
# eSi-SMI

# 1 Contents



 

# 2 Overview

The eSi-SMI core provides a static memory interface, allowing accesses to off-chip memories such as asynchronous RAMs, ROMs and parallel flashes. It supports the following features:

- Configurable number of data pins (8/16/32).
- Configurable number of address pins.
- Configurable number of banks (1-8). Each bank has its own chip-select.
- Each bank has programmable settings, including: data width; wait states, write-protect and privilege-level.
- Dual AMBA 3 AHB-lite slave interface. One is for configuration and one is for memory access.

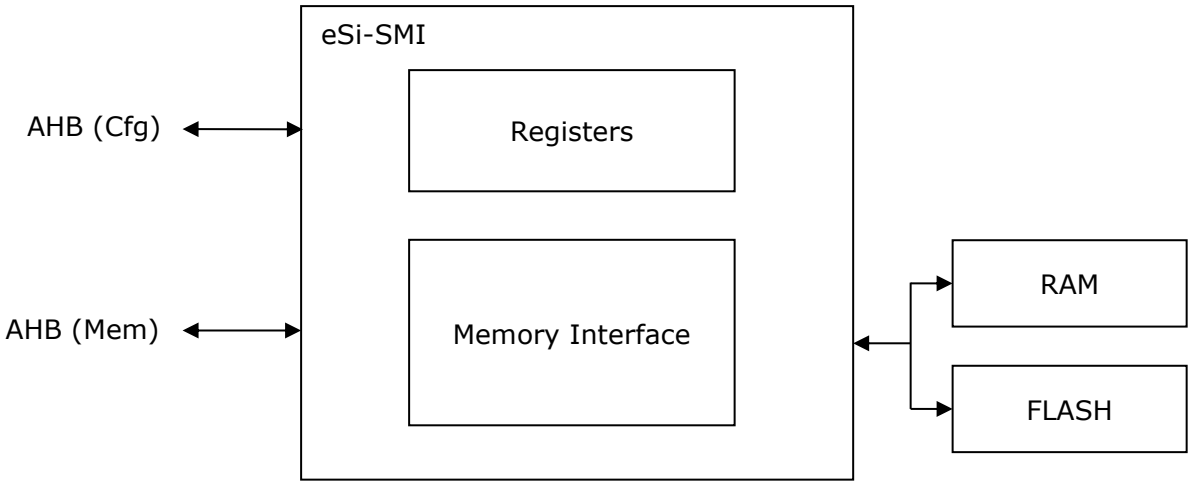


**Figure 1: eSi-SMI**

 

# 3 Hardware Interface

| | |
|---|---|
| **Module Name** | cpu_ahb_smi |
| **HDL** | Verilog |
| **Technology** | Generic |
| **Source Files** | cpu_ahb_smi.v |

| Port | Type | Values | Description |
|---|---|---|---|
| data_pins | Integer | 8, 16, 32 | Specifies the width of the external data bus |
| address_pins | Integer | 2 - BITS | Specifies the width of the external address bus |
| banks | Integer | 1 – 8 | Specifies the number of memory banks |

**Table 1: Parameters**

| Port | Direction | Width | Description |
|---|---|---|---|
| mem_hclk | Input | 1 | Memory interface, AHB clock |
| mem_hresetn | Input | 1 | Memory interface, AHB reset, active-low |
| mem_haddr | Input | BITS | Memory interface, AHB address |
| mem_hburst | Input | 3 | Memory interface, AHB burst type |
| mem_hmastlock | Input | 1 | Memory interface, AHB locked transfer |
| mem_hprot | Input | 4 | Memory interface, AHB protection |
| mem_hsize | Input | 3 | Memory interface, AHB size |
| mem_htrans | Input | 2 | Memory interface, AHB transfer type |
| mem_hwdata | Input | BITS | Memory interface, AHB write data |
| mem_hwrite | Input | 1 | Memory interface, AHB write |
| mem_hready | Input | 1 | Memory interface, AHB ready |
| mem_hsel | Input | 1 | Memory interface, AHB select |
| mem_hready | Output | 1 | Memory interface, AHB ready |
| mem_hrdata | Output | BITS | Memory interface, AHB read data |
| mem_hresp | Output | 1 | Memory interface, AHB response |
| cfg_hclk | Input | 1 | Configuration interface, AHB clock |
| cfg_hresetn | Input | 1 | Configuration interface, AHB reset, active-low |
| cfg_haddr | Input | BITS | Configuration interface, AHB address |
| cfg_hburst | Input | 3 | Configuration interface, AHB burst type |
| cfg_hmastlock | Input | 1 | Configuration interface, AHB locked transfer |
| cfg_hprot | Input | 4 | Configuration interface, AHB protection |
| cfg_hsize | Input | 3 | Configuration interface, AHB size |
| cfg_htrans | Input | 2 | Configuration interface, AHB transfer type |
| cfg_hwdata | Input | BITS | Configuration interface, AHB write data |
| cfg_hwrite | Input | 1 | Configuration interface, AHB write |
| cfg_hready | Input | 1 | Configuration interface, AHB ready |
| cfg_hsel | Input | 1 | Configuration interface, AHB select |
| cfg_hready | Output | 1 | Configuration interface, AHB ready |
| cfg_hrdata | Output | BITS | Configuration interface, AHB read data |
| cfg_hresp | Output | 1 | Configuration interface, AHB response |
| data_in | Input | data_pins | Data input |
| data_out | Output | data_pins | Data output |
| data_out_enable | Output | data_pins/8 | Data output enable |
| chip_enable_n | Output | banks | Chip enable, active-low |
| output_enable_n | Output | 1 | Output enable, active-low |
| write_enable_n | Output | 1 | Write enable, active-low |
| byte_enable_n | Output | data_pins/8 | Byte enable, active-low |
| address | Output | address_pins | Read/write address |

**Table 2: I/O Ports**

 

For complete details of the AHB signals, please refer to the AMBA 3 AHB-Lite Protocol v1.0 Specification available at:

http://www.arm.com/products/system-ip/amba/amba-open-specifications.php

 

# 4 Software Interface

## 4.1 Register Map

Each memory bank has its own set of control registers, as illustrated in Table 3: Register Map. In this table, `N` indicates the bank, which ranges from 0 to `banks`-1.

| Register | Address offset | Access | Description |
|---|---|---|---|
| `control[N]` | 0x10*N+0x00 | R/W | Control register |
| `read_wait[N]` | 0x10*N+0x04 | R/W | Read wait register |
| `write_wait[N]` | 0x10*N+0x08 | R/W | Write wait register |
| `wait[N]` | 0x10*N+0x0c | R/W | Wait register |

**Table 3: Register Map**

### 4.1.1 Control Register

The per-bank control register contains a selection of flags that control the operation of the memory interface when accesses addresses within the corresponding bank.

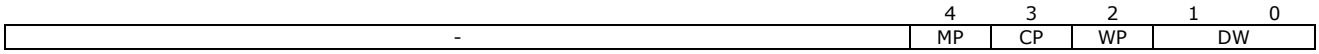


**Figure 2: Format of the `control` register**

| Register | Values | Description |
|---|---|---|
| `DW` | 0 – 8-bit<br>1 – 16-bit<br>2 – 32-bit | Data width |
| `WP` | 0 – Read / write<br>1 – Read-only | Write protection |
| `CP` | 0 – User<br>1 – Supervisor | Privilege level required to access the configuration registers |
| `MP` | 0 – User<br>1 – Supervisor | Privilege level required to access the memory |

**Table 4: Fields of the `control` register**

### 4.1.2 Read Wait Register

The per-bank read wait register contains the number of wait states for memory reads.

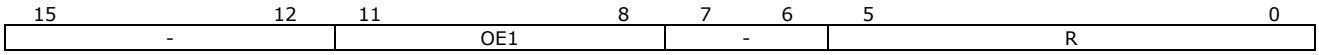


**Figure 3: Format of the `read_wait` register**

| Register | Values | Description |
|---|---|---|
| `R` | 0 – 63 | Read wait states |
| `OE1` | 0 – 15 | Output enable assert wait states |

**Table 5: Fields of the `read_wait` register**

 

### 4.1.3 Write Wait Register

The per-bank write wait register contains the number of wait states for memory writes.

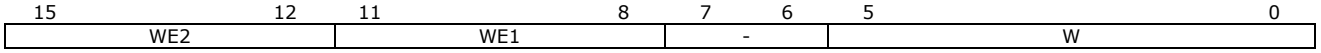


**Figure 4: Format of the `write_wait` register**

| Register | Values | Description |
|---|---|---|
| W | 0 – 63 | Write wait states |
| WE1 | 1 – 15 | Write enable assert wait states |
| WE2 | 1 – 15 | Write enable deassert wait states |

**Table 6: Fields of the `write_wait` register**

### 4.1.4 Wait Register

The per-bank wait register contains the number of wait states for one phase of the memory interface's operation.

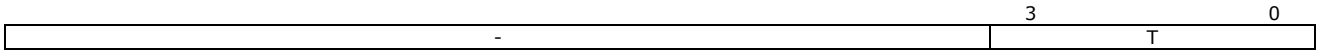


**Figure 5: Format of the `wait` register**

| Register | Values | Description |
|---|---|---|
| T | 1 – 15 | Read-to-write turn-around wait states |

**Table 7: Fields of the `wait` register**

 

# 5 Timing

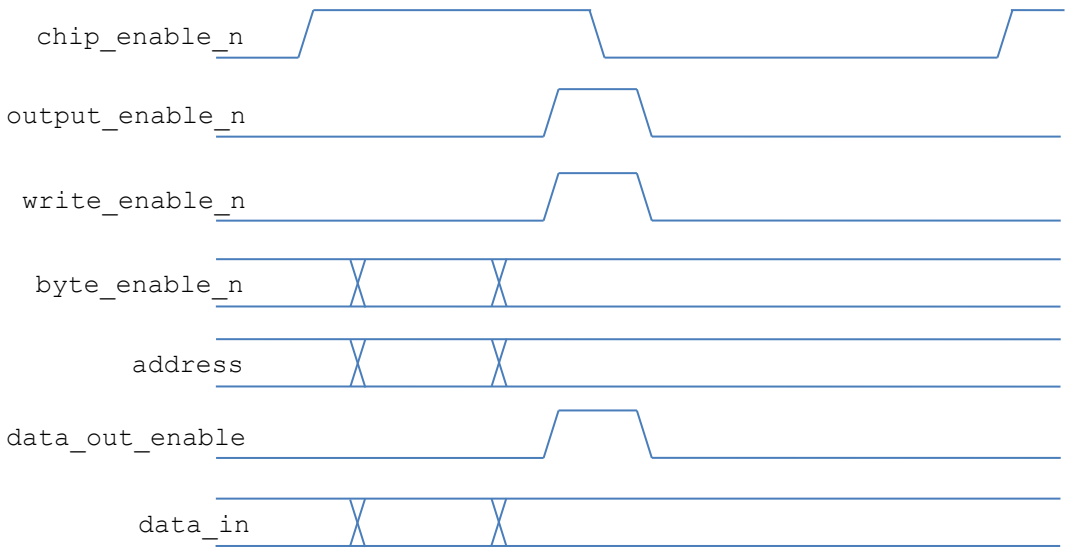


**Figure 6: Read Timing**

 

# 6 Connectivity

Figure 7: SMI Connectivity shows a typical example how the SMI can be connected to multiple banks of off-chip SRAM and flash. In this example, the first bank consists of two 256k x 16-bit SRAMs are connected in parallel to form a 1Mbyte 32-bit RAM, with the second bank consisting of a single 8MByte 16-bit flash.

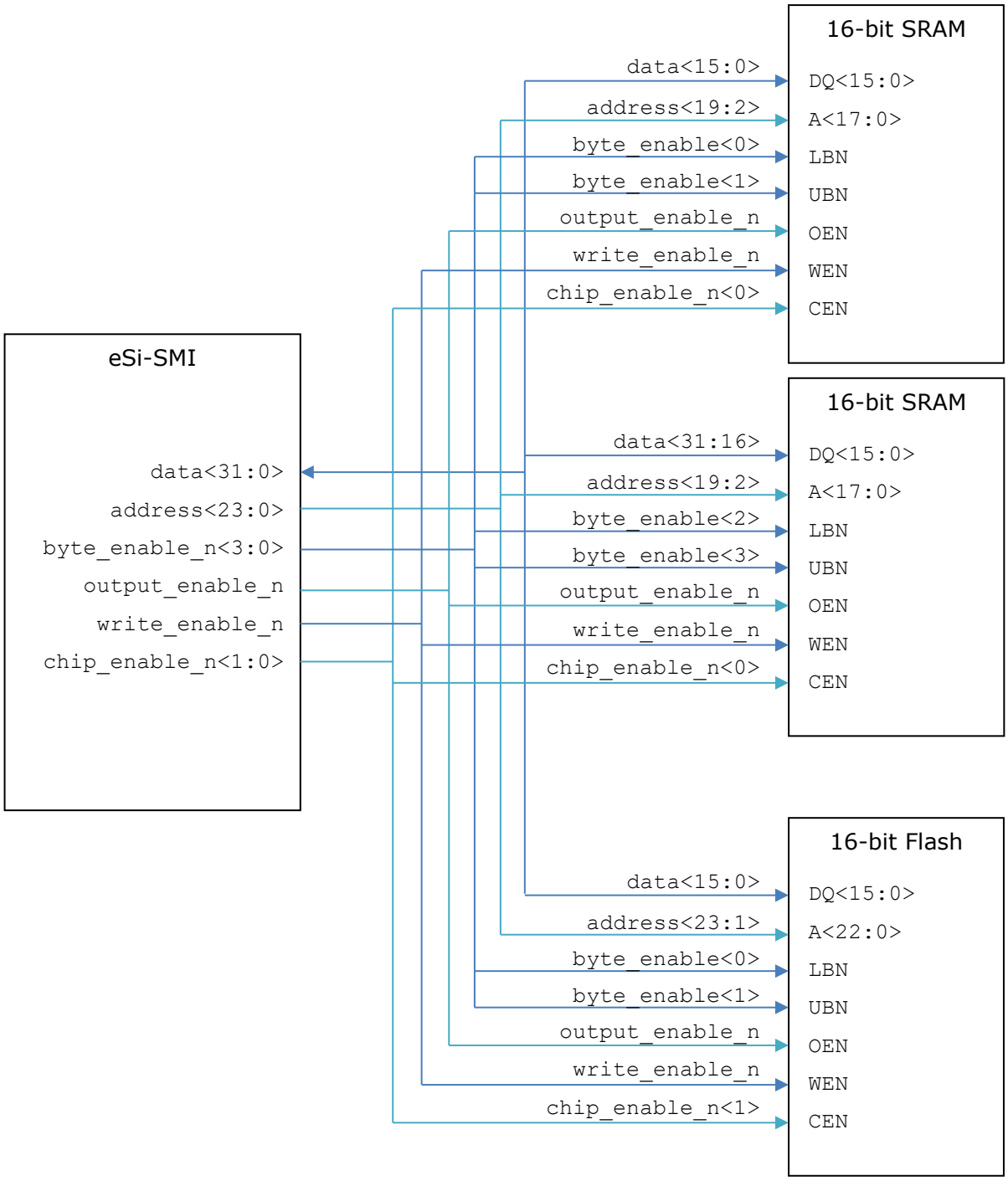


**Figure 7: SMI Connectivity**

 

# 7 Revision History

| Hardware Revision | Software Release | Description |
|---|---|---|
| 1 | 2.4.0 | Initial release |
| 1 | 2.8.10 | Remove unsupported `OE2` field |
| 2 | 6.0.3 | Added `control.MP` field<br>Added `control.WP` field |

**Table 8: Revision History**

 