Freescale Semiconductor
Application Note

AN3215
Rev. 0, 2/2006

Interfacing MPC55xx Microcontrollers to the MFR4200 FlexRay Controller

by: Steven McQuade
MCD Applications, East Kilbride

1 Introduction

Freescale Semiconductor offers a range of standalone FlexRay communication controllers (MFR4xxx) that can be interfaced with 16-bit and 32-bit microcontrollers (MCU). This application note describes the hardware, software and timing considerations necessary for reliable communication between the MFR4200 controller and the MPC55xx family of MCUs.

2 Objective

This document demonstrates the simplicity of the hardware interface between the MFR4200 and the MPC55xx, and discusses hardware and software considerations when configuring the MPC55xx for operation with the MFR4200. The information contained can help you quickly design a FlexRay node using the MPC55xx family of MCUs. Evaluation boards with software are available from Freescale, to assist in the development of FlexRay applications. (See http://www.freescale.com/flexray.)

Table of Contents

Note: In this document, active-low signals are indicated by a “#” at the end of the signal name, e.g. “IRQn#”.

3 Hardware Design Requirements

The MPC55xx family interfaces with the MFR4200 via the external bus interface (EBI). On the MPC55xx the EBI provides individual address, data and control signals. The MFR4200 must be connected to the MPC55xx using the MFR4200 asynchronous memory interface (AMI) mode. The devices can be connected together without additional glue logic, which simplifies the design and reduces the system cost.

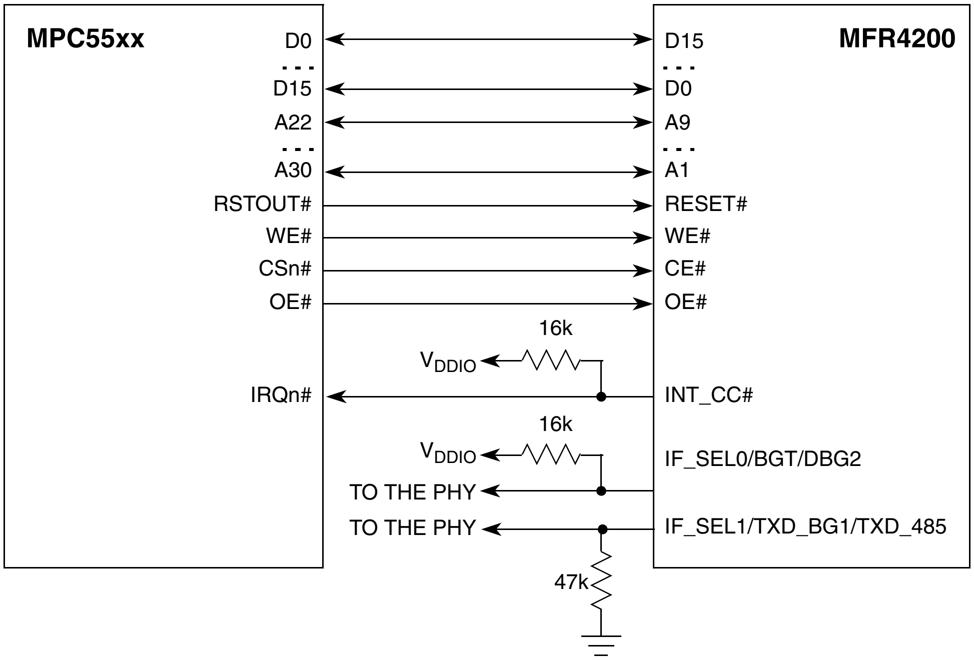

Figure 1. AMI Interface with MPC55xx Family

3.1 Selecting the AMI Mode

There are two modes of operation for the MFR4200 controller host interface (CHI), the AMI mode and the HCS12 mode. The HCS12 mode is used to interface to the HCS12 family of microcontrollers (see reference 4).

To select the AMI mode, IF_SEL0 must be at the logic high level and IF_SEL1 must be at the logic low level. When using 3.3 V V_{DDIO}, IF_SEL0 must be pulled high using a 16 kΩ pullup resistor, and IF_SEL1 must be pulled low using a 47 kΩ pulldown resistor. (Similarly, at 3.3 V V_{DDIO}, INT_CC# must be held high by a 16 kΩ pullup resistor.)

3.2 Bus Signals

3.2.1 Data and Address Pins

On the MFR4200, D0 is the least significant bit (LSB) of the data bus, and A1 is the LSB of the address bus; however, on the MPC55xx, D0 and A1 are the most significant bits. Therefore, the data and address pins must be reverse connected to the MFR4200, i.e., with D0 on the MFR4200 connected to D15 on the MPC55xx, and A1 on the MFR4200 connected to A30 on the MPC55xx.

3.2.2 Control Signals

The MFR4200 has a 16-bit data bus, and either of the following MPC55xx signals can be connected to the MFR4200 WE# signal:

- WE0#
- WE1#

WE0# is asserted if the data lane DATA[0:7] contains valid data

WE1# is asserted if the data lane DATA[8:15] contains valid data

NOTE

Chip selects CS[0:3]# on the MPC55xx can be used to interface to the MFR4200. However, note that CS0# is the global chip select for boot memory.

3.2.3 Voltage Levels

The MFR4200 must be configured for 3.3V I/O (by powering VDDR with 3.3V) to allow correct interfacing to the MPC55xx, whose EBI pins must also be powered from 3.3V. Refer to the electrical specifications provided in the MPC55xx and MFR4200 data sheets (references 1 and 3) for more information.

4 Timing Considerations

For the MPC55xx and the MFR4200 communication controller to communicate reliably, the timing between the MPC55xx EBI and the MFR4200 must be matched.

It can be seen from the respective timing diagrams in the device electrical specifications that the read and write access times to the MFR4200 are greater than the MPC55xx can achieve. To match the timing characteristics of both devices, additional wait states are required. To comply with the MFR4200 t_{CEWE} timing parameter, the MPC55xx EBI must run at 33 MHz or less; however, if the application requires the MPC55xx EBI to operate at a higher frequency, then the MFR4200 chip enable must be tied to ground. More information is provided in Table 2.

Table 1 and Table 2 summarize the read and write timing and the number of wait states (EBI at 33 MHz), required for successful operation. The number of wait states that can be added is controlled in the MPC55xx option registers EBI_OR[0:3], which, in conjunction with the MPC55xx base registers EBI_BR[0:3], control the MPC55xx chip selects CS[0:3]#. Refer to the memory controller section of the appropriate MPC55xx reference manual (reference 1) for more information.

Table 1. MPC55xx to MFR4200 Read Cycle Summary

MFR4200		MPC55xx EBI Timing (ns)		Comments
Description	**Timing (ns)**	**33 MHz Bus, 0 Wait States (Default)**	**33 MHz Bus, 6 Wait States**	
Read cycle time (t_{RC})	155 min	60 ($2T_C$)	180 ($6T_C$)	Minimum of 4 wait states required for t_{RC}. However, due to t_{DOE} a 5th wait state is required at 33 MHz bus operation.
Address setup (t_{SAR})	5 min	5 min	30 (T_C)	Default is OK.
Address hold (t_{HAR})	50 min	20.5 $((T_C\text{-}t_{cov})+t_{coh}))$	140.5 $(5T_C+((T_C\text{-}t_{cov})+t_{coh}))$	However, OK with the minimum of 4 wait states required for a read operation at 33 MHz.
OE# low to data valid (t_{DOE})	145 max	19 min $(T_C\text{-}t_{cov})$	169 $(5T_C$ + (TC-$t_{cov})$	Default is OK.
OE# high (t_{HOE})	30 min	59.5 $((T_C\text{-}t_{coh})+T_C+t_{cov})$	59.5 $((T_C\text{-}t_{coh})+T_C+t_{cov})$	Default is OK. Refer to the MPC5553/4 reference manual (reference 1)
OE# low (t_{LOE})	150 min	15.5 $((T_C\text{-}t_{cov})+t_{coh}))$	165.5	However, OK with the minimum of 5 wait states required for a read operation at 33 MHz.
OE# low to low Z (t_{LZOE})	20 min	Not required for interface	Not required for interface	Not required for interface.
OE# high to high Z (t_{HZOE})	15 max	0	0	Default is OK.
OE# high to CE# high (t_{OEH})	0 min	0	0	Default is OK.
WE# high to OE# low (t_{WEOE})	80 min	69.5 $((T_C\text{-}t_{coh})+T_C+t_{cov})$	69.5 $((T_C\text{-}t_{coh})+T_C+t_{cov})$	Driver software must ensure that timing is met (e.g., insert NOPs between consecutive accesses).

Note:

T_C = System clock period
t_{cov} = CLKOUT positive edge to output signal valid
t_{coh} = CLKOUT positive edge to output signal invalid

Table 2. MPC55xx to MFR4200 Write Cycle Summary

MFR4200		MPC55xx EBI Timing (ns)		Comments
Description	Timing (ns)	33 MHz Memory Access Without Wait States	33 MHz Bus, Wait State Requirement	
Write cycle time (t_{WC})	50 min	60 ($2T_C$)	60 ($2T_C$)	0 wait states required
Address setup (t_{SAW})	30 min	45.5 $((T_C-t_{cov})+T_C+t_{coh})$	45.5 $((T_C-t_{cov})+T_C+t_{coh})$	Default MPC55xx EBI timing OK. 0 wait states required for 33 MHz bus operation.
Address hold (t_{HAW})	5 min	30 $((T_C-t_{coh})+t_{coh})$	30 $((T_C-t_{coh})+t_{coh})$	Default MPC55xx EBI timing OK. 0 wait states required for 33 MHz bus operation.
CE# low to write end (t_{SCE})	50 min	45.5 $((T_C-t_{cov})+T_C+t_{coh})$	75.5 $((T_C-t_{cov})+2T_C+t_{coh})$	1 wait state required for 33 MHz bus operation.
Data setup to write end (t_{SD})	30 min	20.5 $((T_C-t_{cov})+t_{coh})$	40.5 $((T_C-t_{cov})+T_C+t_{coh})$	1 wait state required for 33 MHz bus operation.
Data hold from write end (t_{HD})	5 min	30 $((T_C-t_{coh})+t_{coh})$	30 $((T_C-t_{coh})+t_{coh})$	Default MPC55xx EBI timing OK. 0 wait states required for 33 MHz bus operation.
WE# pulse width (t_{PWE})	30 min	20.5 $((T_C-t_{cov})+t_{coh})$	50.5 $((T_C-t_{cov})+T_C+t_{coh})$	1 wait state required for 33 MHz bus operation.
WE# high time (t_{WEH})	55 min	69.5 $(T_C-t_{coh})+T_C+t_{cov}$	69.5 $(T_C-t_{coh})+T_C+t_{cov}$	Default MPC55xx EBI timing OK. 0 wait states required for 33 MHz bus operation.
Write end to CE# high (t_{CEWE})	30 min	30 $((T_C-t_{coh})+t_{coh})$	30 $((T_C-t_{coh})+t_{coh})$	The MPC5500 EBI frequency must be reduced to 33.3 MHz or less for t_{CEWE} to be 30 ns or greater.
OE# high to WE# low (t_{OEWE})	15 min	69.5 $(T_C-t_{coh})+T_C+t_{cov}$	69.5 $(T_C-t_{coh})+T_C+t_{cov}$	Default MPC55xx EBI timing OK. 0 wait states required for 33MHz bus operation.

Note:

T_C = System clock period
t_{cov} = CLKOUT positive edge to output signal valid
t_{coh} = CLKOUT positive edge to output signal invalid

From Table 1, it can be seen that the read cycle time (t_{RC}) of the MFR4200 device is 155 ns (min) whereas the minimum read cycle time of the MPC55xx EBI operating at 33 MHz (system frequency of 66 MHz) is 60 ns (min). To satisfy the timing requirements of t_{DOE}, five wait states are necessary, resulting in a total read cycle of 210 ns.

From Table 2, it can be seen that one wait state is needed for a write cycle. Care must be taken with the write end to CE# high timing (t_{CEWE}) on the MFR4200. This is specified at 30 ns (min). As the MPC55xx provides one clock cycle between WE# negating and CE# negating, this means that the MPC55xx EBI maximum frequency of operation is 33.3 MHz for correct interfacing to the MFR4200 without any hardware modification. As mentioned previously, if a higher EBI frequency is required, the MFR4200 CE# signal must be tied to ground.

The pause timing between consecutive read/write operations (t_{WEOE}) must also be met; this can be done by controlling the delay during consecutive accesses by, for example, inserting NOPs in the driver code. As the MPC55xx family has an on-chip cache, care must be taken when writing this section of application code, to ensure that the core executes these NOP instructions serially. This can be done by cache inhibiting the code and using the msync and isync instructions to ensure that the code is executed serially. Refer to the e200z6 PowerPC reference manual (reference 2) for detailed information on these instructions.

5 Software

The software setup described in this section is concerned mainly with initializing the MPC55xx to allow the MFR4200 controller host interface to function properly. This involves configuring the EBI pads and memory controller to locate the MFR4200 into the MPC55xx memory map.

Also, note that the MFR4200 magic number register must be read, to ensure that the MFR4200 has completed its internal initialization, before the MPC55xx can access any other register on the MFR4200.

Once both devices are initialized, the MPC55xx can read and write MFR4200 registers.

5.1 Pad Configuration for EBI Operation

Each pin on the MPC55xx must be initialized individually to select the pin function, direction and static electrical attributes. This is controlled by the appropriate MPC55xx system integration unit_pad configuration registers (SIU_PCRs).

NOTE

> The number of the PCR required to configure the pad relates to the GPIO number for the pad, for example CS0 (GPIO0) uses PCR0. Refer to the signal description and system integration unit chapters of the appropriate MPC55xx reference manual (reference 1) for more detailed information.

The MPC55xx memory controller must be initialized for the EBI to use the correct chip select, and the correct number of wait states (which depends on the internal bus frequency). See Section 4, “Timing Considerations” for specific details. Figure 2 shows the steps needed to allow communication between the devices.

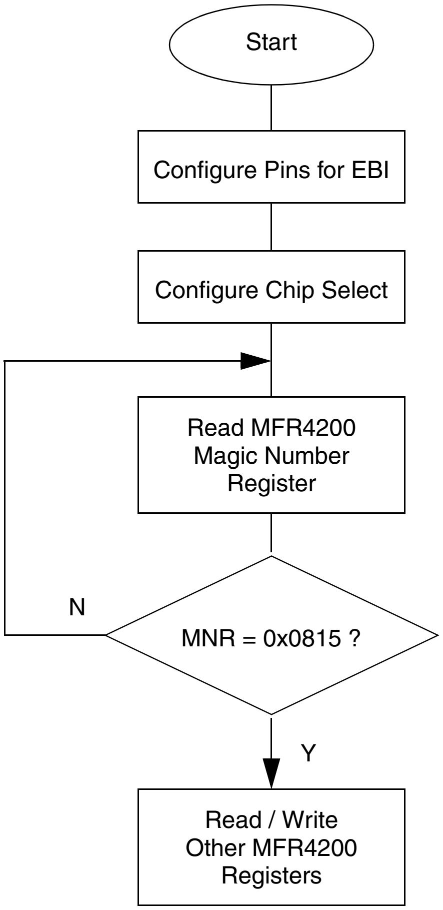
Start
Configure Pins for EBI
Configure Chip Select
Read MFR4200 Magic Number Register
MNR = 0x0815 ?
N
Y
Read / Write Other MFR4200 Registers

Figure 2. Configuration Flow

5.2 Chip Select Configuration

As shown in Figure 2, the second step in the initialization sequence is to configure the chip select on the MPC55xx. The memory controller controls the chip select generation for the MPC55xx. There are four chip selects on the MPC55xx, any one of which can be chosen as the chip select to the MFR4200.

NOTE

> CS0# is the global chip select, which is used primarily for booting from external FLASH memory. If it is used for this primary purpose, it cannot be used for MFR4200 communication.

The option registers (EBI_OR[0:3]) and base registers (EBI_BR[0:3]) are used to configure the chip select. Refer to the memory controller section of the MPC55xx reference manual (reference 1) for more information on chip select configuration.

5.3 MPC55xx Chip Select and FMPLL Settings

From Section 4, “Timing Considerations”, it is shown that wait states are required for successful read/write cycles between the MPC5xx and MFR4200.

It has been shown that a minimum of five wait states are required for a read cycle, and one wait state for a write cycle. This can be set in the option register OR[SCY] field.

For example, if CS1# is being used for communication with the MFR4200, the following register settings could be used:.

Chip Select

```
EBI_BR1          = 0x20080803       ;Base address = 0x20080000, 16 bit port size,
                                    ;burst inhibit, valid chip select
EBI_OR1          = 0xFFF80050       ;Address Mask = 13 bit, 9 wait states
```

FMPLL

```
FMPLL.SYNCR.R    = 0x1E900000       ;MFD = 29, RFD = 2, PREDIV = 1, PLL will lose lock
while (FMPLL.SYNSR.B.LOCK != 1)     ;Wait for PLL to lock before continuing.
FMPLL.SYNCR.R    = 0x1E880000       ;MFD = 29, RFD = 1, PREDIV = 1. The system
                                    ;frequency is now 66 MHz.
```

5.4 MFR4200 Magic Number Register

The MFR4200 magic number register (MNR) contains 0x0000 while the controller is initializing after leaving the hard reset state. Only when this initialization is complete does the MNR contain the value 0x0815. After leaving the hard reset state, the host must not access any of the MFR4200 registers, with the exception of the MNR, which acknowledges the end of the internal initialization procedure. The initialization takes 1025 communication controller clock cycles after negation of Hard Reset.

6 Conclusions

The FlexRay controller can be connected to the MPC55xx family of MCUs, without any glue logic being required, as the EBI on the MPC55xx interfaces directly with the AMI on the MFR4200. Software configuration is also straightforward and the MFR4200 is simply memory mapped into the global address space.

7 References

1. MPC5553/MPC5554 Reference Manual. (MPC5553/4RM)
2. e200z6 PowerPC Core Reference Manual (e200z6RM)
3. MFR4200 FlexRay Communication Controller Data Sheet (MFR4200)

These documents are available on the Freescale Semiconductor web site at http://www.freescale.com.

More information on FlexRay and FlexRay products can be found at http://www.freescale.com/flexray

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor