Motorola Semiconductor Engineering Bulletin

EB263

How to Program Chip Selects on Modular Microcontrollers with a System Integration Module or a Single-Chip Integration Module

By Sharon Darley
       Austin, Texas

Introduction

Modular MCUs with a system integration module (SIM) have 12 chip-select lines, and those with a single-chip integration module (SCIM) have nine chip selects. Both modules'[1] chip selects are programmed in the same fashion.

These chip selects are used to expand the system. A chip-select signal selects and enables a particular peripheral device or memory chip for data transfer. The chip selects can also be programmed to generate data transfer and size acknowledge (DSACK) signals.

The chip-select logic can wait for a certain number of states before generating the acknowledgment. These states are called wait states. The wait states are inserted after the S3 cycle of a read or write bus cycle. A normal bus cycle lasts three clock cycles plus the number of wait clock cycles. Up to and including 13 wait states can be inserted automatically by the chip selects.

EB263

MOTOROLA

Engineering Bulletin

Two problems sometimes encountered when using chip selects are:

- The chip select generates the wrong number of wait states.

- The wrong chip select is enabled.

These problems can be traced to one of two things:

- Either the DSACK lines have been left floating, or

- Two or more chip selects that are programmed to generate different numbers of wait states are responding to the same address space.

This engineering bulletin deals with these two specific problems.

General Information

DSACK Signals

The two DSACK signals are:

- DSACK1
- DSACK0

These are inputs to the MCU that the external device uses to indicate its port size and completion of the bus cycle. The DSACK signal encodings are shown below in **Table 1**. Since these signals are active low, if they are left floating and one of them goes low, the access may be terminated early before the external device can respond. Thus, these signals should be tied high with a pullup resistor or be configured as I/O (input/output) pins. The DSACK pins, along with other bus control signals, are configured as I/O pins by driving data bus pin 8 (DB8) to 0 volts during reset or by programming the appropriate CSPAR bits.

EB263

Freescale Semiconductor, Inc.

Table 1. DSACK Signals

DSACK1	DSACK0	Result
1	1	Insert wait states in current bus cycle
1	0	Complete cycle; 8-bit data bus
0	1	Complete cycle; 16-bit data bus
0	0	Reserved; defaults to 6-bit port

Chip Select Registers

Another reason that a chip select seems to be generating the wrong number of wait states could be that two chip selects are responding to the same address. (Note that it does not matter whether the chip-select pin is connected to anything.) Whether or not two chip selects respond to the same address is determined by both the base address and the block size in the chip-select base address register (CSBARx). **Table 2** shows the format for the chip-select base address registers, and **Table 3** shows the format for the chip-select option registers.

Table 2. Format for CSBARBT and CSBAR0–CSBAR10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	BLKSZ		

Table 3. Format for CSORBT and CSOR0–CSOR10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mode	Byte		R/W		STRB	DSACK				Space		IPL			AVEC

The MCU only uses a base address that lies on an integer multiple of the block size. Here is an example of how to determine if the chip selects are programmed correctly.

1. On a sheet of paper, make a table with four columns as shown in **Table 4**.

    a. Run your initialization code for programming the chip-select base address and option registers.

EB263

b. Then, stop your program, and look at all of the chip-select base address and option registers.

c. For each chip select, fill in the appropriate cell in the table with the value in the corresponding chip-select option register and base address register.

d. In addition, fill in the BLKSZ cell with the block size indicated by the last three bits of the base address register. See **Table 3** for the block size encodings.

Table 4 is filled with example values.

Table 4. Sample Table

CS	CSBARx	CSORx	BLKSZ
CSBOOT	$0003	$68B0	64 k
CS0	$0504	$58FE	128 k
CS1	$1002	$5830	16 k
CS10	$0400	$4C3E	2 k

Table 5. Block Size Encoding

BLKSZ[2:0]	Block Size	Address Lines
000	2 k	ADDR[23:11]
001	8 k	ADDR[23:13]
010	16 k	ADDR[23:14]
011	64 k	ADDR[23:16]
100	128 k	ADDR[23:17]
101	256 k	ADDR[23:18]
110	512 k	ADDR[23:19]
111	1 M*	ADDR[23:20]

* Maximum block size = 512 Kbytes on CPU16-based MCUs, in which ADDR[23:20] = ADDR[19]

2. Compare the base address register values for any overlap. This procedure involves not only comparing the actual base address registers, but also looking at the block sizes and thus the number of address lines compared. The processor does not look at all 24 address lines when it compares for a match. Thus, two chip selects could be interpreted as responding to the same base address, even though their base address registers are programmed differently.

For example, use **Table 4**. Look at the addresses to see any overlap. The answer is yes.

Chip selects 0 and 10 conflict. Both will respond to the address $40000. CS0 is programmed to start at $50000; however, it is programmed incorrectly. It does not lie on an even boundary of 128 Kbytes. (Remember that 1 K = 1024.) As shown in **Table 5**, only address lines 23 through 17 are compared for a block size of 128 Kbytes.

This is how the address decoding process works:

```
CSBAR0: $0504 = base address of $50000 and block size of 128 K.
23 20 19 16 15 12 11 8 7 4 3 0
$5000 = %0000 0101 0000 0000 0000 0000
```

For a 128-K block size, address lines 16–0 are decoded as 0. As a result, bits 19–16 are read as a 4 instead of a 5, and CS0 responds to the address space starting at location $40000.

However, CS10 is also programmed to respond to the address space starting at location $40000. It is acceptable for two chip selects to respond to the same address space; however, in this case, the option registers are programmed for a different number of wait states. As a result, the termination signal will be indeterminate.

To correct the problem, CSBAR0 and/or CSBAR10 must be changed. If CSBAR0 is changed to $0404 and CSBAR10 is changed to a different, non-overlapping base address, then $40000 is an acceptable base address for CS0, since it is an exact multiple of 128 Kbytes.

EB263

That is:

1 K = $400

128 K = $20000

$40000/$20000 = 2 = an integer

Therefore, $40000 is on a 128-Kbyte boundary.

Summary

To summarize, the DSACK lines must not be left floating unless they are configured as I/O pins. Each chip-select base address must lie on an exact multiple of its block size. If two chip selects are programmed to respond to the same address space, then they must be programmed with the same number of wait states and identical STRB and mode fields.

EB263

Freescale Semiconductor, Inc.

EB263

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217, 1-800-441-2447 or 1-303-675-2140. Customer Focus Center, 1-800-521-6274

JAPAN: Nippon Motorola Ltd.: SPD, Strategic Planning Office, 141, 4-32-1 Nishi-Gotanda, Shinagawa-ku, Tokyo, Japan. 03-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

Mfax is a trademark of Motorola, Inc.

© Motorola, Inc., 1998

MOTOROLA

EB263/D

For More Information On This Product,
Go to: www.freescale.com