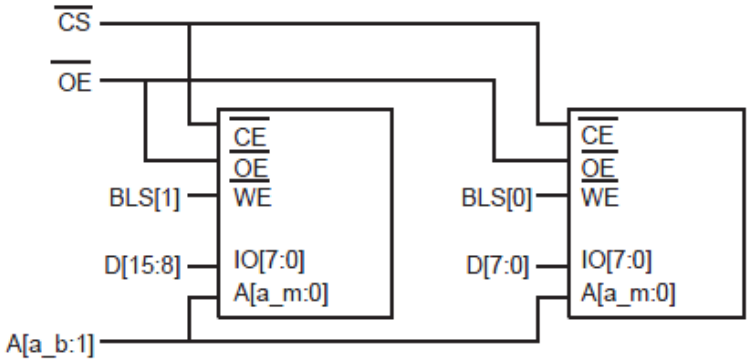

b. 32 bit wide memory bank interfaced to two 16 bit memory chips



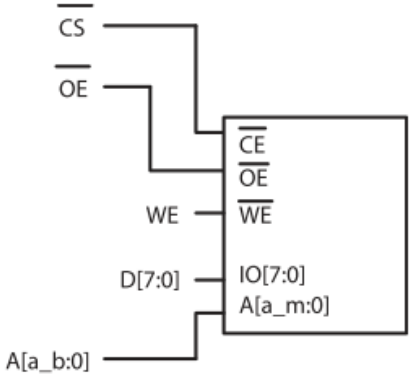

c. 32 bit wide memory bank interfaced to one 8 bit memory chip

# 3.7.2. The EMC: Memory map and connections

- LPC178x: SRAM connection




16 bit wide memory bank interfaced to two 8 bit memory chips

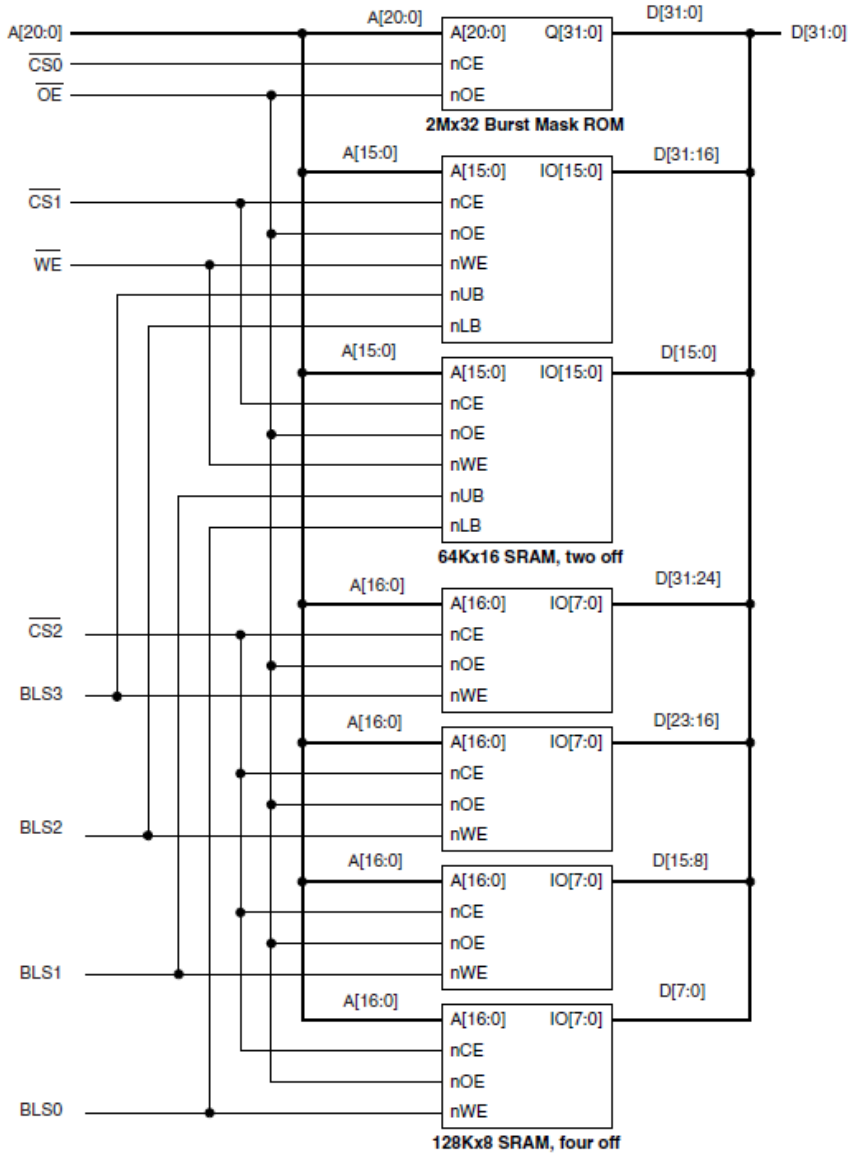



**8 bit bank external memory interface (bits MW = 00)**

# 3.7.2. The EMC: Memory map and connections

- LPC178x:

  Implementation example



# 3.7.3. The EMC: Registers

- EMC configuration (area enabled, data port width, access delays, …)
  - Performed through registers, depending on the external map available, and thus, on the specific device
  - In LPC178x:

| Name | Access | Offset | After reset | After boot | Description | |
|---|---|---|---|---|---|---|
| CONTROL | R/W | 0x000 | 0x3 | 0x1 | Controls operation of the memory controller | GENERAL |
| STATUS | RO | 0x004 | 0x5 | 0x5 | Provides EMC status information | |
| CNFIG | R/W | 0x008 | 0 | 0 | Configures operation of the memory controller | |
| STATICEXTENDEDWAIT | R/W | 0x80 | 0 | 0 | Selects time for long static memory read and write transfers | |
| STATICCONFIG0 | R/W | 0x200 | 0 | 0x81 | Selects the memory configuration for chip select 0 | For each CS |
| STATICWAITWEN0 | R/W | 0x204 | 0 | 0 | Selects the delay from chip select 0 to a write enable | |
| STATICWAITOEN0 | R/W | 0x208 | 0 | 0 | Selects the delay from chip select 0 or address change, whichever is later, to output enable | |
| STATICWAITRD0 | R/W | 0x20C | 0x1F | 0xE | Selects the delay from chip select 0 to a read access | |
| STATICWAITPAGE0 | R/W | 0x210 | 0x1F | 0x1F | Selects the delay for asynchonous page mode sequential accesses for chip select 0 | |
| STATICWAITWR0 | R/W | 0x214 | 0x1F | 0x1F | Selects the delay from chip select 0 to a write access | |
| STATICWAITTURN0 | R/W | 0x218 | 0xF | 0xF | Selects the number of bus tumaround cycles for chip select 0 | |

# 3.7.3. The EMC: Registers

## EMC configuration in LPC178x: General registers

| Name | Description |
|---|---|
| CONTROL | • EMC enable<br>• Low-power mode |
| STATUS | • **Busy.** EMC is performing memory transactions, commands, auto-refresh cycles, etc.<br>• **Write buffer status.** Write buffers contain data<br>• **Self-refresh acknowledge.** Indicates the operating mode of the EMC, self-refresh or normal |
| CNFIG | • **Endian mode.** Little or big |
| STATICEXTENDEDWAIT | • **Extended wait time out** (10 bits). Reset value: 0x0 = 16 clock cycles<br>Value to be programmed = (memory transfer time x CCLK frequency) / 16 -1 |

For example, for a static memory read/write transfer time of 16 µs, and a CCLK frequency of 50 MHz, the following value must be programmed into this register:
$(16 \times 10^{-6} \times 50 \times 10^{6}) / 16 - 1 = 49$.

# 3.7.3. The EMC: Registers

- EMC configuration in LPC178x: For each CS, basic configuration

| Name | Description |
|---|---|
| STATICCONFIG0 | • Memory width (1:0)<br>0x0: 8 bit (reset)<br>0x1: 16 bit<br>0x2: 32 bit<br><br>• Page mode (3). The EMC can burst up to 4 external accesses<br>0x0: Async page mode disable (reset)<br>0x1: Async page mode enabled<br><br>• Chip select polarity (6)<br>0x0: Active LOW chip select<br>0x1: Active HIGH chip select<br><br>• Byte lane state (7). Enables different types of memory to be connected<br>0x0: For reads all the bits in BLSn[3:0] are HIGH. For writes all the bits in BLSn[3:0] are LOW<br>0x1: For reads and writes the respective active bits in BLSn[3:0] are LOW<br><br>• Extended wait (8). Uses StaticExtendedWait register to time both read and write transfers rather than the StaticWaitRd and StaticWaitWr registers<br>0x0: Extended wait disabled (reset value)<br>0x1: Extended wait enabled |

# 3.7.3. The EMC: Registers

- **EMC configuration in LPC178x:** For each CS, basic configuration

**Table 134. Static Memory Configuration registers (STATICCONFIG[0:3], address 0x2009 C200 (STATICCONFIG0), 0x2009 C220 (STATICCONFIG1), 0x2009 C240 (STATICCONFIG2), 0x2009 C260 (STATICCONFIG3)) bit description**

| Bit | Symbol | Value | Description | Reset Value |
|---|---|---|---|---|
| 1:0 | MW | | Memory width. | 0 |
| | | 0x0 | 8 bit (POR reset value). | |
| | | 0x1 | 16 bit. | |
| | | 0x2 | 32 bit. | |
| | | 0x3 | Reserved. | |
| 2 | - | | Reserved. Read value is undefined, only zero should be written. | NA |
| 3 | PM | | Page mode. In page mode the EMC can burst up to four external accesses. Therefore devices with asynchronous page mode burst four or higher devices are supported. Asynchronous page mode burst two devices are not supported and must be accessed normally. | 0 |
| | | 0 | Disabled (POR reset value). | |
| | | 1 | Asynchronous page mode enabled (page length four). | |
| 5:4 | - | | Reserved. Read value is undefined, only zero should be written. | NA |
| 6 | PC | | Chip select polarity. The value of the chip select polarity on power-on reset is 0. | 0 |
| | | 0 | Active LOW chip select. | |
| | | 1 | Active HIGH chip select. | |
| 7 | PB | | Byte lane state. The byte lane state bit, PB, enables different types of memory to be connected. For byte-wide static memories the $\overline{BLS3:0}$ signal from the EMC is usually connected to $\overline{WE}$ (write enable). In this case for reads all the $\overline{BLS3:0}$ bits must be HIGH. This means that the byte lane state (PB) bit must be LOW.<br>16 bit wide static memory devices usually have the $\overline{BLS3:0}$ signals connected to the UBn and LBn (upper byte and lower byte) signals in the static memory. In this case a write to a particular byte must assert the appropriate UBn or LBn signal LOW. For reads, all the $\overline{UB}$ and $\overline{LB}$ signals must be asserted LOW so that the bus is driven. In this case the byte lane state (PB) bit must be HIGH. | 0 |
| | | 0 | For reads all the bits in $\overline{BLS3:0}$ are HIGH. For writes the respective active bits in $\overline{BLS3:0}$ are LOW (POR reset value). | |
| | | 1 | For reads the respective active bits in $\overline{BLS3:0}$ are LOW. For writes the respective active bits in $\overline{BLS3:0}$ are LOW. | |

# 3.7.3. The EMC: Registers

- LPC178x: static memory R/W access

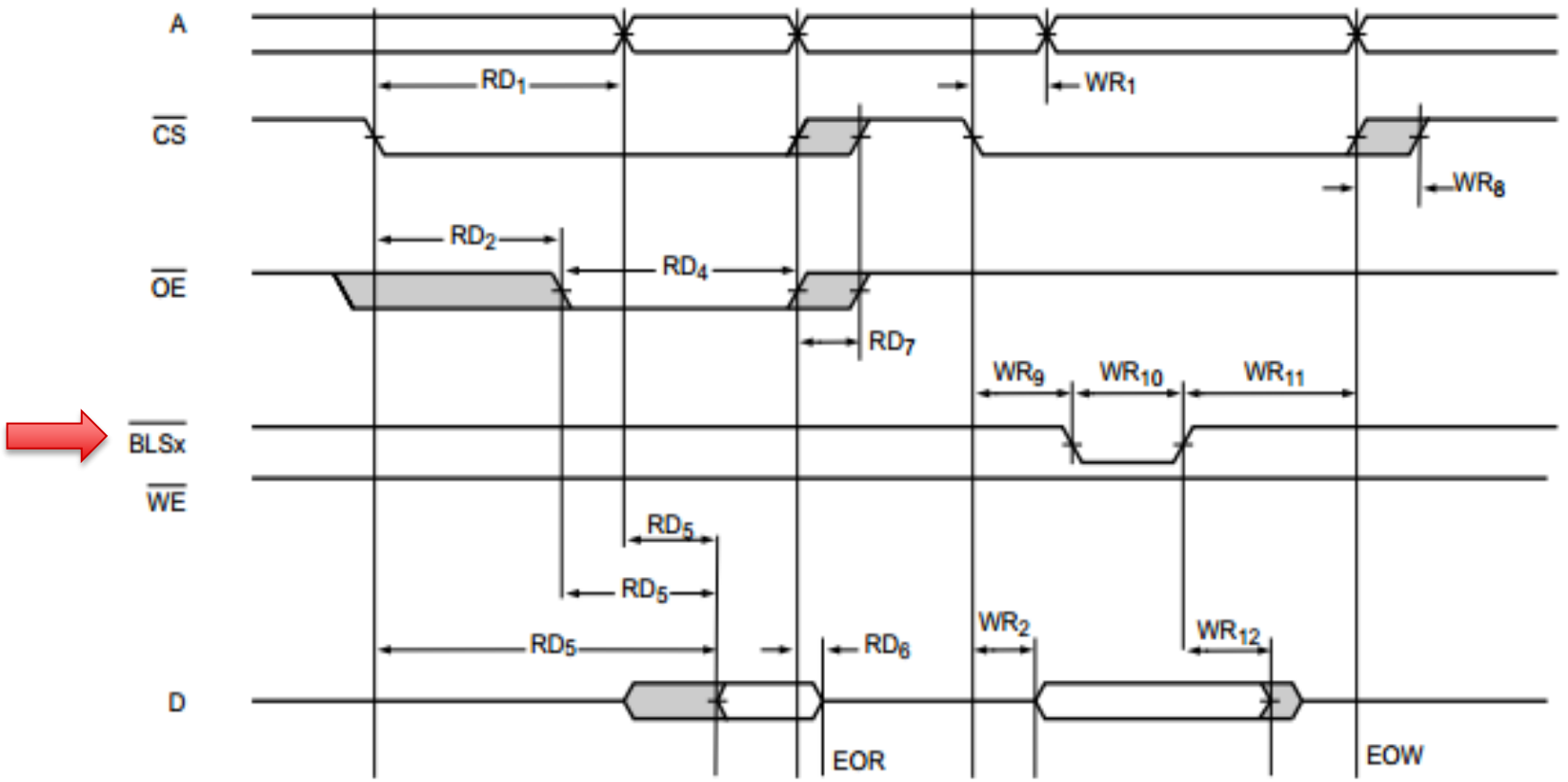



Fig 16. External static memory read/write access (PB = 0)

# 3.7.3. The EMC: Registers

- LPC178x: static memory R/W access

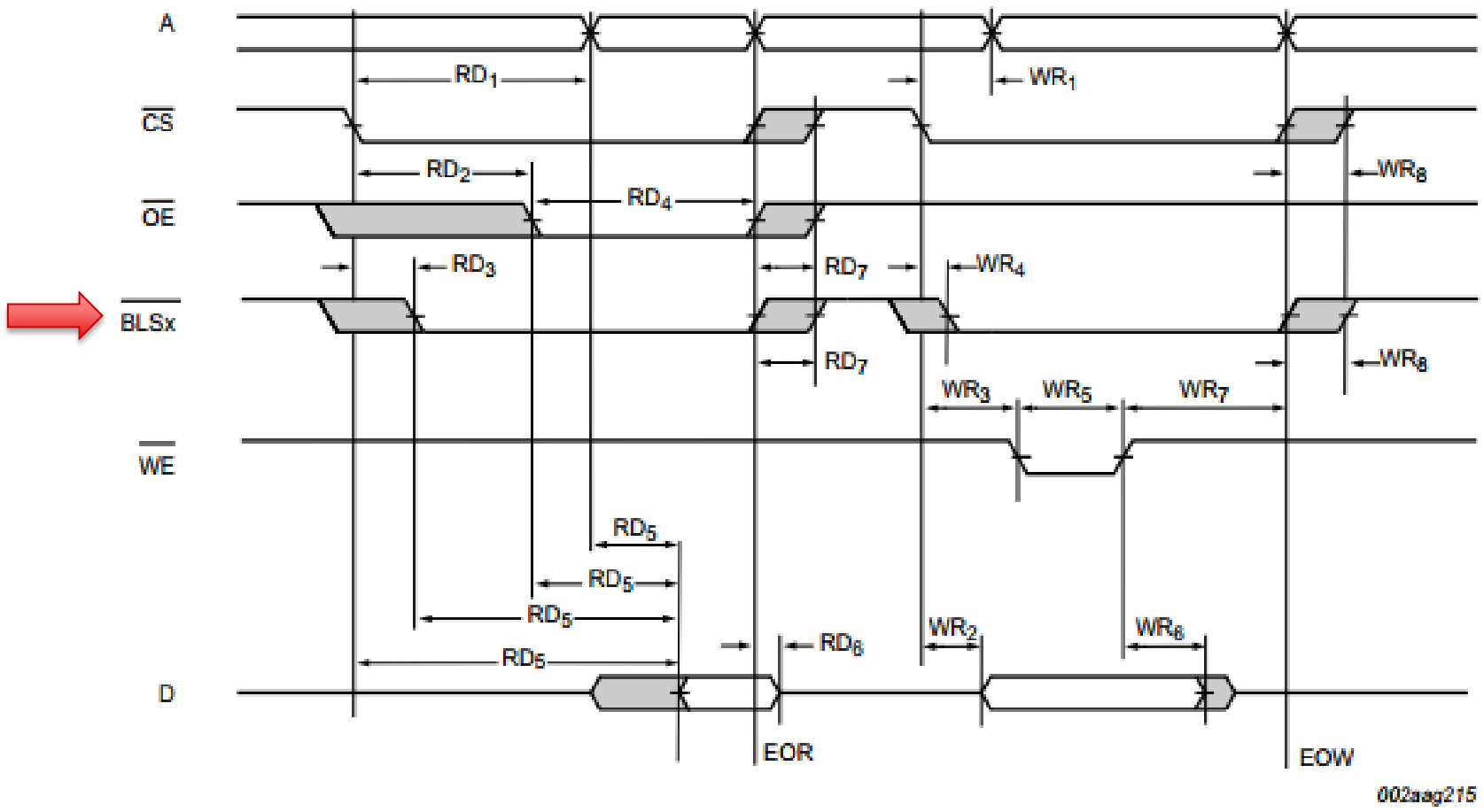



Fig 17. External static memory read/write access (PB =1)

# 3.7.3. The EMC: Registers

- EMC configuration in LPC178x: For each CS, timing configuration

**Table 309. Static Memory Write Enable Delay registers (STATICWAITWEN[0:3], address 0x4000 5204 (STATICWAITWEN0), 0x4000 5224 (STATICWAITWEN1), 0x4000 5244 (STATICWAITWEN2), 0x4000 5264 (STATICWAITWEN3)) bit description**

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 3:0 | WAITWEN | Wait write enable.<br>Delay from chip select assertion to write enable.<br>0x0 = One CCLK cycle delay between assertion of chip select and write enable (POR reset value).<br>0x1 - 0xF = (n + 1) CCLK cycle delay. The delay is (WAITWEN +1) x tCCLK. | 0x0 |
| 31:4 | - | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | - |

# 3.7.3. The EMC: Registers

- EMC configuration in LPC178x: For each CS, timing configuration

**Table 310. Static Memory Output Enable delay registers (STATICWAITOEN[0:3], address 0x4000 5208 (STATICWAITOEN0), 0x4000 5228 (STATICWAITOEN1), 0x4000 5248 (STATICWAITOEN2), 0x4000 5268 (STATICWAITOEN3)) bit description**

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 3:0 | WAITOEN | Wait output enable.<br>Delay from chip select assertion to output enable.<br>0x0 = No delay (POR reset value).<br>0x1 - 0xF = n cycle delay. The delay is WAITOEN x tCCLK. | 0x0 |
| 31:4 | - | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | - |

# 3.7.3. The EMC: Registers

- **EMC configuration in LPC178x:** For each CS, timing configuration

**Table 311. Static Memory Read Delay registers (STATICWAITRD[0:3], address 0x4000 520C (STATICWAITRD0), 0x4000 522C (STATICWAITRD1), 0x4000 524C (STATICWAITRD2), 0x4000 526C (STATICWAITRD3)) bit description**

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 4:0 | WAITRD | Non-page mode read wait states or asynchronous page mode read first access wait state.<br>Non-page mode read or asynchronous page mode read, first read only:<br>0x0 - 0x1E = (n + 1) CCLK cycles for read accesses. For non-sequential reads, the wait state time is (WAITRD + 1) x tCCLK.<br>0x1F = 32 CCLK cycles for read accesses (POR reset value). | 0x1F [1] |
| 31:5 | - | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | - |

[1] The reset value depends on the boot mode.

# 3.7.3. The EMC: Registers

- EMC configuration in LPC178x: For each CS, timing configuration

**Table 312. Static Memory Page Mode Read Delay registers (STATICWAITPAGE[0:3], address 0x4000 5210 (STATICWAITPAGE0), 0x4000 5230 (STATICWAITPAGE1), 0x4000 5250 (STATICWAITPAGE2), 0x4000 5270 (STATICWAITPAGE3)) bit description**

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 4:0 | WAITPAGE | Asynchronous page mode read after the first read wait states. Number of wait states for asynchronous page mode read accesses after the first read:<br>0x0 - 0x1E = (n+ 1) CCLK cycle read access time. For asynchronous page mode read for sequential reads, the wait state time for page mode accesses after the first read is (WAITPAGE + 1) x tCCLK.<br>0x1F = 32 CCLK cycle read access time (POR reset value). | 0x1F |
| 31:5 | - | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | - |

# 3.7.3. The EMC: Registers

- EMC configuration in LPC178x: For each CS, timing configuration

**Table 313. Static Memory Write Delay registers (STATICWAITWR[0:3], address 0x4000 5214 (STATICWAITWR0), 0x4000 5234 (STATICWAITWR1), 0x4000 5254 (STATICWAITWR2), 0x4000 5274 (STATICWAITWR3)) bit description**

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 4:0 | WAITWR | Write wait states.<br>SRAM wait state time for write accesses after the first read:<br>0x0 - 0x1E = (n + 2) CCLK cycle write access time. The wait state time for write accesses after the first read is WAITWR (n + 2) x tCCLK.<br>0x1F = 33 CCLK cycle write access time (POR reset value). | 0x1F |
| 31:5 | - | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | - |

# 3.7.3. The EMC: Registers

- **EMC configuration in LPC178x:** For each CS, timing configuration

**Table 314. Static Memory Turn-around Delay registers (STATICWAITTURN[0:3], address 0x4000 5218 (STATICWAITTURN0), 0x4000 5238 (STATICWAITTURN1), 0x4000 5258 (STATICWAITTURN2), 0x4000 5278 (STATICWAITTURN3)) bit description**

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 3:0 | WAITTURN | Bus turn-around cycles.<br>0x0 - 0xE = (n + 1) CCLK turn-around cycles. Bus turn-around time is (WAITTURN + 1) x tCCLK.<br>0xF = 16 CCLK turn-around cycles (POR reset value). | 0xF |
| 31:4 | - | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | - |

Bus turn-around cycles are generated between external bus transfers in the following situations when at least one of the memory banks is static memory:

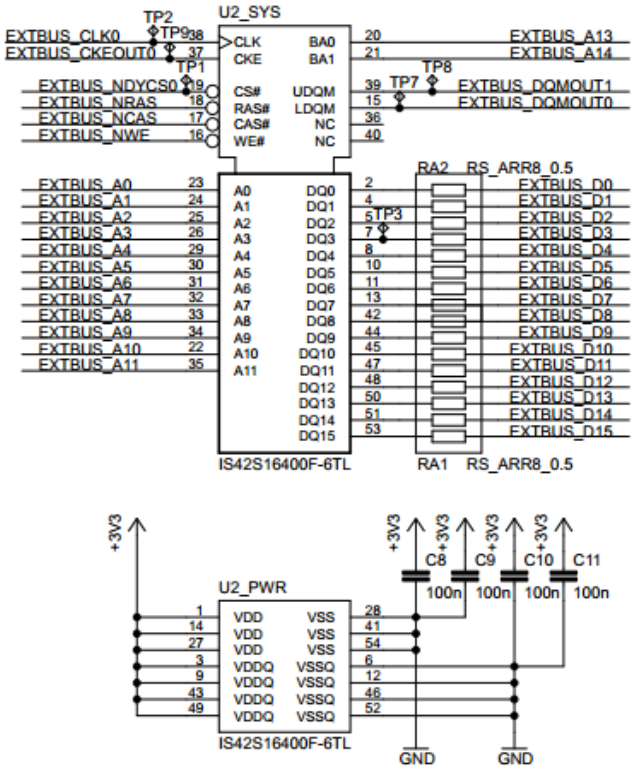
- between read and read to different memory banks
- between read and write to the same memory bank
- between read and write to different memory banks

Bus turn-around cycles prevent bus contention on the external memory data bus.

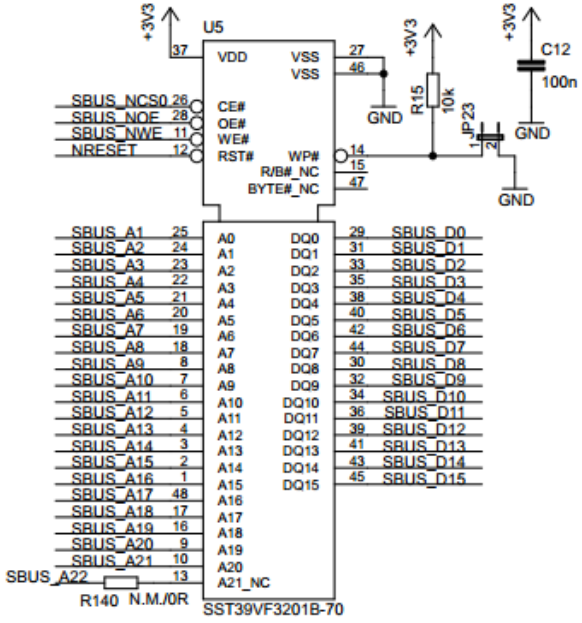
# 3.7.4. The EMC: Example

- Analyze the schematic of external memory in a LPC1850 board:

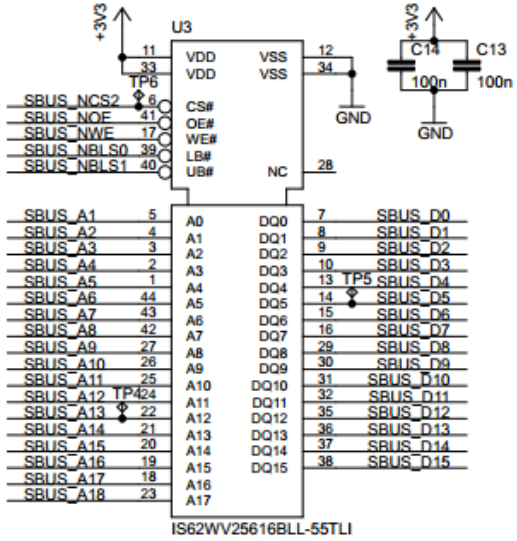
SDRAM




FLASH

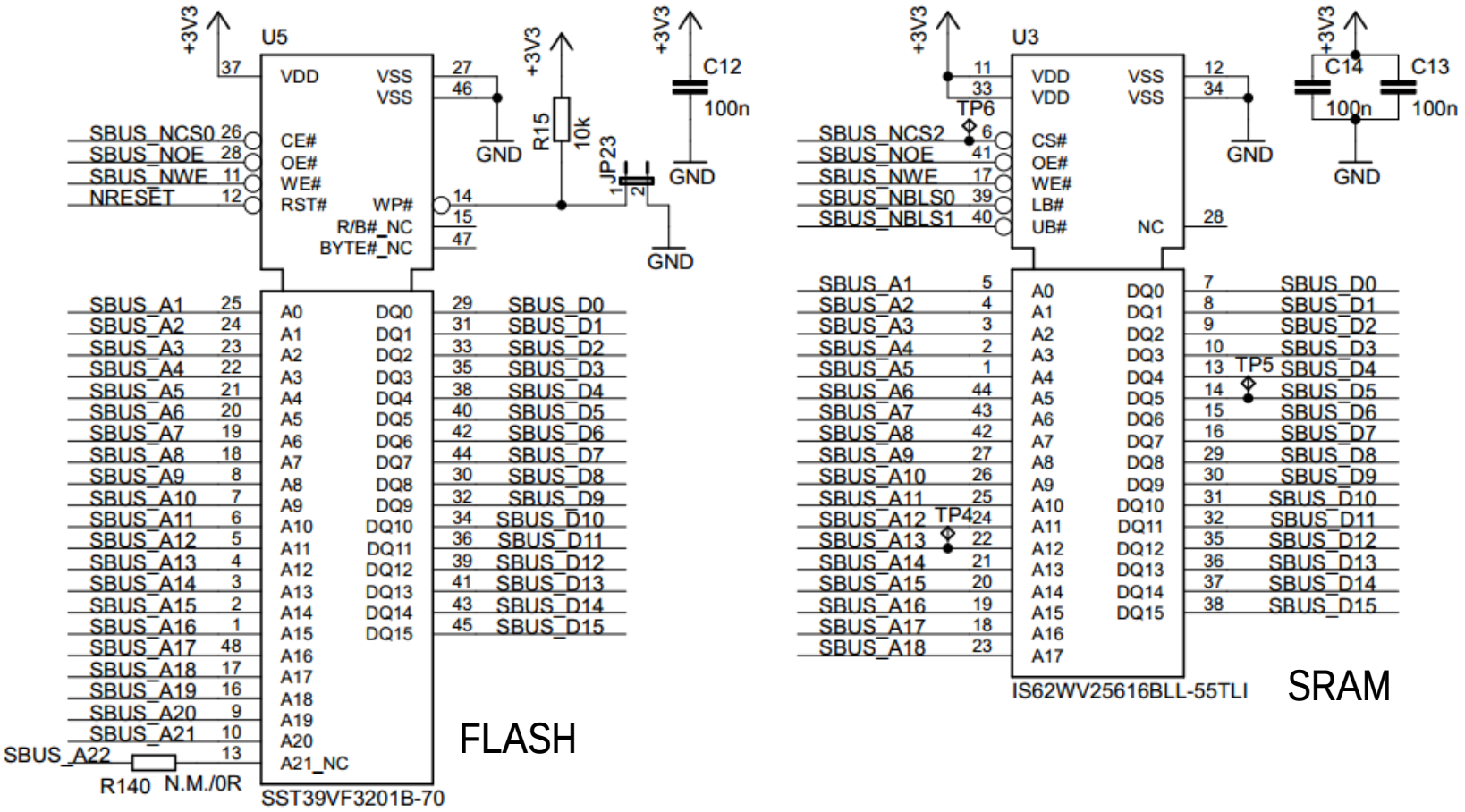



SRAM



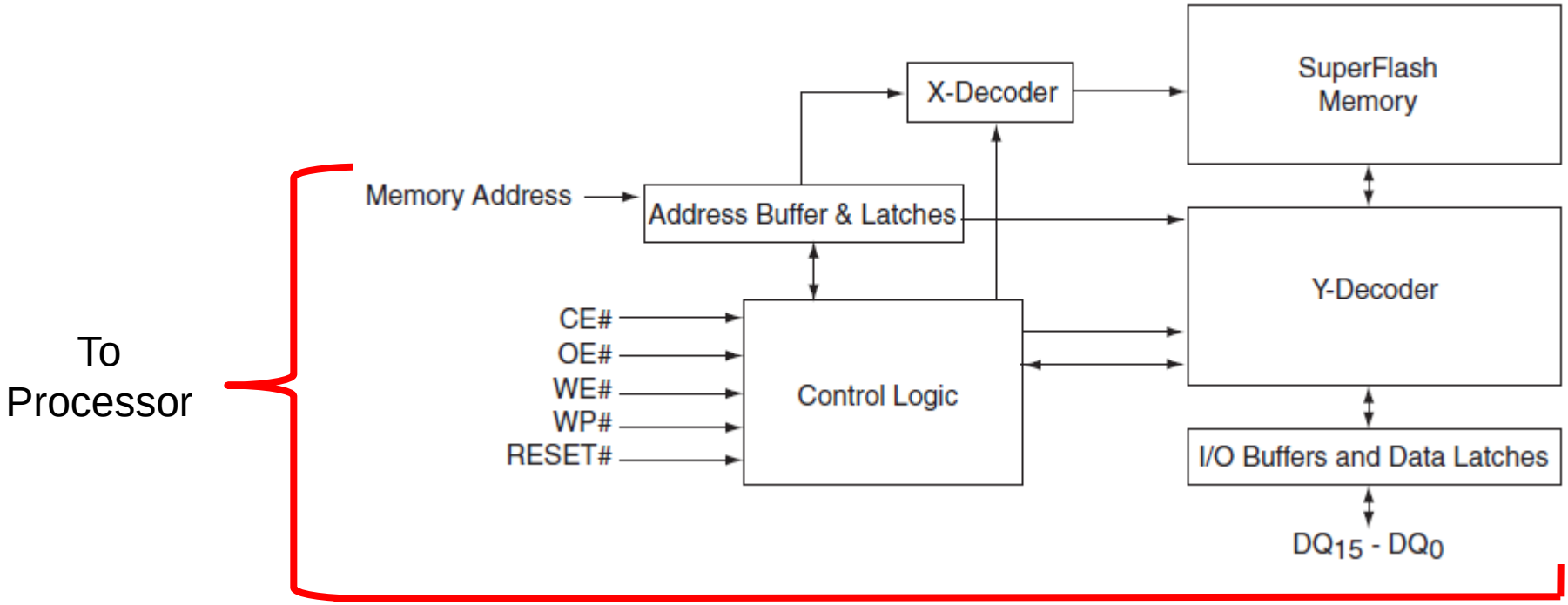
# 3.7.4. The EMC: Example

- Analyze the schematic of external memory in a LPC1850 board:



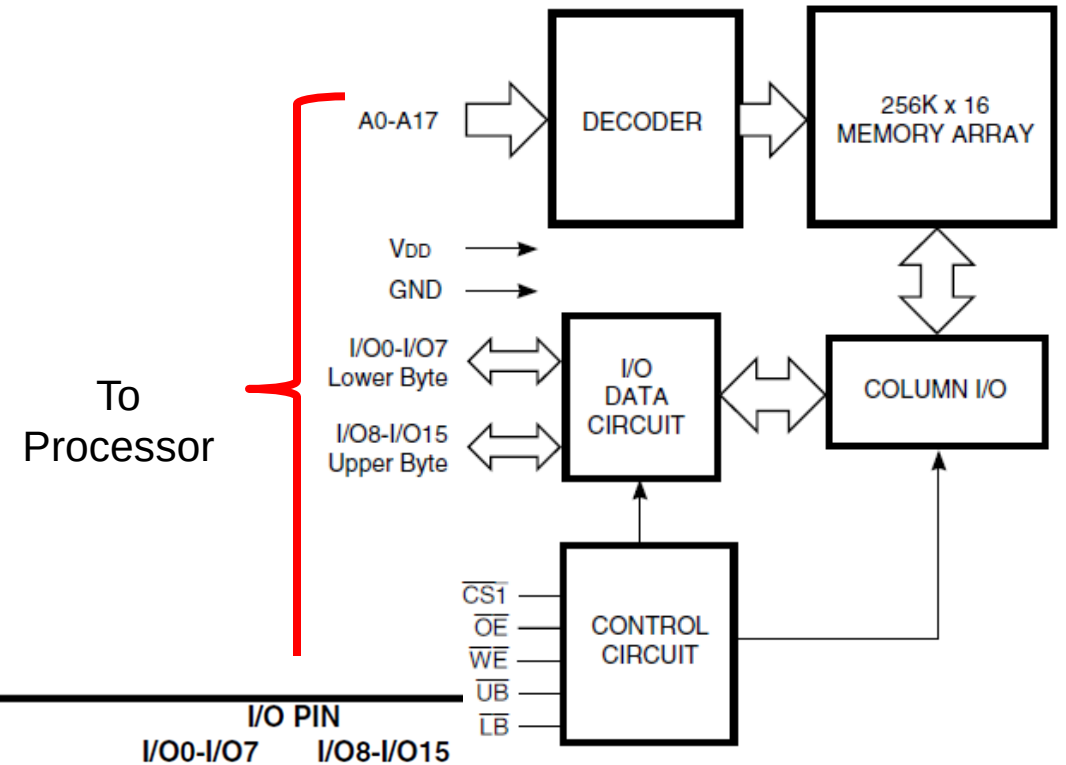
# 3.7.4. The EMC: Example

## SST39VF3201B



# 3.7.4. The EMC: Example

## IS62WV25616




**TRUTH TABLE**

| Mode | WE | CS1 | OE | LB | UB | I/O PIN I/O0-I/O7 | I/O PIN I/O8-I/O15 |
|---|---|---|---|---|---|---|---|
| Not Selected | X | H | X | X | X | High-Z | High-Z |
| | X | X | X | H | H | High-Z | High-Z |
| Output Disabled | H | L | H | L | X | High-Z | High-Z |
| | H | L | H | X | L | High-Z | High-Z |
| Read | H | L | L | L | H | DOUT | High-Z |
| | H | L | L | H | L | High-Z | DOUT |
| | H | L | L | L | L | DOUT | DOUT |
| Write | L | L | X | L | H | DIN | High-Z |
| | L | L | X | H | L | High-Z | DIN |
| | L | L | X | L | L | DIN | DIN |

# 3.7.4. The EMC: Example

1. Fill in the following table related to the used external memory map:

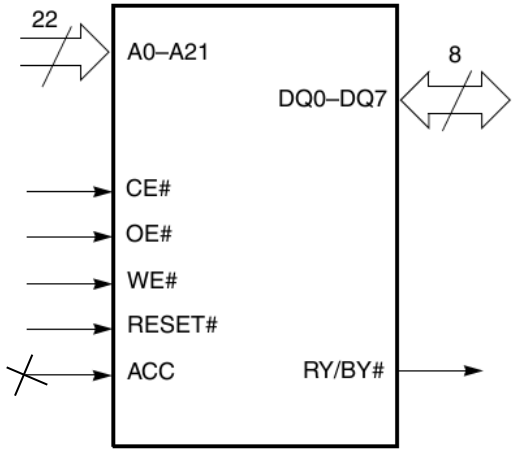
| Memory (chip) | Type | Capacity | Initial Add. | Data length |
|---|---|---|---|---|
| SST39VF3201B | | | | |
| IS62WV25616BLL | | | | |

2. Fill in the following table related to the type of decoding used in the schematic:

| Memory (chip) | Decoding Type | Logic address of the first position in the chip |
|---|---|---|
| SST39VF3201B | | |
| IS62WV25616BLL | | |

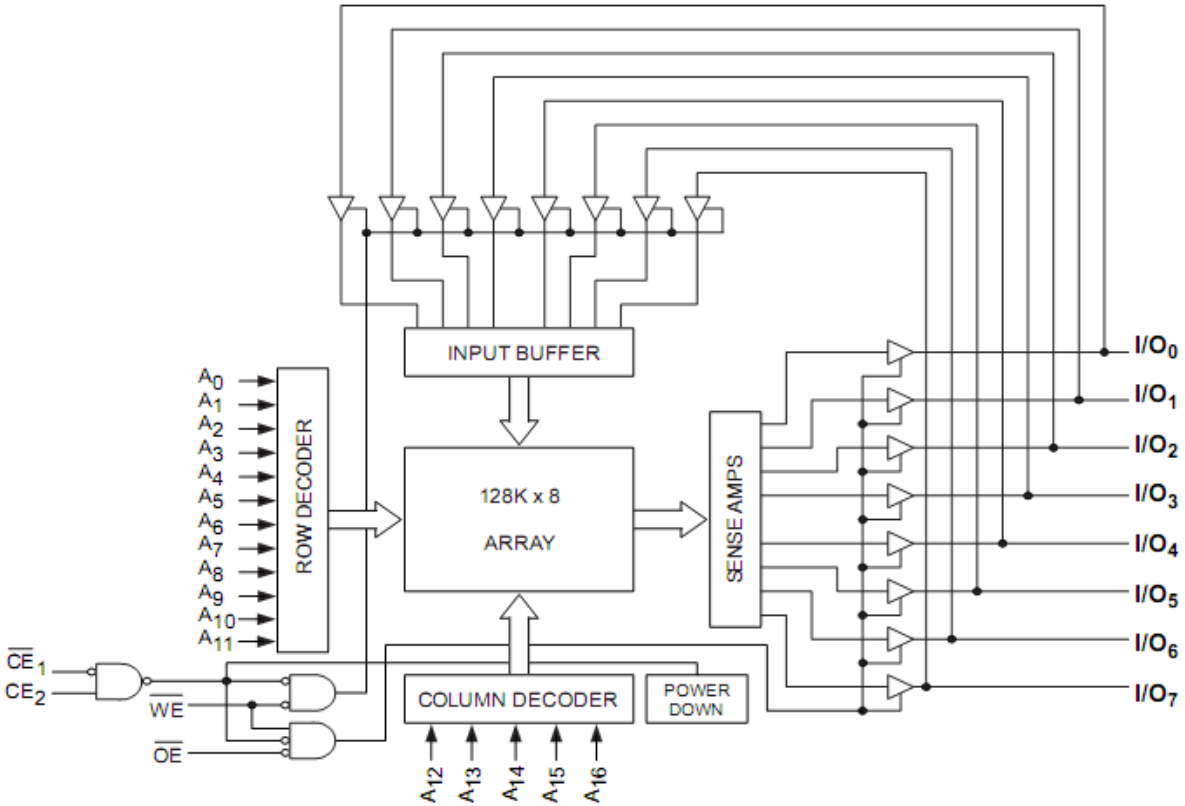
# 3.7.4. The EMC: Example

3. Modify the schematic in order to implement the space occupied by the SST39VF3201B-70 by chips of the S29AL032D70TAI00, preserving the data length used in that space



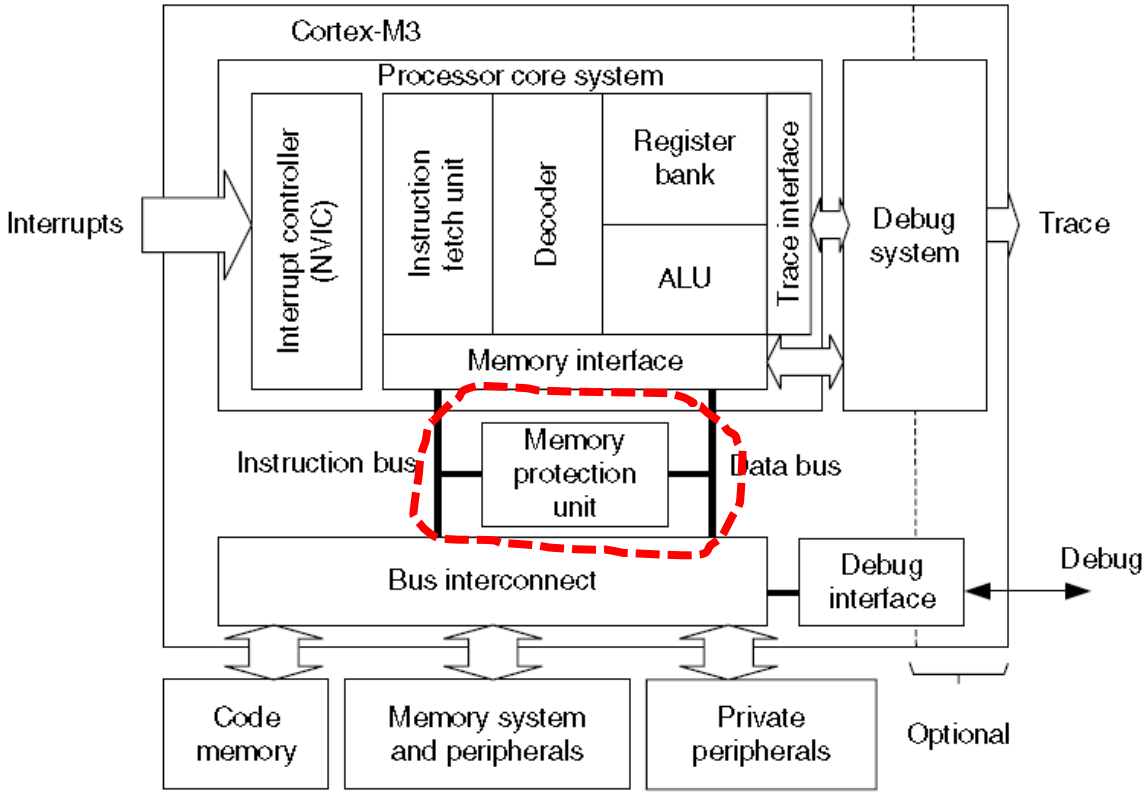
# 3.7.4. The EMC: Example

4. Do the same as in 3, implementing the space occupied by the IS62WV25616BLL-55TLI by chips of the CY62128E, preserving the data length used in that space



# 3.8. The MPU. General description

- How does MPU work?
  - Monitors I-bus, D-bus and System-bus



# 3.8. The MPU. General description

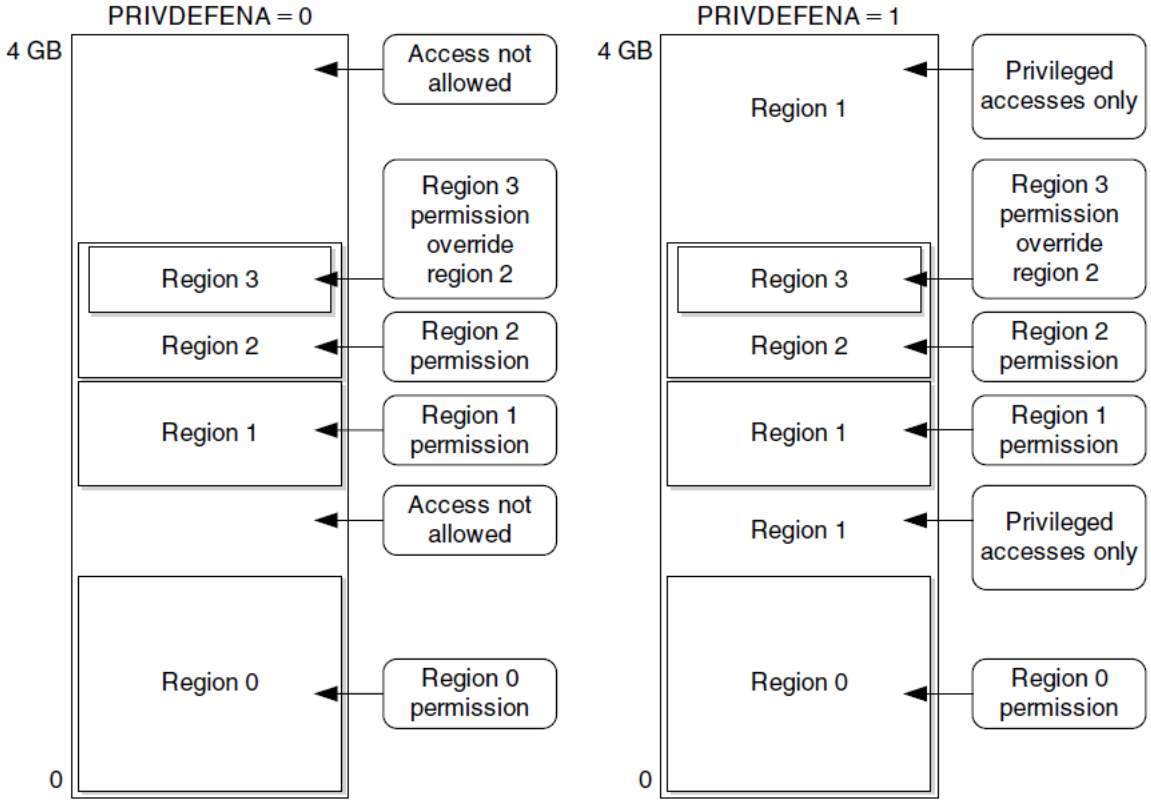
- The Memory Protection Unit (MPU) is a common element in complex embedded systems that…
  - Provides memory protection features, making based systems more robust
  - Included in Cortex-M3 (and LPC17XX specifically) and many other µC
  - Sets up the protection by defining the memory map as a number of regions, with a specific functionality, and type of use and accesses allowed
    - Up to 8 regions can be defined
    - MPU regions can be overlapped. If a memory location falls on two regions, the memory access attributes and permission will be the highest-numbered region
    - It is also possible to define a default background memory map for privileged accesses (PRIVDEFENA)
  - Accesses to memory locations that are not defined in the MPU or not permitted will cause the Memory Management Fault

# 3.8. The MPU. General description

- What is MPU used for?
  - Preventing user applications from corrupting data used by the operating system
  - Separating data between tasks by blocking from accessing others' data
  - Allowing memory regions to be defined as read-only so that data can be protected
  - Detecting unexpected memory accesses (for example, stack corruption)

# 3.8. The MPU. General description

- An example of use:
  - PRIVDEFENA differs between normal map and default background map for privileged accesses



# 3.8.1. The MPU. Registers

- MPU has to be configured in order to be enabled

**Table 13.1** MPU Type Register (0xE000ED90)

| Bits | Name | Type | Reset Value | Description |
|---|---|---|---|---|
| 23:16 | IREGION | R | 0 | Number of instruction regions supported by this MPU; because ARMv7-M architecture uses a unified MPU, this is always 0 |
| 15:8 | DREGION | R | 0 or 8 | Number of regions supported by this MPU; in the Cortex-M3, this is either 0 (MPU not present) or 8 (MPU present) |
| 0 | SEPARATE | R | 0 | This is always 0, as the MPU is unified |

**Table 13.2** MPU Control Register (0xE000ED94)

| Bits | Name | Type | Reset Value | Description |
|---|---|---|---|---|
| 2 | PRIVDEFENA | R/W | 0 | Privileged default memory map enable; when set to 1 and if the MPU is enabled, the default memory map will be used for privileged accesses as a background region. If this bit is not set, the background region is disabled and any access not covered by any enabled region will cause a fault. |
| 1 | HFNMIENA | R/W | 0 | If set to 1, it enables the MPU during the hard fault handler and nonmaskable interrupt (NMI) handler; otherwise, the MPU is not enabled (bypassed) for the hard fault handler and NMI. |
| 0 | ENABLE | R/W | 0 | It enables the MPU if set to 1. |

# 3.8.1. The MPU. Registers

**Table 13.4** MPU Region Base Address Register (0xE000ED9C)

| Bits | Name | Type | Reset Value | Description |
|---|---|---|---|---|
| 31:N | ADDR | R/W | — | Base address of the region; N is dependent on the region size—for example, a 64 KB size region will have a base address field of [31:16]. |
| 4 | VALID | R/W | — | If this is 1, the REGION defined in bit [3:0] will be used in this programming step; otherwise, the region selected by the MPU Region Number register is used. |
| 3:0 | REGION | R/W | — | This field overrides the MPU Region Number register if VALID is 1; otherwise, it is ignored. Because eight regions are supported in the Cortex-M3 MPU, the region number override is ignored if the value of the REGION field is larger than 7. |

**Table 13.5** MPU Region Base Attribute and Size Register (0xE000EDA0)

| Bits | Name | Type | Reset Value | Description |
|---|---|---|---|---|
| 31:29 | Reserved | — | — | — |
| 28 | XN | R/W | — | Instruction Access Disable (1 = disable instruction fetch from this region; an attempt to do so will result in a memory management fault) |
| 27 | Reserved | — | — | — |
| 26:24 | AP | R/W | — | Data Access Permission field |
| 23:22 | Reserved | — | — | — |
| 21:19 | TEX | R/W | — | Type Extension field |
| 18 | S | R/W | — | Shareable |
| 17 | C | R/W | — | Cacheable |
| 16 | B | R/W | — | Bufferable |
| 15:8 | SRD | R/W | — | Subregion disable |
| 7:6 | Reserved | — | — | — |
| 5:1 | REGION SIZE | R/W | — | MPU Protection Region size |
| 0 | ENABLE | R/W | — | Region enable |