Application Note

AN2248/D
Rev. 1, 02/2002

Using the MGT5100 SDRAM Controller

by **Mark Jonas and Davide Santo**
Driver Information Systems
Munich, Germany

Introduction

Synchronous Dynamic RAM (SDRAM) and Double Data Rate Synchronous Dynamic RAM (DDR-SDRAM or simply DDR) are among today's preferred memories where high speed and low power are needed.

One specific advantage of the MGT5100 is the support of fast Synchronous Dynamic RAM by a dedicated memory controller and a physical interface separate from the peripheral bus (also known as the local plus Bus). This dedicated memory controller has its own set of registers to control its operation which should simplify the design of memory subsystem hardware and software. The MGT5100 SDRAM Controller provides all the handshaking signalling for both SDRAM or DDR architectures allowing a glueless and straightforward connection of the memory to the processor.

A total of 5 registers are available to program the memory controller which is very flexible and can easily be adapted to many different types of SDRAM or DDR with minimum effort.

This application note will explore the characteristics and the programming environment of the MGT5100 memory controller and provide some simple examples for reference.

What is SDRAM

SDRAM and the recently introduced DDR-SDRAM are dynamic RAMs (Random Access Memory) operating synchronously to the processor which controls them. Transactions of data are faster and more efficient than with more conventional memory devices, such as FPRAM (Fast Page Mode Dynamic RAM) or EDORAM (Extended Data Output Dynamic RAM) or BEDORAM (Burstable EDORAM). As a matter of fact, SDRAM memory includes all the specific advantages of these other devices (such as bursting, fetching the next block of memory when the previous one is sent over the bus, etc.).

DDR-SDRAM, also called SDRAM-II, uses both clock edges to transfer data during a transaction, allowing a “double data” rate at the same clock speed if compared with SDRAM. Also DDR memory will use only 2.5V supply voltage versus the typical 3.3V used by the SDRAM type, thereby reducing power consumption.

For both kind of memories, the synchronous operation tends to increase the number of instructions the processor can perform in a given time. The main difficulty in using SDRAM and DDR is due to the unusual architecture (different banks divided into rows and columns) of the SDRAMs. Their control can appear to the inexperienced person as cumbersome because many parameters have to be taken into consideration.

Part of the design goal behind the MGT5100’s Memory Controller was to achieve an easy to use program (all under SW control) and an easy to use interface for the user of the processor.

Synchronous Dynamic RAM characteristics

In general, the characteristics of SDRAM/DDR memory can be summarized as follows:

1. Internal matrix organization: address of a memory location is decoded using row and column lines which are time multiplexed as they share the same pins on the controller.
2. Refresh: being a dynamic memory, a refresh mechanism is needed to keep memory content valid. The MGT5100 allows self-refresh to be active even in the so called “deep sleep” mode, when no clocks are running inside the core.
3. Banks: the memory is divided into banks to allow more efficient transactions. A bank must be precharged before any one of its location can be read or written.
4. Memory access is time multiplexed: commands (such as read, active, write, etc.). are sequenced to allow access to a location. For example, a typical access to a new bank’s location would be achieved by the Precharge, Active, Read/Write sequence. It is important that the memory controller “controls” all the specific timings and delays during the sequence.
5. Single and multiple reads (called Burst) or writes operations are allowed
6. Initialization of a SDRAM device consists of a pre-defined procedure which includes writing to dedicated registers located internally in the controlled device (the MODE and EXTENDED MODE registers).
7. CAS Latency (CL): this is the most important parameter of a SDRAM and represents the delay between a Read command issued by the controller and the instant when the data is driven onto the bus by the controlled device, and subsequently can be latched by the core. MGT5100 supports a CL equal to 2 or 3 for SDR, while for DDR the value can be 2, 2.5 or 3.

As already mentioned, MGT5100’s memory controller supports invisible control of the memory device by taking responsibility of all the hand-shaking once the appropriate setup for the memory has been completed.

Supported Architecture

The MGT5100 provides a simpler usage model for DRAMs compared to other processors of the PowerPC family (ex: PPC823, PPC8260, etc.). The price for this ease of use is the restriction that NOT all SDRAM/DDR devices available on the market can be used, but the memory controller covers most of them. The limitations of the controller are shown in detail below.

- MGT5100 supports only a 32-bit wide external memory system architecture (which can be achieved using a single 32-bit device or two 16-bit devices, or even four 8-bit devices). Single 16-bit devices can not be used on MGT5100. Internally, the XLB data bus is 64 bits wide so that the data fetched from the external device is aligned on a long word (64-bit) boundary.
- The MGT5100's internal address bus allocates up to 26 bits to control the external dynamic memories to each of two external chip selects.
- With 26 address bits and 32 data bits, the maximum memory size allowed is 256 MBytes (128 MBytes per chip select). Up to 4 banks are used, leaving a total of 24 internal address lines for row and column (including the internal address line connected to the chip select).
- On the pins, the MGT5100 presents 13 address lines on which rows and columns are multiplexed in time. Up to 13 rows and 12 columns can be controlled but not at the same time. The sum of rows and columns must always be less than or equal to 24.
- A minimum number of eight columns "used" by the memory must be assured for proper operations. For this reason, the memory controller does not support those 4M x 32-bit devices which uses columns 0 through 6, only! (See the note below.)
- The memory controller also will NOT work with devices which have more than 8 columns but less than 12 rows.

Always refer to the MGT5100 User Manual for a detailed table of the supported architectures.

***Note**: There are 32-bit wide devices on the market, such as the MT46V4M32 by Micron, whose internal architecture uses a minimum of 8 columns and therefore, can be used along with the MGT5100's memory controller.*

Chip Selects

There are two physical chip select pins on the MGT5100. One is always used as chip select function (and has no other possible function), while the second has, at boot, the default function of GPIO (GPIO_WKUP_6). The total addressable memory is split into equal parts on the 2 chip selects even when one is not used (by programming of the Address Select Register located at MBAR+MEMCTL+0x100). This implies that up to 128 MBytes can be controlled by each single chip select.

The signals controlling each chip selects are derived directly from the internal bus address lines A3, A4, A5 or A6 depending whether 128 MBytes, 64 MBytes, 32 MBytes or just 16 MBytes are used per chip select.

See Appendix B at the end of this application note for a table reporting some possible devices which are supported by the controller. The table shows how the physical columns and rows are re-mapped on the internal XLB bus. It also shows which internal signal is used, case by case, to control the 2 external chip select pins.

As an example, let’s discuss a 512Mbit DDR memory by MICRON which comes in two different flavours:

a. MT46V64M8 which is a 64Maddress x 8 bits (4 banks)
b. MT46V32M16 which is a 32MAddress x 16 bits (4 banks)

Case (a) needs to be controlled by using A0-A12 for rows and A0-A9 plus A11 for the columns. (A10 is used in connection with the precharge operation; therefore, it is not really part of the address.) With the sum of rows and columns being equal to 24, the control of the device should be possible. Indeed, this is not the case !

In fact, the memory controller internally multiplexes some higher row and column lines onto the internal address lines A3 to A6. The selection of which row/column will be placed on which internal line is done via software at initialization time by writing the “Control register’s addr_sel” bits (bits 6:7) according to the following table. (Please note that bit 7 shall be always 0.)

Table 1Row and Columns Multiplexing Table

addr_sel bits	PPC internal address lines mapped to COL or ROW address			
	A3	A4	A5	A6
00	COL 11	COL 10	COL 9	COL 8
01	COL 10	COL 9	COL 8	ROW 12
10	X	X	X	X
11	X	X	X	X

The obvious choice would be setting the add_sel bits to b’01’. It is clear that Column Address Line 11 is not supported by this multiplexing and therefore the MGT5100 will NOT be able to use this architecture.

Instead, Case (b) needs only columns A0-A9 (see data sheet for more details). In this circumstance, by choosing the option b’01’ because A10 is not used by the device, the internal address line A3 can be “re-routed” externally on one chip select.

Careful analysis of the memory data sheet is highly recommended as the same device (512 MBits DDR) might have different architectures, NOT all of which will be supported.

Maximum number of device on the bus

The MGT5100 dynamic bus interface specifies an absolute maximum load of 25pF on the memory bus with a typical value of 17pF. Considering these values and the high frequency of operation, it is generally recommended for high performance to limit the number of memory devices supported by a single chip select to one or two.

SDRAM Controller Registers

To set up the memory controller for operation with SDR or DDR SDRAM, only 5 registers are needed.

Mode Register

The Mode Register is a general register address to facilitate access to the Mode Register and the Extended Mode Register (for DDR only), respectively, of the SDRAM memory device. It does not have an effect on the memory controller of the MGT5100 directly.

Control Register

This register is used to set up the main characteristics of the memory controller. It can select whether MGT5100 drives SDR or DDR memory, the row / column multiplexing scheme and many other things. The MSB of this register enables write access to the Mode Register.

Config1 Register

Single Read and Single Write timings are covered by this register.

Config2 Register

Burst Read and Burst Write timings are covered by this register.

Config3 Register

This register is also known as Addr_Sel Register. It is used to set the maximum amount of memory accessed by each of the two available SDRAM chip selects. That is, it defines an address boundary where the total memory is split between the two chip selects.

Examples

Example 1: SDRAM

This example demonstrates how to set up the MGT5100 memory controller to access two Micron MT48LC16M16A2 devices, which results in 64 MBytes of memory.

First, the environment of the SDR SDRAM chips must be defined. In this case, the XLB speed is 66MHz and only one chip select is going to be used. Now the configuration values for the SDRAM controller can be calculated.

Mode Register

The mode_code field’s value is defined in the data sheet of the SDRAM and must be taken from there. For this example, the burst length is eight, burst type is sequential, CAS latency is two, operating mode is standard and write burst mode is set to the programmed burst length. Additionally, the Mode Register’s write_strobe bit is set to one to generate a write strobe pulse to write the valid data into the SDRAM. Therefore, the value written to the Mode Register is 0x008D0000.

Control Register

The value written to this register is 0x51470000. This is equivalent to:

- SDRAM clock is enabled
- Memory device is of SDR type
- Refresh counter is enabled
- Fast XLB bus clock is not enabled
- Row / column multiplexing according to **Table 1** is “01” (column 10, 9, 8 and row 12 are multiplexed respectively to internal addresses 4 to 7)
- SDRAM controller drives the data bus always, except during read (to save power)
- Refresh counter is set to 0x07

The refresh counter is calculated as follows: the basic information is the minimum refresh time, which can be found in the SDRAM data sheet. For our example it is 7.81 µs. The formula to calculate the refresh counter value is:

$$\mathrm{refresh\_count} = \left\lfloor \frac{\mathrm{XLB\_freq} \cdot \mathrm{refresh\_time\_period}}{64} \right\rfloor - 1$$

Therefore, refresh_count is equal to 0x07.

Config1 Register

The Config1 Register is used to store the delay values between the single read and single write commands. The SDRAM controller uses these values to generate the proper delays when accessing the device. The delays are expressed in XLB bus clock cycles. Some of these values are fixed by the design of the SDRAM controller and some must be calculated using the memory device’s data sheet. Please refer to the MGT5100 User Manual for the detailed description of each field.

For this example the value written to this register is 0xC2222600. This is equivalent to:

- Single read to read / write delay is 12 XLB clocks (this is a conservative setting)
- Single write to read (or pre charge) delay is 2 XLB clocks
- Read CAS latency is 2 XLB clocks
- Active command to read / write delay is 2 XLB clocks
- Precharge command to active command delay is 2 XLB clocks
- Refresh to active command delay is 6 XLB clocks
- Write latency delay is 0 XLB clocks (SDR does not have a latency on the write command)

Config2 Register

The Config2 Register is used to store the delay values for burst access. The same applies as mentioned previously for the Config1 Register.

For this example the value written to this register is 0x88B70004. This is equivalent to:

- Burst read to read / pre charge delay is 8 XLB clocks
- Burst write to read / write delay is 8 XLB clocks
- Burst read to write delay is 11 XLB clocks
- Burst length is 8 XLB clocks (the value written must be one less)
- Read delay tab is set to default, i.e. 4 XLB clocks

Config3 Register

The Config3 Register, also known as Adr_Sel Register, is used to partition the available memory between the two SDRAM controller chip selects.

For this example the value written to this register is 0x0002. This is equivalent to 64 MBytes per chip select. Therefore, in this example the entire memory is covered by a single chip select.

Example 2: DDR

This example demonstrates how to set up the MGT5100 memory controller to access four Micron MT46V16M8 devices, which results in 64 MBytes of memory.

First, the environment of the DDR SDRAM chips must be defined. In this case, the XLB speed is 66MHz and only one chip select is going to be used. Now the configuration values for the SDRAM controller can be calculated.

This example will only show the differences to the SDR example above.

Mode Register

The Mode Register is used here in the same way it has been used in the SDR example. Refer to the memory data sheet for the detailed meaning of the mode_code field's value. Here the value 0x008D0000 is used.

Extended Mode Register

Physically, the Extended Mode Register is at the same address as the Mode Register. To write to the Extended Mode Register, the two most significant bits of the mode_code field must be set to "01". Here the value 0x40090000 is used (enable DLL, reduced drive strength, QFC function disable, normal operating mode).

Control Register

This examples differs from the SDR example in that the DDR specific fields must be filled. These are:

- ddr_mode bit must be set to "1"
- ddr_32bit bit must be set to "1" in case a 32 bit device is used (32 bit devices use A8 instead of A10 to control refresh operation)
- the bits in the ddr_dqs_en field must set to "1" corresponding to the byte lanes the device uses

Here the value 0x714F0F00 is used (minimum refresh cycle time is 15.625 µs).

Config1 Register

Here the value 0x73622730 is used. The major difference to the SDR example is that some fields are expressed in XLB bus clocks and some use double the frequency of the XLB bus. Please refer to MGT5100's user manual for details.

Config2 Register

Here the value 0x45770004 is used. As in the Config1 registers some of the fields use the XLB bus clock, some use double the frequency of the XLB bus.

Config3 Register

The Config3 register is used in the same way in this example as in the SDR example, i.e. using the value 0x0002.

Appendix A - Software Listing SDRAM Controller Initialization

```
.equ    MBAR,              0x80000000   # Module Base Address
.equ    IPBI_SDRAM_START,  0x0034       # Offset to SDRAM start BAR
.equ    IPBI_SDRAM_END,    0x0038       # Offset to SDRAM end BAR
.equ    IPBI_CS_ENABLE,    0x0054       # Offset to CS enable register
.equ    MEMCTL_MODE,       0x0100       # Offset to SDRAM Mode register
.equ    MEMCTL_CONTROL,    0x0104       # Offset to SDRAM Control register
.equ    MEMCTL_CONFIG1,    0x0108       # Offset to SDRAM Config1 register
.equ    MEMCTL_CONFIG2,    0x010C       # Offset to SDRAM Config2 register
.equ    MEMCTL_CONFIG3,    0x0110       # Offset to SDRAM Config2 register

.equ    DRAM_START,        0x00000000   # SDRAM Start address
.equ    DRAM_STOP,         0x04000000   # SDRAM stop address

#.equ   RAMTYPE_DDR        # activate this switch for DDR, else SDR
.if RAMTYPE_DDR
   # Settings for four Micron MT46V32M8 DDR devices, XLB = 66 MHz
   .equ    DRAM_EMODE,    0x40090000
   .equ    DRAM_MODE,     0x008D0000
   .equ    DRAM_CONTROL,  0x714F0F00
   .equ    DRAM_CONFIG1,  0x73622730
   .equ    DRAM_CONFIG2,  0x45770004
   .equ    DRAM_CONFIG3,  0x02000000   # 64 MB total
.else
   # Settings for two Micron MT48LC16M16A2 SDR devices, XLB = 66 MHz
   .equ    DRAM_MODE,     0x008D0000
   .equ    DRAM_CONTROL,  0x51470000
   .equ    DRAM_CONFIG1,  0xC2222600
   .equ    DRAM_CONFIG2,  0x88B70004
   .equ    DRAM_CONFIG3,  0x02000000   # 64 MB total
.endif

########## here the initialization code starts ##########

addis   r8,r0,(MBAR)@h                  # MBAR held in R8
ori     r8,r8,(MBAR)@l

# Step 1) setup SDRAM start / stop address and enable chip select

addis   r9,r0,(DRAM_START >> 15)@h      # set SDRAM start address
ori     r9,r9,(DRAM_START >> 15)@l
stw     r9,IPBI_SDRAM_START(r8)

addis   r9,r0,(DRAM_STOP >> 15)@h       # set SDRAM stop address
ori     r9,r9,(DRAM_STOP >> 15)@l
stw     r9,IPBI_SDRAM_END(r8)

lwz     r9,IPBI_CS_ENABLE(r8)           # enable SDRAM CS
oris    r9,r9,0x0040
stw     r9,IPBI_CS_ENABLE(r8)
```

```
# Step 2) write CONFIG1

addis    r4,r0,(DRAM_CONFIG1)@h
ori      r4,r4,(DRAM_CONFIG1)@l
stw      r4,MEMCTL_CONFIG1(r8)           # write CONFIG1

# Step 3) write CONFIG2

addis    r4,r0,(DRAM_CONFIG2)@h
ori      r4,r4,(DRAM_CONFIG2)@l
stw      r4,MEMCTL_CONFIG2(r8)           # write CONFIG2

# Step 4) write CONFIG3

addis    r4,r0,(DRAM_ADRSEL)@h
ori      r4,r4,(DRAM_ADRSEL)@l
stw      r4,MEMCTL_CONFIG3(r8)           # write CONFIG3

# Step 5) write CONTROL

addis    r4,r0,(DRAM_CONTROL)@h
ori      r4,r4,(DRAM_CONTROL)@l

mr       r5,r4
oris     r5,r5,0x8000
stw      r5,MEMCTL_CONTROL(r8)           # write CONTROL

# Step 6) issue precharge all

mr       r6,r5
ori      r6,r5,0x0002
stw      r6,MEMCTL_CONTROL(r8)           # write CONTROL

# Step 7) write EXTENDED MODE (for DDR only)

.if RAMTYPE_DDR
addis    r6,r0,(DRAM_EMODE)@h
ori      r6,r6,(DRAM_EMODE)@l
stw      r6,MEMCTL_MODE(r8)              # write MODE
.endif

# Step 8) write MODE

addis    r6,r0,(DRAM_MODE)@h
ori      r6,r6,(DRAM_MODE)@l
oris     r6,r6,0x0400
stw      r6,MEMCTL_MODE(r8)              # write MODE

# Step 9) issue precharge all followed by a refresh

mr       r6,r5
ori      r6,r5,0x0002
stw      r6,MEMCTL_CONTROL(r8)           # write CONTROL

ori      r6,r5,0x0004
```

```
    stw      r6,MEMCTL_CONTROL(r8)           # write CONTROL

    # Step 10) issue refresh

    stw      r6,MEMCTL_CONTROL(r8)           # write CONTROL

    # Step 11) write MODE

    addis    r6,r0,(DRAM_MODE)@h
    ori      r6,r6,(DRAM_MODE)@l
    stw      r6,MEMCTL_MODE(r8)              # write MODE

    # Step 12) write CONTROL

    stw      r4,MEMCTL_CONTROL(r8)           # write CONTROL
```

Appendix B - SDRAM SDR Configuration Examples

The following table reports just some possible architectures supported by the MGT5100 Memory Controller. It must be considered as an example only.

SDR Devices	Device Architecture	Row x Column x Banks	Data width per single chip (in bits)	Min. number of devices needed per CS	Min. total available memory per CS (Mbytes)	Physical Address Multiplexing (on the internal XLB Bus)							Control Register bits 6:7	XLB_Sel 6:7	Note
						3	4	5	6	7-18	19-20	21-28			
MT48LC1M16A1	16 Mbit	11x8x2	16	2	4	X	X	X	X	X	X	X	X	X	**Not Supported** because it uses only 2 banks
MT48LC8M8A2	64 Mbit	12x9x4	8	4	32	X	X	CS	Col 8	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘01’	It is recommended NOT to use more than 4 devices allocated on the Memory controller bus!
MT48LC4M16A2		12x8x4	16	2	16	X	X	X	CS	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘00’	
MT48LC2M32B2		11x8x4	32	1	8	X	X	X	CS	Row 10-0	Bank 1-0	Col 7-0	‘00’	‘00’	Row 11 is not driven!!
HYB 39S64800ET-7.5		12x9x4	8	4	32	X	X	CS	Col 8	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘01’	
HYB 39S64160ET-7.5		12x8x4	16	2	16	X	X	X	CS	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘00’	
K4S643232E-TI		11x8x4	32	1	8	X	X	X	CS	Row 10-0	Bank 1-0	Col 7-0	‘00’	‘00’	Row 11 is not driven!!
MT48LC16M8A2 HYB 39S128800CT-7.5	128 Mbit	12x10x4	8	4	64	X	CS	Col 9	Col 8	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘10’	Not recommended to use with both CS
MT48LC8M16A2 HYB39S128160CT-7.5		12x9x4	16	2	32	X	X	CS	Col 8	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘01’	64 Mbytes can be mounted using both CS
MT48LC4M32B2		12x8x4	32	1	16	X	X	X	CS	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘00’	32 Mbytes can be obtained using both CS
MT48LC32M8A2 HYB 39S256800CT-7.5	256 Mbit	13x10x4	8	4	128	X	CS	Col 8	Row 12	Row 11-0	Bank 1-0	Col 7-0	‘01’	‘11’	Not recommended to use with both CS
MT48LC16M16A2 HYB 39S256160CT-7.5		13x9x4	16	2	64	X	CS	Col 8	Row 12	Row 11-0	Bank 1-0	Col 7-0	‘01’	‘10’	128 Mbytes possible when both CS are used
MT48LC64M8A2 HYB 39S512800AT-7.5	512 Mbit	13x11x4	8	4	256	X	X	X	X	X	X	X	X	X	Not Supported. There can’t be 256Mbytes on a single CS
MT48LC32M16A2 HYB 39S512160AT-7.5		13x10x4	16	2	128	CS	Col 9	Col 8	Row 12	Row 11-0	Bank 1-0	Col 7-0	‘01’	‘11’	Max available memory (256 Mbytes) is reached when using both CS

Appendix C - SDRAM DDR Configuration Examples

The following table reports just some possible architectures supported by the MGT5100 Memory Controller. It must be considered as an example only.

DDR Devices	Device Architecture	Row x Column x Banks	Data width per single chip (in bits)	Min. number of devices needed per CS	Min. total available memory per CS (Mbytes)	Physical Address Multiplexing (on the internal XLB Bus)							Control Register bits 6:7	Xlb_Sel 6:7	Note
						3	4	5	6	7-18	19-20	21-28			
MT46V2M32	64 Mbit	11x8x4	32	1	8	X	X	X	CS	Row 10-0	Bank 1-0	Col 7-0	‘00’	‘00’	Row 11 is not driven!!
MT46V16M8 IBMN612804GT3B	128 Mbit	12x10x4	8	4	64	X	CS	Col 9	Col 8	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘10’	Not recommended to use with both CS
MT46V8M16		12x9x4	16	2	32	X	X	CS	Col 8	Row 11-0	Bank 1-0	Col 7-0	‘00’	‘01’	64 Mbytes can be mounted using both CS
MT46V32M8 HYB25D256800AT	256 Mbit	13x10x4	8	4	128	X	CS	Col 8	Row 12	Row 11-0	Bank 1-0	Col 7-0	‘01’	‘11’	Not recommended to use with both CS
MT46V16M16		13x9x4	16	2	64	X	CS	Col 8	Row 12	Row 11-0	Bank 1-0	Col 7-0	‘01’	‘10’	128 Mbytes possible when both CS are used
MT46V64M8	512 Mbit	13x11x4	8	4	256	X	X	X	X	X	X	X	X	X	Not Supported. There can’t be 256Mbytes on a single CS
MT46V32M16		13x10x4	16	2	128	CS	Col 9	Col 8	Row 12	Row 11-0	Bank 1-0	Col 7-0	‘01’	‘11’	Max available memory (256 Mbytes) is reached when using both CS

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center, 3-20-
1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://www.motorola.com/semiconductors

Information in this document is provided solely to enable system and software implementers to use Motorola 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.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 which may be provided in Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part.

MOTOROLA