LH79520 System-on-Chip SDRAM Connection and Usage

PURPOSE

The purpose of this Application Note is to discuss the LH79520 Core, the LH79520 internal and external bus systems, the LH79520 Synchronous Dynamic RAM (SDRAM) Controller, how these interact in two specific realizations of an LH79520-SDRAM device interconnection, and how the user can configure and use such realizations. This discussion, considered in conjunction with the NXP LH79520 User's Guide, is intended to assist a user in understanding the LH79520 SDRAM controller system sufficiently to design and use their own specific realization of the LH79520 interconnected to their specific SDRAM devices.

SCOPE

This Application Note gives an overview and provides details and examples of the use of the LH79520 SDRAM Controller (SDRC) and External Bus Interface (EBI) for a specific SDRAM device, the Micron Technology, Inc. MT48LC8M16A2.

An overview is presented first, followed by a theory of Operation divided into the following sections:

- A block diagram of the LH79520-SDRAM interface
- General characteristics of SDRAM devices
- The Micron MT48LC8M16A2 SDRAM device
- The LH79520 ARM720T Core
- LH79520 Advanced High-performance Bus (AHB)
- External buses:
  - The LH79520 External Bus Interface
  - The LH79520 SDRAM Controller (SDRC)

Two examples are presented and discussed throughout the document, a realization using a 32-bit bus with two MT48LC8M16A2 devices, and a realization using a 16-bit bus with one MT48LC8M16A2 device.

OVERVIEW

SDRAM is similar to older, slower Asynchronous Dynamic RAM (DRAM). SDRAM uses a Row-Column addressing scheme similar to DRAM where the logical address of a memory location is parsed into a row part and a column part on multiplexed address lines. SDRAM and DRAM each have data lines which are multiplexed or non-multiplexed depending on the bus width.

DRAM and SDRAM differ in the number and function of their control lines. DRAM works Asynchronously, which means the edge transitions of the control signals register operations the device or controller is to perform, and the sequencing of those transitions identify the operations to be performed.

On the other hand, SDRAM operates Synchronously. This means that all operations are registered on an edge of the memory bus clock SDCLK, specifically the rising edge. The complete waveform for SDCLK is a 50% duty cycle square wave. The term ‘SDCLK’ in this document will be used to refer to the rising edge part of the SDCLK waveform unless specifically stated otherwise or the context clearly implies an alternative meaning.

For a ‘single-beat’ Read or Write operation, DRAM and SDRAM with similar ‘speed ratings’ have similar access times. However, for burst operations, the SDRAM is usually faster. Note that ‘speed ratings’ of SDRAM and DRAM cannot be compared directly unless the rating is specified in bus clock frequency. Typical manufacturer's values are derived from different criteria for SDRAM and DRAM when specifying the speed rating in nanoseconds.

When ‘bursting’, a DRAM may require ‘5-2-2-2’ or 11 bus clocks for a burst-of-four 32-bit words access while an SDRAM running on the same bus clock may only require ‘5-1-1-1’ or 8 bus clocks for the burst-of-four access.

The operating configuration of Asynchronous DRAM is fixed and not programmable at run-time. However, the operating configuration of Synchronous DRAM is programmable within limits specified by each manufacturer.

Using SDRAM with the LH79520 can be considered in three phases:

1. Hardware Implementation: Selecting the SDRAM device to be used, and designing the hardware interconnection of the LH79520 and the selected SDRAM, the external bus structure.

   Major considerations during the hardware implementation phase include the number of SDRAM devices, the specifications of the SDRAM devices, the SDRC parameters available and their range, the external bus width, and the bus clock frequency that will be used.

2. SDRC Initialization: Programming the SDRAM controller.

   SDRC initialization is performed at run-time with the object of initializing the SDRC to establish its desired operating configuration. Major considerations during the SDRC initialization phase include the external bus structure, bus clocking speed, the specifications of the actual SDRAM devices implemented during the hardware implementation phase, the SDRAM device parameters available and their range, the SDRC parameters available and their range, establishing the actual SDRC operating configuration, and power modes.

3. SDRAM Initialization: Programming the SDRAM device.

   SDRAM initialization is performed at run-time with the object of initializing the SDRAM device itself to establish its desired operating configuration. Major considerations during the SDRAM initialization phase include the external bus structure, bus clocking speed, the actual SDRC configuration, the SDRAM device parameters and their range, and establishing the actual SDRAM configuration.

While the various elements and phases of the LH79520-SDRAM realization may be discussed separately, it should be remembered that they must work together during initialization and at run-time. The separate stages involved in the SDRC and SDRAM initialization phases are usually interleaved in actual practice.

LH79520 — SDRAM Interface Overview

A Block Diagram showing the relationship of the SDRAM Controller and EBI to SDRAM and other elements of the LH79520 is shown in Figure 1.

CONVENTIONS

See the Glossary Section for Signal Name Conventions.

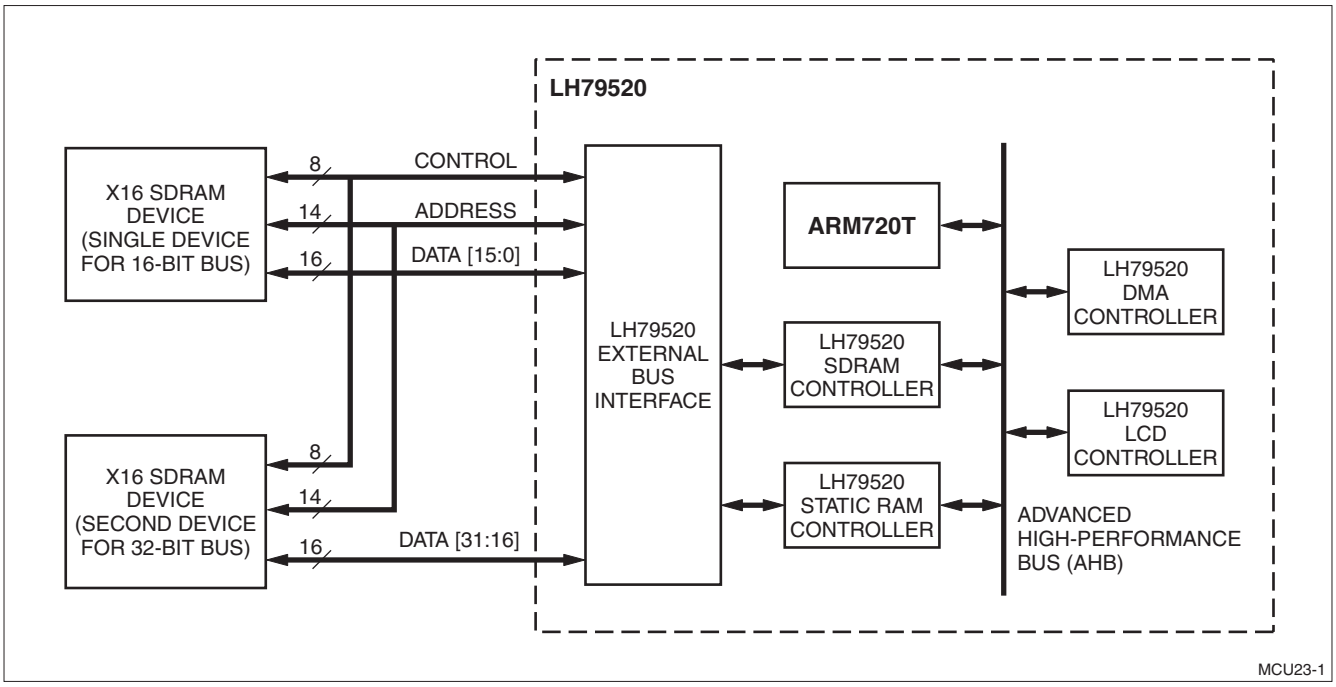

Figure 1. LH79520 ARM720T Core, SDRC, and EBI

Theory of Operation

Power connections are assumed throughout this document.

The interconnection of an LH79520 and two Micron MT48LC8M16A2 devices to accommodate a 32-bit data bus is shown in Figure 2.

Since the LH79520 SDRAM Controller is capable of handling 32-bit Reads, Writes, and bursts to a 16-bit bus, a single Micron MT48LC8M16A2 device can be used to accommodate a 16-bit data bus. The interconnection for this hardware configuration is shown in Figure 3.

Internally, the LH79520 functions as a 32-bit device. Therefore the LH79520 SDRAM Controller must control the 16-bit bus and single SDRAM in such a way that words, halfwords, and bytes are written to the appropriate locations in the ×16 device.

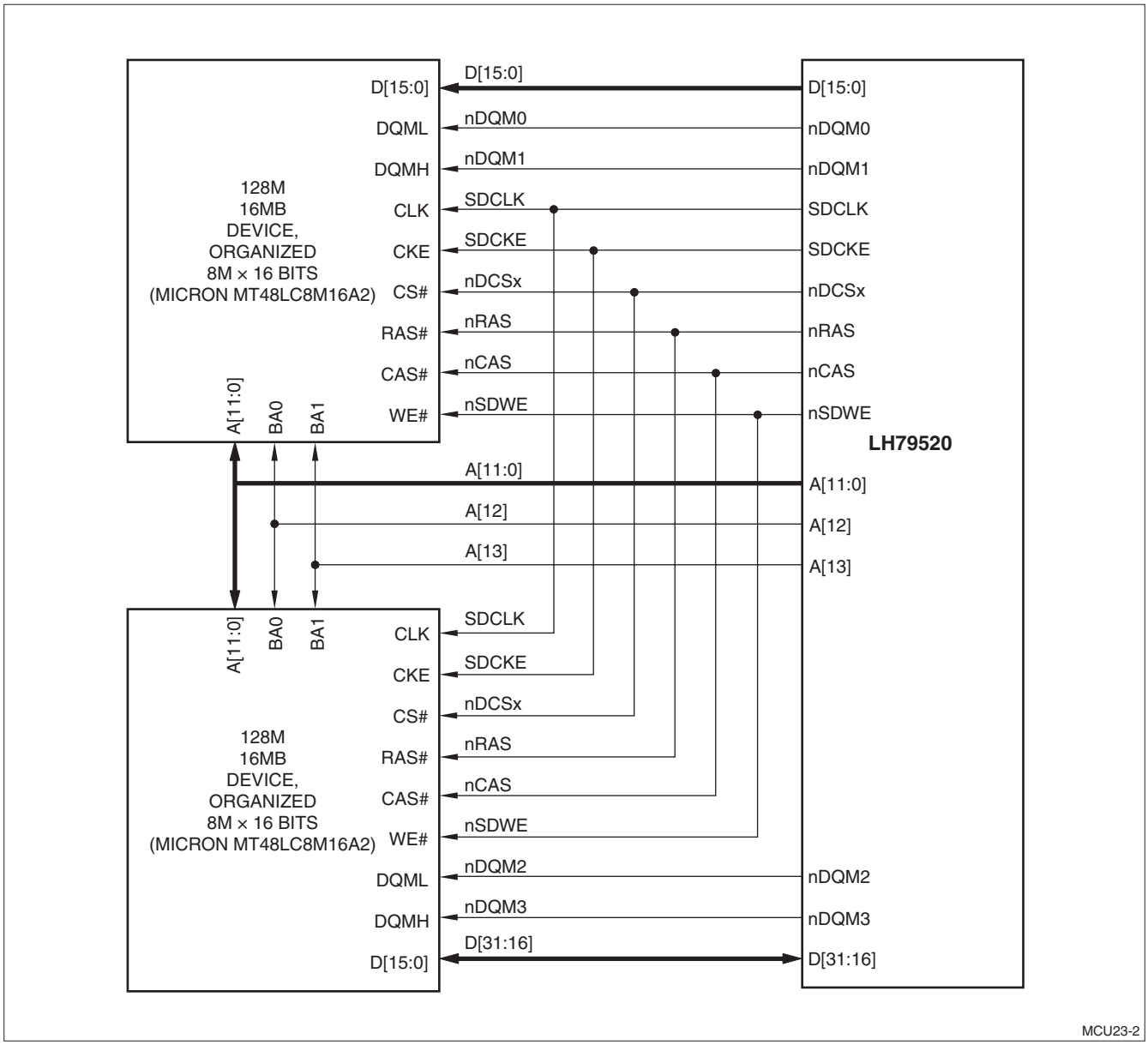

Figure 2. 32-bit Data Bus SDRAM Interconnection

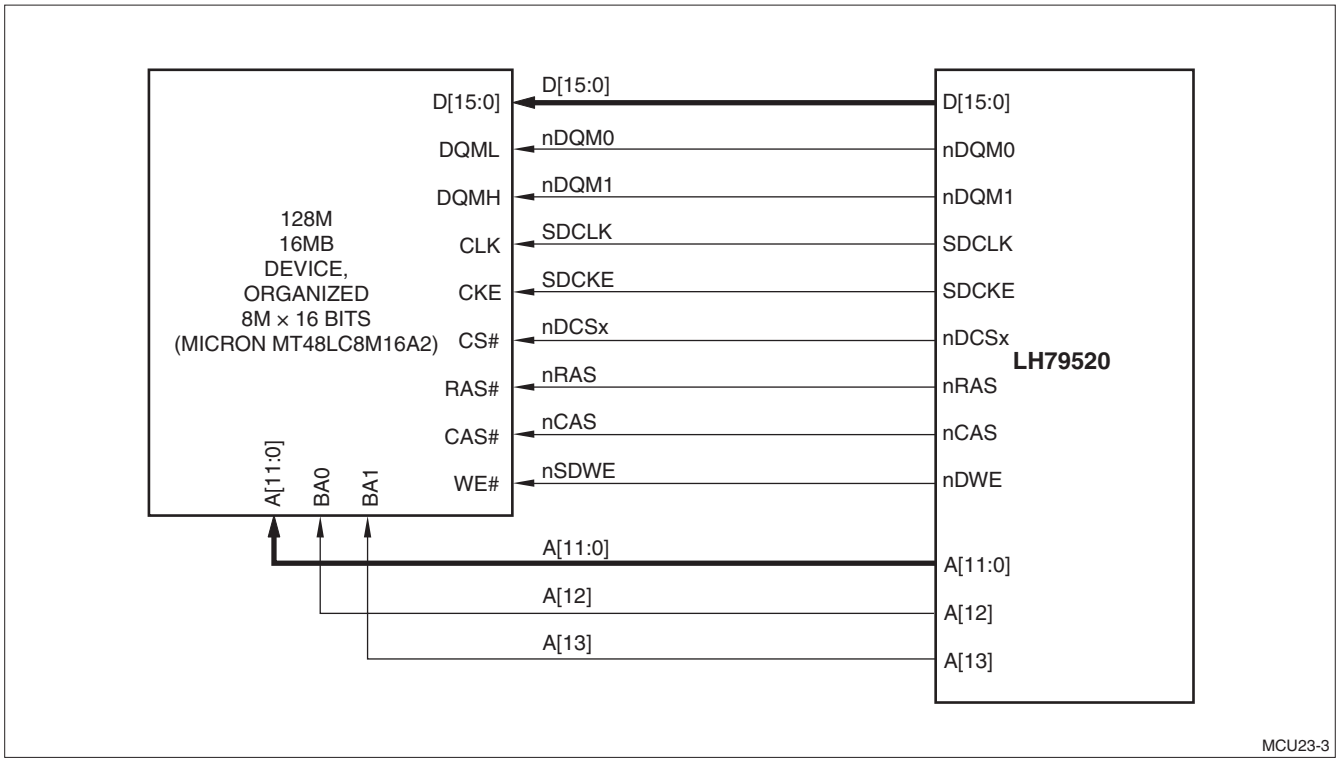

Figure 3. LH79520 16-bit Data Bus SDRAM Interconnection

LH79520 SDRAM Controller and SDRC-SDRAM Interface

In general, the SDRAM Controller controls SDRAM by issuing commands to the SDRAM. The SDRC implements these commands through hardware by setting the state of several signal lines and holding the desired state across a rising edge of SDCLK. The SDRAM registers these commands on the rising edge of SDCLK. Once registered, the SDRAM responds to these commands.

The SDRAM controller can issue several commands. The example SDRAM can respond to these commands. These commands are described in later in this document. Unfortunately, although there is some commonality among vendors, not all vendors observe the same command naming conventions. Furthermore, there is not necessarily a clear one-to-one mapping between the commands generated by the SDRC and the commands understood by the SDRAM of interest. This lack of standardization may generate some confusion when designing and initializing the SDRAM interface.

COMMANDS

The following general list of commands are supported by the SDRAM Controller and understood by the example SDRAM. Some commands are automatically generated by the SDRC during normal operation independent of program control, while others are generated by the SDRC only under program control.

- Command Inhibit
- No Operation
- Active
- Read
- Write
- Burst Terminate
- Precharge, including Auto Precharge
- Precharge All
- Auto Refresh
- Self Refresh
- Load Mode Register
- Normal Operation
- Power-down Mode
- Clock Suspend Mode

Command Signal Lines

The following signal lines are significant as control lines for commands.

- nCS
- nRAS
- nCAS
- nWE
- DQM (DQML, DQMH)
- CKE
- A10

Normal Operation

The fundamental operations performed during normal operation are Read and Write.

During a Read operation, the SDRAM is commanded to enable data onto the data lines which is then latched by the SDRC. During a Write operation, the SDRC enables data onto the data lines and commands the SDRAM to latch the data into SDRAM storage.

There are several variants of Read and Write operations. Among the variants, Auto Precharge can be enabled or disabled, a Read or Write may be Single or Burst, an access can be Single or Multiple Byte. All of these variants are selected by the SDRAM controller using the control lines.

AUTO REFRESH, SELF REFRESH, AUTO PRECHARGE, PRECHARGE

Auto Refresh

Auto Refresh is used during normal operation and ensures that every address is refreshed at least once every 64 ms. This is done to ensure that the data in the SDRAM does not decay. For example, the example SDRAM requires 4,096 Auto Refresh commands every 64 ms, or one Refresh every 15.625 μs. These refresh commands are issued automatically once the SDRC is properly configured.

Self Refresh

Self Refresh is a feature of some SDRAMs that is used during low power modes to minimize power consumption. It allows data to be retained in the SDRAM even if the rest of the system is powered down. After the SDRAM is commanded to enter Self Refresh mode, no external clocking need be applied because the SDRAM provides its own internal refresh clocking. Normal Read and Write operations are not possible when the SDRAM is in Self Refresh mode. When the SDRAM is brought out of Self Refresh mode, Auto Refresh commands must once again be applied to ensure that data are preserved.

Auto Precharge and Precharge

Auto Precharge is the ‘automatic’ version of the Precharge command, which directs the SDRAM to deactivate the open row in a particular bank. Once a Precharge command is issued, the bank is in an idle state and must be reactivated prior to issuing Read or Write commands to that bank. ‘Auto Precharge’ ensures that the Precharge command is issued upon the completion of a Read or Write burst.

Note that Auto Refresh and Auto Precharge are not the same thing.

SPECIFIC SDRAM — MICRON MT48LC8M16A2

The Micron MT48LC8M16A2, the example SDRAM, is a 128Mb Synchronous Dynamic RAM device. It is internally configured as a quad-bank DRAM with a synchronous interface, with all signals registered on the rising edge of SDCLK. Each internal bank contains 32Mb. The 32Mb in each bank is organized as 4,096 rows by 512 columns by 16 bits. Each bank has 2Mb distinct addresses for 8Mb distinct addresses per device. Four two-byte wide banks yields 16MB of storage in each device.

The row address width is 12 bits **(A11-A0)** and the column address width is 9 bits **(A8-A0)**. The four individual banks are selected with signal lines **(BA0-BA1)**.

Programming the Mode Register of the Micron MT48LC8M16A2 allows the user to establish Burst Length, burst type, CAS latency, and write burst mode. The values chosen for these parameters must be consistent with the hardware implementation, the bus clock frequency, and the values chosen for the SDRC configuration parameters.

The Micron MT48LC8M16A2 can run at a bus clock speed of 100 MHz. This easily accommodates the LH79520 maximum bus clock speed of 51.6096 MHz and allows setting the CAS latency = 2 in both the SDRC configuration registers and the SDRAM mode registers.

LH79520 ARM720T Core

This discussion is limited to those aspects of the LH79520 ARM720T Core that are necessary for accessing external SDRAM memory.

The LH79520 ARM720T Core, the LH79520 DMA Controller, and the Color LCD Controller access SDRAM through their respective AHB bus interfaces. See Figure 1 for a block diagram showing the Core-SDRAM interface. Only Core accesses will be discussed because these are the most comprehensive. Core accesses include SDRC and SDRAM initialization, configuration, and mode control. The DMA and LCD accesses follow an analogous route but are more limited in their interactions with SDRAM with the DMA Controller limited to Read/Write operations and the Color LCD Controller limited to Read operations.

The Core initiates an SDRAM Read or Write operation by executing an instruction involving a memory access in the SDRAM physical address range. The instruction request is communicated via hardware and internal logic out the Core AHB interface across the AHB bus to the SDRAM Controller. The SDRAM Controller operates on the request with further hardware and logic, translating the request through the SDRAM Controller to the External Bus Interface (EBI), and thence to the SDRAM. The internal operation of the Core-AHB-SDRC interface just described is transparent to the LH79520 user. Only signals on the External Bus can be observed by the user.

The Core initiates other operations, e.g., configuration and initialization, by writing explicit values to the SDRC control registers at explicitly defined addresses.

DATA TYPES AND ALIGNMENT

The LH79520 supports the following data types:

- Byte (8-bit)
- Halfword (16-bit)
- Word (32-bit)

Addresses of these data types must be aligned as follows:

- Word entities must be on a 4-byte boundary
- Halfword entities must be on a 2-byte boundary
- Byte entities may be aligned on any byte boundary

LH79520 Advanced High-performance Bus (AHB)

This discussion is limited to those aspects of the LH79520 AHB that are significant for accessing external SDRAM memory.

Control and arbitration of the AHB are transparent to the user.

The width of the data bus on the AHB is 32 bits, and all 32 bits are used for a word read/write.

For Half-Word writes, LH79520 internal logic enables the least significant 16 bits of the register onto the least significant bits of the AHB, and replicates these bits onto the upper 16 bits of the AHB. For writes, these data will also appear on the external bus when gated by the SDRC. The logic of the SDRC is responsible for interpreting the address to determine the commands necessary to write the half-word on the appropriate half of the AHB and external bus to its identified half-word storage location in SDRAM. The lower half-word (bits 0-15) of the 32-bit data word is written to an ‘even’ half-word address in the ×16 device, and the upper half-word (bits 16-31) is written to an ‘odd’ half-word boundary in the ×16 device.

For half-word reads, the addressed half-word does *not* appear on both halves of the external bus. The SDRC accepts the addressed half-word data and enables it onto both halves of the AHB, and the LH79520 internal logic gates the lower half of the AHB into the lower half of the 32-bit general purpose register designated in the instruction.

A similar operation applies to byte writes. For byte writes, the least significant 8 bits of the register are replicated on all four bytes of the AHB 32-bit data bus, and these data appear on the external bus. The SDRC is responsible for generating the appropriate commands to store the byte into the designated address.

Byte Reads and Writes are similarly controlled so that the byte 0, 1, 2, and 3 in a word is written and read to the appropriate byte address in the device.

LH79520 External Bus Interface (EBI)

The External Bus signal lines may be divided into three separate groups: Address, Data, and Control.

Deciding the number of physical address lines to be used for the Address group depends on the type of SDRAM device being implemented. For the example device, the Micron MT48LC8M16A2, the number of address lines used is 14 to accommodate the 12 row address lines and the 2 bank select lines.

The Data group may comprise either 32 data bits or 16 data bits. The SDRC logic is responsible for generating the appropriate commands to write to and read from the addressed storage locations on a bus of either size. Since word width write (32 bits) can be made to an SDRAM device on a 16-bit bus, the sequence of SDRC generated commands to write a word to a SDRAM devices on a 16-bit bus is significantly different than it is for a 32-bit bus. For a word write on a 16-bit bus, the SDRC must 'time-division multiplex' the external data bus by splitting the word into two half-words and writing each word sequentially to the device(s). The user must configure the SDRC to establish the external bus width in use.

The Control group comprises the lines previously identified as control lines for commands. The state of these control lines at an SDCLK establish and control the behavior of the SDRAM device:

- nCS
  - **nCS** is Chip Select. If it is asserted, the device is 'selected'. If it is de-asserted, the device is 'deselected'.
- nRAS
  - **nRAS** is Row Address Select. If it is asserted concurrently with nCS, the device is made 'Active', and the Row Address is latched by SDCLK.
- nCAS
  - **nCAS** is Column Address Select. If it is asserted concurrently with nCS, the Column Address is latched by SDCLK , and a Read or Write operation starts.
- nWE
  - **nWE** is Write Enable. If it is asserted concurrently with nCS and nCAS, a Write operation starts. If it is de-asserted concurrently with nCS and nCAS, a Read operation starts.
- DQM (DQML, DQMH)
  - **DQML** and **DQMH** control the width of the Read/Write operation. For the example SDRAM, **DQML** corresponds to the LSB of the two-byte wide device, and **DQMH** corresponds to the MSB of the two-byte wide device. They can be thought of as 'byte selects'. Whether they will be asserted depends on the width of the Read/Write operation. For a word-width operation, **DQMH** and **DQML** are both asserted on the each device. For a half-word-width operation, **DQMH** and DQML are both asserted on the appropriate device. For a byte-width operation, the appropriate DQMx is asserted on the appropriate device. (On ×8 SDRAM devices, the signal is simply *DQM* on the device, and **DQML** and **DQMH** from the LH79520 are each connected to the appropriate device *DQM*.)
  - When **DQML** and **DQMH** will be asserted depends on the type of the operation, that is, whether the Operation Is A Read Or A Write. If The Operation Is A Read, **DQML** and **DQMH** are asserted concurrently with the Read command. If the operation is a Write, **DQML** and **DQMH** are asserted two SDCLKs after the Write command.
- CKE
  - **CKE** is the Clock Enable signal. During normal operation, the **CKE** signal is continuously asserted. De-asserting **CKE** concurrently with the Auto Refresh command will cause the SDRAM to enter Self Refresh Mode.
- A10
  - **A10** is an address line during the Active command. **A10** is used by the SDRC during the Precharge command and during the Read/Write command. If **A10** is asserted concurrently with the Read/Write command, it signals an Auto Precharge command is to be applied automatically at the end of the burst. If it is asserted during a Precharge command, it the state of the Bank Select pins determine which bank is to be precharged. If **A10** is de-asserted during a Precharge command, the bank select pins are ignored and all banks are precharged.

SIGNAL VISIBILITY

All signals on the External Bus may be viewed with a logic analyzer which displays the time-based behavior of the signals.

Basic Memory Access Functionality

SDRAM differs from earlier types of DRAM in that it is made to run synchronously to the bus clock SDCLK, resulting in an improvement in access speed.

Asynchronous DRAM and Synchronous DRAM command mechanisms both use commands on signal lines. However, Asynchronous DRAM commands are the signal line transitions themselves considered independently of any external clocking mechanism. The order in which transitions occur, and the direction of the transitions are the means used to command Asynchronous DRAM.

The Synchronous DRAM command mechanism considers the state of the signal lines at the rising edge of SDCLK, registering commands, addresses, and data only on this rising edge. This decreases the total time necessary to register data on the second and subsequent element in a burst.

Bursting

SDRAM's speed advantage is best realized when it is operated in a burst mode. 'Bursting' an SDRAM facilitates a registration of data on every SDCLK after the first access. The nominal time difference is that DRAM typically takes at least the equivalent of two SDCLK clock periods for the second through N elements, while SDRAM registers data on every SDCLK after the first.

Single-beat Reads and Writes

For the example SDRAM, the sequence of operation for a typical 'single-beat' (single datum) Read or Write proceeds as follows:

The SDRC enables the row and bank part of the logical address onto the address lines, usually coincident with the chip select signal (nCSx) and the row address (nRAS) signal. This is done by the SDRC prior to the next rising edge of SDCLK early enough to allow the address lines time to settle before the SDCLK rising edge ('SDCLK') registers the device active and simultaneously latches the row and bank address into the SDRAM device. The RAS-to-CAS delay is commenced, and then nCSx and nRAS are de-asserted, nominally half a bus clock cycle after registration.

After the RAS-to-CAS-delay, the SDRC again asserts nCSx, only this time the column address (nCAS) signal and the column and bank part of the logical address is concurrently asserted. If the action is a Read from the SDRAM device, the nWE signal remains de-asserted; if the action is a Write to the SDRAM device, the nWE signal is asserted concurrently with nCSx and nCAS. When SDCLK registers the column address, the CAS latency period commences and then the SDRC de-asserts nCSx, nRAS, nCAS, and nWE (if applicable). After the CAS latency period has elapsed, data on the data lines is valid, and is latched into the LH79520 for a Read, or into the SDRAM device for a Write.

Both the RAS-to-CAS latency and the CAS latency values are established by the programmer during SDRC and SDRAM Initialization. Determining the optimal values for the latencies is discussed infra.

Burst Reads and Writes

For 'burst' Reads and Writes, data are registered into the SDRAM or LH79520 on each succeeding SDCLK after the first registration during the CAS cycle. The column address is incremented after each registration so the data are stored sequentially into the column addresses of the selected row and bank. The number of Reads or Writes in a burst are established by the programmer during SDRC and SDRAM Initialization.

Initializing the SDRAM Controller and the SDRAM

The LH79520 SDRAM Controller is configured by entering the proper parameter values into its four configuration registers in the proper sequence during initialization. This is done by writing the configuration data directly to their memory-mapped location under program control. The four registers are:

- SDRCConfig0
- SDRCConfig1
- SDRCRefTimer
- SDRCWBTimeout

SDRCConfig1 is used as a command register during the initialization sequence, so its value changes during initialization. Although a final configuration value is written to SDRCConfig1 prior to completion of initialization, this register is also used under program control to enable and disable the read and write buffers. SDRCConfig0, SDRCRefTimer, and SDRCWBTimeout are set during initialization to values that usually do not change until an LH79520 Reset.

The SDRAM devices are configured by programming their onboard Mode Registers. This is done by using bits [1:0] of SDRCConfig1 to cause the SDRC to generate the Load Mode Register command to the SDRAM, then performing a read operation to the SDRAM device. The value to be programmed is coded into the address signals during the read. The coded address is formed by combining the base address of the applicable Chip Select with the bit pattern with the values to be loaded into corresponding bits of the Mode Register in a way described infra.

DETERMINING PARAMETER VALUES

For initialization, the user must establish each of the following:

- Decide whether or not to enable Auto Pre-charge.
- Determine the permitted RAS to CAS latency.
- Determine the permitted CAS latency.
- Define the external bus width.
- Determine the necessary burst length.
- Determine the necessary burst type: sequential or interleaved.
- Identify the specific type of SDRAM attached to each of the Chip Select signals, that is, determine the number of internal banks in the respective SDRAM devices, and determine the address multiplexing scheme they use.
- Determine the Write Burst Mode: either programmed burst length or single location access should be selected in conjunction with enabling or disabling the SDRC Write Buffers.
- Decide how to set two closely coupled parameters (Clock Enable and Clock Control) that control the SDRAM clock.

Manifest constants to facilitate programming the SDRC configuration registers are provided in the header file LH79502_sdrc.h, available as a part of a zipfile on the NXP Semiconductors website through the ARM and BlueStreak library, at: http://www.nxp.com.

Auto Pre-charge

During normal operation, it is desirable to use Auto Pre-charge, so Auto Pre-charge should usually be enabled. In general, manual pre-charge is used for page size bursting transfers under program control. Discussion of when and how to use manual pre-charge is beyond the scope of this document. Specifically, bit [24] of SDRCConfig0 should be set.

RAS to CAS latency

The RAS to CAS latency is the number of SDCLKs between the ACTIVE command and the Read or Write command.

The LH79520 only allows RAS to CAS latency (RCD) values of 2 or 3.

Of these two values, the RAS to CAS latency is selected by inspecting the SDRAM AC operating conditions in the manufacturer's data sheet. For our example SDRAM, Micron uses the ‘Command’ nomenclature to identify its requirements, not ‘RAS-to-CAS’. The value we need to consider is the MIN value for ‘Active to Read or Write delay’ for the specific device used on the EVB, the MT48LC8M16A2-8E. The Micron data sheet shows this value, tRCD, to be 20 ns.

The RAS-to-CAS setting in the LH79520 SDRCConfig0 register is specified in integral counts of SDCLK. The required count value is the least integral number of SDCLKs times the bus clock period that results in a value greater than or equal to the minimum Active to Read or Active to Write delay.

That is, the following inequality must be satisfied:

$$Period_{(SDCLK)} \times RCD \geq tRCD$$

Assuming the run-time bus clock is 51.6096 MHz, the SDCLK period is then ~19.3 ns. An RCD value of 1, even if we could use that value, would not satisfy the inequality. The next integral value is 2 which does satisfy the inequality, and thus for our example SDRAM, the RAS to CAS delay may be set to a minimum of 2 SDCLKs. An RCD setting of 3 is acceptable, but there will be an unnecessary degradation of access performance if 3 is used with the example device.

Note that if the RAS-to-CAS setting in the LH79520 SDRCConfig0 register is 2, and the operation is a Read, the CAS will be registered on the second SDCLK after the Active command, i.e., when RAS is registered. If the operation is a Write, CAS will not be registered until the third SDCLK (CAS latency setting + 1).

CAS Latency

The CAS latency identifies the number of SDCLKs between registration of the Read or Write command and the registration of valid data. A CAS latency of 2 means that data are registered on the second SDCLK after the Read or Write SDCLK.

Understanding CAS latency is somewhat confusing because the actual latency for Writes is different than the actual latency for Reads for a specific configured value of CAS latency. If CAS latency is set to 2, then the actual CAS latency for a Read is 2 SDCLKs, while the actual CAS latency for a Write is 3 SDCLKS.

The CAS latency is set in bits [21:20] of SDRCConfig0 as well as in bits [6:4] of the SDRAM Device Mode Register. The latencies identified in both devices must be equal.

The SDRC can accommodate CAS latencies of 1, 2, or 3. For our example SDRAM device, only CAS latencies or 2 or 3 SDCLKs are applicable and that for SDCLK frequencies below 100 MHz, a CAS latency of 2 is acceptable. Since our example SDCLK is ~51.6 MHz, a CAS latency of 2 should be used for optimal performance. There will be an unnecessary degradation of access performance is 3 is used.

External Bus Width

External bus width is established by the physical hardware. The LH79520 EVB permits usage of a single example SDRAM device on a 16-bit bus or two example SDRAM devices on a 32-bit bus. This permits meaningful performance measurements for the target bus, which will be one or the other.

SDRCConfig0 bit [19] establishes the external bus width from the LH79520 viewpoint. SDRCConfig0 [19] set to 1 identifies a 16-bit bus; set to 0 (default), it identifies a 32-bit bus. The value selected also determines which Burst Length must be programmed into the SDRAM device.

Burst Length

The burst length programmed into the SDRAM device must be coordinated with the external bus width selected. If the bus width selected is 16 bits, the burst length programmed into the SDRAM must be 8. If the bus width selected is 32 bits, the burst length programmed must be 4.

Burst Type

The burst type is nomenclature associated with the SDRAM device. The ‘burst type’ is fixed by the LH79520 SDRC and must be programmed into the Mode Register of the SDRAM device. The LH79520 reads and writes bursts using sequential address locations, and thus sequential burst type should be selected. Interleaved Burst Type is reserved for Full Page accesses under program control.

SDRAM Type Specification

The SDRAM Type is specified separately for each of the four available chip select lines. See the LH79520 User's Guide for the specific bits in the SDRCConfig0 Register that are applicable.

For our example SDRAM device, the manufacturer's data sheet identifies the MT48LC8M16A2 as:

- a four bank device
- an ‘×16’ (16 data bits) device
- a 128Mbit device, and thus not a 256Mbit device.

The configuration bits for CS0 and CS1 should be programmed accordingly. Manifest constants are defined in the file LH79520_sdrc.h to readily facilitate programming the SDRC.

Write Buffers and Write Burst Mode

The LH79520 provides a feature called ‘Merging Write Buffers’. A merging write buffer compacts Writes of all widths into quad-word bursts which efficiently transfer data to SDRAM. See the LH79520 User’s Guide for details.

The write burst mode is determined by the LH79520 SDRC write buffer configuration. If the write buffers are enabled, data are written to all devices by bursts. Disabling the write buffers is done to facilitate programming Synchronous Flash devices, so that each memory location is written individually, thus allowing a program controlled delay between writing to each location. Once the Flash is programmed, the write buffers should be enabled to allow burst writing and the SDRAM devices should be programmed accordingly. The SDRAM device permits writes in either write burst mode, so that if it is necessary to program flash and thus require single location write access, the program with its read/write data locations can be executing in SDRAM concurrently with programming the flash on a different chip select. When the write buffers are enabled, the SDRAM devices Mode Register must be programmed to indicate that Writes are done by programmed burst length. Write buffers enabled and programmed burst length are the normal operation configuration of the example SDRAM.

Clock Enable and Clock Control

Clock Enable and Clock Control are configured using SDRCConfig0 [18:17]. The Clock Enable bit [18] affects the CKE signal. When Clock Enable is cleared, the SDRC de-asserts the CKE signal when all SDRAM devices are idle to conserve power. When Clock Enable is set, the CKE signal is asserted to all SDRAM devices continuously. Clock Control affects the SDCLK signal. If Clock Control is set, SDCLK is inhibited when all devices are idle. If Clock Control is cleared, SDCLK runs continuously.

Note that simultaneously setting Clock Control and Clock Enable is a forbidden condition. That is, configuring SDCLK to stop when idle while configuring CKE to assert continuously should not be done.

Configuring the SDRAM Device Load Mode Register

Using SDRCConfig1 under program control, the SDRC can generate one of three specific commands, or resume normal operation. The commands are:

- Generate a NOP command and wait
- Generate a PRE-CHARGE ALL command and wait
- Generate a LOAD MODE REGISTER command and wait
- Resume normal operation

Once the SDRAM has been given the LOAD MODE REGISTER command using SDRCConfig1, the values on the address lines the next time the device is given the ACTIVE command will be loaded into the Mode Register. A convenient way to do this is to perform a read of the device. This will cause the SDRC to generate an ACTIVE command followed by a Read command. Since the row address component of the physical address is latched during the ACTIVE command, the row component of the address must contain the Mode Register configuration pattern during the read operation.

For the example device, bits [9:2] of the physical memory address provide the column address, bits [11:10] provide the bank select, the DQM lines provide signals that decode address bits [1:0], and the row part of the address comprises bits [23:12]. These values are derived in substantial part from the SDRAM address mapping table in the User's Guide describing a 128Mb, 8Mb × 16 SDRAM device. Thus, the pattern to be written into the Mode Register for our example device must be shifted left 12 bits and OR'd with the Chip Select base address to generate the necessary address signals.

Assume that the external bus width is 32 bits so the burst length will be 4, the burst type is sequential, the CAS latency is 2, the operating mode is Standard Operation, and programmed burst length will be in effect. For the example, the base pattern derived from the Mode Register definition in the SDRAM manufacturer's documentation is therefore:

b000000100010 = 0x22

The devices connected to CS0 are physically mapped to 0x20000000 base address. The devices connected to CS1 are physically mapped to 0x28000000 base address. Thus, the C style pseudo-code for programming the mode register is:

```
Write Generate LOAD MODE REGISTER command to SDRCConfig1
Read location (0x20000000 | (0x22 << 12))
Read location (0x28000000 | (0x22 << 12))
Wait for the SDRAM device to become idle using SDRCConfig1
Write Resume Normal Operation to SDRCConfig1.
```

This pseudo-code fragment provides an illustrative example of how to program the SDRAM device during the initialization sequence. It will be integrated with other configuration code elsewhere in this document. The numerical values will be have to be derived for the user's own specific realization.

Configuring the Refresh Timer Register

As with any DRAM, Synchronous DRAM must be refreshed periodically to maintain its data. The number of SDCLKs between Auto Refreshes is determined by the value in SDRCRefTimer. This value is dependent upon the required SDRAM refresh rate and the SDCLK frequency.

The example device requires 4,096 Auto Refresh signals every 64 milliseconds per manufacturer's specification. Providing a distributed AUTO REFRESH command every 15.625 µs will meet the refresh requirement ($4096 \times (64 \times 10^{-3})$) The value which must be programmed into the SDRCRefTimer register is the number of SDCLK cycles between refresh cycles. Thus, the value depends on the SDCLK frequency according the following formula:

SDRCRefTimer value = (SDCLK cycles per second) × (period between refreshes)

For the example bus clock frequency of 51.6096 MHz:

SDRCRefTimer value = $(51.6096 \times 10^{6}) \times 15.625 \times 10^{-6}$ = 806 = 0x326

Configuring the Write Buffer Timeout Register

If the Merging Write Buffer is enabled, the data which has been written to it should be written to SDRAM without undue delay in order to preserve coherency, if necessary. To accommodate this, the user may program the number of SDCLKs which may elapse without anything being written to the write buffer before the data in the write buffer is flushed to the SDRAM. The SDRCWBTimeout register is set by the programmer to the delay in AHB clocks(SDCLKs) that must occur with no writes before the SDRC's Merging Write Buffer is flushed.

The range of the SDRCWBTimeout register is 0 to 65535. 0 disables the Write Buffer Timeout function. When selecting the value, the user should choose an optimal value that balances the requirement that memory coherency be maintained with the requirement to allow other controllers timely access to and efficiently use the external memory bus.

For the example SDCLK, the maximum value of 65535 yields a time before a write of ~1.2 ms. A value of 128 will ensure a write buffer flush after ~2.5 µs.

Programming Examples

```
static void init_sdram (unsigned int clock_idx)
{
	volatile int tmp;

	/* pseudo-code for Micron MC48LC8M16A2TG-8EL SDRAMs
	 * Issue two NOP commands
	 * Delay at least 100 usecs.
	 * Issue precharge to all banks
	 * Once in idle state, issue two auto refresh cycles to each bank
	 * Program the Mode register when all banks are idle
	 * Wait the "specified time" before initiating subsequent operation
	 * */

	/*            DELAY  to allow power and clocks to stabilize */
	/* load ~100 us value to timer1 */
	mem_timer (200);

	SDRAM->config1 = SDRAM_INIT_NOP;
	SDRAM->config1 = SDRAM_INIT_NOP;

	/*            DELAY  to allow SDRAM clocks to settle */
	/* load ~200 us value to timer1 */
	mem_timer (200);

	/* issue a "Pre-charge All" command */
	SDRAM->config1 = SDRAM_INIT_PALL;

	/* load ~250 us value to timer1 */
	mem_timer (250);

	/* refresh every 16 clock cycles */
	SDRAM->reftimer = SDRAM_SET_REFRESH(16);
	/*            DELAY  for at least two auto refresh cycles */
	/* load ~250 us value to timer1 */
	mem_timer (250);

	/* set the refresh timer */
	switch (clock_idx)
	{
		case RCPC_CLKIDX_78_78:
			SDRAM->reftimer = SDRAM_SET_REFRESH(REFTIMER_78);
			break;

		case RCPC_CLKIDX_78_52:
		case RCPC_CLKIDX_52_52:
		case RCPC_CLKIDX_DEFAULT: /* Assumes 52 MHz default clock */
			SDRAM->reftimer = SDRAM_SET_REFRESH(REFTIMER_52);
			break;
```

```
            case RCPC_CLKIDX_78_39:
            case RCPC_CLKIDX_52_39:
            case RCPC_CLKIDX_39_39:
                  SDRAM->reftimer = SDRAM_SET_REFRESH(REFTIMER_39);
                  break;
            case RCPC_CLKIDX_10_10:
            default:
                  SDRAM->reftimer = SDRAM_SET_REFRESH(REFTIMER_10);
                  break;
      }

      /* load ~250 us value to timer1 */
      mem_timer (250);

      /* Program the SDRAM internal mode registers on bank nSDCS0-1
       * Burst Length - 4 (A2:A0 = 0b010)
       * Burst Type - Sequential (A3 = 0)
       * CAS Latency - 2 (A6:A4 = 0x010)
       * Operating Mode - Standard (A8:A7 = 0)
       * Write Burst Mode - Programmed Burst Length (A9 = 0)
       */

      /* Select mode register update mode */
      SDRAM->config1 = SDRAM_INIT_MODE;

      tmp = *((int *)(SDRAM_BANK0_BASE | (0x22 << 12 )));
      tmp = *((int *)(SDRAM_BANK1_BASE | (0x22 << 12 )));

      /* Wait until idle */
      while (SDRAM->config1 & SDRAM_STATUS)
            ;
      /* Configure SDRAM Controller Configuration Register 0 */
      SDRAM->config0 =SDRAM_A_AUTO |
      SDRAM_C_CONT |
      SDRAM_CLAT2 |
      SDRAM_CS0_4BANK |
      SDRAM_CS1_4BANK |
      SDRAM_CS0_X32 |
      SDRAM_CS1_X32 |
      SDRAM_RCLAT2 |
      SDRAM_EXTBUS32;

      /* Wait until idle */
      while (SDRAM->config1 & SDRAM_STATUS)
            ;
      /* select normal operating mode */
      SDRAM->config1 = SDRAM_INIT_NORMAL;

      /* Wait until idle */
      while (SDRAM->config1 & SDRAM_STATUS)
            ;
      /* Wait ~100 us before using */
      mem_timer (128);
}
```

GLOSSARY

Signal Name Conventions

Physical signal names on the physical external memory bus are represented in a bold emphasis vertical font, e.g., A[5], nDCS0, D[3].

Physical signal names on the Micron SDRAM device are represented in a regular emphasis vertical font, e.g., A0, nCS0, D3.

Logical signal names internal to the LH79520 are represented in a regular emphasis italic font, e.g., A[5], nDCS0, D[3].

If the LH79520 logical signal names refer to the signals on the AHB, the name will be preceded by a lower-case ‘h’.

Signal names preceded by a lower-case ‘n’ refer to signals that are asserted active LOW.

Definitions

nRAS, and nCAS

Row address signal and Column address signal, respectively. Historically, from the usage associated with DRAM, ‘RAS’ is an acronym for ‘Row Address Strobe’, and ‘CAS’ is an acronym for ‘Column Address Strobe’. This nomenclature leads to confusion when it is applied to SDRAM because the ‘strobing’ action for SDRAM is accomplished by the rising edge transition of SDCLK, not by nRAS and nCAS. For SDRAM, nRAS and nCAS signal States, not signal Transitions.

RAS-to-CAS Latency

The number of SDCLKs from registering the row address to registering the column address. Minimum = 1 SDCLK. Other possible values: 2, 3.

CAS Latency

The number of SDCLKs from registering the column address until data are registered. Minimum = 1 SDCLK. Other possible values: 2, 3. Note that after the initial CAS latency to data valid, each subsequent data-out element will be valid on the next rising clock edge.

Rising Edge, Positive Edge, Leading Edge

Equivalent terms for a LOW-to-HIGH transition of a signal.

SDCLK

SDRAM Clock signal. In this document, ‘SDCLK’ used alone means ‘the rising edge of SDCLK’ unless the context clearly establishes otherwise.

SDRC

SDRAM Controller

Program Control

SDRC commands to SDRAM which occur under program control are those issued as a result of explicit instructions executed by the ARM720T Core. That is, the Core explicitly requests that the SDRC issue a specific command to SDRAM. An example is ‘Precharge All’. Commands ‘not under program control’ are the sequence of commands to SDRAM issued by the SDRC during normal operation. These are initiated by the SDRC internal state machine as an indirect result of a Read or Write request from the ARM720T Core, but are otherwise independent of the Core.

ADDITIONAL INFORMATION

See the LH79520 User's Manual and the SDRAM vendor Data Sheet for additional details on the implementation, operation, and use of the LH79520 MMU and Cache.

ANNEX A: Disclaimers (11)

1. t001dis100.fm: General (DS, AN, UM, errata)

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

2. t001dis101.fm: Right to make changes (DS, AN, UM, errata)

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

3. t001dis102.fm: Suitability for use (DS, AN, UM, errata)

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

4. t001dis103.fm: Applications (DS, AN, UM, errata)

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

5. t001dis104.fm: Limiting values (DS)

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) may cause permanent damage to the device. Limiting values are stress ratings only and operation of the device at these or any other conditions above those given in the Characteristics sections of this document is not implied. Exposure to limiting values for extended periods may affect device reliability.

6. t001dis105.fm: Terms and conditions of sale (DS)

Terms and conditions of sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by NXP Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

7. t001dis106.fm: No offer to sell or license (DS)

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

8. t001dis107.fm: Hazardous voltage (DS, AN, UM, errata; if applicable)

Hazardous voltage — Although basic supply voltages of the product may be much lower, circuit voltages up to 60 V may appear when operating this product, depending on settings and application. Customers incorporating or otherwise using these products in applications where such high voltages may appear during operation, assembly, test etc. of such application, do so at their own risk. Customers agree to fully indemnify NXP Semiconductors for any damages resulting from or in connection with such high voltages. Furthermore, customers are drawn to safety standards (IEC 950, EN 60 950, CENELEC, ISO, etc.) and other (legal) requirements applying to such high voltages.

9. t001dis108.2.fm: Bare die (DS; if applicable)

Bare die (if applicable) — Products indicated as Bare Die are subject to separate specifications and are not tested in accordance with standard testing procedures. Product warranties and guarantees as stated in this document are not applicable to Bare Die Products unless such warranties and guarantees are explicitly stated in a valid separate agreement entered into by NXP Semiconductors and customer.

10. t001dis109.fm: AEC unqualified products (DS, AN, UM, errata; if applicable)

AEC unqualified products — This product has not been qualified to the appropriate Automotive Electronics Council (AEC) standard Q100 or Q101 and should not be used in automotive critical applications, including but not limited to applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer’s own risk.

11. t001dis110.fm: Suitability for use in automotive applications only (DS, AN, UM, errata; if applicable)

Suitability for use in automotive applications only — This NXP Semiconductors product has been developed for use in automotive applications only. The product is not designed, authorized or warranted to be suitable for any other use, including medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer’s own risk.