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.

AN205256

F²MC-16FX, MB96340, MB88121, Interfacing MB96340 to MB88121

Cypress offers a standalone Flex Ray Communication Controller, MB88121 series, which supports parallel and serial connectivity to Host MCU. The MB88121 series supports a parallel Bus interface modes, 16-bit multiplexed mode, especially for 16FX MCU.

Contents

1 Introduction....1
2 Connection example MB96F348RS – MB88121B....2
3 Configuration....6
3.1 MB96340 Series....6
3.2 MCU Operation Mode....8
3.3 MB88121 CC operation mode....10
3.4 Configuration of the External Bus Interface....10
3.5 Simplified Timing Diagram of MB88121B....19
3.6 Initialization Sequence of MB88121....21
3.7 Interrupts....22
3.8 Debugging Support....27
3.9 Stop Watch pin....27
4 Reference....28
Document History....29

1. Introduction

Cypress offers a standalone Flex Ray Communication Controller, MB88121 series, which supports parallel and serial connectivity to Host MCU. The MB88121 series supports a parallel Bus interface modes, 16-bit multiplexed mode, especially for 16FX MCU. The 16FX MCU MB96340 series is supporting a 16-bit multiplexed bus interface. The following discusses Hardware and Software requirement based on MB96F348RS series.

Note:
Not all MB96300 devices support a multiplexed external bus interface.

2. Connection example MB96F348RS – MB88121B

A connection example is summarized in the following table. All pins of the communication controller (CC) are divided into 7 categories:

- External bus interface: CC pins in connection with MCU pins
- CC mode pins
- CC debug pins
- CC stop watch trigger pin
- CC pins in connection with physical layer transceiver
- CC power supply pins
- CC external clock pin

Note:

1. The connection of the interrupt pins is application-specific. The table here shows only an example.
2. The CC reset pin can be controlled by MCU reset pin or by an independent circuitry.
3. In this application note external bus area 2 is configured for MB88121B. Other areas can also be used.
4. For the power supply it should be noticed that MB88121B is a single voltage supply chip. All power supply pins should have the same supply voltage between 3.3V and 5V.

External bus interface: CC pins in connection with MCU pins					
CC pin Nr	**Name**	**Function**		**MCU pin Nr**	**Name**
53	AD10	Address/data bus MCU⇔CC		87	P01_2/AD10
54	AD9			86	P01_1/AD09
55	AD8			85	P01_0/AD08
56	AD7			84	P00_7/AD07
57	AD6			83	P00_6/AD06
58	AD5			82	P00_5/AD05
59	AD4			81	P00_4/AD04
60	AD3			80	P00_3/AD03
61	AD2			79	P00_2/AD02
62	AD1			78	P00_1/AD01
63	AD0			77	P00_0/AD00
CC pin Nr	Name	Function		MCU pin Nr	Name
46	D15	Data bus MCU⇔CC		96	P01_7/AD15
47	D14			95	P01_6/AD14
50	D13			94	P01_5/AD13
51	D12			89	P01_4/AD12
52	D11			88	P01_3/AD11
CC pin Nr	Name	Function		MCU pin Nr	Name
10	INT0 [1]	Interrupt output CC=>MCU		46	P07_1/INT1
34	INT1			47	P07_2/INT2
35	INT2			48	P07_3/INT3
36	INT3			49	P07_4/INT4
37	INT4			50	P07_5/INT5
CC pin Nr	Name	Function		MCU pin Nr	Name
6	$\overline{RST}$ [2]	Reset	MCU=>CC	\	\
15	BCLK	Bus clock		12	P03_7/ECLK
19	$\overline{CS}$	Chip select		72	P09_5/CS2 [3]
20	$\overline{RD}$	Read enable		6	P03_1/RDX
21	$\overline{WR}$	Write enable		7	P03_2/WRLX/WRX
22	ALE	Address latch enable		5	P03_0/ALE
27	RDY	Ready signal	CC=>MCU	11	P03_6/RDY
31	DMA_REQ	DMA request		55	P07_6/INT6
CC mode pins					
CC pin Nr	Name	Function	I/O type	Logic value	
64	MD2	Mode selection	IN	1	16bit parallel bus interface
4	MD1			0	
5	MD0			0	
28	MDE2	Extended mode selection		0	Multiplexed mode for 16FX
29	MDE1			0	
30	MDE0			1	

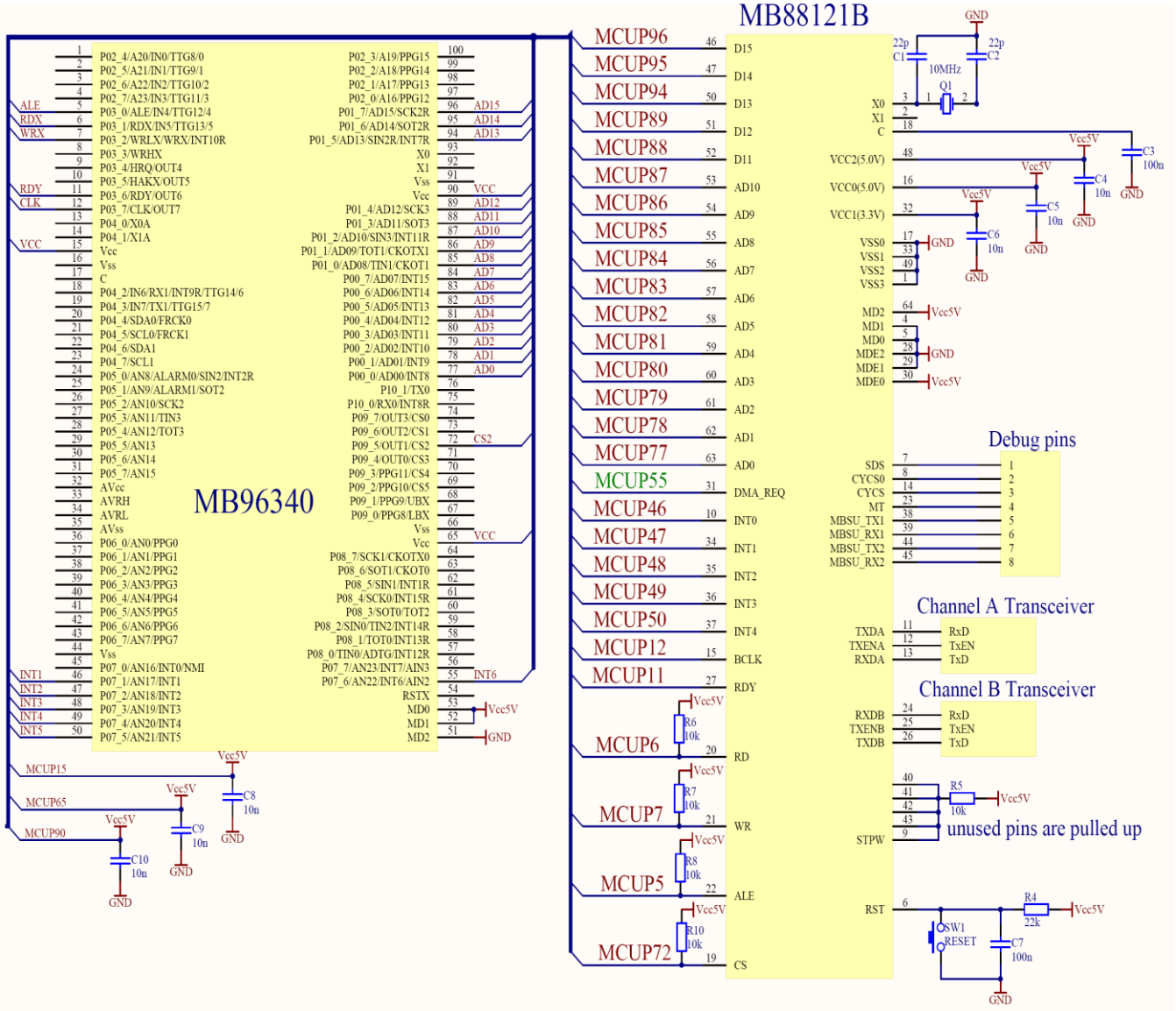
CC debug pins				
CC pin Nr	**Name**	**Function**	**I/O type**	**Configuration register**
7	SDS	Start of dynamic segment	OUT	Debug support register DBGS
8	CYCS0	Cycle 0 start	OUT	
14	CYCS	Cycle start	OUT	
23	MT	Macro tick start	OUT	
38	MBSU_TX1	Message buffer status update port	OUT	
39	MBSU_RX1			
44	MBSU_TX2			
45	MBSU_RX2			

CC stop watch trigger pin			
CC pin Nr	**Name**	**I/O type**	**Function**
9	STPW	IN	After stop watch is triggered (rising or falling edge), register STPW1 captures actual cycle counter and macro tick value, register STPW2 captures slot counter values for channel A and B.

CC pins in connection with physical layer transceiver				
CC pin Nr	**Name**	**Function**	**I/O type**	**Connection**
11	TXDA	Data transmission	OUT	Channel A transceiver
12	$\overline{TXENA}$	Channel transmission enable	OUT	
13	RXDA	Data reception	IN	
24	RXDB	Data reception	IN	Channel B transceiver
25	$\overline{TXENB}$	Channel transmission enable	OUT	
26	TXDB	Data transmission	OUT	

CC power pins		
CC pin Nr	**Name**	**Function**
1,17,33,49	VSS	0V Ground
16, 32, 48	VCC[4]	Single power supply input between 3.3v and 5v
18	C	Power supply stabilization capacitor (≥100nF)

CC external clock pin			
CC pin Nr	**Name**	**Function**	
2	X1	Oscillation output	4MHz, 5MHz,8MHz,10MHz crystal or ceramic oscillator
3	X0	Oscillation input	


MB96340
MB88121B
Debug pins
Channel A Transceiver
Channel B Transceiver
unused pins are pulled up
MCUP96
MCUP95
MCUP94
MCUP89
MCUP88
MCUP87
MCUP86
MCUP85
MCUP84
MCUP83
MCUP82
MCUP81
MCUP80
MCUP79
MCUP78
MCUP77
MCUP55
MCUP46
MCUP47
MCUP48
MCUP49
MCUP50
MCUP12
MCUP11
MCUP6
MCUP7
MCUP5
MCUP72
MCUP15
MCUP65
MCUP90
Vcc5V
GND
10MHz
Q1
RESET
SW1

3. Configuration

This chapter introduces the concrete configuration of MB96F348RS series

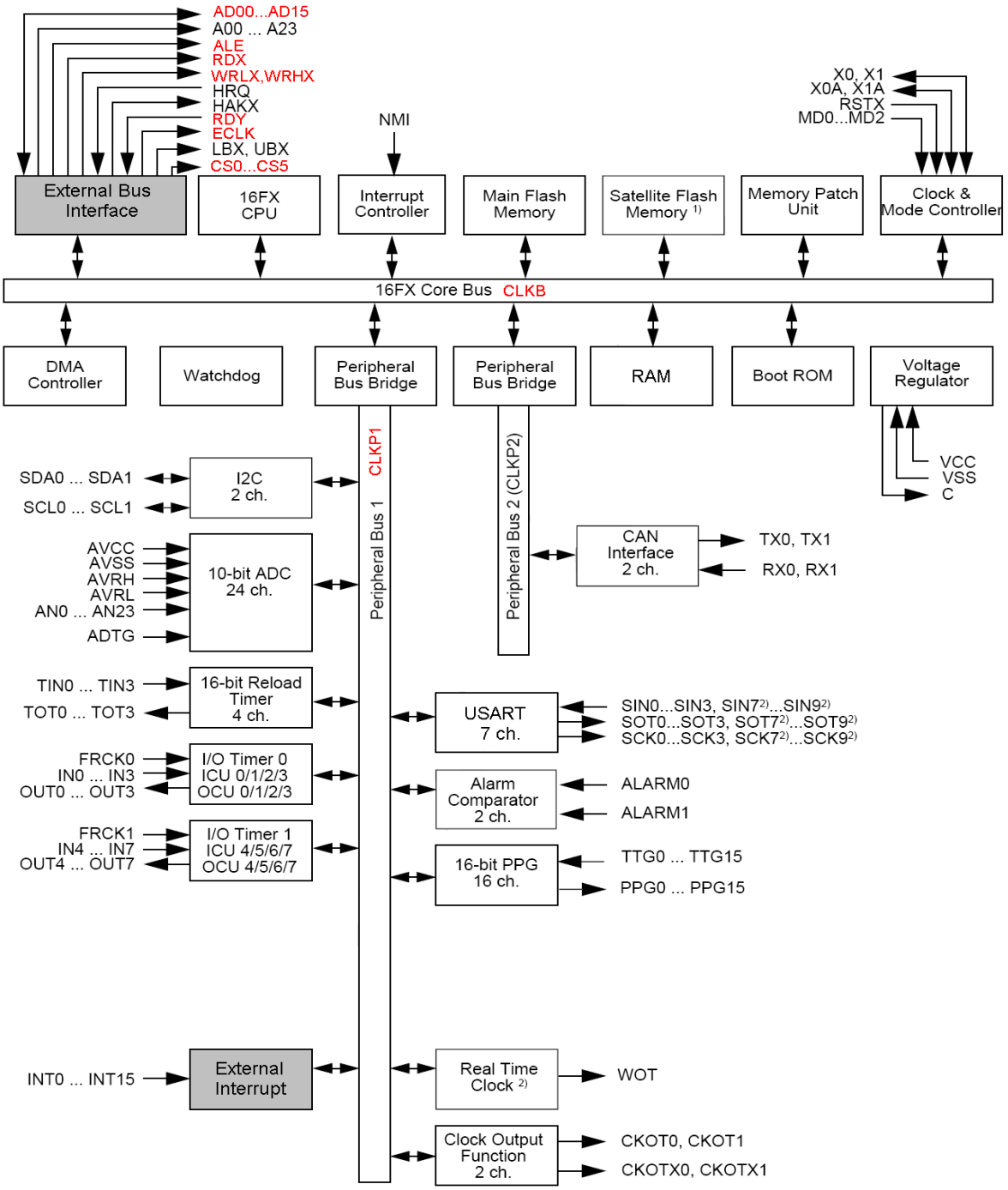
3.1 MB96340 Series

MB96340 series is based on Cypress's 16FX architecture. Some features of this MCU family are listed below.

Table 1. MB96340 feature

Flash/ROM	RAM	MB96340 series
128KB	6KB	MB96344R,MB96344Y
288KB	16KB	MB96F346R, MB96346R, MB96F346Y, MB96346Y, MB96F346A
416KB	16KB	MB96F347R, MB96347R, MB96F347Y, MB96347Y, MB96F347A
544KB	24KB	MB96F348R, MB96F348Y, MB96F348A
Main Flash 544KB Satellite Flash 32KB	24KB	MB96F348C, MB96F348H, MB96F348T
External interrupts		16 channels
External bus interface		Multiplexed mode
Chip select signal		6

Figure 1. Block diagram of the MB96340 series



3.2 MCU Operation Mode

The MCU operates in the internal ROM external bus mode. That means the internal Flash is accessible and in addition the external bus interface might be enabled during initialization phase of the MCU

The configuration of mode pins and file start.asm are listed below.

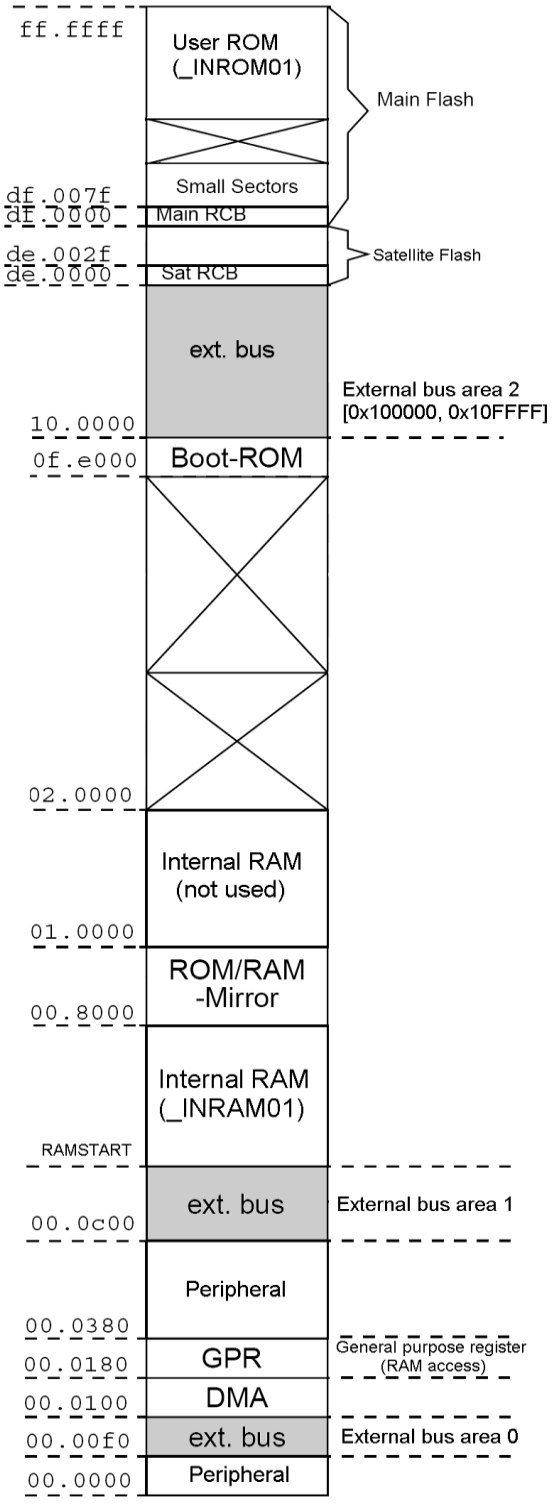
Table 2. Configuration of internal ROM external bus mode

Mode pin setting	Operation mode	Description
MD2=0	Internal vector mode	The boot vector (user program start address) is read from internal memory address 0xFFFFDC.
MD1=1		
MD0=1		

```
;=====================================================================
; 4.1 Controller Series, Device
;=====================================================================
#set       SERIES    MB96340
#set       DEVICE    MB96348R
;=====================================================================
; 4.14 Boot Vector
;=====================================================================
#set       BOOT_VECTOR   BOOT_VECTOR_TABLE
#if BOOT_VECTOR == BOOT_VECTOR_TABLE
           .SECTION        RESVECT, CONST, LOCATE=H'FFFFDC
           .DATA.E _start
           .SECTION        BOOT_SELECT, CONST, LOCATE=H'DF0030
           .DATA.L 0xFFFFFFFF
#endif
;=====================================================================
; 6.1 Import external symbols
;=====================================================================
           .IMPORT   _main                    ; user code entrance
           …
           .EXPORT   _start
;=====================================================================
; 6.2 Program start (the boot vector should point here)
;=====================================================================
_start:
```

The memory map of this operation mode is illustrated in the next page.

Figure 2. Memory map of internal RAM external bus mode



3.3 MB88121 CC operation mode

The mode pins MD [2:0] and MDE [2:0] select between different bus types. The multiplexed mode is fixed encoded.

Table 3. Mode selection input of the communication controller

MD2	MD1	MD0	Mode	MDE2	MDE1	MDE0	Mode extension	
1	0	0	16bit parallel communication	0	0	0	Multiplexed bus	FR460
				0	0	1		16FX
				1	0	0	Non-multiplexed bus	FR460
				1	1	0		FR360

3.4 Configuration of the External Bus Interface

MB96340 has 6 chip select areas. The configuration registers are listed in the table.

Table 4. External bus interface registers

Address	+0	+1
	External bus area [5:0] configuration register	
0x0006E0	EACL0	EACH0
0x0006E2	EACL1	EACH1
0x0006E4	EACL2	EACH2
0x0006E6	EACL3	EACH3
0x0006E8	EACL4	EACH4
0x0006EA	EACL5	EACH5
	External bus area [5:2] select register	
0x0006EC	EAS2	EAS3
0x0006EE	EAS4	EAS5
0x0006F0	EBM (external bus mode register)	EBCF (external bus clock and function register)
	External bus address output enable register	
0x0006F2	EBAE0	EBAE1
0x0006F4	EBAE2	EBCS (external bus control signal register)
Gray Settings used for MB88121 connectivity		

The MB88121 CC requires a minimum address range of 2K (0 – 0x7FF).

That means that CS0 cannot be used due to insufficient address range. In this example CS2 is used

3.4.1 External Bus Mode Register (EBM)

Register EBM defines the bus mode of the external bus interface. Bit function is listed below.

Table 5. Register EBM

Bit	Name		Function
7	NMS	0	Multiplexed bus mode for all external areas (parameter ADDRESSMODE in start.asm)
		1	Non-multiplexed bus mode for all external areas
6	ERE	0	Internal ROM mode (parameter BUSMODE in start.asm)
		1	External ROM mode
5	EAE5	0	External area 5 is disabled (parameter CHIP_SELECT5 in start.asm)
		1	External area 5 is enabled
4	EAE4	0	External area 4 is disabled (parameter CHIP_SELECT4 in start.asm)
		1	External area 4 is enabled
3	EAE3	0	External area 3 is disabled (parameter CHIP_SELECT3 in start.asm)
		1	External area 3 is enabled
2	EAE2	0	External area 2 is disabled
		1	External area 2 is enabled (parameter CHIP_SELECT2 in start.asm)
1	EAE1	0	External area 1 is disabled (parameter CHIP_SELECT1 in start.asm)
		1	External area 1 is enabled
0	EAE0	0	External area 0 is disabled (parameter CHIP_SELECT0 in start.asm)
		1	External area 0 is enabled
Gray Settings used for MB88121 connectivity			

3.4.2 External Bus Clock and Function Register

Register EBCF controls the external bus clock, the external READY and HOLD function. Bit function is listed below.

Table 6. Register EBCF

Bit	Name		Function
15	HDE	0	HOLD function disabled (parameter HOLD_REQ in start.asm)
		1	HOLD function enabled
14	RYE	0	READY function disabled
		1	READY function enabled (parameter EXT_READY in start.asm)
13	CKE	0	External bus clock output disabled
		1	External bus clock output enabled (parameter EXT_CLOCK_ENABLE in start.asm)
12	CKI	0	External bus clock is not inverted the inactive level of ECLK is '0' and the rising edge is the active edge
		1	External bus clock is inverted , the inactive level of ECLK is '1' and the falling edge is the active edge (parameter EXT_CLOCK_INVERT in start.asm)
11	CSM	0	External bus clock is always output (parameter EXT_CLOCK_SUSPEND in start.asm)
		1	External bus clock is only output during transfer
10:8	DIV[2:0]	External bus clock (ECLK) divider (parameter EXT_CLOCK_DIVISION in start.asm)	
Gray Settings used for MB88121 connectivity			

Table 7. Bit DIV [2:0] of register EBCF

DIV2	DIV1	DIV0	External bus clock (ECLK) divider setting
0	0	0	CLKB (CPU clock)
0	0	1	CLKB/2
0	1	0	CLKB/4
0	1	1	CLKB/8
1	0	0	CLKB/16
1	0	1	CLKB/32
1	1	0	CLKB/64
1	1	1	CLKB/128
Gray Settings used for MB88121 connectivity			

Note:

1. All operations of the external bus are based on ECLK even if the output of this clock is disabled.
2. CSM=0, ECLK is always output, also in HOLD state.

The 16FX MCU offers two methods to extend the read/write access to an external device: the auto wait function (bits EACL_R [2:0]) and the READY function through RDY pin (bit EBCF_RYE). They enable the access to low-speed memory and peripheral circuits.

The auto wait function can insert 1 to 32 wait cycles to extend the data cycle. By READY function the data cycle is extended as long as pin RDY =0.

3.4.3 External Bus Address output Enable Registers (EBAE [2:0])

Registers EBAE [2:0] control the output function of each address line.

0: disable the address line
1: enable the address line

Table 8. Register EBAE [2:0]

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Corresponding parameter in file start.asm
0x0006F2 (EBAE0)	A7	A6	A5	A4	A3	A2	A1	A0	ADDR_PINS_7_0
	1	1	1	1	1	1	1	1	
0x0006F3 (EBAE1)	A15	A14	A13	A12	A11	A10	A9	A8	ADDR_PINS_15_8
	1	1	1	1	1	1	1	1	
0x0006F4 (EBAE2)	A23	A22	A21	A20	A19	A18	A17	A16	ADDR_PINS_23_16
	0	0	0	0	0	0	0	0	
Gray Settings used for MB88121 connectivity									

In the multiplexed bus mode register EBAE0 and EBAE1 control the address output of the pins AD15—AD00. As the 16-bit multiplexed mode is used AD00... AD15 needs to be enabled.

3.4.4 External Bus Control Signal Register (EBCS)

Register EBCS enables/disables the control signals of the external bus.

Table 9. Register EBCS

Bit	Name	Function		Corresponding parameter in file start.asm
15	\	\		\
14	ASL	0	Address strobe is low active ($\overline{AS}$ function)	ADDRESS_STROBE_LVL
		1	Address strobe is high active (ALE function)	
13	ASE	0	Address strobe output (ALE/ $\overline{AS}$) disabled	ADDRESS_STROBE
		1	Address strobe output (ALE/ $\overline{AS}$) enabled	
12	RDE	0	Read strobe output ($\overline{RD}$) disabled	READ_STROBE
		1	Read strobe output ($\overline{RD}$) enabled	
11	WRHE	0	Write strobe output ($\overline{WRH}$) disabled	HIGH_WRITE_STROBE
		1	Write strobe output ($\overline{WRH}$) enabled	
10	WRLE	0	Write strobe output ($\overline{WRL}$ / $\overline{WR}$) disabled	LOW_WRITE_STROBE
		1	Write strobe output ($\overline{WRL}$ / $\overline{WR}$) enabled	
9	UBE	0	Byte select output ($\overline{UB}$) disabled	HIGH_BYTE_SIGNAL
		1	Byte select output ($\overline{UB}$) enabled	
8	LBE	0	Byte select output ($\overline{LB}$) disabled	LOW_BYTE_SIGNAL
		1	Byte select output ($\overline{LB}$) enabled	
Gray Settings used for MB88121 connectivity				

3.4.5 External Area Configuration Registers (EACH/EACL [5:0])

Function of register EACH and EACL is listed below. In this application note external area 2 is used. The corresponding parameter is CS2_CONFIG.

Table 10. Register EACH/EACL

Register EACH2			
Bit	**Name**	**Function**	
15:14	\	\	
13	ATL	0	Code and data read possible
		1	Only data read possible
12	CSL	0	Chip select signal is low active ($\overline{CS}$)
		1	Chip select signal is high active (CS)
11	CSE	0	Chip select output disabled
		1	Chip select output enabled
10:8	EASZ[2:0]	External area size	
Register EACL2			
7	BW	0	16bit data bus width
		1	8bit data bus width
6	ES	0	Little-endian
		1	Big-endian
5	WSF	0	$\overline{WRL}$ write strobe is activated
		1	$\overline{WR}$ write strobe is activated
4	STS	0	Strobe timing scheme 0 is selected
		1	Strobe timing scheme 1 is selected
3	ACE	0	Address cycle is not extended
		1	Address cycle is extended by one ECLK cycle
2:0	R[2:0]	Automatic wait cycles	
Gray Settings used for MB88121 connectivity			

Note:

1. Bit ACE decides the length of the address cycle in multiplexed bus mode
2. Bit STS selects the timing of the signal ALE, $\overline{WR}$ and $\overline{RD}$ (for details please see the hardware manual of MB96300 series)

Table 11. EASZ [2:0] of register EACH

EASZ2	EASZ1	EASZ0	Area size	Valid bit of register EAS
0	0	0	64KB	EAS_A[7:0]
0	0	1	128KB	EAS_A[7:1]
0	1	0	256KB	EAS_A[7:2]
0	1	1	512KB	EAS_A[7:3]
1	0	0	1MB	EAS_A[7:4]
1	0	1	2MB	EAS_A[7:5]
1	1	0	4MB	EAS_A[7:6]
1	1	1	8MB	EAS_A[7]

Table 12. R [2:0] of register EACL

R2	R1	R0	Automatic wait cycles
0	0	0	No wait cycle
0	0	1	1 cycle
0	1	0	2 cycle
0	1	1	3 cycle
1	0	0	4 cycle
1	0	1	8 cycle
1	1	0	16 cycle
1	1	1	32 cycle

3.4.6 Registers EAS [5:2]

MB96340 series support 6 external bus areas. Among them area 0 and area 1 have fixed address range.

Table 13. Address range of area 0 and area 1

External area	Address range
CS0	0x0000F0—0x0000FF
CS1	0x000C00—(RAM_start-1)

Table 14. RAM start address of different devices

Device	RAM size	RAM_start
MB96344	6KB	0x006A40
MB96(F)346, MB96(F)347	16KB	0x004240
MB96F348	24KB	0x002240

The address range of area 2 to area 5 is programmable through register EAS [5:2] and register EACH [5:2].

Table 15. Configuration registers for area range

External area	Available address area	Corresponding register
CS2	0x100000—0xFFFFFF	EAS2, EACH2_EASZ
CS3		EAS3, EACH3_EASZ
CS4		EAS4, EACH4_EASZ
CS5		EAS5, EACH5_EASZ

Register EAS defines the upper 8-bit address (start bank) of the external bus area. Together with bits EASZ [2:0] of register EACH the address range of each external bus area can be specified.

The configurations in file start.asm are listed below. Area 2 is modified for MB88121B. Area 3/4/5 are not used.

Table 16. Configurations of area range in file start.asm

EAS register	EACH register	Area range
EAS2=0x10 (parameter CS2_START)	EACH2_EASZ[2:0]=0 (parameter CS2_CONFIG)	0x100000—0x10FFFF (64KB)
EAS3=0x40 (parameter CS3_START)	EACH3_EASZ[2:0]=6 (parameter CS3_CONFIG)	0x400000—0x7FFFFF (4MB)
EAS4=0x80 (parameter CS4_START)	EACH4_EASZ[2:0]=6 (parameter CS4_CONFIG)	0x800000—0xBFFFFF (4MB)
EAS5=0xC0 (parameter CS5_START)	EACH5_EASZ[2:0]=6 (parameter CS5_CONFIG)	0xC00000—0xFFFFFF (4MB)

3.4.7 Settings in file start.asm

Cypress offers Software examples for MB96340 series, including a template project. The start.asm file is used to initialize the MCU. All settings are done via defines, according to the selected defines a code content is generated during compilation.

Following find the required bus interface settings for start.asm file using MB96340 and MB88121 series.

```
;===================================================================
; 4.8  Clock Selection
;===================================================================
#set      CRYSTAL         FREQ_4MHZ
#set      CPU_48MHZ_CLKP2_16MHZ              0x08
#set      CLOCK_SPEED     CPU_48MHZ_CLKP2_16MHZ
;===================================================================
; 4.10 External Bus Interface
;===================================================================
#set      BUSMODE INTROM_EXTBUS    ; bus mode
#set      ADDRESSMODE MULTIPLEXED  ; address-mode
#set      CHIP_SELECT2       ON    ; enable chip select area
#set      HOLD_REQ           OFF   ; HOLD function
#set      EXT_READY          ON    ; Ready function
#set      EXT_CLOCK_ENABLE   ON    ; external bus clock output
#set      EXT_CLOCK_INVERT   ON    ; clock inversion
#set      EXT_CLOCK_SUSPEND  OFF   ;
#set      EXT_CLOCK_DIVISION  EXT_CLOCK_DIV2 ; select clock divider
#set      ADDR_PINS_23_16    B'00000000     ; select used address lines
#set      ADDR_PINS_15_8     B'11111111     ; A10..A0
#set      ADDR_PINS_7_0      B'11111111     ;
#set      LOW_BYTE_SIGNAL    OFF   ;
#set      HIGH_BYTE_SIGNAL   OFF   ;
#set      LOW_WRITE_STROBE   ON    ; write strobe signal WRLX/WRX
#set      HIGH_WRITE_STROBE  OFF   ;
#set      READ_STROBE        ON    ; read strobe signal RDX
#set      ADDRESS_STROBE     ON    ; address strobe signal ALE/ASX
#set      ADDRESS_STROBE_LVL OFF   ; address strobe active low
#set      CS2_CONFIG  B'0000100000111010  ; Chip Select Area 2 configuration
#set      CS2_START  0x10       ; start bank of chip select area2
```

```
;===================================================================
; 6.6 Set external bus configuration
;===================================================================
MOV  EBCF, #((HOLD_REQ << 7) | (EXT_READY << 6) | (EXT_CLOCK_ENABLE << 5) |
(EXT_CLOCK_INVERT << 4) | (EXT_CLOCK_SUSPEND << 3) | EXT_CLOCK_DIVISION)
MOV  EBAE0,#ADDR_PINS_7_0
MOV  EBAE1,#ADDR_PINS_15_8
MOV  EBAE2,#ADDR_PINS_23_16
MOV  EBCS, #((ADDRESS_STROBE_LVL << 6) | (ADDRESS_STROBE << 5) | (READ_STROBE
<< 4) | (HIGH_WRITE_STROBE << 3) | (LOW_WRITE_STROBE << 2) | (HIGH_BYTE_SIGNAL
<< 1) | LOW_BYTE_SIGNAL)
MOVW EACL2,#CS2_CONFIG
MOV  EAS2, #CS2_START
MOV  EBM,  #((ADDRESSMODE << 7) | ((BUSMODE-1) << 6) | (CHIP_SELECT5 << 5) |
(CHIP_SELECT4 << 4) | (CHIP_SELECT3 << 3) | (CHIP_SELECT2 << 2) |
(CHIP_SELECT1 << 1) | CHIP_SELECT0)

MOV  PIER00,#0xFF ;MCU port 0
MOV  PIER01,#0xFF ;MCU port 1
SETB PIER03:6     ;READY function
ROMM_CONFIG    .EQU     ((MIRROR_BANK << 4) | (MIRROR_SIZE << 1) |
(ROMMIRROR))
```

3.4.8 Used Bus Interface Functions at pins on MB96F348RS

All the used pins of the external bus interface are listed below. Data output function of pins AD [15:00] is automatically enabled. Pin ALE is used to distinguish value on the address/data lines. Pin RDY is used by slow external devices to insert wait cycles. The initial value of register DDR (data direction register) and PIER (port input enable register) is 0.

Table 17. Configuration registers of I/O port function

MCU pin	Description	Register PIER	Register DDR	Other registers
RDY	ready signal	PIER03_IE6=1	DDR03_D6=0	EBCF_RYE
AD[15:0]	address output	\	\	EBAE[2:0]
	data output	\	\	\
	data input	PIER00=0xFF PIER01=0xFF	DDR00=0 DDR01=0	\
ALE	address latch enable signal	\	\	EBCS_ASE
CS2	chip select signal	\	\	EACH_CSE
RDX	read signal	\	\	EBCS_RDE
WRX	Write signal	\	\	EBCS_WRLE
ECLK	external bus clock	\	\	EBCF_CKE

MB96348RS provides maximal 24bit address-line and 16bit data-line for the external bus interface. Since the communication controller MB88121B has 11bit address-line and 16bit data-line, the bus width is fixed.

- 16bit data bus: MCU pins AD [15:00].
- E-Ray registers are 16bit addressed from 0x0000 to 0x07FC. Each register is 32bit long and begins at even address. Since the data bus is restricted to 16bit, the high order 16bit data and low order 16bit data are distinguished by address bit a [1]. The first five bits and the last bit of a 16bit address are always zero. The 11bit address bus (MCU pins AD [10:00]) is shown below. Address A [23:16] are not used.

MCU **port 0** and **port 1** are reused for data and address bus.

Table 18. 11bit address line

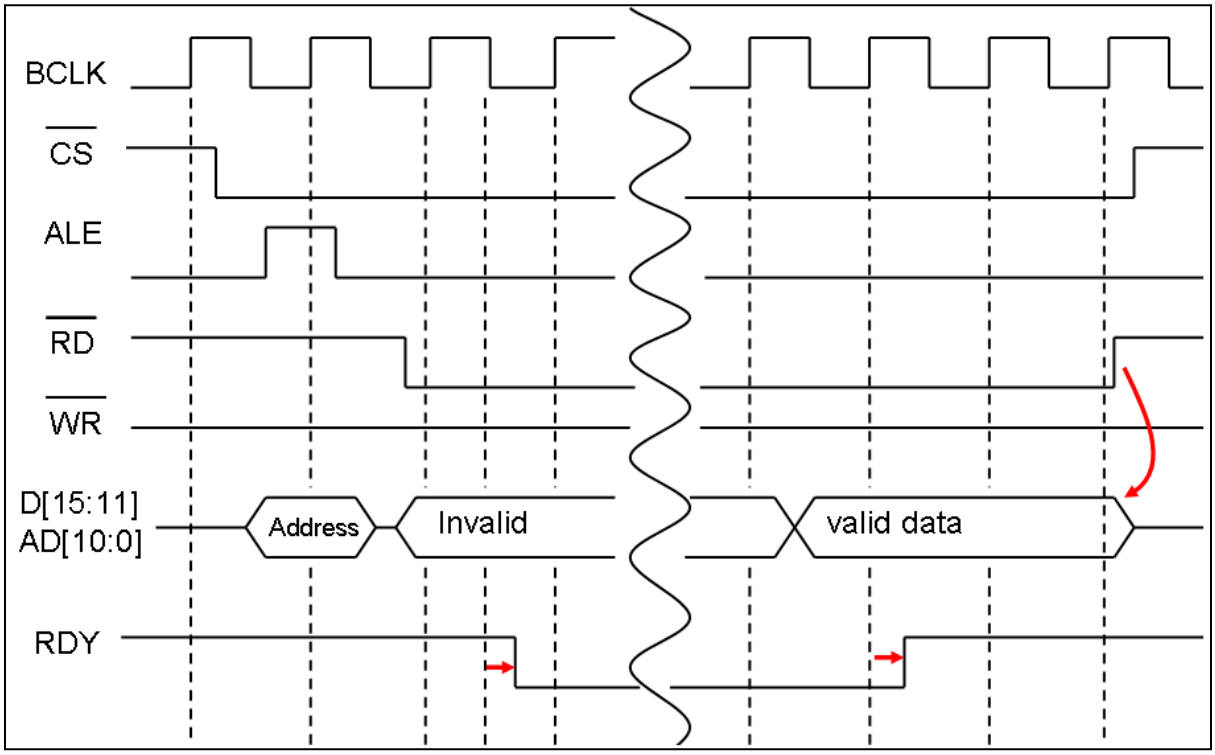
0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	0
Always zero					Address A[10:2]									A[1]	A[0]

3.5 Simplified Timing Diagram of MB88121B

The below shown timing diagram is required to achieve a correct access to MB88121. The settings discussed for the bus interface register will generate this timing. Depending on PCB layout the number of wait states may differ. This needs to be rechecked and optimized case by case.

3.5.1 MB88121 read timing

Figure 3. Read operation

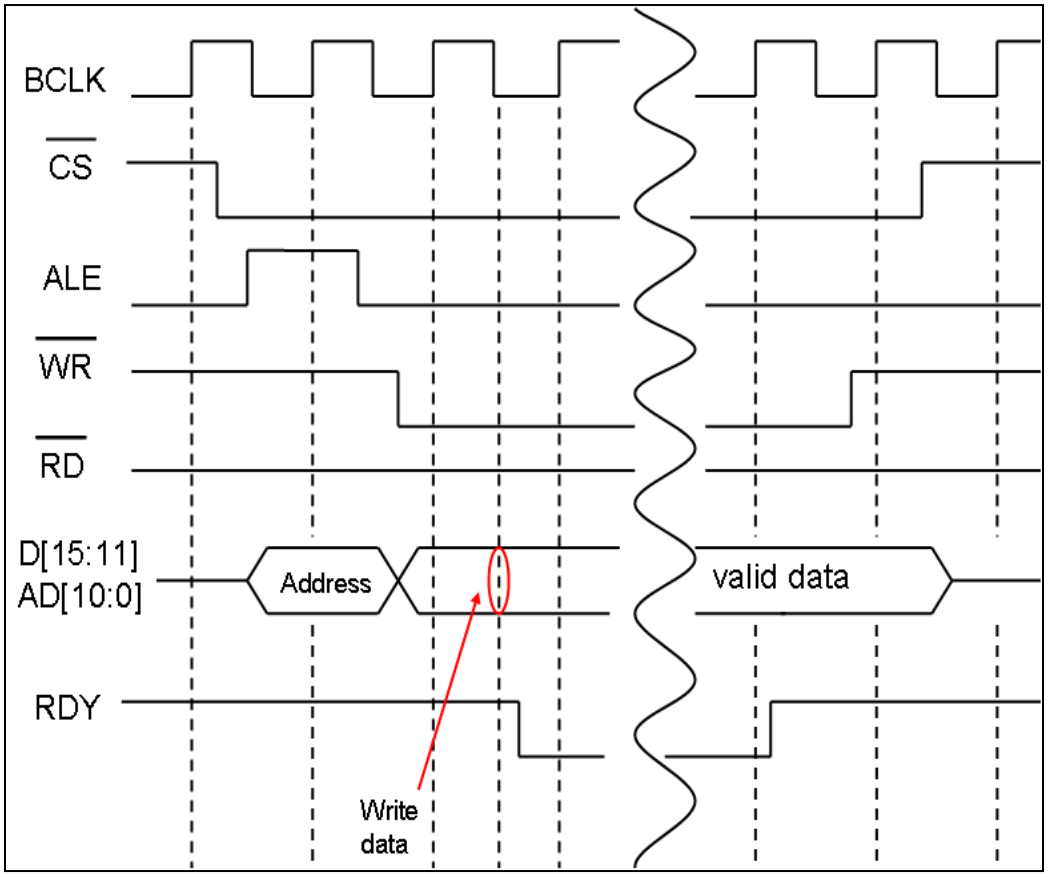


Operation sequence:

1. The address is latched by the falling edge of ALE pin.
2. The RD pin (signal $\overline{RD}$) becomes low level, the output data on pins D [15:11] and AD [10:0] is invalid.
3. After that the RDY pin becomes low level at the next falling edge of the clock signal BCLK (pin BCLK), which causes the MCU to wait.
4. After several wait cycles the RDY pin becomes high level at the rising edge of the BCLK pin and the valid data is output from the pins D[15:11] and AD[10:0].
5. When the RD pin becomes high level again, the read access is finished. Pins D [15:11] and AD [10:0] become Hi-Z.

3.5.2 MB88121 write timing

Figure 4. Write operation



Operation sequence:

1. The address is latched by the falling edge of ALE pin.
2. The WR pin (signal $\overline{WR}$) becomes low level. At the next falling edge of the BCLK signal the data on pins D [15:11] and AD [10:0] is written to a temporary register and the RDY pin becomes low level, which causes the MCU to wait.
3. The data in the temporary register is written to the addressed register. The RDY pin becomes high level again.

> **Note:**
> Details about RDY wait cycle and byte ordering are located in the MB88121 data sheet.

3.6 Initialization Sequence of MB88121

After the proper setup of the bus interface MB88121 is visual in the MCU address range starting at 0x10.0000.

The MB88121 requires internal operation frequency of 80 MHz this frequency is generated via the internal PLL of the MB88121.

First access to MB88121 should setup the PLL. The PLL is set in the CLOCK CONTROL REGISTER (CCNT) (Address 0x04 => 0x10.0004).

The example is using a 10 MHz crystal. The PLL must be setup to multiplication ratio x8 (PMUL [1:0]) to achieve the 80 MHz the PLL must be enabled via PON bit. To supply the Flex Ray CC with clock the STOP bit needs to be set to '0'.

Other bits shall be set to '0'.

The stabilization time of the PLL is 600us. After this time the MB88121 the PLL clock usage can be selected via the CCNT.SEEL bit (set to '1').

The wait time can be assured using a reload timer of the MCU.

The MB88121 is operating with 80 MHz It is now possible to initialize the MB88121 with the Flex Ray bus parameter settings.

```
/*                          SAMPLE CODE                                  */

#define CCNT ((uint32_t *)0x100004)   /* address of CCNT ERAY Register
*/
void start_rldtmr_3 (void){
   TMCSR3_TRG = 1;          /* start count operation */
}
/* setup of reload timer 3 */
  TMCSR3_CNTE = 0;          /* stop reload Timer */
  TMRLR3 = 600;             /* set reload value */
  TMCSR3 = 0x0802;

/* set MB88121 Clock */
*CCNT = 0x0000000D;         /* enable PLL, PLLx8 */
start_rldtmr_3();           /* start wait time */
while (!TMCSR3_UF);         /* PLL stabilization wait time */
*CCNT = 0x0000000F;         /* switch to PLL clock */
```

3.7 Interrupts

The MB88121 Flex Ray CC is supporting also interrupt events. These events are available at pins and can be connected to MCU external interrupt pins.

3.7.1 CC Interrupt

CC interrupt pins signal different interrupt request. For details please refer to the document “MB88121B preliminary data sheet” and “E-Ray user manual”.

Table 19. CC Interrupt output by 16bit multiplexed mode

CC pin	Interrupt type
INT0	E-Ray interrupt line0 (eray_int0)
INT1	E-Ray interrupt line1 (eray_int1)
INT2	E-Ray timer0 interrupt
INT3	E-Ray timer1 interrupt
INT4	Low voltage detection interrupt

After power-on / Reset these pins are set to output driving Low level. Interrupt requests are indicated by High Level output at the pins.

To enable the output of the signals INT2 to INT 4 the INT (Interrupt register) at offset address 0x0C (Address 0x10.000C in our example configuration) must be used. The dedicated E-Ray timer interrupts are indication a request just a few Macro ticks, It is recommended to use rising edge detection for the external interrupts.

The INT0 and INT1 pins are connected to the Eray_int0 and Eray_int1 interrupts. The functions are configured with the E-Ray interrupt registers. In case of an interrupt request the pins output High-level and remaining the level until the Interrupt Flag is clear in the corresponding register.

Following interrupt registers are available:

- Error Interrupt Register (EIR): indicates an error interrupt request
- Status Interrupt Register (SIR): indicates a status interrupt request
- Error Interrupt Line Select (EILS): Selects which error interrupt is output at which eray_int (0 or 1) line.
- Status Interrupt Line Select (SILS): Selects which status interrupt is output at which eray_int (0 or 1) line.
- Error Interrupt Enable Set / Reset (EIES, EIER): Enable / disable error interrupts
- Status Interrupt Enable Set / Reset (SIES, SIER) : Enable / disable status interrupts
- Interrupt Line Enable (ILE): Enable interrupt lines

3.7.2 MCU External Interrupt

Following show the setup and usage of the external interrupt (rising edge) for MB88121 connection.

Following registers are relevant for external interrupt:

- ELVR: interrupt request level register
- EIRR: interrupt request register
- ENIR: interrupt request enable register
- DDR: data direction register
- PIER: port input enable register
- ICR: interrupt control register

3.7.2.1 External Interrupt Configuration registers

Register ELVR, EIRR and ENIR specify the external interrupt function. ELVR defines how to detect interrupt request signal. EIRR shows if an external interrupt request is detected. ENIR enables/disables interrupt request.

Table 20. EIRR, ENIR ELVR overview

External interrupt 0-7									
		Bit value							
register	**address**	7	6	5	4	3	2	1	0
ENIR0	0x000058	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
EIRR0	0x000059	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
ELVR0	0x00005B	15	14	13	12	11	10	9	8
		LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4
	0x00005A	7	6	5	4	3	2	1	0
		LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
External interrupt 8-15									
		Bit value							
register	**address**	7	6	5	4	3	2	1	0
ENIR1	0x00005C	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
EIRR1	0x00005D	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8
ELVR1	0x00005F	15	14	13	12	11	10	9	8
		LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12
	0x00005E	7	6	5	4	3	2	1	0
		LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8

Table 21. ELVR register

LBn	LAn	Description
0	0	Detect “L” level and generate an interrupt request
0	1	Detect “H” level and generate an interrupt request
1	0	Detect the rising-edge and generate an interrupt request
1	1	Detect the falling-edge and generate an interrupt request

CC interrupt pins are high-level active. Therefore signal high-level or rising-edge should be chosen for external interrupt detection.

Table 22. EIRR register

ERn	Read register	Write register
0	No external interrupt request present	Clear external interrupt request flag
1	External interrupt request present	No effect

Table 23. ENIR register

ENn	Description
0	External interrupt request is disabled
1	External interrupt request is enabled

3.7.2.2 I/O Port Register DDR and PIER

MB96F348RS have altogether 16 external interrupt channels. User should choose five of them for MB88121B.

The external interrupt function is assigned to MCU port00 and port07. Each of them has an 8bit data direction register (DDR) and an 8bit port input enable register (PIER). The default value of register DDR and PIER after reset is zero, respectively all ports are input mode and disabled. To use the external interrupt function only register PIER should be set to one.

Table 24. Register DDR and PIER

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	description
PIERxx	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0	IEx=0 disable digital input IEx=1 enable digital input
DDRxx	D7	D6	D5	D4	D3	D2	D1	D0	Dx=0 input mode Dx=1 output mode

An initialization example of the external interrupt 0 is shown below.

```
void Init_extint_0 (void){
   ENIR0_EN0 = 0;    /* disable extInt0 interrupt */
   DDR07_D0 = 0;     /* port07.0 Data direction: input */
   PIER07_IE0 = 1;   /* enable digital input */
   ELVR0_LB0 = 1;    /* rising edge detection */
   ELVR0_LA0 = 0;
   EIRR0_ER0 = 0;    /* clear interrupt request flag */
   ENIR0_EN0 = 1;    /* enable interrupt */
}
```

Before enabling the external interrupt request (ENn = 1) it is recommended to clear the request flag of the external interrupt (ERn = 0) to avoid interrupts caused by previous trigger (ERn is set independently of the setting of ENn).

3.7.2.3 Interrupt Vector Table

Register ICR defines the interrupt level. Bits IX [7:0] select the interrupt resource and bits IL [2:0] specify the corresponding interrupt level. 7 priority levels are programmable. Level 0 has the highest priority and level 7 disables the interrupt. The address of each interrupt service routine is set in the interrupt vector. The vector address is calculated by adding the offset (listed in the following table) to the table base register value (TBR).

Register TBR defines the most significant 14 bits (TB [23:10]) of the 24bit start address of the interrupt vector table. The least significant bits TB [9:0] are fixed to 0. The initial value after reset is 0xFFFC, which results in a table start address 0xFFFC00.

Table 25. Register TBR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TB23	TB22	TB21	TB20	TB19	TB18	TB17	TB16	TB15	TB14	TB13	TB12	TB11	TB10	0	0

Table 26. Register ICR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IX7	IX6	IX5	IX4	IX3	IX3	IX2	IX1	IX0	\	\	\	\	IL2	IL1	IL0

Table 27. Interrupt vector table

External interrupt channel	vector number (decimal)	Index IX[7:0] in register ICR (decimal)	Interrupt request cleared by DMA	Interrupt vector	
				Offset	Default vector address
0	17	17	YES	0x3B8	0xFFFC00+ offset
1	18	18		0x3B4	
2	19	19		0x3B0	
3	20	20		0x3AC	
4	21	21		0x3A8	
5	22	22		0x3A4	
6	23	23		0x3A0	
7	24	24		0x39C	
8	25	25		0x398	
9	26	26		0x394	
10	27	27		0x390	
11	28	28		0x38C	
12	29	29		0x388	
13	30	30		0x384	
14	31	31		0x380	
15	32	32		0x37C	

A default interrupt vector table for MB96340 series is located in file **vectors.c** (provided in the template project). An example for the external interrupt 3 is shown below.

```
#define MIN_ICR  12
#define MAX_ICR  96
#define DEFAULT_ILM_MASK 7
void InitIrqLevels(void) /* interrupt level definition */
{
  volatile int irq;
  for (irq = MIN_ICR; irq <= MAX_ICR; irq++)
  {
    ICR = (irq << 8) | DEFAULT_ILM_MASK;
  }
  ICR = 0x1403; /* Ext. INT3 */
}
…
__interrupt void ExtInt3_IRQHandler(void); /* interrupt service routine */
…
#pragma intvect ExtInt3_IRQHandler 20    /* interrupt vector definition */
…
```

3.8 Debugging Support

The MB88121 offering also debug support at some pins.

Table 28. Debug pins on MB88121

CC debug pins				
CC pin Nr	**Name**	**Function**	**I/O type**	**Configuration register**
7	SDS	Start of dynamic segment	OUT	Debug support register DBGS
8	CYCS0	Cycle 0 start	OUT	
14	CYCS	Cycle start	OUT	
23	MT	Macro tick start	OUT	
38	MBSU_TX1	Message buffer status update port	OUT	
39	MBSU_RX1			
44	MBSU_TX2			
45	MBSU_RX2			

After power on / reset these pins are set to output driving Low level.

The upper 16-bit (Bit 31.16) of CUS2 Register is called Debug support register (DBGS), having the offset address 0x08 (0x10.0008). Via these pins the output of the dedicated debug signal is controlled.

The output of the signals is enabled by setting ‘1’ to the corresponding bit position in

3.9 Stop Watch pin

The MB88121 supports the Stop Watch function. The function is similar to Input Capture Unit, the time base is Flex Ray global time. In case an external Signal, connected to Stop Watch pin, is changing its Level the Macro tick value of Flex Ray channel A and B is stored in register and can be read out by host MCU.

This pin is an input pin. If the function is not used, a pull-up or pull-down resistor must be connected to this pin.

4. Reference

- MB96340 series Datasheet
- MB96300 super series Hardware Manual
- Application note: hardware set up for 16FX series (AN204772)
- Application note: external bus interface for 16FX series (AN205546)
- Application note: external interrupts for 16FX series (AN205496)
- Application note: Interrupts for 16FX series (AN205548)
- Application note: I/O-port for 16FX series (AN205493)
- MB88121B preliminary data sheet Ver1.3
- MB88121B User's Manual

Document History

Document Title: AN205256 - F²MC-16FX, MB96340, MB88121, Interfacing MB96340 to MB88121

Document Number: 002-05256

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	03/25/2008	V1.0, MSt First draft
*A	5082297	NOFL	04/21/2016	Converted Spansion Application Note “MCU-AN-300046-E-V10” to Cypress format
*B	5870350	AESATP12	09/01/2017	Updated logo and copyright.

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

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

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

© Cypress Semiconductor Corporation, 2008-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.