The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205490

FCR4, ATLAS Series MB9DF126, External Bus Interface (EBI) - Accessing SRAM

This application note describes the usage of the External Bus Interface (EBI) for R4 Family. As an example the MB9DF126 MCU is used, hardware related matters will be described for the SK-MB9EF120-001 starter kit in combination with the MEM-FCR4-004 memory module.

Contents

1 Introduction

The chapter gives short overview of this application note.

This application note describes the usage of the External Bus Interface (EBI) for R4 Family. As an example the MB9DF126 MCU is used, hardware related matters will be described for the SK-MB9EF120-001 starter kit in combination with the MEM-FCR4-004 memory module.

The External Bus Interface (EBI) is a generic memory controller, which can access external memories like SRAM, FLASH or SDRAM as well as peripheral devices having a parallel bus interface. It supports access to up to 8 SRAM/FLASH and one SDRAM memory banks supporting following features:

- Supports 8/16/32-bit bus width for SRAM/FLASH
- Supports 16/32-bit SDRAM bus widths
- Supports big-endian access for SRAM/FLASH
- Provides 8 chip selects lines for SRAM/FLASH, 1 chip select line for SDRAM
- Supports NOR FLASH page accesses
- Supports SDRAM power down mode (automatic and interval setting are available)

The application note will focus on using SRAM memory with an accompanying Software Samples available at http://www.cypress.com/cypress-mcu-product-softwareexamples illuminating typical software settings based on the Low Level Library L3.

For more details on the External Bus Interface (EBI), further memory interface and timing examples, register settings as well as device specific limitations see the FCR4 hardware manual and to the device specific datasheet. Also consider the Low Level Library (L3) help files and the software examples full source code for software programming related issues.

2 Hardware Interface

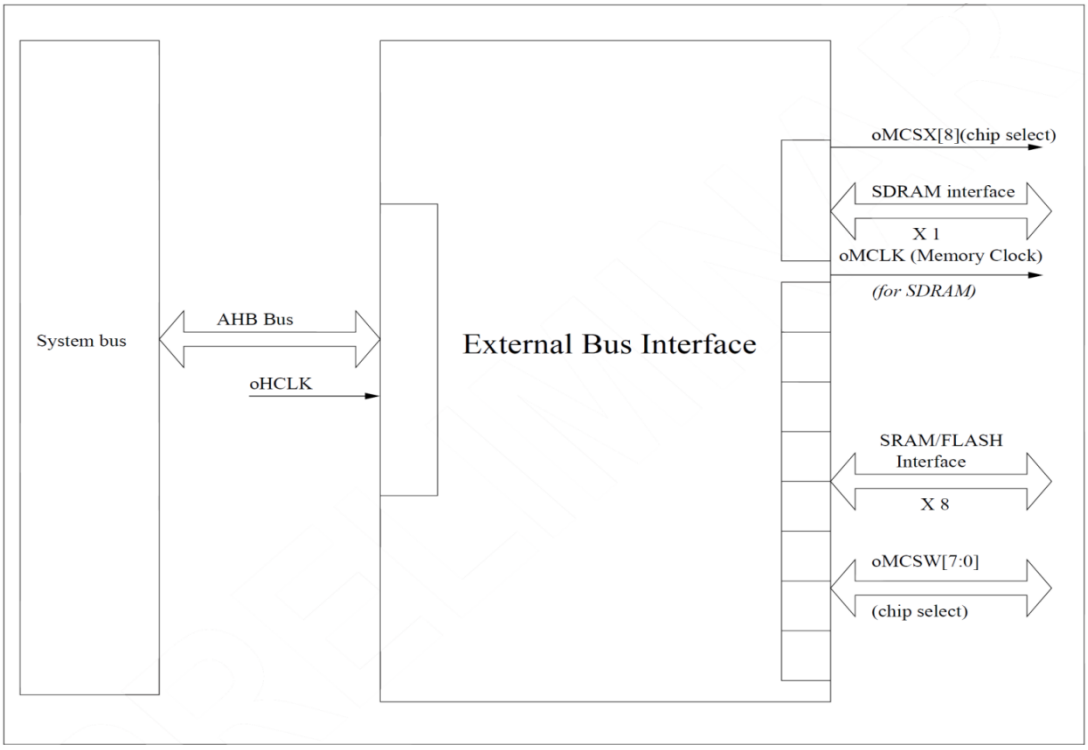
The chapter describes the hardware requirements for the External Bus Interface module.

2.1 Interface Description

The External Bus Interface module as a generic memory controller provides all necessary signals to attach various types of external memory devices like SRAM, SDRAM and FLASH memory as well as generic memory mapped peripheral devices e.g. Fujitsu MB88121 FLEXRAY[tm] controller.

This chapter provides general hardware related information about the FCR4 External Bus Interface module (EBI) while device specific matters will be described for the SK-MB9EF120-001 starter kit in combination with the MEM-FCR4-004 memory module in the following chapter.

Figure 2-1: General overview of the External Bus Interface (EBI)



The figure above shows the block diagram and signal groups provided:

- Address and Data Bus
- Chip Select Signals
- Control and Clock Signals grouped for the various memory types supported

For a more detailed view, see table 2-1 on next page:

Table 2-1: External Bus Interface (EBI) Module pins and Functions

Pin name	I/O	Function
oMAD[24:0]	O	Address output: Lower 16-bit are shared by SDRAM/SRAM
oMWDatb[31:0]	O	Data Port: write data
iMRDatb[31:0]	I	Data Port: read data
oMDQM[3:0]	O	SRAM: Byte mask signal
oMWEX	O	SRAM: Write enable signal
oMOEX	O	SRAM: Read enable
iMRDY	I	SRAM: Memory Ready
oMCSX[7:0]	O	SRAM: Chip select, shared with NAND
oMCLK	O	SDRAM: Clock Gated version of EBI HCLK
oMCASX	O	SDRAM: Column Address Select
oMRASX	O	SDRAM: Row Address Select
oMDWEX	O	SDRAM: Write Enable
oMCKE	O	SDRAM: Clock Enable
oMCSX[8]	O	SDRAM: Chip select
oMNALE	O	NAND Flash: Address Latch Enable
oMNCLE	O	NAND Flash: Command Latch Enable
oMNWEX	O	NAND Flash: Write Enable
oMNREX	O	NAND Flash: Read Enable
oMCSX[7:0]	O	NAND: Chip select, shared with SRAM

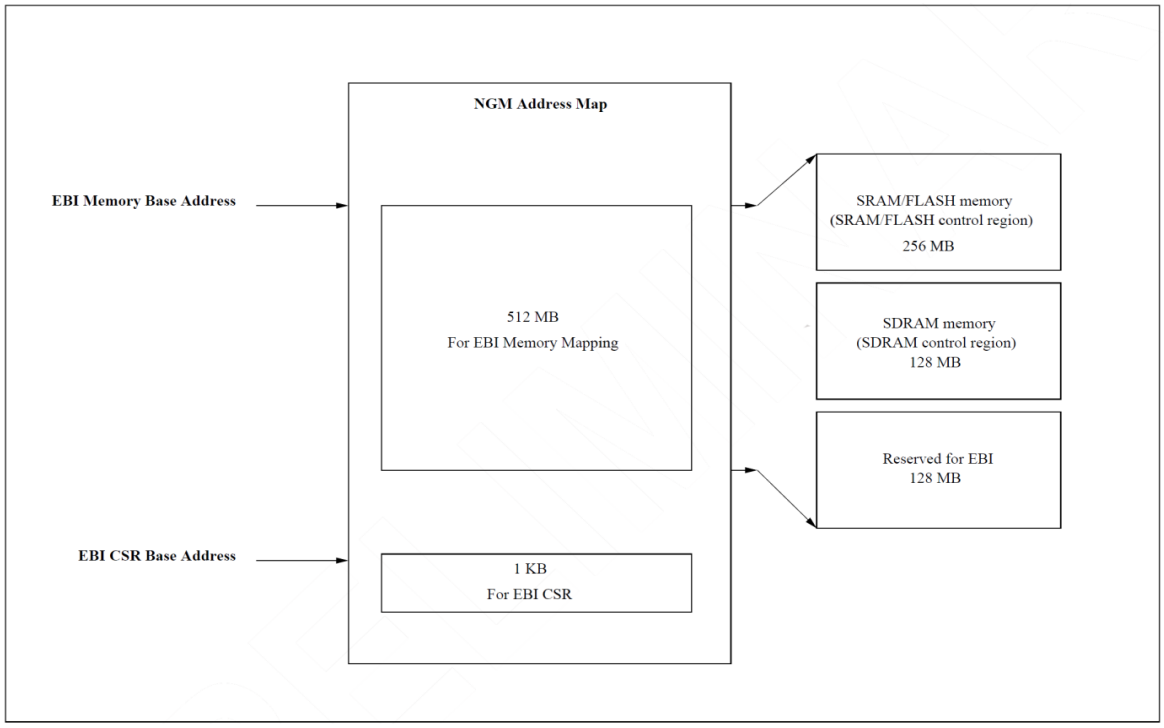
The MB9DF126 MCU supports up to 4 chip select signals and 2 byte mask signals; the data bus is 16 bit wide:

- oMCSX[3:0], oMCSX[8]
- oMAD[23:0]
- oMDATA[31:0]
- oMDQM[1:0]

2.2 Memory Regions

The External Bus Interface is configured by setting distinct memory regions. The control region is divided into the SRAM control region and the SDRAM control region and specific register sets. Up to 8 SRAM/FLASH regions and 1 SDRAM region (corresponding to above described MCSX chip select signals) can be individually mapped to the available address space as illustrated with the following diagram using the register sets described on the following page:

Figure 2-2: Memory Regions of the Extern al Bus Interface (EBI)



For more details on EBI Memory regions refer to the device datasheet

Table 2-2: External Bus Interface (EBI) Register Overview

Register name	Function
EBI_UNLOCK	General: Unlock Register, protects the EBI registers from being modified accidentally by software.
EBI_LSTSR	Status: Provides the status of External Bus Interface lock feature
EBI_ERRR	Status: Provides the Error status for SRAM/FLASH or SDRAM access
EBI_SFMR0..7	SRAM/FLASH: Provides the mode configuration: • Data width and endianess • Normal, NAND, NOR (page) and RDY (slow) access modes
EBI_SFACCR0..7	SRAM/FLASH: Provides the configuration to control the access timing of each memory bank
EBI_SFADDCR0..7	SRAM/FLASH: Provides the address and address mask to select specific memory banks
EBI_SDMODCR	SDRAM: Provides the mode configuration: • Bank address, RAS and CAS configuration • Refresh mode and clock configuration • Power save configuration in conjunction with EBI_SDPCR
EBI_SDRCR	SDRAM: Provides the refresh timing configuration
EBI_SDPCR	SDRAM: Power Control Register
EBI_SDTCR	SDRAM: Provides the configuration bits to control various timing modes related to SDRAM
EBI_SDCOMDR	SDRAM: Command Register

2.3 Memory Characteristic

The SRAM/FLASH controller has an address space of 256 MB. Each address can be set optionally with a different timing and endianness to be specified for each chip select.

NAND FLASH and NOR FLASH can be connected. NOR FLASH is accessed by usual SRAM accessing, dedicated pins are provided for NAND FLASH (for processing NAND flash access see the FCR4 hardware manual and Flash device datasheet). Low-speed devices can be connected by connecting the iMRDY pin of EBI to the peripheral ready signal.

The SDRAM controller has an address space of 128 MB. It accesses the memory device according to the correspondence among the internal bus, row, column and burst access specified by the user. This enables connections to SDRAMs of various configurations.

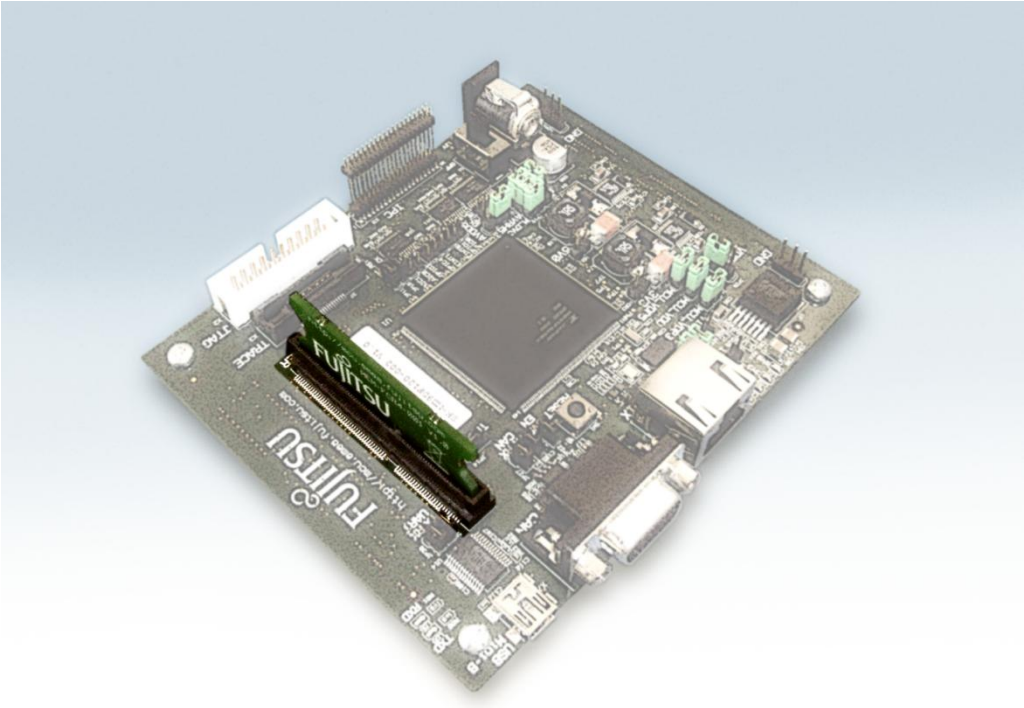
3 SK-MB9EF120-001 External Bus Interface Connector

This chapter describes the SK-MB9EF120-001 External Bus Interface Connector

3.1 External Bus Interface Connector Specification

The SK-MB9EF120-001 starter kit is equipped with a connector (X6, shown below) providing necessary External Bus Interface signals listed next page to connect SRAM, FLASH and SDRAM memory devices:

Figure 3 1: External Bus Interface Connector Location



For schematics and details on the SK-MB9EF120-001 External Bus Interface Connector X6 refer to the SK-MB9EF120-001 starter kit documentation

Table 3-1: External Bus Interface Connector X6 Pinout

Pin	Pad	Signal	EBI output pins	EBI input pins	EBI output function
2	B1	GND	-	-	-
4	B2	GND	-	-	-
6	B3	RST_X_3V3	-	-	-
8	B4	GND	-	-	-
10	B5	P1_34	EBI0_MAD19 (H)	-	Address output
12	B6	P1_35	EBI0_MAD20 (H)	-	Address output
14	B7	P1_36	EBI0_MAD21 (H)	-	Address output
16	B8	P1_37	EBI0_MAD22 (H)	-	Address output
18	B9	P1_38	EBI0_MAD23 (H)	-	Address output
20	B10	P1_30_B*	EBI0_MAD15 (H)	-	Address output
22	B11	P1_31_B*	EBI0_MAD16 (H)	-	Address output
24	B12	P1_32_B*	EBI0_MAD17 (H)	-	Address output
26	B13	P1_33_B*	EBI0_MAD18 (H)	-	Address output
28	B14	P3_0	EBI0_MAD00 (H)	-	Address output
30	B15	P3_1	EBI0_MAD01 (H)	-	Address output
32	B16	P3_2	EBI0_MAD02 (H)	-	Address output
34	B17	GND	-	-	-
36	B18	P3_3	EBI0_MAD03 (H)	-	Address output
38	B19	P3_4	EBI0_MAD04 (H)	-	Address output
40	B20	P3_5	EBI0_MAD05 (H)	-	Address output
42	B21	P3_6	EBI0_MAD06 (H)	-	Address output
44	B22	P3_7	EBI0_MAD07 (H)	-	Address output
46	B23	P3_8	EBI0_MAD08 (H)	-	Address output
48	B24	P3_9	EBI0_MAD09 (H)	-	Address output
50	B25	P3_10	EBI0_MAD10 (H)	-	Address output
52	B26	P3_11	EBI0_MAD11 (H)	-	Address output
54	B27	P3_12	EBI0_MAD12 (H)	-	Address output
56	B28	P1_24_B*	EBI0_MAD13 (G)	-	Address output
58	B29	P1_25_B*	EBI0_MAD14 (G)	-	Address output
60	B30	GND	-	-	-
62	B31	P3_33	EBI0_MAD18 (F) EBI0_MNALE (G) EBI0_MCASX (H)	-	Address output NAND Address Latch SDRAM CAS
64	B32	P3_34	EBI0_MAD19 (F) EBI0_MNCLE (G) EBI0_MRASX (H)	-	Address output NAND CLE SDRAM RAS
66	B33	P3_35	EBI0_MAD20 (F) EBI0_MNWEX (G)EBI0_MDWEX (H)	-	Address output NAND Write Enable SDRAM Write Enable
68	B34	P3_36	EBI0_MAD21 (F) EBI0_MNREX (G) EBI0_MCKE (H)	-	Address output NAND read enable SDRAM clock enable
70	B35	P3_37	EBI0_MAD18 (F) EBI0_MCASX (G) EBI0_MDQM0 (H)	-	Address output SDRAM CAS SRAM Byte Mask

Pin	Pad	Signal	EBI output pins	EBI input pins	EBI output function
72	B36	P3_38	EBI0_MAD19 (F) EBI0_MRASX (G) EBI0_MDQM1 (H)	-	Address output SDRAM RAS SRAM Byte Mask
74	B37	P3_39	EBI0_MAD20 (F) EBI0_MDWEX (G) EBI0_MWEX (H)	-	Address output SDRAM write Enable SRAM write enable
76	B38	P3_40	EBI0_MAD21 (F) EBI0_MCKE (G) EBI0_MOEX (H)	-	Address output SDRAM clock enable SRAM read enable
78	B39	P3_41	EBI0_MCLK (H)	-	SDRAM clock
80	B40	P3_42	EBI0_MAD22 (F) EBI0_MDQM0 (G)	EBI0_RDY	Address output SRAM Byte Mask
82	B41	P3_13	EBI0_MCSX8 (G) EBI0_MCSX0 (H)	-	SDRAM Select SRAM/NAND Select
84	B42	P3_14	EBI0_MCSX1 (H)	-	SRAM/NAND Select
86	B43	GND	-	-	-
88	B44	P3_15	EBI0_MDQM1 (G) EBI0_MCSX2 (H)	-	SRAM Byte Mask SRAM/NAND Select
90	B45	P3_16	EBI0_MCSX3 (H)	-	SRAM/NAND Select
92	B46	P3_17	EBI0_MDATA00 (H)	EBI0_MDATA00	Data Port
94	B47	P3_18	EBI0_MDATA01 (H)	EBI0_MDATA01	Data Port
96	B48	P3_19	EBI0_MDATA02 (H)	EBI0_MDATA02	Data Port
98	B49	P3_20	EBI0_MDATA03 (H)	EBI0_MDATA03	Data Port
100	B50	P3_21	EBI0_MDATA04 (H)	EBI0_MDATA04	Data Port
102	B51	P3_22	EBI0_MDATA05 (H)	EBI0_MDATA05	Data Port
104	B52	P3_23	EBI0_MDATA06 (H)	EBI0_MDATA06	Data Port
106	B53	P3_24	EBI0_MDATA07 (H)	EBI0_MDATA07	Data Port
108	B54	P3_25	EBI0_MDATA08 (H)	EBI0_MDATA08	Data Port
110	B55	P3_26	EBI0_MDATA09 (H)	EBI0_MDATA09	Data Port
112	B56	GND	-	-	-
114	B57	P3_27	EBI0_MDATA10 (H)	EBI0_MDATA10	Data Port
116	B58	P3_28	EBI0_MDATA11 (H)	EBI0_MDATA11	Data Port
118	B59	P3_29	EBI0_MDATA12 (H)	EBI0_MDATA12	Data Port
120	B60	P3_30	EBI0_MDATA13 (H)	EBI0_MDATA13	Data Port
122	B61	P3_31	EBI0_MDATA14 (H)	EBI0_MDATA14	Data Port
124	B62	P3_32	EBI0_MDATA15 (H)	EBI0_MDATA15	Data Port
126	B63	P1_26_B*	-	-	GPIO
128	B64	P1_27_B*	-	-	GPIO
130	B65	P1_28_B*	-	-	GPIO
132	B66	P1_29_B*	-	-	GPIO
134	B67	3V3	-	-	-
136	B68	3V3	-	-	-
138	B69	3V3	-	-	-
140	B70	3V3	-	-	-

Letter ‘F’ ... ‘H’ in the table above corresponds to the Port Output Function Setting for the listed GPIO Pin, refer to the device specific datasheet for more details.

The corresponding Port Input function is selected with this setting automatically for the data bus; for the EBI0_RDY input general GPIO input parameters have to be set as indicated by the programming example extracts further down.

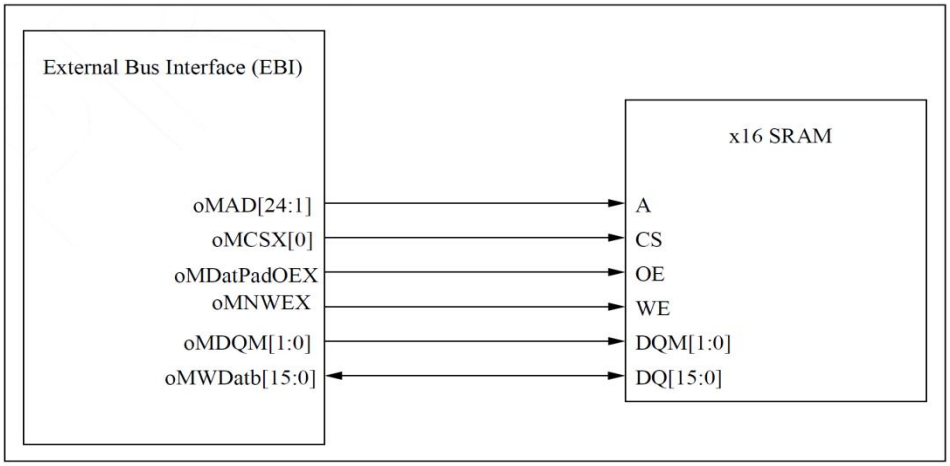
Port Pins marked with ‘*’ are shared with the trace port and will not be available when using the trace port at full data width.

3.2 SRAM and NOR FLASH Example

This application note refers to the MEM-FCR4-004 memory module as an example for adding SDRAM and NOR FLASH memory to the SK-MB9EF120-001 starter kit. In General, following signals have to be provided for SRAM/FLASH memory as indicated in following diagram:

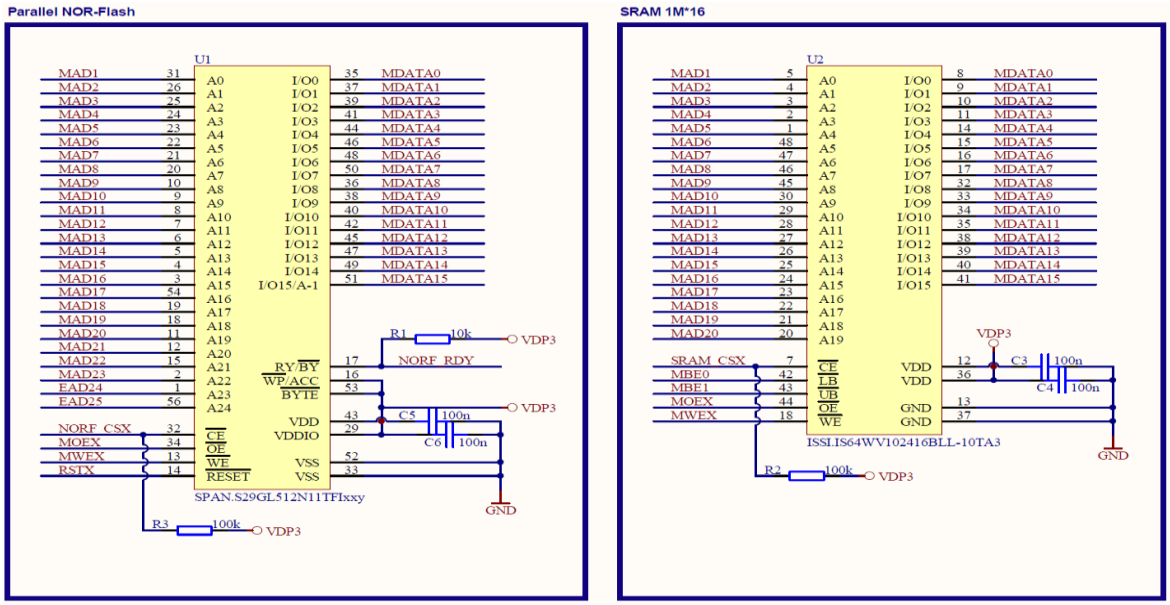
- Address and Data Bus
- Chip Select Signals
- Signals for Read and Write Enable, also a Byte mask signal can be supplied

Figure 3-1: General Example for 16 bit SRAM Attachment



Next Figure shows the EM-FCR4-004 memory module schematic as an example for adding SDRAM and NOR FLASH memory to the SK-MB9EF120-001 starter kit External Bus Interface Connector X6. Some considerations have to be employed regarding 16bit wide memory devices e.g. using Byte mask signals to control upper and lower byte access to the SRAM and providing right signal levels for access with to the FLASH memory. For the FLASH memory also RESET and RDY (ready) signal being used:

Figure 3-2: MEM-FCR4-004 Memory Module Schematic



Both devices supporting 16 bit data with, addressing has to use EBI0_MDATA01 (MAD01) as least significant bit connecting to devices A0 address lines. For FLASH memory additional GPIO pins (EAD24/25) being used to extend the available address range (bank switching).

Write enable EBI0_MWEX (MWEX) and read enable EBI0_MOEX (MOEX) controls are provided to both memory devices, devices (and corresponding memory to be set with EBI_SFADDCR0 and EBI_SFADDCR0 registers for Bank 0 and 1) are selected using EBI0_MCSX0 (NORF_CSX) and EBI0_MCSX1 (SRAM_CSX) signals. Using L3 Low Level Library all required register settings (mode, address mapping and timing parameters) will be done by using L3 function 'Ebi_InitMemoryBank(*Banknumber*, *&stcMemorybankConfig*);' as described next chapter.

For SRAM, the Byte mask signals EBI0_MDQM0/1 (MBE0/1) are used. When the access is made using a bit width narrower than the targeted one (e.g. byte access to a 16-bit width target), the byte access will be executed by being controlled by MDQM (byte mask) signal if it is in write operation (The SRAM/FLASH controller outputs only necessary data). In read, if the target device has output mask signal, unnecessary outputs from the device can be avoided by letting the device output only the data of necessary bytes using MDQM control. As a result, power dissipation is reduced.

For the FLASH memory device being limited in access speed, the ready signal RY provided by the device is used to drive EBI0_RDY (NORF_RDY). Also reset line RST_X_3V3 (RSTX) being needed for FLASH.

For details on the required signals and timing parameters of the employed memory devices see corresponding datasheets.

4 Software Requirements

This chapter describes the programming of the External Bus Interface

4.1 General Considerations

This Chapter refers to the fcr4_ebi_sram_mb9d126-vxx software sample discussing main programming steps to access the SRAM device on the FCR4-004 Memory Module to be place into the External Bus Interface Connector X6 of the SK-MB9EF120-001 starter kit. The software sample is based on the L3 Low Level Library.

Derived from Table 3-1: External Bus Interface Connector X6 Pinout and Figure 3-3: MEM-FCR4-004 Memory Module Schematic (see last chapter) following Pins and memory regions need to be defined for the software sample:

Figure 4-1: External Bus Interface Software Sample Definitions

```
// EBI Settings for SRAM access
#define EBI_ADR_PORT_A      3
#define EBI_ADR_PORT_B      1

#define EBI_ADR0_PIN        0
#define EBI_ADR13_PIN       24
#define EBI_ADR15_PIN       30
#define EBI_ADR24_PIN       33

#define EBI_DATA_PORT       3
#define EBI_DATA0_PIN       17

#define EBI_CTRL_PORT       3
#define EBI_MDQM0_PIN       37
#define EBI_MDQM1_PIN       38
#define EBI_MWEX_PIN        39
#define EBI_MOEX_PIN        40
#define EBI_MCLK_PIN        41
#define EBI_RDY_PIN         42

#define EBI_CS_PORT         3
#define EBI_CS0_PIN         13

#define NOR_FLASH_START     0x10000000
#define NOR_FLASH_END       0x108FFFFF

#define SRAM_START          0x11000000
#define SRAM_END            0x111FFFFF
```

Using those defines, access to the SRAM device on the FCR4-004 Memory Module can be programmed next passages.

4.2 Low Level Library GPIO Configuration for EBI

First, all GPIO Pins have to be set up (expand general pin port definitions in main.c), starting with the address bus pins EBI0_MAD01 - EBI0_MAD23 (see Table 3-1: External Bus Interface Connector X6 Pinout and definitions above):

Figure 4-2: Low Level Library GPIO Configuration for EBI – Address Bus

```
// EBI Address lines
{.u8Port            = EBI_ADR_PORT_A,
 .u8Pin             = EBI_ADR0_PIN,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = FALSE
                      }
},  // EBI0_MAD00

{.u8Port            = EBI_ADR_PORT_A,
 .u8Pin             = EBI_ADR0_PIN + 1,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = FALSE
                      }
},  // EBI0_MAD01

 . . .

{.u8Port            = EBI_ADR_PORT_A,
 .u8Pin             = EBI_ADR0_PIN + 11,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = FALSE
                      }
},  // EBI0_MAD12
```

Do the same configuration for the remaining address lines:

.u8Port = EBI_ADR_PORT_B, .u8Pin = EBI_ADR13_PIN (EBI0_MAD13)
.u8Port = EBI_ADR_PORT_B, .u8Pin = EBI_ADR13_PIN + 1 (EBI0_MAD14)

.u8Port = EBI_ADR_PORT_B, .u8Pin = EBI_ADR15_PIN (EBI0_MAD15)

. . .

.u8Port = EBI_ADR_PORT_B, .u8Pin = EBI_ADR15_PIN + 8 (EBI0_MAD15)

.u8Port = EBI_ADR_PORT_A, .u8Pin = EBI_ADR24_PIN (EAD24)
.u8Port = EBI_ADR_PORT_A, .u8Pin = EBI_ADR24_PIN + 1 (EAD25)

For EBI0_MAD13 and EBI0_MAD14 set .enOutputFunction = PortOutputResourceG

Next set up the GPIO port pins for the data bus (need to have input enabled)...:

Figure 4-3: Low Level Library GPIO Configuration for EBI – Data Bus

```
// EBI Data lines
{.u8Port            = EBI_DATA_PORT,
 .u8Pin             = EBI_DATA0_PIN,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = TRUE
                      }
},  // EBI0_MDATA00

{.u8Port            = EBI_DATA_PORT,
 .u8Pin             = EBI_DATA0_PIN + 1,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = TRUE
                      }
},  // EBI0_MDATA01

 . . .

{.u8Port            = EBI_DATA_POR
 .u8Pin             = EBI_DATA0_PIN + 15,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = TRUE
                      }
},  // EBI0_ EBI0_MDATA15
```

... and chip select signals (output signals, same as address bus):

Figure 4-4: Low Level Library GPIO Configuration for EBI – Chip Select Lines

```
// EBI Chip Select signals
{.u8Port            = EBI_CS_PORT,
 .u8Pin             = EBI_CS0_PIN,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = FALSE
                      }
},  // EBI0_MCSX0

 . . .

{.u8Port            = EBI_CS_PORT,
 .u8Pin             = EBI_CS0_PIN + 3,
 .enGpioDirection   = GpioOutput,
 .bGpioOutputHigh   = FALSE,
 .stcPortPinConfig  = {.enOutputFunction = PortOutputResourceH,
                       .bInputEnable = FALSE
                      }
},  // EBI0_MCSX3
```

Last not least do the same configuration for the remaining EBI control lines:

.u8Port = EBI_CTRL_PORT, . EBI_MDQM0_PIN (EBI_MDQM0)
.u8Port = EBI_CTRL_PORT, . EBI_MDQM1_PIN (EBI_MDQM1)
.u8Port = EBI_CTRL_PORT, . EBI_MWEX _PIN (EBI_MWEX)
.u8Port = EBI_CTRL_PORT, . EBI_MOEX _PIN (EBI_MOEX)
.u8Port = EBI_CTRL_PORT, . EBI_MCLK_PIN (EBI_MCLK)
.u8Port = EBI_CTRL_PORT, . EBI_RDY_PIN (EBI_RDY)

For EBI0_RDY set .bInputEnable = TRUE

4.3 Low Level Library EBI Configuration

Now the External Bus Interface can get initialized using ‘Ebi_InitMemoryBank(). The second code block reflects the external bus interface timing and has to be adjusted to clock settings within ‘mcu.h’ and SRAM device characteristics (see timing diagram below)

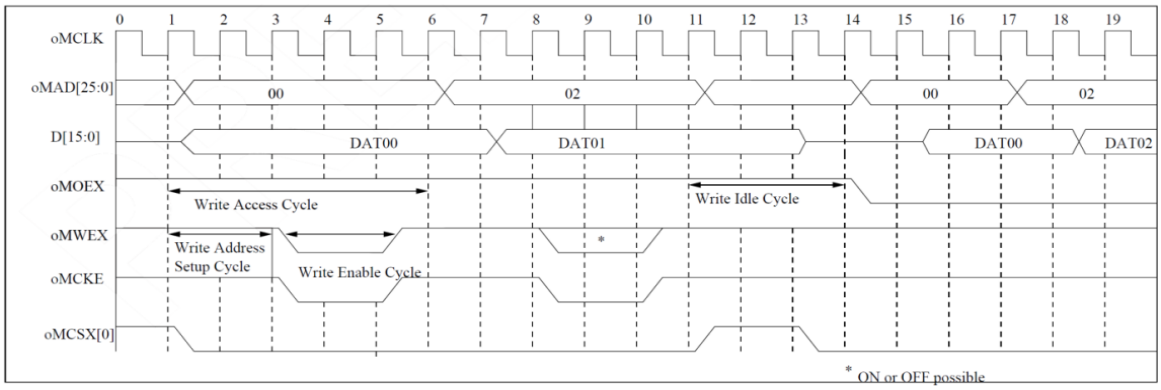
Figure 4-5: Low Level Library External Bus Interface Initialization

```
// Variable for SRAM configuration
stc_ebi_memorybankconfig_t  stcEbiRamMemorybankConfig =
                                    { .bBigEndian          = FALSE,
                                      .bReadyMode          = FALSE,
                                      .bPageAccessMode     = FALSE,
                                      .bNandFlashMode      = FALSE,
                                      .bEnableWrite        = TRUE,
                                      .bEnableReadByteMask = TRUE,
                                      .enDataWidth         = EbiDataWidth16bit,

                                      .enWriteIdleCycles   = EbiCycles4,
                                      .enWriteEnableCycles = EbiCycles4,
                                      .enWriteAddressSetupCycles = EbiCycles4,
                                      .enWriteAccessCycles       = EbiCycles10,
                                      .enReadIdleCycles          = EbiCycles4,
                                      .enFirstReadAddressCycles  = EbiCycles3,
                                      .enReadAddressSetupCycles  = EbiCycles3,
                                      .enReadAccessCycles        = EbiCycles4,

                                      .u8BankAddress = 0x10,
                                      .enBankSize = EbiBankSize2MB
                                     };
// Initialise EBI SRAM memory bank 1
Ebi_InitMemoryBank(1, &stcEbiRamMemorybankConfig);
```

Figure 4-6: External Bus Interface Timing Diagram



4.4 Provided Low Level Library Interface

Following Figure gives an overview of the provided software interface:

Figure 4-7: L3 External Bus Interface (EBI) Software Interface

```
typedef enum en_ebi_data_width
{
    EbiDataWidth8bit  = 0,
    EbiDataWidth16bit = 1,
    EbiDataWidth32bit = 2,
} en_ebi_data_width_t;

typedef enum en_ebi_cycles
{
    EbiCycles0  = 0,
    EbiCycles1  = 1,
      ...
    EbiCycles15 = 15,
    EbiCycles16 = 16,
} en_ebi_cycles_t;

typedef enum en_ebi_bank_size
{
    EbiBankSize1MB   = 0x00,
    EbiBankSize2MB   = 0x01,
    EbiBankSize4MB   = 0x03,
    EbiBankSize8MB   = 0x07,
    EbiBankSize16MB  = 0x0f,
    EbiBankSize32MB  = 0x1f,
    EbiBankSize64MB  = 0x3f,
    EbiBankSize128MB = 0x7f,
} en_ebi_bank_size_t;

typedef struct stc_ebi_memorybankconfig
{
    boolean_t bBigEndian;
    boolean_t bReadyMode;
    boolean_t bPageAccessMode;
    boolean_t bNandFlashMode;
    boolean_t bEnableWrite;
    boolean_t bEnableReadByteMask;
    en_ebi_data_width_t enDataWidth;
    en_ebi_cycles_t enWriteIdleCycles;
    en_ebi_cycles_t enWriteEnableCycles;
    en_ebi_cycles_t enWriteAddressSetupCycles;
    en_ebi_cycles_t enWriteAccessCycles;
    en_ebi_cycles_t enReadIdleCycles;
    en_ebi_cycles_t enFirstReadAddressCycles;
    en_ebi_cycles_t enReadAddressSetupCycles;
    en_ebi_cycles_t enReadAccessCycles;
    uint8_t u8BankAddress;
    en_ebi_bank_size_t enBankSize;
} stc_ebi_memorybankconfig_t;

typedef struct stc_ebi_config
{
    stc_ebi_memorybankconfig_t astcMemoryBankConfig[8];
} stc_ebi_config_t;

extern en_result_t Ebi_InitMemoryBank(uint8_t u8MemoryBank, const
                                      stc_ebi_memorybankconfig_t* pstcConfig);
```

Document History

Document Title: AN205490 - FCR4, ATLAS series MB9DF126, External Bus Interface (EBI) - Accessing SRAM

Document Number: 002-05490

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	MKEA	03/30/2012	V1.0; MKoe Initial version
*A	5071020	MKEA	01/07/2016	Converted Spansion Application Note “MCU-AN-300136-E-V10” to Cypress format
*B	5862825	AESATMP9	08/24/2017	Updated logo and copyright.
*C	6054547	NOFL	02/12/2018	Updated links Updated template

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709