# 8M x 8 HyperRAM™

**ADVANCED INFORMATION**
**MAY 2015**

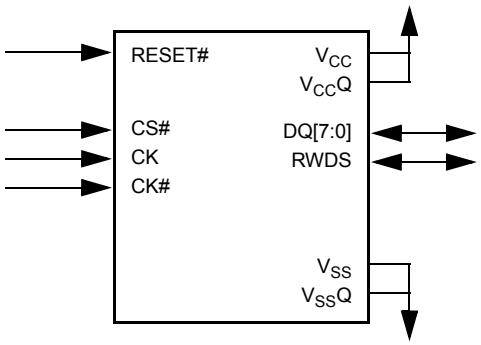
## Overview

The IS66/67WVH8M8ALL/BLL are integrated memory device containing 64Mbit Pseudo Static Random Access Memory using a self-refresh DRAM array organized as 8M words by 8 bits. The device supports a HyperBus interface, Very Low Signal Count (Address, Command and data through 8 DQ pins), Hidden Refresh Operation, and Automotive Temperature Operation, designed specially for Mobile and Automotive applications.

## Distinctive Characteristics

### HyperBus™ Low Signal Count Interface

- **3.0V I/O, 11 bus signals**
  - Single ended clock (CK)
- **1.8V I/O, 12 bus signals**
  - Differential clock (CK, CK#)
- **Chip Select (CS#)**
- **8-bit data bus (DQ[7:0])**
- **Read-Write Data Strobe (RWDS)**
  - Bidirectional Data Strobe / Mask
  - Output at the start of all transactions to indicate refresh latency
  - Output during read transactions as Read Data Strobe
  - Input during write transactions as Write Data Mask




### High Performance

- **Up to 333MB/s**
- **Double-Data Rate (DDR) - two data transfers per clock**
- **166-MHz clock rate (333 MB/s) at 1.8V $V_{CC}$**
- **100-MHz clock rate (200 MB/s) at 3.0V $V_{CC}$**
- **Sequential burst transactions**
- **Configurable Burst Characteristics**
  - Wrapped burst lengths:
    - 16 bytes (8 clocks)
    - 32 bytes (16 clocks)
    - 64 bytes (32 clocks)
    - 128 bytes (64 clocks)
  - Linear burst
  - Hybrid option - one wrapped burst followed by linear burst
  - Wrapped or linear burst type selected in each transaction
  - Configurable output drive strength
- **Package and Die Options**
  - 25-ball FBGA footprint
  - 24-ball FBGA footprint

Copyright © 2015 Integrated Silicon Solution, Inc. All rights reserved. ISSI reserves the right to make changes to this specification and its products at any time without notice. ISSI assumes no liability arising out of the application or use of any information, products or services described herein. Customers are advised to obtain the latest version of this device specification before relying on any published information and before placing orders for products.

Integrated Silicon Solution, Inc. does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless Integrated Silicon Solution, Inc. receives written assurance to its satisfaction, that:
a.) the risk of injury or damage has been minimized;
b.) the user assume all such risks; and
c.) potential liability of Integrated Silicon Solution, Inc is adequately protected under the circumstances

# Performance Summary

| Read Transaction Timings | |
|---|---|
| Maximum Clock Rate at 1.8V $V_{CC}/V_{CC}Q$ | 166MHz |
| Maximum Clock Rate at 3.0V $V_{CC}/V_{CC}Q$ | 100MHz |
| Maximum Access Time, ($t_{ACC}$ @ 166MHz) | 36 ns |
| Maximum CS# Access Time to first word @ 166MHz (excluding refresh latency) | 56 ns |

| Maximum Current Consumption | |
|---|---|
| Burst Read or Write (linear burst at 166MHz, 1.8V) | 60mA |
| Power On Reset | 50mA |
| Standby (CS# = High, 3V, 105°C) | 300uA |
| Deep Power Down (CS# = High, 3V, 105°C) | 20uA |
| Standby (CS# = High, 1.8V, 105°C) | 300uA |
| Deep Power Down (CS# = High, 1.8V, 105°C) | 10uA |

# 1. General Description

The ISSI HyperRAM family of products are high-speed CMOS, Self-refresh Dynamic RAM (DRAM) devices, with a HyperBus interface.

The Random Access Memory (RAM) array uses dynamic cells that require periodic refresh. Refresh control logic within the device manages the refresh operations on the RAM array when the memory is not being actively read or written by the HyperBus interface master (host). Since the host is not required to manage any refresh operations, the DRAM array appears to the host as though the memory uses static cells that retain data without refresh. Hence, the memory can also be described as Pseudo Static RAM (PSRAM).

Because the DRAM cells cannot be refreshed during a read or write transaction, there is a requirement that the host not perform read or write burst transfers that are long enough to block the necessary internal logic refresh operations when they are needed. The host is required to limit the duration of transactions and allow additional initial access latency, at the beginning of a new transaction, if the memory indicates a refresh operation is needed.

HyperBus is a low signal count, Double Data Rate (DDR) interface, that achieves high speed read and write throughput. The DDR protocol transfers two data bytes per clock cycle on the DQ input/output signals. A read or write transaction on HyperBus consists of a series of 16-bit wide, one clock cycle data transfers at the internal HyperRAM core with two corresponding 8-bit wide, one-half-clock-cycle data transfers on the DQ signals. All inputs and outputs are LV-CMOS compatible. Ordering Part Number (OPN) device versions are available for core ($V_{CC}$) and IO buffer ($V_{CC}Q$) supplies of either 1.8V or 3.0V (nominal).

Command, Address, and Data information is transferred over the eight HyperBus DQ signals. The clock is used for information capture by a HyperBus device when receiving Command-Address/Data on the DQ signals. Command-Address values are center aligned with clock edges.

The Read/Write Data Strobe (RWDS) is a bidirectional signal that indicates:

- when data will start to transfer from the memory to the host in read transactions (initial read latency),
- when data is being transferred from the memory to the host during read data transfers (source synchronous read data strobe),
- when data will start to transfer from the host to the memory in write transactions (initial write latency),
- and data masking during write data transfers.

During the command and address cycles of a read or write transaction, RWDS acts as an output from the memory to indicate whether additional initial access latency is needed to perform a dynamic memory refresh operation.

During read data transfers, RWDS is a read data strobe with data values edge aligned with the transitions of RWDS driven by the memory device.

During write data transfers, RWDS indicates whether a data byte is masked (prevented from changing the byte location in memory) or not masked (written to memory). Data masking may be used by the host to byte align write data within the memory or to enable merging of multiple non-word aligned writes in a single burst write. During write transactions, data is center aligned with the clock.

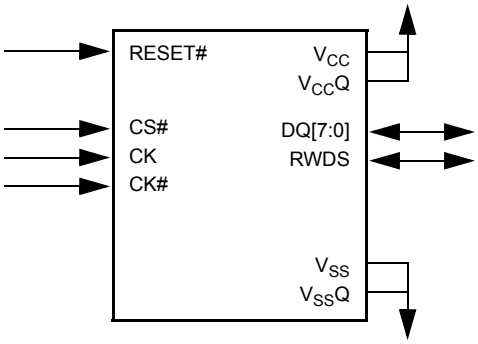
Read and write transactions are burst oriented, transferring the next sequential word during each clock cycle. Each individual read or write transaction can use either a wrapped or linear burst sequence. During wrapped transactions, accesses start at a selected location and continue to the end of a configured word group aligned boundary, then wrap to the beginning location in the group, then continue back to the starting location. Wrapped bursts are generally used for critical word first instruction or data cache line fill read accesses. During linear transactions, accesses start at a selected location and continue in a sequential manner until the transaction is terminated when CS# returns High. Linear transactions are generally used for large contiguous data transfers such as graphic image moves. Since each transaction command selects the type of burst sequence for that access, wrapped and linear burst transactions can be dynamically intermixed as needed.

For additional information on HyperBus interface operation, please refer to the HyperBus specification.

# 2. HyperRAM Product Overview

The HyperRAM™ Family consists of multiple density option, 1.8V or 3.0V core and I/O, synchronous self-refresh Dynamic RAM (DRAM) memory devices. This family provides a HyperBus slave interface to the host system. HyperBus has an 8 bit (1 byte) wide DDR data bus and uses only word-wide (16-bit data) address boundaries. Read transactions provide 16 bits of data during each clock cycle (8 bits on both clock edges). Write transactions take 16 bits of data from each clock cycle (8 bits on each clock edge).

**Figure 2.1** HyperRAM Interface




Read and write transactions require two clock cycles to define the target row address and burst type, then an initial access latency of $t_{ACC}$ . During the Command-Address (CA) part of a transaction, the memory will indicate whether an additional latency for a required refresh time ($t_{RFH}$) is added to the initial latency; by driving the RWDS signal to the High state. During the CA period the third clock cycle will specify the target word address within the target row. During a read (or write) transaction, after the initial data value has been output (or input), additional data can be read from (or written to) the row on subsequent clock cycles in either a wrapped or linear sequence. When configured in linear burst mode, the device will automatically fetch the next sequential row from the memory array to support a continuous linear burst. Simultaneously accessing the next row in the array while the read or write data transfer is in progress, allows for a linear sequential burst operation that can provide a sustained data rate of 333 MB/s (1 byte (8 bit data bus) * 2 *(data clock edges)* * 166 MHz = 333 MB/s).

# 3. HyperBus™ Interface

For the general description of how the HyperBus interface operates in HyperRAM memories, refer to the HyperBus specification. The following section describes HyperRAM device dependent aspects of HyperBus interface operation.

All bus transactions can be classified as either read or write. A bus transaction is started with CS# going Low with CK = Low and CK# = High. The transaction to be performed is presented to the HyperRAM device during the first three clock cycles in a DDR manner using all six clock edges. These first three clocks transfer three words of Command / Address (CA0, CA1, CA2) information to define the transaction characteristics:

- Read or write transaction.
- Whether the transaction will be to the memory array or to register space.
- Whether a read transaction will use a linear or wrapped burst sequence.
- The target half-page address (row and upper order column address).
- The target Word (within half-page) address (lower order column address).

Once the transaction has been defined, a number of idle clock cycles are used to satisfy initial read or write access latency requirements before data is transferred. During the Command-Address portion of all transactions, RWDS is used by the memory to indicate whether additional initial access latency will be inserted for a required refresh of the memory array.

When data transfer begins, read data is edge aligned with RWDS transitions or write data is center aligned with clock transitions. During read data transfer, RWDS serves as a source synchronous data timing strobe. During write data transfer, clock transitions provide the data timing reference and RWDS is used as a data mask. When RWDS is Low during a write data transfer, the data byte is written into memory; if RWDS is High during the transfer the byte is not written.

Data is transferred as 16-bit values with the first eight bits transferred on a High going CK (write data or CA bits) or RWDS edge (read data) and the second eight bits being transferred on the Low going CK or RWDS edge. Data transfers during read or write operations can be ended at any time by bringing CS# High when CK = Low and CK# = High.

The clock may stop in the idle state while CS# is High.

The clock may also stop in the idle state for short periods while CS# is Low, as long as this does not cause a transaction to exceed the CS# maximum time low ($t_{CSM}$) limit. This is referred to as Active Clock Stop mode. In some HyperBus devices this mode is used for power reduction. However, due to the relatively short $t_{CSM}$ period for completing each data transfer, the Active Clock Stop mode is generally not useful for power reduction but, may be used for short duration data flow control by the HyperBus master.

## 3.1 Command-Address Bit Assignments

**Table 3.1** Command-Address Bit Definitions

| CA Bit# | Bit Name | Bit Function |
|---|---|---|
| 47 | R/W# | Identifies the transaction as a read or write.<br>R/W#=1 indicates a Read transaction<br>R/W#=0 indicates a Write transaction |
| 46 | Address Space (AS) | Indicates whether the read or write transaction accesses the memory or register spaces.<br>AS=0 indicates memory space<br>AS=1 indicates the register space<br>The register space is used to access device ID and Configuration registers. |
| 45 | Burst Type | Indicates whether the burst will be linear or wrapped.<br>Burst Type=0 indicates wrapped burst<br>Burst Type=1 indicates linear burst |
| 44-37 (256Mb)<br>44-36 (128Mb)<br>44-35 (64Mb) | Reserved | Reserved for future row address expansion.<br>Reserved bits should be set to 0 by the HyperBus master. |
| 36-22 (256Mb)<br>35-22 (128Mb)<br>34-22 (64Mb) | Row Address | Row component of the target address: System word address bits A23-A9 |
| 21-16 | Upper Column Address | Upper Column component of the target address: System word address bits A8-A3 |
| 15-3 | Reserved | Reserved for future column address expansion.<br>Reserved bits should be set to 0 by the HyperBus master. |
| 2-0 | Lower Column (word) Address | Lower Column component of the target address: System word address bits A2-0 selecting the starting word within a row. |

## 3.2 Read Transactions

**Table 3.2** Maximum Operating Frequency For Latency Code Options

| Latency Code | Latency Clocks | Maximum Operating Frequency (MHz) |
|---|---|---|
| 0000 | 5 | 133 |
| 0001 | 6 | 166 |
| 0010 | Reserved | NA |
| 0011 | Reserved | NA |
| 0100 | Reserved | NA |
| 0101 | Reserved | NA |
| 0110 | Reserved | NA |
| 0111 | Reserved | NA |
| 1000 | Reserved | NA |
| 1001 | Reserved | NA |
| 1010 | Reserved | NA |
| 1011 | Reserved | NA |
| 1100 | Reserved | NA |
| 1101 | Reserved | NA |
| 1110 | 3 | 83 |
| 1111 | 4 | 100 |

***Note:***

*1. The Latency Code is the value loaded into Configuration Register bits CR0[7:4].*

## 3.3 Write to Memory Space Transactions

When a linear burst write reaches the last address in the array, continuing the burst beyond the last address has undefined results.

# 4. Memory Space

When CA[46] is 0 a read or write transaction accesses the DRAM memory array.

**Memory Space Address Map**

| Unit Type | Count | System Word Address Bits | CA Bits | Notes |
|---|---|---|---|---|
| Rows within 64Mb device | 8192 (Rows) | A21 - A9 | 34 - 22 | |
| Rows within 32Mb device | 4096 (Rows) | A20 - A9 | 33 - 22 | |
| Row | 1 (row) | A8 - A3 | 21 - 16 | 512 (word addresses) 1 kbytes |
| Half-Page | 8 (word addresses) | A2 - A0 | 2 - 0 | 16 bytes |

# 5. Register Space

When CA[46] is 1 a read or write transaction accesses the Register Space.

**Register Space Address Map**

| Register | System Address | - | - | - | 31-27 | 26-19 | 18-11 | 10-3 | - | 2-0 |
|---|---|---|---|---|---|---|---|---|---|---|
| | CA Bits | 47 | 46 | 45 | 44-40 | 39-32 | 31-24 | 23-16 | 15-8 | 7-0 |
| Identification Register 0 (read only) | | C0h or E0h | | | | 00h | 00h | 00h | 00h | 00h |
| Identification Register 1 (read only) | | C0h or E0h | | | | 00h | 00h | 00h | 00h | 01h |
| Configuration Register 0 Read | | C0h or E0h | | | | 00h | 01h | 00h | 00h | 00h |
| Configuration Register 0 Write | | 60h | | | | 00h | 01h | 00h | 00h | 00h |
| Configuration Register 1 Read | | C0h or E0h | | | | 00h | 01h | 00h | 00h | 01h |
| Configuration Register 1 Write | | 60h | | | | 00h | 01h | 00h | 00h | 01h |

***Note:***

*1. CA45 may be either 0 or 1 for either wrapped or linear read. CA45 must be 1 as only linear single word register writes are supported.*

## 5.1 Device Identification Registers

There are two read only, non-volatile, word registers, that provide information on the device selected when CS# is low. The device information fields identify:

- Manufacturer
- Type
- Density
  - Die Stack Address
  - Row address bit count
  - Column address bit count

.

**Table 5.1** ID Register 0 Bit Assignments

| Bits | Function | Settings (Binary) |
|---|---|---|
| 15-14 | Die Address | 00 - Die 0 (Lowest address die or single die)<br>01 - Die 1<br>10 - Die 2<br>11 - Die 3 |
| 13 | Reserved | 0 - default |
| 12-8 | Row Address Bit Count | 00000 - One Row address bit<br>...<br>11111 - Thirty-two row address bits |
| 7-4 | Column Address Bit Count | 0000 - One column address bit<br>...<br>1111 - Sixteen column address bits |
| 3-0 | Manufacturer | 0000 - Reserved<br>0011 - ISSI<br>0010 to 1111 - Reserved |

.

**Table 5.2** ID Register 1 Bit Assignments

| Bits | Function | Settings (Binary) |
|---|---|---|
| 15-4 | Reserved | 0000_0000_0000b (default) |
| 3-0 | Device Type | 0000 - HyperRAM<br>0001 to 1111 - Reserved |

### 5.1.1 Density and Row Boundaries

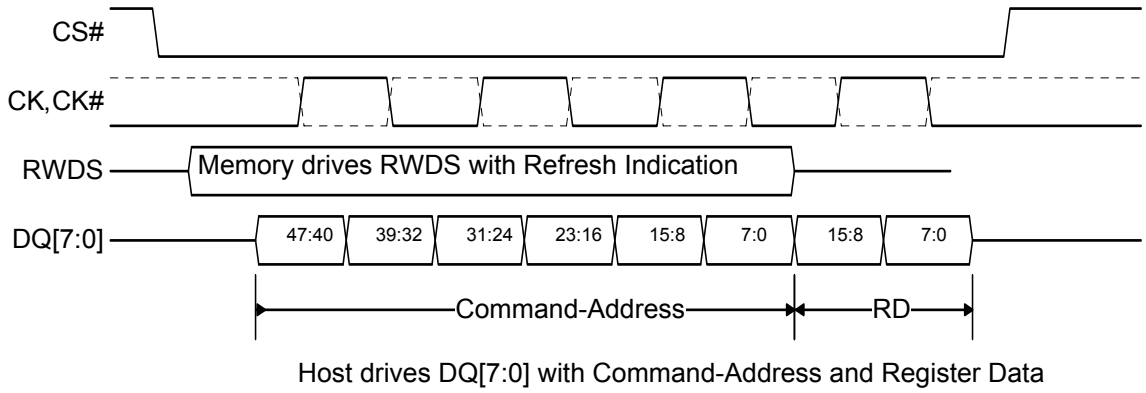
The DRAM array size (density) of the device can be determined from the total number of system address bits used for the row and column addresses as indicated by the Row Address Bit Count and Column Address Bit Count fields in the ID0 register. For example: a 64 Mbit HyperRAM has 9 column address bits and 13 row address bits for a total of 22 word address bits = 2^22 = 4 Mwords = 8 MBytes. The 9 column address bits indicate that each row holds 2^9 = 512 words = 1 kbytes. The row address bit count indicates there are 8196 rows to be refreshed within each array refresh interval. The row count is used in calculating the refresh interval.

## 5.2 Register Space Access

Register default values are loaded upon power-up or hardware reset. The registers can be altered at any time while the device is in the standby state.

Loading a register is accomplished with a single 16-bit word write transaction as shown in Figure 5.1. CA[47] is zero to indicate a write transaction, CA[46] is a one to indicate a register space write, CA[45] is a one to indicate a linear write, lower order bits in the CA field indicate the register address.

**Figure 5.1** Loading a Register




***Notes:***

1. *The host must not drive RWDS during a write to register space.*
2. *The RWDS signal is driven by the memory during the Command-Address period based on whether the memory array is being refreshed. This refresh indication does not affect the writing of register data. RWDS is driven immediately after CS# goes low, before CA[47:46] are received to indicate that the transaction is a write to register space, for which the RWDS refresh indication is not relevant.*
3. *The register value is always provided immediately after the CA value and is not delayed by a refresh latency.*
4. *The the RWDS signal returns to high impedance after the Command-Address period. Register data is never masked. Both data bytes of the register data are loaded into the selected register.*

Each register is written with a separate single word write transaction. Register write transactions have zero latency, the single word of data immediately follows the Command-Address. RWDS is not driven by the host during the write because RWDS is always driven by the memory during the CA cycles to indicate whether a memory array refresh is in progress. Because a register space write goes directly to a register, rather than the memory array, there is no initial write latency, related to an array refresh that may be in progress. In a register write, RWDS is also not used as a data mask because both bytes of a register are always written and never masked.

Reserved register fields must be written with their default value. Writing reserved fields with other than default values may produce undefined results.

Reading of a register is accomplished with a single 16 bit read transaction with CA[46]=1 to select register space. If more than one word is read, the same register value is repeated in each word read. The CA[45] burst type is “don’t care” because only a single register value is read. The contents of the register is returned in the same manner as reading array data, with one or two latency counts, based on the state of RWDS during the Command-Address period. The latency count is defined in the Configuration Register 0 Read Latency field (CR0[7:4]).

### 5.2.1 Configuration Register 0

Configuration Register 0 (CR0) is used to define the power mode and access protocol operating conditions for the HyperRAM device. Configurable characteristics include:

- Wrapped Burst Length (16, 32, 64 or 128-byte aligned and length data group)
- Wrapped Burst Type
  - Legacy wrap (sequential access with wrap around within a selected length & aligned group)
  - Hybrid wrap (Legacy wrap once then linear burst at start of the next sequential group)
- Initial Latency
- Variable Latency
  - Whether an array read or write transaction will use fixed or variable latency. If fixed latency is selected the memory will always indicate a refresh latency and delay the read data transfer accordingly. If variable latency is selected, latency for a refresh is only added when a refresh is required at the same time a new transaction is starting.
- Output Drive Strength
- Deep Power Down Mode

.

**Table 5.3** Configuration Register 0 Bit Assignments

| CR0 Bit | Function | Settings (Binary) |
|---|---|---|
| 15 | Deep Power Down Enable | 1 - Normal operation (default)<br>0 - Writing 0 to CR[15] causes the device to enter Deep Power Down. |
| 14-12 | Drive Strength | 000 - 34 ohms (default)<br>001 - 115 ohms<br>010 - 67 ohms<br>011 - 46 ohms<br>100 - 34 ohms<br>101 - 27 ohms<br>110 - 22 ohms<br>111 - 19 ohms |
| 11-8 | Reserved | 1 - Reserved (default)<br>Reserved for Future Use. When writing this register, these bits should be set to 1 for future compatibility. |
| 7-4 | Initial Latency | 0000 - 5 Clock Latency<br>0001 - 6 Clock Latency (default)<br>0010 - Reserved<br>0011 - Reserved<br>0100 - Reserved<br>...<br>1101 - Reserved<br>1110 - 3 Clock Latency<br>1111 - 4 Clock Latency |
| 3 | Fixed Latency Enable | 0 - Variable Latency - 1 or 2 times Initial Latency depending on RWDS during CA cycles.<br>1 - Fixed 2 times Initial Latency (default) |
| 2 | Hybrid Burst Enable | 0: Wrapped burst sequences to follow hybrid burst sequencing<br>1: Wrapped burst sequences in legacy wrapped burst manner (default) |
| 1-0 | Burst Length | 00 - 128 bytes<br>01 - 64 bytes<br>10- 16 bytes<br>11 - 32 bytes (default) |

#### 5.2.1.1 Wrapped Burst

A wrapped burst transaction accesses memory within a group of words aligned on a word boundary matching the length of the configured group. Wrapped access groups can be configured as 16, 32, 64 or 128 bytes alignment and length. During wrapped transactions, access starts at the Command-Address selected location within the group, continues to the end of the configured word group aligned boundary, then wraps around to the beginning location in the group, then continues back to the starting location. Wrapped bursts are generally used for critical word first instruction or data cache line fill read accesses.

#### 5.2.1.2 Hybrid Burst

The beginning of a hybrid burst will wrap within the target address wrapped burst group length before continuing to the next half-page of data beyond the end of the wrap group. Continued access is in linear burst order until the transfer is ended by returning CS# high. This hybrid of a wrapped burst followed by a linear burst starting at the beginning of the next burst group, allows multiple sequential address cache lines to be filled in a single access. The first cache line is filled starting at the critical word. Then the next sequential line in memory can be read in to the cache while the first line is being processed.

**Table 5.4** CR0[2] Control of Wrapped Burst Sequence

| Bit | Default Value | Name |
|---|---|---|
| 2 | 1 | Hybrid Burst Enable<br>CR[2]= 0: Wrapped burst sequences to follow hybrid burst sequencing<br>CR[2]= 1: Wrapped burst sequences in legacy wrapped burst manner |

.

**Table 5.5** Example Wrapped Burst Sequences (Sheet 1 of 2)

| Burst Selection | | Burst Type | Wrap Boundary (bytes) | Start Address (Hex) | Address Sequence (Hex) (Words) |
|---|---|---|---|---|---|
| CA[45] | CR0[2:0] | | | | |
| 0 | 000 | Hybrid 128 | 128 Wrap once then Linear | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 00, 01, 02<br>(wrap complete, now linear beyond the end of the initial 128 byte wrap group)<br>40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, ... |
| 0 | 001 | Hybrid 64 | 64 Wrap once then Linear | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 00, 01, 02,<br>(wrap complete, now linear beyond the end of the initial 64 byte wrap group)<br>20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, ... |
| 0 | 001 | Hybrid 64 | 64 Wrap once then Linear | XXXXXX2E | 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D,<br>(wrap complete, now linear beyond the end of the initial 64 byte wrap group)<br>40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, ... |
| 0 | 010 | Hybrid 16 | 16 Wrap once then Linear | XXXXXX02 | 02, 03, 04, 05, 06, 07, 00, 01,<br>(wrap complete, now linear beyond the end of the initial 16 byte wrap group)<br>08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, ... |
| 0 | 010 | Hybrid 16 | 16 Wrap once then Linear | XXXXXX0C | 0C, 0D, 0E, 0F, 08, 09, 0A, 0B,<br>(wrap complete, now linear beyond the end of the initial 16 byte wrap group)<br>10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, ... |
| 0 | 011 | Hybrid 32 | 32 Wrap once then Linear | XXXXXX0A | 0A, 0B, 0C, 0D, 0E, 0F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, ... |
| 0 | 011 | Hybrid 32 | 32 Wrap once then Linear | XXXXXX1E | 1E, 1F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, ... |
| 0 | 100 | Wrap 128 | 128 | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 00, 01, 02, ... |
| 0 | 101 | Wrap 64 | 64 | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 00, 01, 02, ... |

**Table 5.5** Example Wrapped Burst Sequences (Sheet 2 of 2)

| Burst Selection | | Burst Type | Wrap Boundary (bytes) | Start Address (Hex) | Address Sequence (Hex) (Words) |
|---|---|---|---|---|---|
| CA[45] | CR0[2:0] | | | | |
| 0 | 101 | Wrap 64 | 64 | XXXXXX2E | 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, ... |
| 0 | 110 | Wrap 16 | 16 | XXXXXX02 | 02, 03, 04, 05, 06, 07, 00, 01, ... |
| 0 | 110 | Wrap 16 | 16 | XXXXXX0C | 0C, 0D, 0E, 0F, 08, 09, 0A, 0B, ... |
| 0 | 111 | Wrap 32 | 32 | XXXXXX0A | 0A, 0B, 0C, 0D, 0E, 0F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, ... |
| 0 | 111 | Wrap 32 | 32 | XXXXXX1E | 1E, 1F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, ... |
| 1 | XXX | Linear | Linear Burst | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, ... |

#### 5.2.1.3 Initial Latency

Memory Space read and write transactions or Register Space read transactions require some initial latency to open the row selected by the Command-Address. This initial latency is $t_{ACC}$ . The number of latency clocks needed to satisfy $t_{ACC}$ depends on the HyperBus frequency and can vary from 3 to 6 clocks. The value in CR0[7:4] selects the number of clocks for initial latency. The default value is 6 clocks, allowing for operation up to a maximum frequency of 166MHz prior to the host system setting a lower initial latency value that may be more optimal for the system.

In the event a distributed refresh is required at the time a Memory Space read or write transaction or Register Space read transaction begins, the RWDS signal goes high during the Command-Address to indicate that an additional initial latency is being inserted to allow a refresh operation to complete before opening the selected row.

Register Space write transactions always have zero initial latency. RWDS may be High or Low during the Command-Address period. The level of RWDS during the Command-Address period does not affect the placement of register data immediately after the Command-Address, as there is no initial latency needed to capture the register data. A refresh operation may be performed in the memory array in parallel with the capture of register data.

#### 5.2.1.4 Fixed Latency

A configuration register option bit CR0[3] is provided to make all Memory Space read and write transactions or Register Space read transactions require the same initial latency by always driving RWDS high during the Command-Address to indicate that two initial latency periods are required. This fixed initial latency is independent of any need for a distributed refresh, it simply provides a fixed (deterministic) initial latency for all of these transaction types. The fixed latency option may simplify the design of some HyperBus memory controllers or ensure deterministic transaction performance. Fixed latency is the default POR or reset configuration. The system may clear this configuration bit to disable fixed latency and allow variable initial latency with RWDS driven high only when additional latency for a refresh is required.

#### 5.2.1.5 Drive Strength

DQ signal line loading, length, and impedance vary depending on each system design. Configuration register bits CR0[14:12] provide a means to adjust the DQ[7:0] signal output impedance to customize the DQ signal impedance to the system conditions to minimize high speed signal behaviors such as overshoot, undershoot, and ringing. The default POR or reset configuration value is 000b to select the mid point of the available output impedance options.

The impedance values shown are typical for both pull-up and pull-down drivers at typical silicon process conditions, nominal operating voltage (1.8Vor 3V) and 50°C. The impedance values may vary by up to +/- 80% from the typical values depending on the Process, Voltage, and Temperature (PVT) conditions. Impedance will increase with slower process, lower voltage, or higher temperature. Impedance will decrease with faster process, higher voltage, or lower temperature.

Each system design should evaluate the data signal integrity across the operating voltage and temperature ranges to select the best drive strength settings for the operating conditions.

#### 5.2.1.6 Deep Power Down

When the HyperRAM device is not needed for system operation, it may be placed in a very low power consuming mode called Deep Power Down (DPD), by writing 0 to CR0[15]. When CR0[15] is cleared to 0, the device enters the DPD mode within $t_{DPDIN}$ time and all refresh operations stop. The data in RAM is lost, (becomes invalid without refresh) during DPD mode. The next access to the device driving CS# Low then High, POR, or a reset will cause the device to exit DPD mode. Returning to Standby mode requires $t_{DPDOUT}$ time. For additional details see Section 6.1.3, *Deep Power Down* on page 15.

### 5.2.2 Configuration Register 1

Configuration Register 1 (CR1) is used to define the distributed refresh interval for this HyperRAM device. The core DRAM array requires periodic refresh of all bits in the array. This can be done by the host system by reading or writing a location in each row within a specified time limit. The read or write access copies a row of bits to an internal buffer. At the end of the access the bits in the buffer are written back to the row in memory, thereby recharging (refreshing) the bits in the row of DRAM memory cells.

However, the host system generally has better things to do than to periodically read every row in memory and keep track that each row is visited within the required refresh interval for the entire memory array. The HyperRAM family devices include self-refresh logic that will refresh rows automatically so that the host system is relieved of the need to refresh the memory. The automatic refresh of a row can only be done when the memory is not being actively read or written by the host system. The refresh logic waits for the end of any active read or write before doing a refresh, if a refresh is needed at that time. If a new read or write begins before the refresh is completed, the memory will drive RWDS high during the Command-Address period to indicate that an additional initial latency time is required at the start of the new access in order to allow the refresh operation to complete before starting the new access.

The required refresh interval for the entire memory array varies with temperature as shown in Table 5.6, *Array Refresh Interval per Temperature* on page 13. This is the time within which all rows must be refreshed. Refresh of all rows could be done as a single batch of accesses at the beginning of each interval, in groups (burst refresh) of several rows at a time, spread throughout each interval, or as single row refreshes evenly distributed throughout the interval. The self-refresh logic distributes single row refresh operations throughout the interval so that the memory is not busy doing a burst of refresh operations for a long period, such that the burst refresh would delay host access for a long period.

**Table 5.6** Array Refresh Interval per Temperature

| Device Temperature (Degrees C) | Array Refresh Interval (ms) | Array Rows | Recommended $t_{CMS}$ (µs) |
|---|---|---|---|
| 85 | 64 | 8192 | 4 |
| 105 | 16 | 8192 | 1 |

.

**Table 5.7** Configuration Register 1 Bit Assignments

| CR1 Bit | Function | Settings (Binary) |
|---|---|---|
| 15-2 | Reserved | 000000h — Reserved (default)<br>Reserved for Future Use. When writing this register, these bits should be cleared to 0 for future compatibility. |
| 1-0 | Distributed Refresh Interval | 10b — default<br>4 µs for Industrial temperature range devices<br>1 µs for Industrial Plus temperature range devices<br>11b — 1.5 times default<br>00b — 2 times default<br>01b — 4 times default |

The distributed refresh method requires that the host does not do burst transactions that are so long as to prevent the memory from doing the distributed refreshes when they are needed. This sets an upper limit on the length of read and write transactions so that the refresh logic can insert a refresh between transactions. This limit is called the CS# low maximum time ($t_{CMS}$). The $t_{CMS}$ value is determined by the array refresh interval divided by the number of rows in the array, then reducing this calculation by half to ensure that a distributed refresh interval cannot be entirely missed by a maximum length host access starting immediately before a distributed refresh is needed. Because $t_{CMS}$ is set to half the required distributed refresh interval, any series of maximum length host accesses that delay refresh operations will be catching up on refresh operations at twice the rate required by the refresh interval divided by the number of rows.

The host system is required to respect the $t_{CMS}$ value by ending each transaction before violating $t_{CMS}$. This can be done by host memory controller logic splitting long transactions when reaching the $t_{CMS}$ limit, or by host system hardware or software not performing a single read or write transaction that would be longer than $t_{CMS}$.

As noted in Table 5.6, *Array Refresh Interval per Temperature* on page 13 the array refresh interval is longer at lower temperatures such that $t_{CMS}$ could be increased to allow longer transactions. The host system can either use the $t_{CMS}$ value from the table for the maximum operating temperature or, may determine the current operating temperature from a temperature sensor in the system in order to set a longer distributed refresh interval.

The host system may also effectively increase the $t_{CMS}$ value by explicitly taking responsibility for performing all refresh and doing burst refresh reading of multiple sequential rows in order to catch up on distributed refreshes missed by longer transactions.

# HyperRAM Hardware Interface

For the general description of the HyperBus hardware interface of HyperFlash memories refer to the HyperBus Specification. The following section describes HyperRAM device dependent aspects of hardware interface.

# 6. Interface States

## 6.1 Power Conservation Modes

### 6.1.1 Interface Standby

Standby is the default, low power, state for the interface while the device is not selected by the host for data transfer (CS#= High). All inputs, and outputs other than CS# and RESET# are ignored in this state.

### 6.1.2 Active Clock Stop

The Active Clock Stop mode reduces device interface energy consumption to the $I_{CC6}$ level during the data transfer portion of a read or write operation. The device automatically enables this mode when clock remains stable for $t_{ACC}$ + 30 ns. While in Active Clock Stop mode, read data is latched and always driven onto the data bus. $I_{CC6}$ shown in Section 7.6, *DC Characteristics* on page 22.

Active Clock Stop mode helps reduce current consumption when the host system clock has stopped to pause the data transfer. Even though CS# may be Low throughout these extended data transfer cycles, the memory device host interface will go into the Active Clock Stop current level at $t_{ACC}$ + 30 ns. This allows the device to transition into a lower current mode if the data transfer is stalled. Active read or write current will resume once the data transfer is restarted with a toggling clock. The Active Clock Stop mode must not be used in violation of the $t_{CSM}$ limit. CS# must go high before $t_{CSM}$ is violated.

### 6.1.3 Deep Power Down

In the Deep Power Down (DPD) mode, current consumption is driven to the lowest possible level ($i_{DPD}$). DPD mode is entered by writing a 0 to CR0[15]. The device reduces power within $t_{DPDIN}$ time and all refresh operations stop. The data in Memory Space is lost, (becomes invalid without refresh) during DPD mode. The next access to the device, driving CS# Low then High, will cause the device to exit DPD mode. A read or write transaction used to drive CS# Low then High to exit DPD mode is a dummy transaction that is ignored by the device. Also, POR, or a hardware reset will cause the device to exit DPD mode. Only the CS# and RESET# signals are monitored during DPD mode. Returning to Standby mode following a dummy transaction or reset requires $t_{DPDOUT}$ time. Returning to Standby mode following a POR requires $t_{VCS}$ time, as with any other POR. Following the exit from DPD due to any of these events, the device is in the same state as following POR.

**Table 6.1** Deep Power Down Timing Parameters

| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| $t_{DPDIN}$ | Deep Power Down CR0[15]=0 register write to DPD power level | 10 | - | µs |
| $t_{DPDCSL}$ | Length of CS# Low period to cause an exit from Deep Power Down | 200 | - | ns |
| $t_{DPDOUT}$ | CS# Low then High to Standby wakeup time | - | 150 | µs |

**Figure 6.1** Deep Power Down Entry Timing

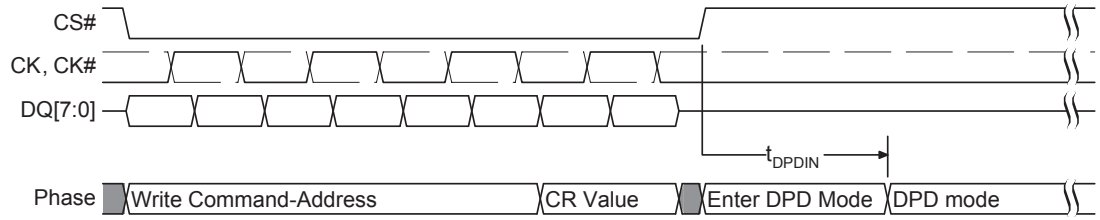
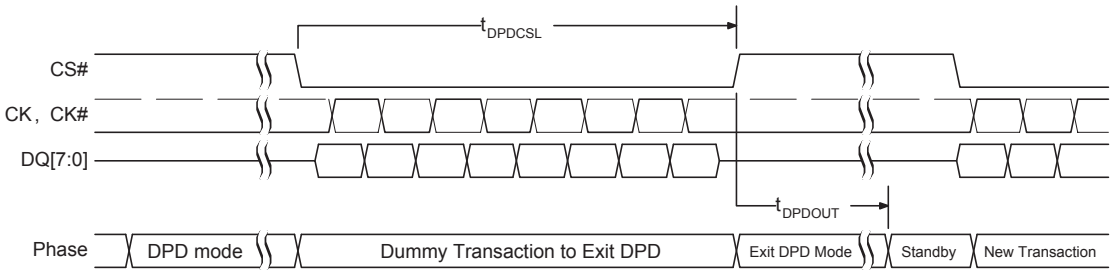



**Figure 6.2** Deep Power Down CS# Exit Timing



# 7. Electrical Specifications

## 7.1 Absolute Maximum Ratings

| | |
|---|---|
| Storage Temperature Plastic Packages | -65°C to +150°C |
| Ambient Temperature with Power Applied | Device Dependent |
| Voltage with Respect to Ground | |
| All signals (1) | -0.5 V to +($V_{CC}$ + 0.5 V) |
| Output Short Circuit Current (2) | 100 mA |
| $V_{CC}$ | -0.5 V to +4.0 V |

**Notes:**

1. Minimum DC voltage on input or I/O signal is -1.0 V. During voltage transitions, input or I/O signals may undershoot $V_{SS}$ to -1.0 V for periods of up to 20 ns. See Figure 7.1. Maximum DC voltage on input or I/O signals is $V_{CC}$ +1.0 V. During voltage transitions, input or I/O signals may overshoot to $V_{CC}$ +1.0 V for periods up to 20 ns. See Figure 7.2.
2. No more than one output may be shorted to ground at a time. Duration of the short circuit should not be greater than one second.
3. Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this data sheet is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.

### 7.1.1 Input Signal Overshoot

During DC conditions, input or I/O signals should remain equal to or between $V_{SS}$ and $V_{DD}$. During voltage transitions, inputs or I/Os may overshoot $V_{SS}$ to -1.0V or overshoot to $V_{DD}$ +1.0V, for periods up to 20 ns.

**Figure 7.1** Maximum Negative Overshoot Waveform

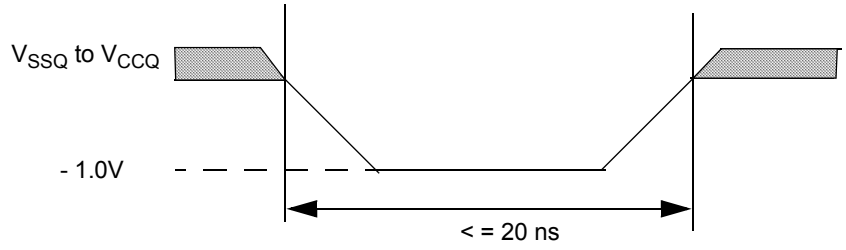
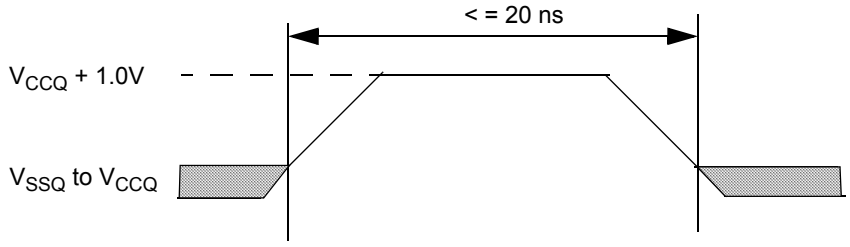



**Figure 7.2** Maximum Positive Overshoot Waveform



## 7.2 Latchup Characteristics

**Table 7.1** Latchup Specification

| Description | Min | Max | Unit |
|---|---|---|---|
| Input voltage with respect to $V_{SS}Q$ on all input only connections | - 1.0 | $V_{CCQ}$ + 1.0 | V |
| Input voltage with respect to $V_{SS}Q$ on all I/O connections | - 1.0 | $V_{CCQ}$ + 1.0 | V |
| $V_{CCQ}$ Current | -100 | +100 | mA |

***Note:***

1. *Excludes power supplies $V_{CC}$/$V_{CCQ}$. Test conditions: $V_{CC}$ = $V_{CCQ}$ = 1.8 V, one connection at a time tested, connections not being tested are at $V_{SS}$.*

## 7.3 Operating Ranges

Operating ranges define those limits between which the functionality of the device is guaranteed.

### 7.3.1 Temperature Ranges

Ambient Temperature ($T_A$)

| | |
|---|---|
| Industrial, Automtive A1 | – 40°C to +85°C |
| Extended, Automotive A2 | – 40°C to +105°C |

### 7.3.2 1.8V Power Supply Voltages

| | |
|---|---|
| $V_{CC}$ and $V_{CCQ}$ | 1.7 V to 1.95 V |

### 7.3.3 3.0V Power Supply Voltages

| | |
|---|---|
| $V_{CC}$ and $V_{CCQ}$ | 2.7V to 3.6V |

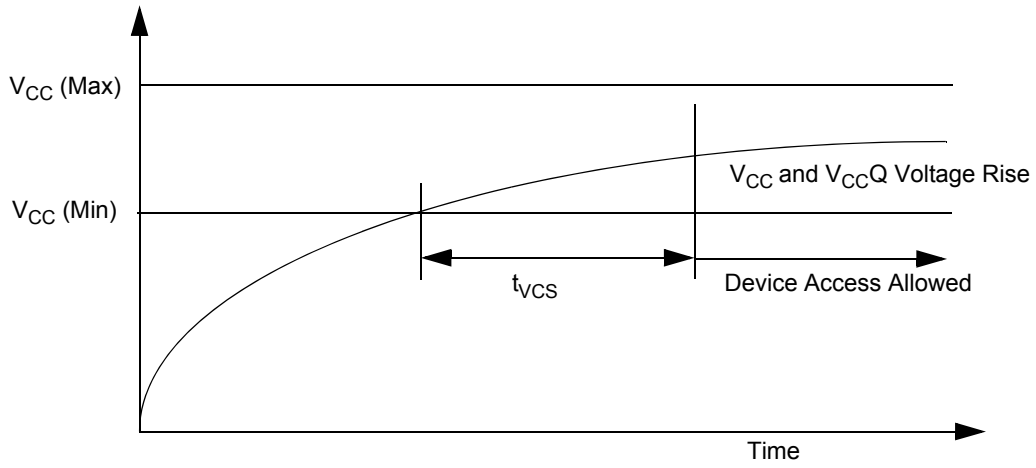
## 7.4 Power-On Reset

HyperBus products include an on-chip voltage sensor used to launch the Power On Reset (POR) process. Vcc and $V_{CCQ}$ must be applied simultaneously. When the power supply reaches a level at or above Vcc (min), the device will require $t_{VCS}$ time to complete its POR process. During the POR period, CS# is ignored. When POR is complete, CS# must be High, the device is ready for normal operation.

The $t_{VCS}$ value and other signal level requirements during POR are device dependent, refer to the individual device data sheets for additional information.

However, as a guideline for a power on sequence compatible with both HyperFlash and HyperRAM, the following diagrams & values may be used.

**Figure 7.3** Power On Reset Timing




***Notes:***
*1. $V_{CC}Q$ must be the same voltage as Vcc.*

HyperFlash and HyperRAM memories have different POR timing characteristics.

HyperFlash memory will start initialization when the power supply reaches $V_{CC}$ minimum independent of CS# or RESET# being High or Low and takes roughly twice as long as HyperRAM to initialize. HyperFlash memory is not sensitive to CS# until RSTO# stops driving Low. CS# for HyperFlash memory must remain High $t_{CSHI}$ time after RSTO# stops driving Low.

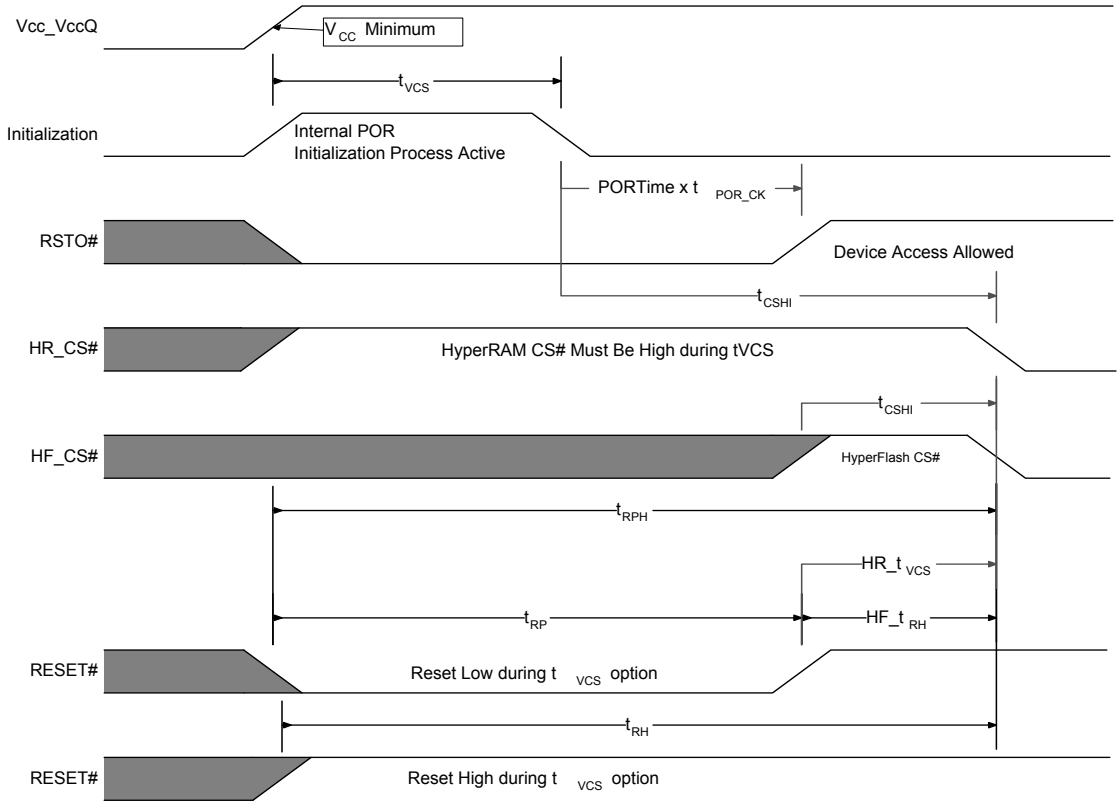
HyperFlash also provides a Reset Output (RSTO#) that the host system may use to drive all host system hardware Reset inputs, including the RESET# input of the HyperFlash driving RSTO#. In this case, the HyperFlash memory will be ready for access following the rise of RSTO# and after the HyperFlash $t_{RH}$ time is satisfied. HyperRAM will be ready for access following the rise of RSTO# and after the HyperRAM $t_{RH}$ time.

HyperRAM will start initialization when the power supply reaches $V_{CC}$ minimum and RESET# is high but, will delay the start of initialization if RESET is Low. HyperRAM initialization will not start until RESET# returns High. HyperRAM is sensitive to CS# during initialization and CS# must be high during $t_{VCS}$.

The general recommendation is to ramp up CS# and RESET# with $V_{CC}Q$ via a pull-up resistor so that CS# and RESET# are High during the HyperFlash $t_{VCS}$ period. HyperFlash and HyperRAM memory will then both be ready for access after the end of the $t_{VCS}$ period.

If RESET# must be driven Low by the host system during some or all of the $t_{VCS}$ period, it is required that CS# be High $t_{CSHI}$ time after RSTO# stops driving Low. In this situation HyperFlash will be ready for access after the HyperFlash $t_{RPH}$, $t_{RH}$ and $t_{CSHI}$ times are satisfied. HyperRAM will be ready for access later, after the HyperRAM $t_{VCS}$ time is satisfied. The host system HyperBus master may begin reading boot code from the HyperFlash memory while the HyperRAM memory completes its initialization process.

**Figure 7.4** Power On Reset Signal Diagram




***Notes:***

1. *$V_{CC}Q$ must be the same voltage as Vcc.*
2. *Vcc ramp rate may be non-linear.*
3. *HyperFlash and HyperRAM device internal POR initialization processes are active after $V_{CC}$ and $V_{CC}Q$ reach $V_{CC}$ minimum until the end of $t_{VCS}$. If RESET# is low during initialization device access may begin after $t_{RH}$ is satisfied.*
4. *Bus transactions (read and write) are not allowed during the power-up reset time ($t_{VCS}$). HyperRAM requires CS# high for $t_{CSHI}$ before going low for the first access.*
5. *PORTime is a customer programmed configuration register intended to allow HyperFlash RSTO# assertion beyond $t_{VCS}$. HyperFlash access is allowed only after RSTO# returns High. If PORTime value is at default FFFFh, RSTO# returns High at the end of $t_{VCS}$.*

**Table 7.2** Power On Reset Parameters

| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| Vcc | 1.8V Vcc Power Supply | 1.7 | 1.95 | V |
| Vcc | 3V Vcc Power Supply | 2.7 | 3.6 | V |
| HyperFlash $t_{VCS}$ | $V_{CC}$ and $V_{CCQ}$ >= minimum to first access | - | 300 | µs |
| HyperRAM $t_{VCS}$ | $V_{CC}$ and $V_{CCQ}$ >= minimum to first access | - | 150 | µs |
| HyperFlash $t_{RPH}$ | RESET# Low to CS# Low | 30 | - | µs |
| HyperRAM $t_{RPH}$ | RESET# Low to CS# Low | 150 | - | µs |
| $t_{RP}$ | RESET# Pulse Width | 200 | - | ns |
| HyperFlash $t_{RH}$ | Time between RESET# (High) and CS# (Low) | 150 | - | ns |
| HyperRAM $t_{RH}$ | Time between RESET# (High) and CS# (Low) | 140 | - | µs |
| $t_{CSHI}$ | Chip Select High Between Operations | 10 | - | ns |

Refer to the individual device data sheets for additional information.

## 7.5 Power Down

HyperBus devices are considered to be powered-off when the core power supply ($V_{CC}$) drops below the $V_{CC}$ Lock-Out voltage ($V_{LKO}$). During a power supply transition down to the $V_{SS}$ level, $V_{CC}Q$ should remain less than or equal to $V_{CC}$. At the $V_{LKO}$ level, HyperRAM will have lost configuration or array data.

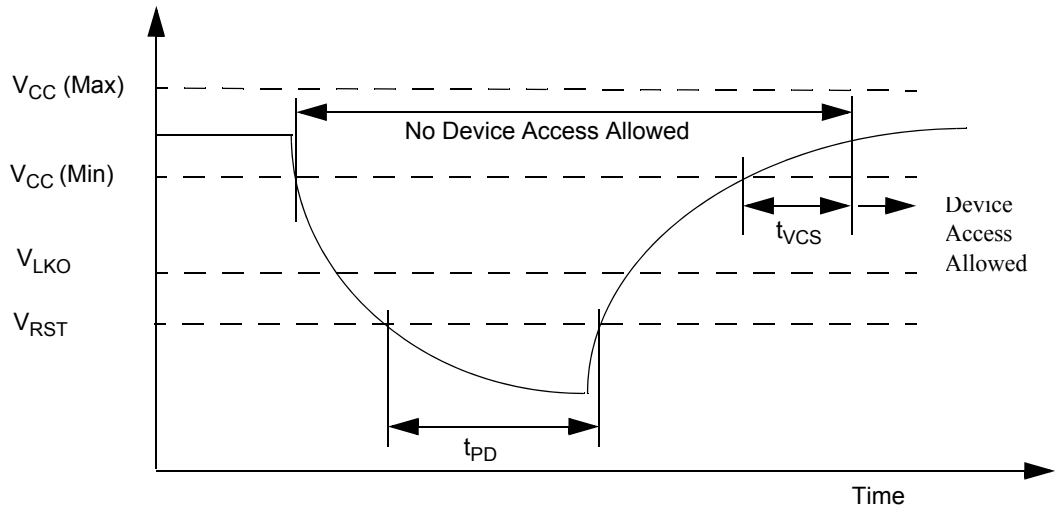
$V_{CC}$ must always be greater than or equal to $V_{CC}Q$ ($V_{CC} \geq V_{CC}Q$).

During Power-Down or voltage drops below $V_{LKO}$, the core power supply voltages must also drop below $V_{CC}$ Reset ($V_{RST}$) for a Power Down period ($t_{PD}$) for the part to initialize correctly when the power supply again rises to $V_{CC}$ minimum. See Figure 7.5, *Power Down or Voltage Drop* on page 21.

If during a voltage drop the $V_{CC}$ stays above $V_{LKO}$ the part will stay initialized and will work correctly when $V_{CC}$ is again above $V_{CC}$ minimum. If $V_{CC}$ does not go below and remain below $V_{RST}$ for greater than $t_{PD}$, then there is no assurance that the POR process will be performed. In this case, a hardware reset will be required ensure the HyperBus device is properly initialized.

The values for the power down parameters are device dependent, refer to the individual device data sheets for additional information.

**Figure 7.5** Power Down or Voltage Drop



## 7.6 DC Characteristics

**Table 7.3** DC Characteristics (CMOS Compatible)

| Parameter | Description | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $I_{LI}$ | Input Leakage Current 3V Device Reset Signal Only | $V_{IN} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | — | — | ±10.0 | µA |
| $I_{LI}$ | Input Leakage Current 1.8V Device Reset Signal Only | $V_{IN} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | — | — | ±5.0 | µA |
| $I_{CC1}$ | $V_{CC}$ Active Read Current | CS# = $V_{IL}$, @166 MHz, $V_{CC}$ = 1.9V | - | TBD | 60 | mA |
| | | CS# = $V_{IL}$, @100 MHz, $V_{CC}$ = 3.6V | - | 20 | 35 | mA |
| $I_{CC2}$ | $V_{CC}$ Active Write Current | CS# = $V_{IL}$, @166 MHz, $V_{CC}$ = 1.9V | - | TBD | 60 | mA |
| | | CS# = $V_{IL}$, @100 MHz, $V_{CC}$ = 3.6V | - | 15 | 35 | mA |
| $I_{CC4I}$ | $V_{CC}$ Standby Current for Industrial (-40C to +85C) | CS#, $V_{CC} = V_{CC}$ max, | - | 135 | 200 | µA |
| $I_{CC4IP}$ | $V_{CC}$ Standby Current for Extended (-40C to +105C) | CS#, $V_{CC} = V_{CC}$ max | - | 135 | 300 | µA |
| $I_{CC5}$ | Reset Current | CS# = $V_{IH}$,, RESET# = $V_{SS}$ +/- 0.3V, $V_{CC} = V_{CC}$ max | - | 10 | 20 | mA |
| $I_{CC6I}$ | Active Clock Stop Current for Industrial (-40C to +85C) | CS# = $V_{IL}$,, RESET# = $V_{CC}$ +/- 0.3V, $V_{CC} = V_{CC}$ max | - | 5.3 | 8 | mA |
| $I_{CC6IP}$ | Active Clock Stop Current for Extended(-40C to +105C) | CS# = $V_{IL}$,, RESET# = $V_{CC}$ +/- 0.3V, $V_{CC} = V_{CC}$ max | - | 5.3 | 12 | mA |
| $I_{CC7}$ | $V_{CC}$ Current during power up | CS#, = H, $V_{CC} = V_{CC}$ max, $V_{CC}$=$V_{CC}$Q= *1.95V or 3.6V* (Note 7.6.1) | — | — | 35 | mA |
| $I_{DPD}$ | Deep Power Down Current 3V | CS#, $V_{CC}$ = 3.6V | — | — | 20 | µA |
| $I_{DPD}$ | Deep Power Down Current 1.8V | CS#, $V_{CC}$ = 1.9V | — | — | 10 | µA |

***Note:***
*1. Not 100% tested.*

### 7.6.1 Capacitance Characteristics

**Table 7.4** 1.8V Capacitive Characteristics

| Description | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| Input Capacitance (CK, CK#, CS#) | CI | 3 | 4.5 | pF |
| Delta Input Capacitance (CK, CK#) | CID | - | 0.25 | pF |
| Output Capacitance (RWDS) | CO | 3 | 4 | pF |
| IO Capacitance (DQx) | CIO | 3 | 4 | pF |
| IO Capacitance Delta (DQx) | CIOD | - | 0.5 | pF |

***Notes***

1. *These values are guaranteed by design and are tested on a sample basis only.*
2. *Contact capacitance is measured according to JEP147 procedure for measuring capacitance using a vector network analyzer. $V_{CC,}$ $V_{CC}Q$ are applied and all other signals (except the signal under test) floating. DQ's should be in the high impedance state.*
3. *Note that the capacitance values for the CK, CK#, RWDS and DQx signals must have similar capacitance values to allow for signal propagation time matching in the system. The capacitance value for CS# is not as critical because there are no critical timings between CS# going active (Low) and data being presented on the DQs bus.*

**Table 7.5** 3.0V Capacitive Characteristics

| Description | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| Input Capacitance (CK, CS#) | CI | 3 | 4.5 | pF |
| Output Capacitance (RWDS) | CO | 3 | 4 | pF |
| IO Capacitance (DQx) | CIO | 3 | 4 | pF |
| IO Capacitance Delta (DQx) | CIOD | - | 0.5 | pF |

***Notes:***

1. *These values are guaranteed by design and are tested on a sample basis only.*
2. *Contact capacitance is measured according to JEP147 procedure for measuring capacitance using a vector network analyzer. $V_{CC,}$ $V_{CC}Q$ are applied and all other signals (except the signal under test) floating. DQ's should be in the high impedance state.*
3. *Note that the capacitance values for the CK, RWDS and DQx signals must have similar capacitance values to allow for signal propagation time matching in the system. The capacitance value for CS# is not as critical because there are no critical timings between CS# going active (Low) and data being presented on the DQs bus.*

## 7.7 Power-Up Initialization

HyperRAM™ Family products include an on-chip voltage sensor used to launch the power-up initialization process. $V_{CC}$ and $V_{CC}Q$ must be applied simultaneously. When the power supply reaches a stable level at or above $V_{CC}$(min), the device will require $t_{VCS}$ time to complete its self-initialization process.

The device must not be selected during power-up. CS# must follow the voltage applied on $V_{CC}Q$ until $V_{CC}$ (min) is reached during power-up, and then CS# must remain high for a further delay of $t_{VCS}$. A simple pull-up resistor from $V_{CC}Q$ to Chip Select (CS#) can be used to insure safe and proper power-up.

If RESET# is Low during power up, the device delays start of the $t_{VCS}$ period until RESET# is High. The $t_{VCS}$ period is used primarily to perform refresh operations on the DRAM array to initialize it.

When initialization is complete, the device is ready for normal operation.

**Figure 7.6** Power-up with RESET# High

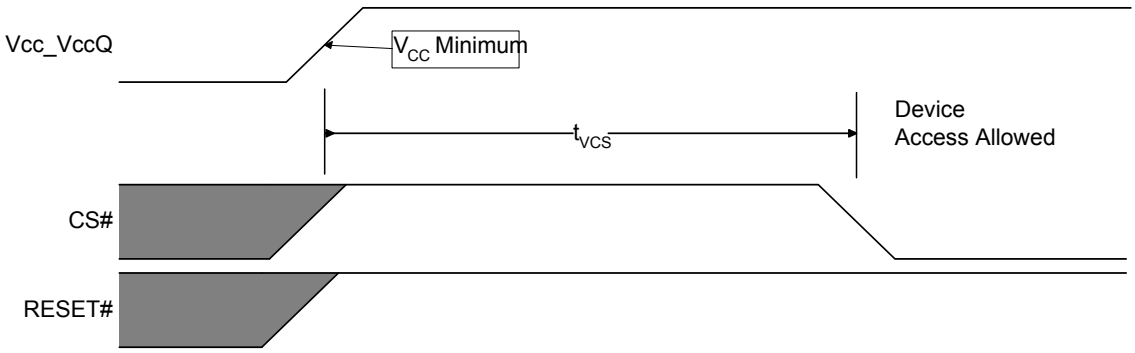



**Figure 7.7** Power-up with RESET# Low

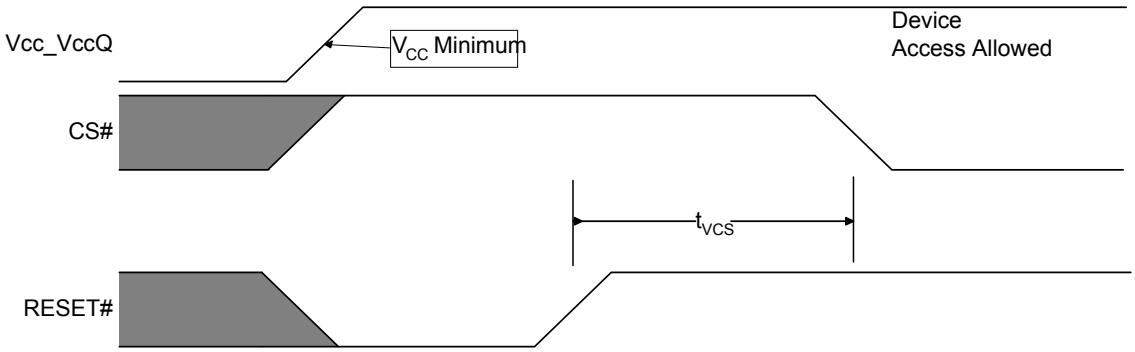



**Table 7.6** Power Up and Reset Parameters

| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| $V_{CC}$ | 1.8V $V_{CC}$ Power Supply | 1.7 | 1.95 | V |
| $V_{CC}$ | 3V $V_{CC}$ Power Supply | 2.7 | 3.6 | V |
| $t_{VCS}$ | $V_{CC}$ and $V_{CC}Q$ >= minimum and RESET# High to first access | - | 150 | µs |

***Notes:***

1. *Bus transactions (read and write) are not allowed during the power-up reset time ($t_{VCS}$).*
2. *$V_{CC}Q$ must be the same voltage as $V_{CC}$.*
3. *$V_{CC}$ ramp rate may be non-linear.*

## 7.8 Power Down

For the general description of the HyperBus interface power down specifications refer to the HyperBus Specification. The following section describes HyperRAM device dependent aspects of power down specifications.

**Table 7.7** 1.8V Power-Down Voltage and Timing

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $V_{CC}$ | $V_{CC}$ Power Supply | 1.7 | 1.95 | V |
| $V_{LKO}$ | $V_{CC}$ Lock-out below which re-initialization is required | 1.7 | – | V |
| $V_{RST}$ | $V_{CC}$ Low Voltage needed to ensure initialization will occur | 1.2 | – | V |
| $t_{PD}$ | Duration of $V_{CC} \le V_{RST}$ | 10 | – | µs |

***Note:***
*1. $V_{CC}$ ramp rate can be non-linear.*

**Table 7.8** 3.0V Power-Down Voltage and Timing

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $V_{CC}$ | $V_{CC}$ Power Supply | 2.7 | 3.6 | V |
| $V_{LKO}$ | $V_{CC}$ Lock-out below which re-initialization is required | 2.7 | – | V |
| $V_{RST}$ | $V_{CC}$ Low Voltage needed to ensure initialization will occur | 1.2 | – | V |
| $t_{PD}$ | Duration of $V_{CC} \le V_{RST}$ | 10 | – | µs |

***Note:***
*1. $V_{CC}$ ramp rate can be non-linear.*

## 7.9 Hardware Reset

The RESET# input provides a hardware method of returning the device to the standby state.

During $t_{RPH}$ the device will draw $I_{CC5}$ current. If RESET# continues to be held Low beyond $t_{RPH}$, the device draws CMOS standby current ($I_{CC4}$). While RESET# is Low (during $t_{RP}$), and during $t_{RPH}$, bus transactions are not allowed.

A hardware reset will:

- cause the configuration registers to return to their default values,
- halt self-refresh operation while RESET# is low,
- and force the device to exit the Deep Power Down state.

After RESET# returns High, the self-refresh operation will resume. Because self-refresh operation is stopped during RESET# Low, and the self-refresh row counter is reset to its default value, some rows may not be refreshed within the required array refresh interval per Table 5.6, *Array Refresh Interval per Temperature* on page 13. This may result in the loss of DRAM array data during or immediately following a hardware reset. The host system should assume DRAM array data is lost after a hardware reset and reload any required data.

**Figure 7.8** Hardware Reset Timing Diagram

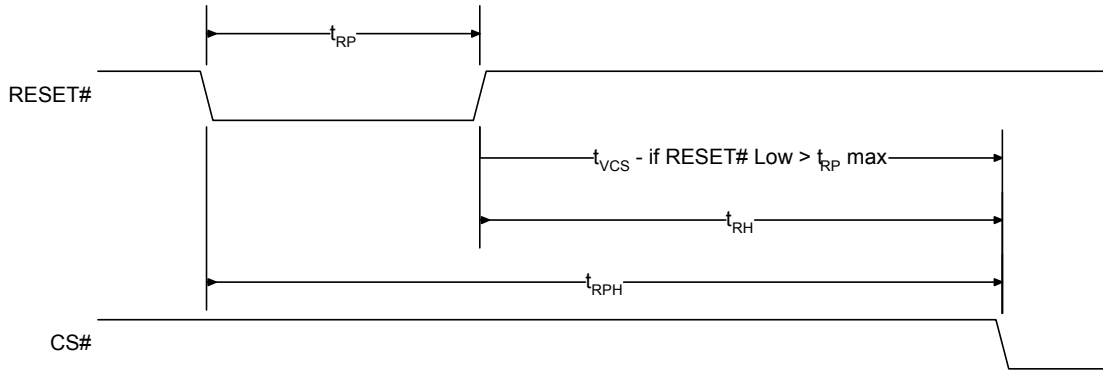


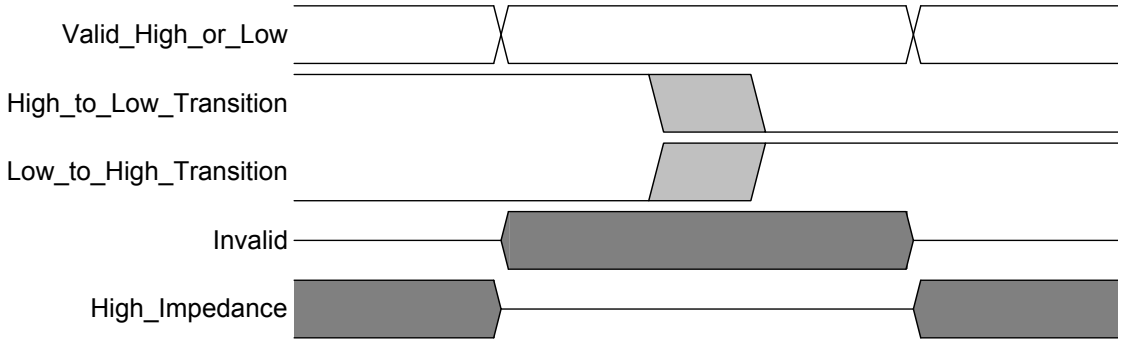

**Table 7.9** Power Up and Reset Parameters

| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| $t_{RP}$ | RESET# Pulse Width | 200 | — | ns |
| $t_{RH}$ | Time between RESET# (high) and CS# (low) | 200 | — | ns |
| $t_{RPH}$ | RESET# Low to CS# Low | 400 | — | ns |

# 8. Timing Specifications

For the general description of the HyperBus interface timing specifications refer to the HyperBus Specification. The following section describes HyperRAM device dependent aspects of timing specifications.

## 8.1 Key to Switching Waveforms




## 8.2 AC Test Conditions

**Figure 8.1** Test Setup

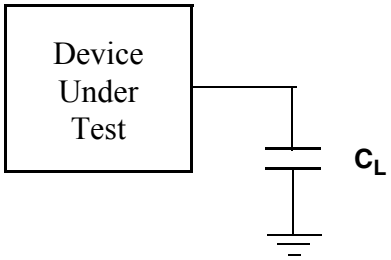


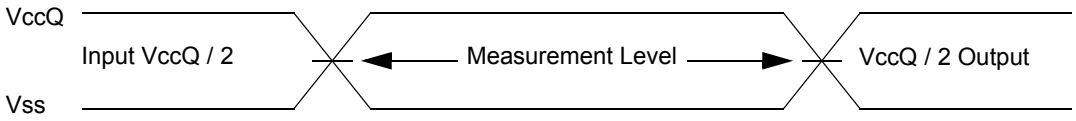

**Table 8.1** Test Specification

| Parameter | All Speeds | Units |
|---|---|---|
| Output Load Capacitance, $C_L$ | 20 | pF |
| Minimum Input Rise and Fall Slew Rates (Note 1) | 2.0 | V/ns |
| Input Pulse Levels | 0.0-$V_{CCQ}$ | V |
| Input timing measurement reference levels | $V_{CCQ}$/2 | V |
| Output timing measurement reference levels | $V_{CCQ}$/2 | V |

***Note:***

1. *All AC timings assume an input slew rate of 2V/ns. CK/CK# differential slew rate of at least 4V/ns.*
2. *Input and output timing is referenced to $V_{CC}Q/2$ or to the crossing of CK/CK#.*

**Figure 8.2** Input Waveforms and Measurement Levels




***Note:***
1. *Input timings for the differential CK/CK# pair are measured from clock crossings.*

## 8.3 AC Characteristics

### 8.3.1 CLK Characteristics

**Figure 8.3** Clock Characteristics

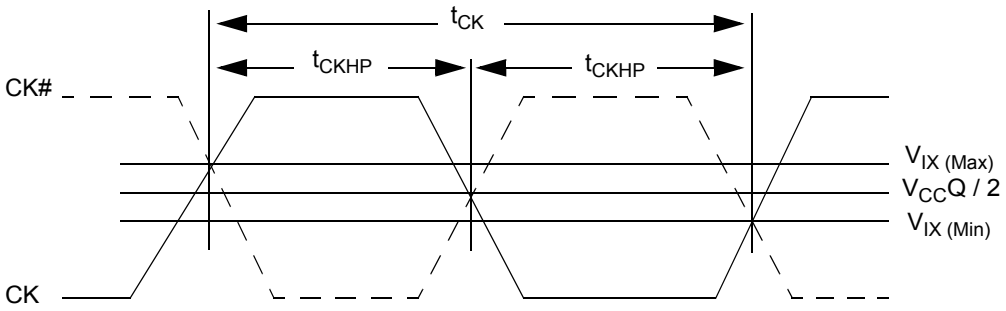



**Table 8.2** Clock Timings

| Parameter | Symbol | 166 MHz | | 133 MHz | | 100 MHz | | 50 MHz (2) | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | Min | Max | |
| CK Period | $t_{CK}$ | 6 | – | 7.5 | – | 10 | – | 20 | – | ns |
| CK Half Period - Duty Cycle | $t_{CKHP}$ | 0.45 | 0.55 | 0.45 | 0.55 | 0.45 | 0.55 | 0.45 | 0.55 | $t_{CK}$ |
| CK Half Period at Frequency<br>Min = 0.45 $t_{CK}$ Min<br>Max = 0.55 $t_{CK}$ Min | $t_{CKHP}$ | 2.7 | 3.3 | 3.375 | 4.125 | 4.5 | 5.5 | 9 | 11 | ns |

***Notes:***
1. *Clock jitter of ±5% is permitted.*
2. *50 MHz timings are only relevant when a burst write is used to load data during a HyperFlash Word Program command.*
3. *CK# is only used on the 1.8V device and is shown as a dashed waveform.*
4. *The 3V device uses a single ended clock input.*

**Table 8.3** Clock AC/DC Electrical Characteristics

| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| DC Input Voltage | $V_{IN}$ | -0.3 | $V_{CC}Q$+ 0.3 | V |
| DC Input Differential Voltage | $V_{ID(DC)}$ | $V_{CC}Q$x 0.4 | $V_{CC}Q$+ 0.6 | V |
| AC Input Differential Voltage | $V_{ID(AC)}$ | $V_{CC}Q$x 0.6 | $V_{CC}Q$ + 0.6 | V |
| AC Differential Crossing Voltage | $V_{IX}$ | $V_{CC}Q$ x 0.4 | $V_{CC}Q$ x 0.6 | V |

***Notes***
1. *CK and CK# input slew rate must be >=1V/ns (2V/ns if measured differentially).*
2. *$V_{ID}$ is the magnitude of the difference between the input level on CK and the input level on CK#.*
3. *The value of VIX is expected to equal $V_{CC}Q$/2 of the transmitting device and must track variations in the DC level of $V_{CC}Q$.*

### 8.3.2 Read Transaction Diagrams

**Figure 8.4** Read Timing Diagram - No Additional Latency Required

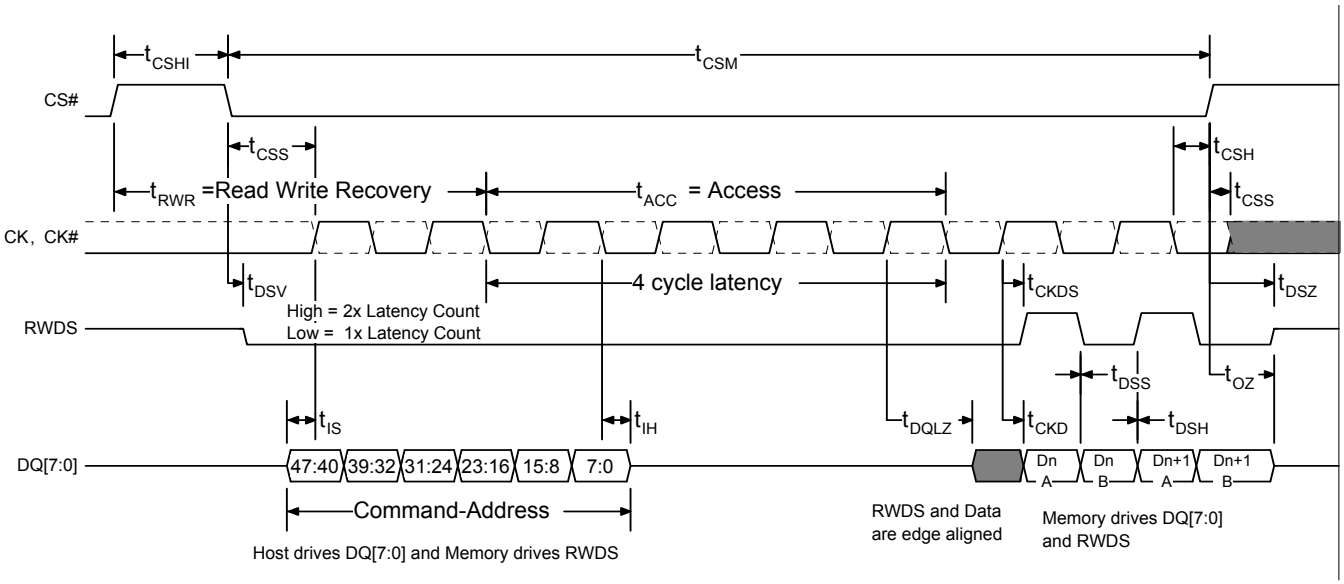
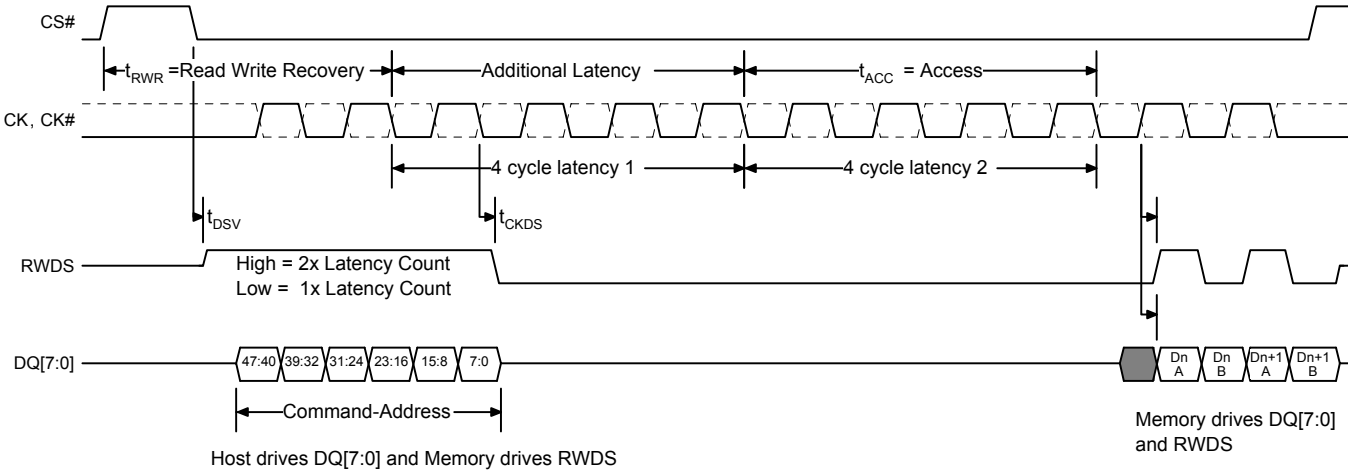


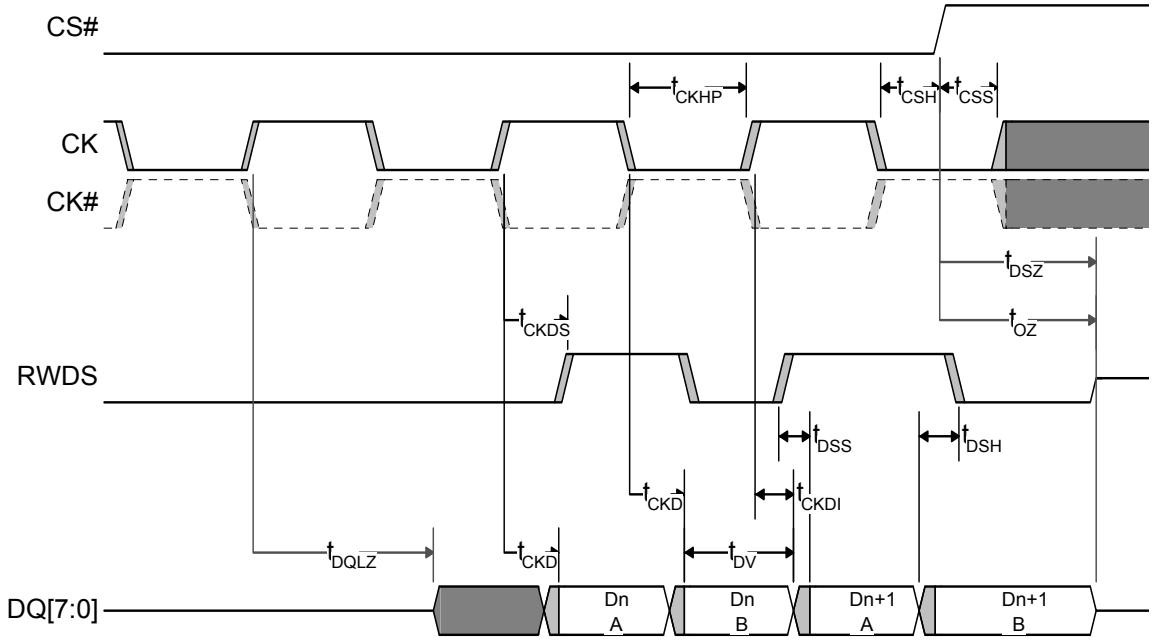

**Figure 8.5** Read Timing Diagram - With Additional Latency




***Notes:***

1. *Transactions must be initiated with CK = Low and CK# = High.*
2. *CS# must return High before a new transaction is initiated.*
3. *The memory drives RWDS during the entire Read transaction.*
4. *During read transactions RWDS is used as an additional latency indicator initially and is then used as a data strobe during data transfer.*
5. *Transactions without additional latency count have RWDS Low during CA cycles. Transactions with additional latency count have RWDS High during CA cycles and RWDS returns low at $t_{DSH}$. All other timing relationships are the same for both figures although they are not shown in the second figure. A four cycle latency is used for illustration purposes only. The required latency count is device and clock frequency dependent.*
6. *HyperFlash does not require $t_{RWR}$ or $t_{CMS}$. These parameters are required by HyperRAM.*
7. *CK# is only used on the 1.8V device and is shown as a dashed waveform.*
8. *The 3V device uses a single ended CK clock input.*

**Figure 8.6** Data Valid Timing




RWDS and Data are edge aligned and driven by the memory

***Notes:***

1. *This figure shows a closer view of the data transfer portion of read transaction diagrams in order to more clearly show the Data Valid period as affected by clock jitter and clock to output delay uncertainty.*
2. *CK# is only used on the 1.8V device. The 3V device uses a single ended CK input.*
3. *The $t_{CKD}$, and $t_{CKDI}$ timing parameters define the beginning and end position of the data valid period.*
4. *The $t_{DSS}$, and $t_{DSH}$ timing parameters define how early or late RWDS may transition relative to the transition of data. This is the potential skew between the clock to data delay $t_{CKD}$, and clock to data strobe delay $t_{CKDS}$. Aside from this skew, the $t_{CKD}$, $t_{CKDI}$, and $t_{CKDS}$. values track together (vary by the same ratio) because RWDS and Data are outputs from the same device under the same voltage and temperature conditions.*

### 8.3.3 Read AC Parameters

**Table 8.4** HyperRAM 1.8V Device Common Read Timing Parameters

| Parameter | Symbol | 166MHz | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| Chip Select High Between Transactions | $t_{CSHI}$ | 6 | – | ns |
| Read-Write Recovery Time | $t_{RWR}$ | 36 | - | ns |
| Chip Select Setup to next CK Rising Edge | $t_{CSS}$ | 3 | – | ns |
| Data Strobe Valid | $t_{DSV}$ | – | 8 | ns |
| Input Setup | $t_{IS}$ | 0.6 | – | ns |
| Input Hold | $t_{IH}$ | 0.6 | – | ns |
| Read Initial Access Time | $t_{ACC}$ | – | 36 | ns |
| Refresh Time | $t_{RFH}$ | 36 | — | ns |
| Clock to DQs Low Z | $t_{DQLZ}$ | 0 | – | ns |
| CK transition to DQ Valid | $t_{CKD}$ | 1 | 5.5 | ns |
| CK transition to DQ Invalid | $t_{CKDI}$ | 0 | 4.6 | ns |
| Data Valid ($t_{DV}$ *min = the lessor of:* $t_{CKHP}$ *min -* $t_{CKD}$ *max +* $t_{CKDI}$ *max*) or ($t_{CKHP}$ *min -* $t_{CKD}$ *min +* $t_{CKDI}$ *min*) | $t_{DV}$ | 1.7 | – | ns |
| CK transition to RWDS valid | $t_{CKDS}$ | 1 | 5.5 | ns |
| RWDS transition to DQ Valid | $t_{DSS}$ | -0.45 | +0.45 | ns |
| RWDS transition to DQ Invalid | $t_{DSH}$ | -0.45 | +0.45 | ns |
| Chip Select Hold After CK Falling Edge | $t_{CSH}$ | 0 | – | ns |
| Chip Select Inactive to RWDS High-Z | $t_{DSZ}$ | – | 6 | ns |
| Chip Select Inactive to DQ High-Z | $t_{OZ}$ | – | 6 | ns |
| Chip Select Maximum Low Time - Industrial Temperature | $t_{CSM}$ | - | 4.0 | us |
| Chip Select Maximum Low Time - Industrial Plus Temperature | | - | 1.0 | us |

***Note:***

1. *A HyperBus device operates correctly with the $t_{CSH}$ value shown, however, CS# must generally remain driven Low (active) by the HyperBus master longer, so that data remains valid long enough to account for $t_{CKD}$, $t_{CKDS}$, and the master interface phase shifting of RWDS to capture the last data transfer from the DQ signals. The HyperBus master will need to drive CS# Low for one or more additional clock periods to ensure capture of valid data from the last desired data transfer.*

**Table 8.5** HyperRAM 3V Device Common Read Timing Parameters

| Parameter | Symbol | 100MHz | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| Chip Select High Between Transactions | $t_{CSHI}$ | 10.0 | – | ns |
| Read-Write Recovery Time | $t_{RWR}$ | 40 | - | ns |
| Chip Select Setup to next CK Rising Edge | $t_{CSS}$ | 3 | – | ns |
| Data Strobe Valid | $t_{DSV}$ | – | 8 | ns |
| Input Setup | $t_{IS}$ | 1.0 | – | ns |
| Input Hold | $t_{IH}$ | 1.0 | – | ns |
| Read Initial Access Time | $t_{ACC}$ | | 40 | ns |
| Refresh Time | $t_{RFH}$ | 40 | — | ns |
| Clock to DQs Low Z | $t_{DQLZ}$ | 0 | – | ns |
| CK transition to DQ Valid | $t_{CKD}$ | 1 | 7 | ns |
| CK transition to DQ Invalid | $t_{CKDI}$ | 0.5 | 5.2 | ns |
| Data Valid ($t_{DV}$ *min = the lessor of:* $t_{CKHP}$ *min -* $t_{CKD}$ *max +* $t_{CKDI}$ *max*) or ($t_{CKHP}$ *min -* $t_{CKD}$ *min +* $t_{CKDI}$ *min*) | $t_{DV}$ | 2.7 | | ns |
| CK transition to RWDS valid | $t_{CKDS}$ | 1 | 7 | ns |
| RWDS transition to DQ Valid | $t_{DSS}$ | -0.8 | +0.8 | ns |
| RWDS transition to DQ Invalid | $t_{DSH}$ | -0.8 | +0.8 | ns |
| Chip Select Hold After CK Falling Edge | $t_{CSH}$ | 0 | – | ns |
| Chip Select Inactive to RWDS High-Z | $t_{DSZ}$ | – | 7 | ns |
| Chip Select Inactive to DQ High-Z | $t_{OZ}$ | – | 7 | ns |
| Chip Select Maximum Low Time - Industrial Temperature | $t_{CSM}$ | - | 4.0 | us |
| Chip Select Maximum Low Time - Industrial Plus Temperature | | - | 1.0 | us |

***Note:***

1. *A HyperBus device operates correctly with the* $t_{CSH}$ *value shown, however, CS# must generally remain driven Low (active) by the HyperBus master longer, so that data remains valid long enough to account for* $t_{CKD}$, $t_{CKDS}$, *and the master interface phase shifting of RWDS to capture the last data transfer from the DQ signals. The HyperBus master will need to drive CS# Low for one or more additional clock periods to ensure capture of valid data from the last desired data transfer.*

### 8.3.4 Write Transaction Diagrams

**Figure 8.7** Write Timing Diagram - No Additional Latency

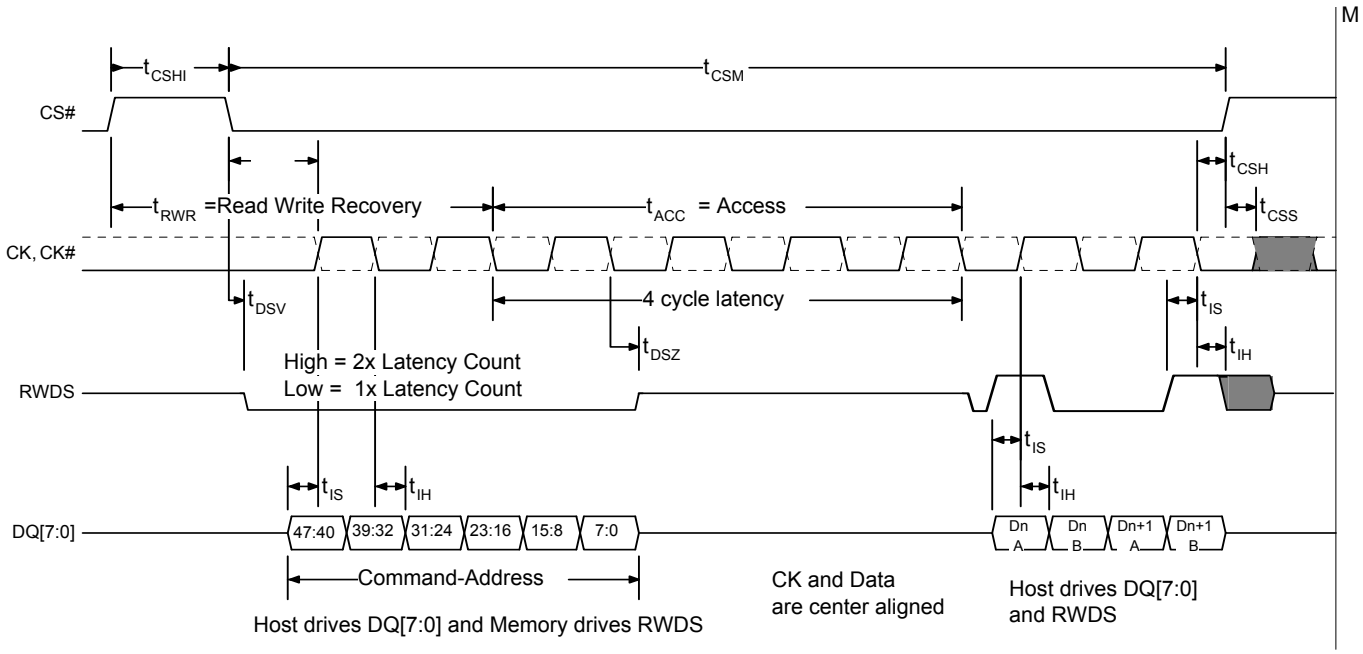
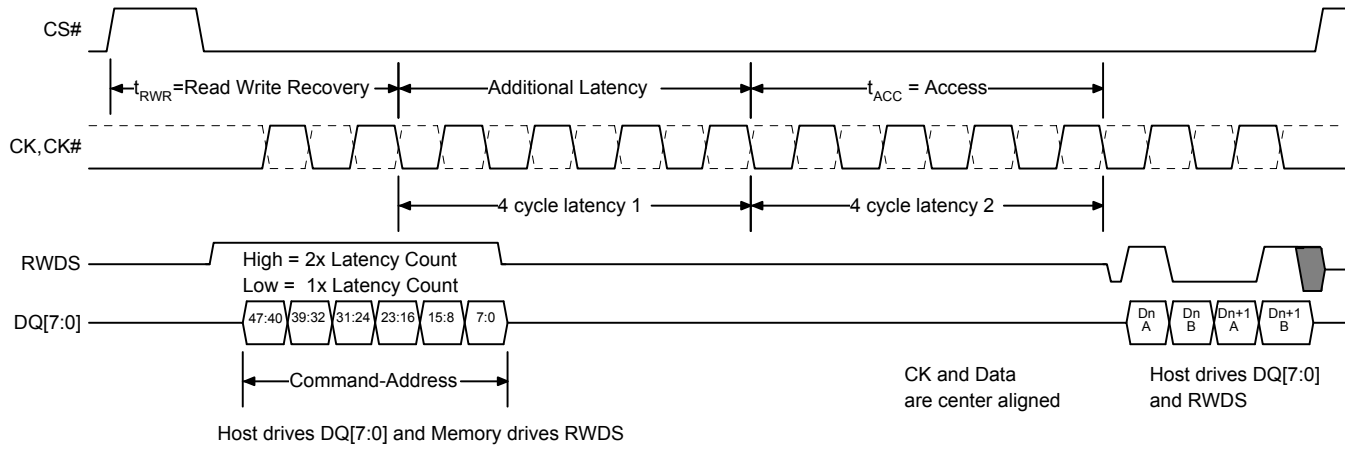


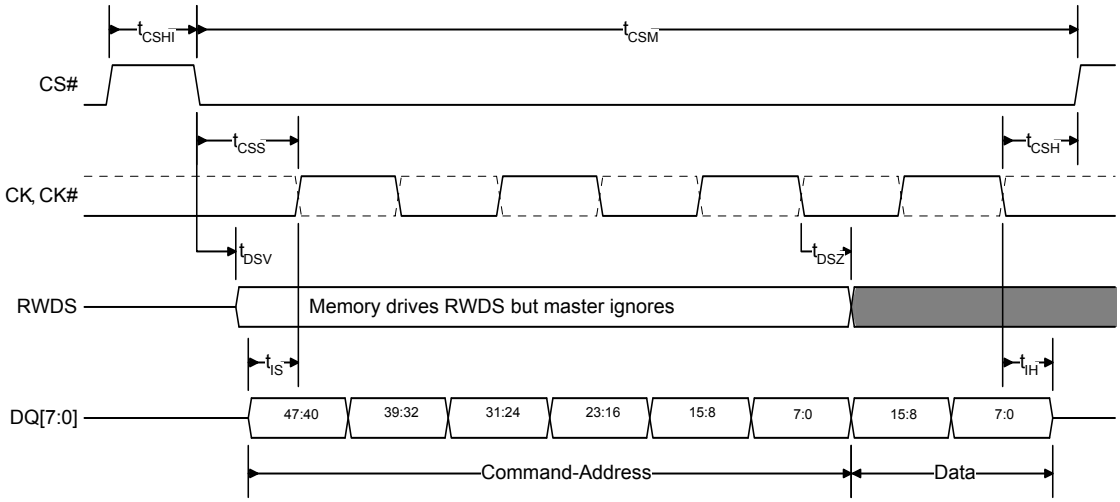

**Figure 8.8** Write Timing Diagram - With Additional Latency




***Notes:***

1. *Transactions must be initiated with CK=Low and CK#=High. CS# must return High before a new transaction is initiated.*
2. *During write transactions with latency, RWDS is used as an additional latency indicator initially and is then used as a data mask during data transfer.*
3. *Transactions without additional latency count have RWDS Low during CA cycles. Transactions with additional latency count have RWDS High during CA cycles and RWDS returns low at $t_{DSH}$. All other timing relationships are the same for both figures although they are not shown in the second figure. A four cycle latency is used for illustration purposes only. The required latency count is device and clock frequency dependent.*
4. *The write transaction shown demonstrates the A byte of the first word and the B byte of the second word being masked. Only Dn B and Dn+1 A is modified in the array. Dn A and Dn+1 B remain unchanged.*
5. *CK# is only used on the 1.8V device and is shown as a dashed waveform.*
6. *The 3V device uses a single ended CK clock input.*

**Figure 8.9** Write Operation without Initial Latency




Host drives DQ[7:0] with Command-Address and Write Data

***Notes:***

1. *Transactions must be initiated with CK=Low and CK#=High. CS# must return High before a new transaction is initiated.*
2. *Writes with zero initial latency, do not have a turn around period for RWDS. The slave device will always drive RWDS during the Command-Address period to indicate whether extended latency is required for a transaction that has initial latency. However, the RWDS is driven before the slave device has received the first byte of CA i.e. before the slave knows whether the transaction is a read or write, to memory space or register space. In the case of a write with zero latency, the RWDS state during the CA period does not affect the initial latency of zero. Since master write data immediately follows the Command-Address period in this case, the slave may continue to drive RWDS Low or may take RWDS to High-Z during write data transfer. The master must not drive RWDS during Writes with zero latency. Writes with zero latency do not use RWDS as a data mask function. All bytes of write data are written (full word writes).*
3. *CK# is only used on the 1.8V device and is shown as a dashed waveform.*
4. *The 3V device uses a single ended CK clock input.*

### 8.3.5 Write AC Parameters

**Table 8.6** HyperRAM 1.8V Device Common Write Timing Parameters

| Parameter | Symbol | 166MHz | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| Chip Select High Between Transactions | $t_{CSHI}$ | 6 | – | ns |
| Read-Write Recovery Time | $t_{RWR}$ | 36 | - | ns |
| Chip Select Setup to next CK Rising Edge | $t_{CSS}$ | 3 | – | ns |
| Data Strobe Valid | $t_{DSV}$ | – | 8 | ns |
| Input Setup | $t_{IS}$ | 0.6 | – | ns |
| Input Hold | $t_{IH}$ | 0.6 | – | ns |
| Memory Space Write Initial Access Time | $t_{ACC}$ | 36 | – | ns |
| Refresh Time | $t_{RFH}$ | 36 | – | ns |
| Chip Select Hold After CK Falling Edge | $t_{CSH}$ | 0 | – | ns |
| Chip Select Inactive or clock to RWDS High-Z | $t_{DSZ}$ | – | 6 | ns |
| HyperRAM Chip Select Maximum Low Time - Industrial Temperature | $t_{CSM}$ | - | 4.0 | us |
| HyperRAM Chip Select Maximum Low Time - Industrial Plus Temperature | | - | 1.0 | us |

***Note:***
*1. Sampled, not 100% tested.*

**Table 8.7** HyperRAM 3V Device Common Write Timing Parameters

| Parameter | Symbol | 100MHz | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| Chip Select High Between Transactions | $t_{CSHI}$ | 10.0 | – | ns |
| Read-Write Recovery Time | $t_{RWR}$ | 40 | - | ns |
| Chip Select Setup to next CK Rising Edge | $t_{CSS}$ | 3 | – | ns |
| Data Strobe Valid | $t_{DSV}$ | – | 8 | ns |
| Input Setup | $t_{IS}$ | 1.0 | – | ns |
| Input Hold | $t_{IH}$ | 1.0 | – | ns |
| Memory Space Write Initial Access Time | $t_{ACC}$ | 40 | – | ns |
| Refresh Time | $t_{RFH}$ | 40 | – | ns |
| Chip Select Hold After CK Falling Edge | $t_{CSH}$ | 0 | – | ns |
| Chip Select Inactive or clock to RWDS High-Z | $t_{DSZ}$ | – | 7 | ns |
| Chip Select Maximum Low Time - Industrial Temperature | $t_{CSM}$ | - | 4.0 | us |
| Chip Select Maximum Low Time - Industrial Plus Temperature | | - | 1.0 | us |

# HyperBus Specification

## 9. General Description

HyperBus products use the high performance HyperBus for connection between a host system master and one or more slave interfaces. HyperBus is used to connect microprocessor, microcontroller, or ASIC devices with random access NOR Flash memory, RAM, or peripheral devices.

HyperBus is an interface that draws upon the legacy features of both parallel and serial interface memories, while enhancing system performance, ease of design, and system cost reduction.

HyperBus has a low signal count, Double Data Rate (DDR) interface, that achieves high read and write throughput while reducing the number of device I/O connections and signal routing congestion in a system.

HyperBus memories provide the fast random access of a parallel interface, for code eXecution-in-Place (XiP) directly from flash memory, to reduce or eliminate duplication of memory space between non-volatile and volatile memories in the system. HyperBus devices transfer all control (command), address, and data information sequentially, one byte per clock edge, at high frequency, to minimize signal count, yet deliver 3 to 6 times the throughput of legacy parallel or serial interfaces.

All HyperBus inputs and outputs are LV-CMOS compatible. Operation at either 1.8V or 3.0V (nominal) I/O voltage supply ($V_{CC}Q$) is supported. Devices using 3.0V $V_{CC}Q$ use a single ended clock (CK) and those using 1.8V $V_{CC}Q$ use a differential clock pair (CK, CK#). Depending on the interface voltage selected, the signal count for HyperBus is 11 or 12 signals.

The DDR protocol transfers two data bytes per clock cycle on the DQ input/output (I/O) signals. A read or write transaction on HyperBus consists of a sequential series of 16-bit, one clock cycle, data transfers - via two corresponding 8-bit wide, one-half-clock-cycle data transfers, one on each single ended clock edge or differential clock crossing. Read and write transactions always transfer full 16-bit words of data. Read data words always contain two valid bytes. Write data words may have one or both bytes masked to prevent writing of individual bytes within a write burst.

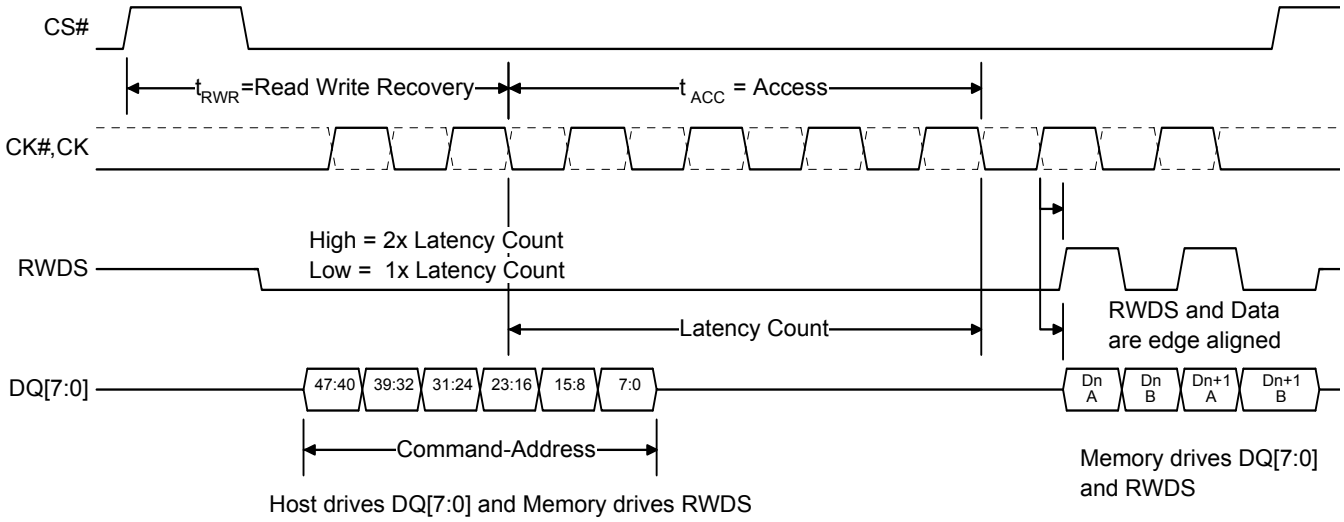
HyperBus has a single master (host system) interface and one or more slave interface devices.

Each slave device on the bus is selected during a transaction by an individual, low active, Chip Select (CS#) from the master.

Command, address, and data information is transferred over the eight HyperBus DQ[7:0] signals. The clock is used for information capture by a HyperBus slave device when receiving command, address, or data on the DQ signals. Command or Address values are center aligned with clock transitions.

Every transaction begins with the assertion of CS# and Command-Address (CA) signals, followed by the start of clock transitions to transfer six CA bytes, followed by initial access latency and either read or write data transfers, until CS# is deasserted.

**Figure 9.1** Read Transaction, Single Initial Latency Count



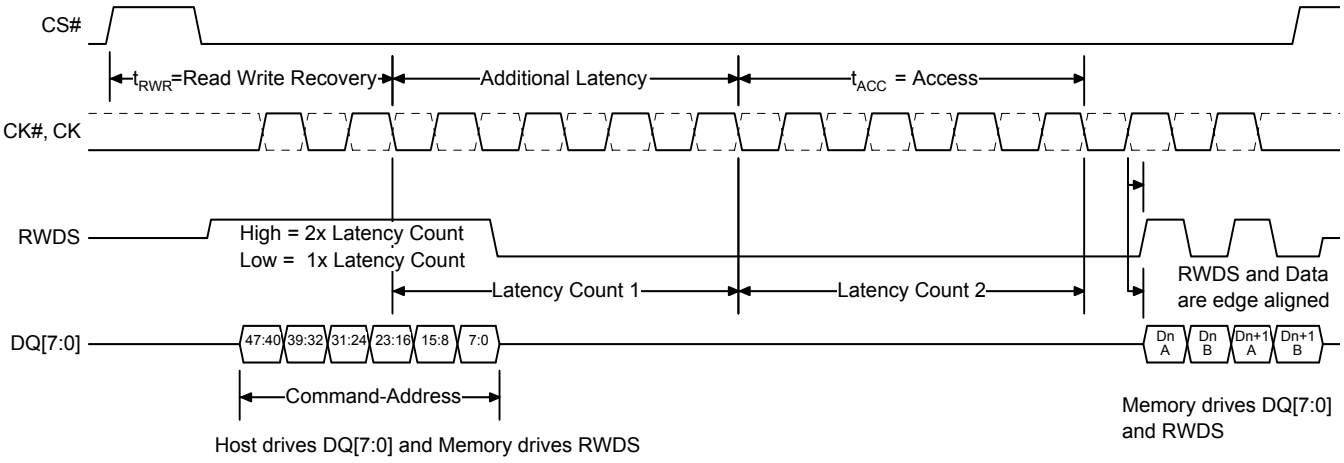

The Read/Write Data Strobe (RWDS) is a bidirectional signal that indicates:

- when data will start to transfer from a slave to the master device in read transactions (initial read latency)
- when data is being transferred from a slave to the master during read transactions (as a source synchronous read data strobe)
- when data may start to transfer from the master to a slave in write transactions (initial write latency)
- data masking during write data transfers

During the CA transfer portion of a read or write transaction, RWDS acts as an output from a slave to indicate whether additional initial access latency is needed in the transaction.

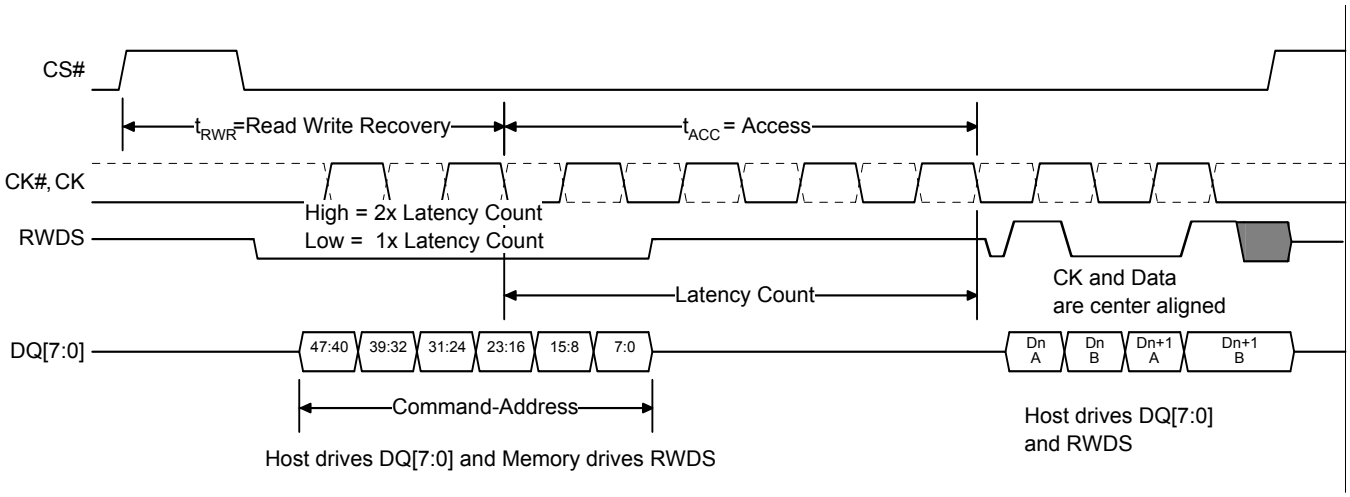
During read data transfers, RWDS is a read data strobe with data values edge aligned with the transitions of RWDS.

**Figure 9.2** Read Transaction, Additional Latency Count



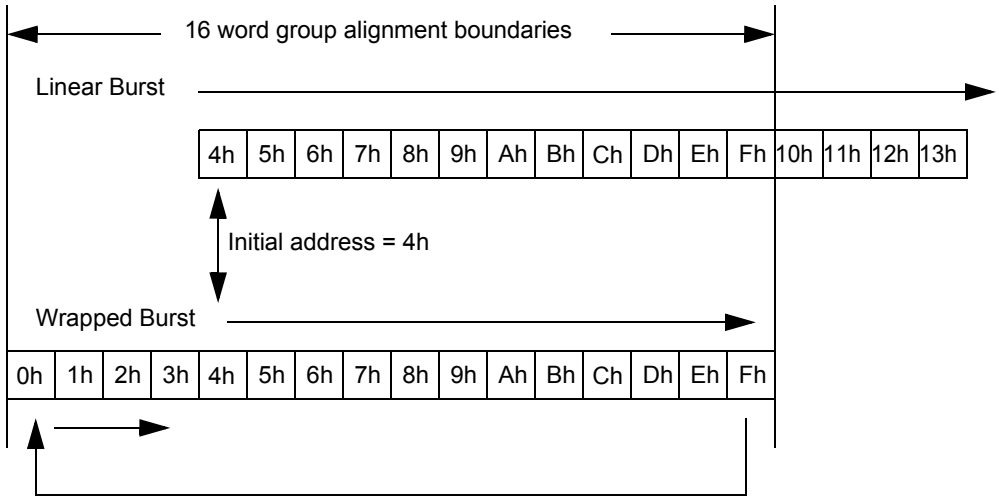
During write data transfers, RWDS indicates whether each data byte transfer is masked with RWDS High (invalid and prevented from changing the byte location in a memory) or not masked with RWDS Low (valid and written to a memory). Data masking may be used by the host to byte align write data within a memory or to enable merging of multiple non-word aligned writes in a single burst write. During write transactions, data is center aligned with clock transitions.

**Figure 9.3** Write Transaction, Single Initial Latency Count




Read and write transactions are burst oriented, transferring the next sequential word during each clock cycle. Each individual read or write transaction can use either a wrapped or linear burst sequence.

**Figure 9.4** Linear Versus Wrapped Burst Sequence




During wrapped transactions, accesses start at a selected location and continue to the end of a configured word group aligned boundary, then wrap to the beginning location in the group, then continue back to the starting location. Wrapped bursts are generally used for critical word first cache line fill read transactions. During linear transactions, accesses start at a selected location and continue in a sequential manner until the transaction is terminated when CS# returns High. Linear transactions are generally used for large contiguous data transfers such as graphic images. Since each transaction command selects the type of burst sequence for that transaction, wrapped and linear bursts transactions can be dynamically intermixed as needed.

# 10. HyperBus Signal Descriptions

## 10.1 Input/Output Summary

HyperBus has a mandatory set of signals as well as optional signals that are supported by some but not all HyperBus compatible devices. Active Low signal names have a hash symbol (#) suffix.

**Table 10.1** Mandatory I/O Summary

| Symbol | Type | Description |
|---|---|---|
| CS# | Master Output, Slave Input | **Chip Select -** Bus transactions are initiated with a High to Low transition. Bus transactions are terminated with a Low to High transition. The master device has a separate CS# for each slave. |
| CK, CK# | Master Output, Slave Input | **Differential Clock -** Command, address, and data information is output with respect to the crossing of the CK and CK# signals. Differential clock is used on 1.8V I/O devices.<br>**Single Ended Clock -** CK# is not used on 3.0V devices, only a single ended CK is used.<br>The clock is not required to be free-running. |
| DQ[7:0] | Input/Output | **Data Input/Output -** Command, Address, and Data information is transferred on these signals during Read and Write transactions. |
| RWDS | Input/Output | **Read Write Data Strobe -** During the Command/Address portion of all bus transactions RWDS is a slave output and indicates whether additional initial latency is required. Slave output during read data transfer, data is edge aligned with RWDS. Slave input during data transfer in write transactions to function as a data mask. (High = additional latency, Low = no additional latency). |

**Table 10.2** Optional I/O Summary

| Symbol | Type | Description |
|---|---|---|
| RESET# | Master Output, Slave Input, Internal Pull-up | **Hardware RESET** - When Low the slave device will self initialize and return to the Standby state. RWDS and DQ[7:0] are placed into the High-Z state when RESET# is Low. The slave RESET# input includes a weak pull-up, if RESET# is left unconnected it will be pulled up to the High state. |
| RSTO# | Master Input, Slave Output, Open Drain | **RESET Output.** RSTO# is an open-drain output used to indicate when a Power-On-Reset (POR) is occurring within the slave device and can be used as a system reset signal. Upon completion of the internal POR, a user defined timeout period can extend the RSTO# Low time. Following the POR and user defined extension time, RSTO# signal will cease driving Low so the master or external resistance can pull the signal to High . There is no internal pull-up required on this signal, however, some slave devices may provide an internal pull-up as a device specific feature. |
| INT# | Master Input, Slave Output, Open Drain | **Interrupt Output.** When INT# is Low the slave device is indicating that an internal event has occurred. This signal is intended to be used as a system interrupt from the slave device to indicate that an on-chip event has occurred. INT# is an open-drain output. There is no internal pull-up required on this signal, however, some slave devices may provide an internal pull-up as a device specific feature. |

# 11. HyperBus Protocol

All bus transactions can be classified as either read or write. A bus transaction is started with CS# going Low with clock in idle state (CK=Low and CK#=High). The first three clock cycles transfer three words of Command/Address (CA0, CA1, CA2) information to define the transaction characteristics. The Command/ Address words are presented with DDR timing, using the first six clock edges. The following characteristics are defined by the Command/Address information:

- Read or Write transaction
- Address Space: memory array space or register space
  - Register space is used to access Device Identification (ID) registers and Configuration Registers (CR) that identify the device characteristics and determine the slave specific behavior of read and write transfers on the HyperBus interface.
- Whether a transaction will use a linear or wrapped burst sequence.
- The target row (and half-page) address (upper order address)
- The target column (word within half-page) address (lower order address)

During the Command/Address (CA) portion of all transactions, RWDS is used by the memory to indicate whether additional initial access latency will be inserted.

Following the Command/Address information, a number of clock cycles without data transfer may be used to satisfy any initial latency requirements before data is transferred. The number of clock cycles inserted is defined by a slave configuration register (latency count) value. If RWDS was High during the CA period an additional latency count is inserted.

Some slave devices may require write transactions with zero latency between the CA cycles and following write data transfers. Writes with zero initial latency, do not have a turn around period for RWDS. The slave device will always drive RWDS during the Command-Address period to indicate whether extended latency is required for a transaction that has initial latency. However, the RWDS is driven before the slave device has received the first byte of CA i.e. before the slave knows whether the transaction is a read or write, to memory space or register space. In the case of a write with zero latency, the RWDS state during the CA period does not affect the initial latency of zero. Since master write data immediately follows the Command-Address period in this case, the slave may continue to drive RWDS Low or may take RWDS to High-Z during write data transfer. The master must not drive RWDS during Writes with zero latency. Writes with zero latency do not use RWDS as a data mask function. All bytes of write data are written (full word writes). Writes without initial latency are generally used for register space writes but the requirement for writes with zero latency is slave device dependent. Writes with zero latency may be required for memory space or register space or neither, depending on the slave device capability.

When data transfer begins, read data is edge aligned with RWDS transitions or write data is center aligned with single-ended clock edges or differential clock CK/CK# crossings. During read data transfer, RWDS serves as a source synchronous read data timing strobe. During write data transfer, clock edges or crossings provide the data timing reference and RWDS is used as a data mask. When RWDS is low during a write data transfer, the data byte is written into memory; if RWDS is high during the transfer the byte is not written.

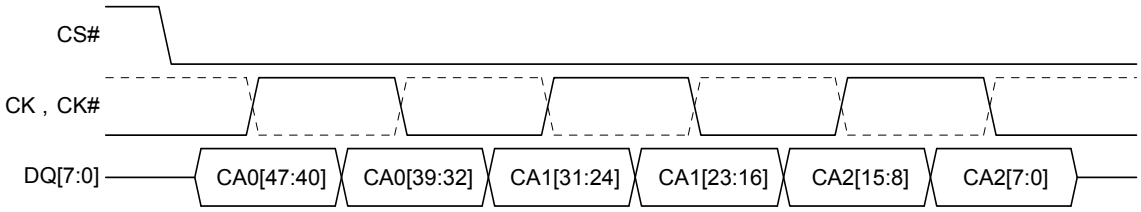
Data is always transferred as 16 bit values with the first eight bits (byte A) transferred on a High going CK (write data) or following a RWDS rising edge (read data) and the second eight bits (byte B) is transferred on the Low going CK edge or following the falling RWDS edge.

Once the target data has been transferred, the host completes the transaction by driving CS# High with clock idle. Data transfers can be ended at any time by bringing CS# High when clock is idle.

The clock is not required to be free-running. The clock may be idle while CS# is High or may stop in the idle state while CS# is Low (this is called Active Clock Stop). Support for Active Clock Stop is slave device dependent. It is an optional HyperBus device feature.

## 11.1 Command/Address Bit Assignments

**Figure 11.1** Command-Address Sequence




***Notes:***

1. *Figure shows the initial three clock cycles of all transactions on the HyperBus.*
2. *CK# of differential clock is shown as dashed line waveform.*
3. *Command-Address information is "center aligned" with the clock during both Read and Write transactions.*

**Table 11.1** Command-Address Bit Assignment to DQ Signals

| Signal | CA0[47:40] | CA0[39:32] | CA1[31:24] | CA1[23:16] | CA2[15:8] | CA2[7:0] |
|---|---|---|---|---|---|---|
| **DQ[7]** | CA[47] | CA[39] | CA[31] | CA[23] | CA[15] | CA[7] |
| **DQ[6]** | CA[46] | CA[38] | CA[30] | CA[22] | CA[14] | CA[6] |
| **DQ[5]** | CA[45] | CA[37] | CA[29] | CA[21] | CA[13] | CA[5] |
| **DQ[4]** | CA[44] | CA[36] | CA[28] | CA[20] | CA[12] | CA[4] |
| **DQ[3]** | CA[43] | CA[35] | CA[27] | CA[19] | CA[11] | CA[3] |
| **DQ[2]** | CA[42] | CA[34] | CA[26] | CA[18] | CA[10] | CA[2] |
| **DQ[1]** | CA[41] | CA[33] | CA[25] | CA[17] | CA[9] | CA[1] |
| **DQ[0]** | CA[40] | CA[32] | CA[24] | CA[16] | CA[8] | CA[0] |

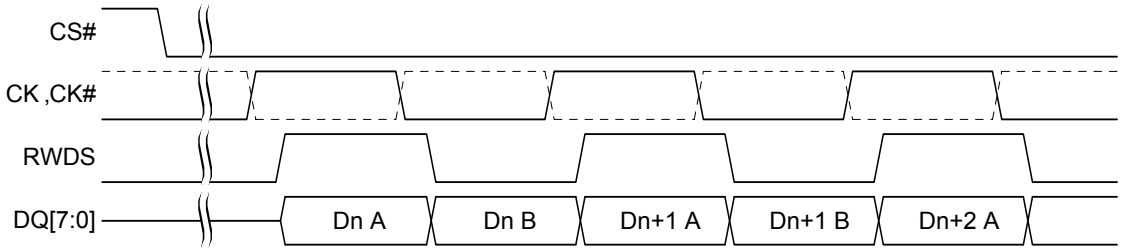
**Table 11.2** Command/Address Bit Assignments

| CA Bit# | Bit Name | Bit Function |
|---|---|---|
| 47 | R/W# | Identifies the transaction as a read or write.<br>R/W#=1 indicates a Read transaction<br>R/W#=0 indicates a Write transaction |
| 46 | Address Space (AS) | Indicates whether the read or write transaction accesses the memory or register space.<br>AS=0 indicates memory space<br>AS=1 indicates the register space<br>The register space is used to access device ID and Configuration registers. |
| 45 | Burst Type | Indicates whether the burst will be linear or wrapped.<br>Burst Type=0 indicates wrapped burst<br>Burst Type=1 indicates linear burst |
| 44-16 | Row & Upper Column Address | Row & Upper Column component of the target address: System word address bits A31-A3<br>Any upper Row address bits not used by a particular device density should be set to 0 by the host controller master interface. The size of Rows and therefore the address bit boundary between Row and Column address is slave device dependent. |
| 15-3 | Reserved | Reserved for future column address expansion.<br>Reserved bits are don't care in current HyperBus devices but should be set to 0 by the host controller master interface for future compatibility. |
| 2-0 | Lower Column Address | Lower Column component of the target address: System word address bits A2-0 selecting the starting word within a half-page. |

***Notes:***

1. *A Row is a group of words relevant to the internal memory array structure and additional latency may be inserted by RWDS when crossing Row boundaries - this is device dependent behavior, refer to each HyperBus device data sheet for additional information. Also, the number of Rows may be used in the calculation of a distributed refresh interval for HyperRAM memory.*
2. *A Page is a 16-word (32-byte) length and aligned unit of device internal read or write access and additional latency may be inserted by RWDS when crossing Page boundaries - this is device dependent behavior, refer to each HyperBus device data sheet for additional information.*
3. *The Column address selects the burst transaction starting word location within a Row. The Column address is split into an upper and lower portion. The upper portion selects an 8-word (16-byte) Half-page and the lower portion selects the word within a Half-page where a read or write transaction burst starts.*
4. *The initial read access time starts when the Row and Upper Column (Half-page) address bits are captured by a slave interface. Continuous linear read burst is enabled by memory devices internally interleaving access to 16 byte half-pages.*
5. *HyperBus protocol address space limit, assuming:*
   *29 Row &Upper Column address bits*
   *3 Lower Column address bits*
   *Each address selects a word wide (16 bit = 2 byte) data value*
   *29 + 3 = 32 address bits = 4G addresses supporting 8Gbyte (64Gbit) maximum address space*
   *Future expansion of the column address can allow for 29 Row &Upper Column + 16 Lower Column address bits = 35 Tera-word = 70 Tera-byte address space.*

**Figure 11.2** Data Placement During a Read Transaction




***Notes:***

1. *Figure shows a portion of a Read transaction on the HyperBus. CK# of differential clock is shown as dashed line waveform.*
2. *Data is "edge aligned" with the RWDS serving as a read data strobe during read transactions.*
3. *Data is always transferred in full word increments (word granularity transfers).*
4. *Word address increments in each clock cycle. Byte A is between RWDS rising and falling edges and is followed by byte B between RWDS falling and rising edges, of each word.*
5. *Data bits in each byte are always in high to low order with bit 7 on DQ7 and bit 0 on DQ0.*

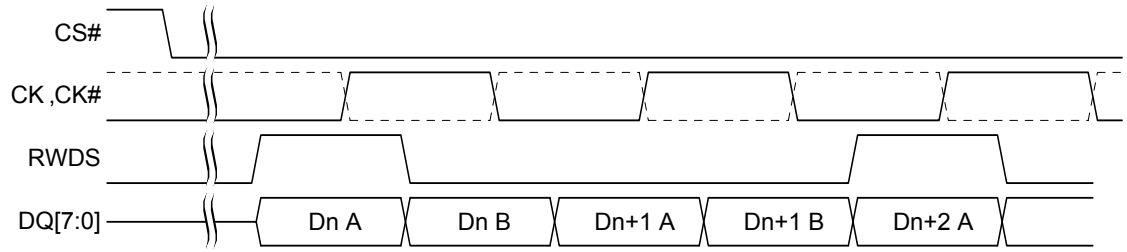
**Table 11.3** Data Bit Placement During Read or Write Transaction

| Address Space | Byte Order | Byte Position | Word Data Bit | DQ | Bit Order |
|---|---|---|---|---|---|
| Memory | Big-endian | A | 15 | 7 | When data is being accessed in memory space:<br>*The first byte of each word read or written is the "A" byte and the second is the "B" byte.*<br>*The bits of the word within the A and B bytes depend on how the data was written. If the word lower address bits 7-0 are written in the A byte position and bits 15-8 are written into the B byte position, or vice versa, they will be read back in the same order.*<br><br>So, memory space can be stored and read in either little-endian or big-endian order. |
| | | | 14 | 6 | |
| | | | 13 | 5 | |
| | | | 12 | 4 | |
| | | | 11 | 3 | |
| | | | 10 | 2 | |
| | | | 9 | 1 | |
| | | | 8 | 0 | |
| | | B | 7 | 7 | |
| | | | 6 | 6 | |
| | | | 5 | 5 | |
| | | | 4 | 4 | |
| | | | 3 | 3 | |
| | | | 2 | 2 | |
| | | | 1 | 1 | |
| | | | 0 | 0 | |
| | Little-endian | A | 7 | 7 | |
| | | | 6 | 6 | |
| | | | 5 | 5 | |
| | | | 4 | 4 | |
| | | | 3 | 3 | |
| | | | 2 | 2 | |
| | | | 1 | 1 | |
| | | | 0 | 0 | |
| | | B | 15 | 7 | |
| | | | 14 | 6 | |
| | | | 13 | 5 | |
| | | | 12 | 4 | |
| | | | 11 | 3 | |
| | | | 10 | 2 | |
| | | | 9 | 1 | |
| | | | 8 | 0 | |

**Table 11.3** Data Bit Placement During Read or Write Transaction

| Address Space | Byte Order | Byte Position | Word Data Bit | DQ | Bit Order |
|---|---|---|---|---|---|
| Register | Big-endian | A | 15 | 7 | When data is being accessed in register space:<br>During a Read transaction on the HyperBus two bytes are transferred on each clock cycle. The upper order byte A (Word[15:8]) is transferred between the rising and falling edges of RWDS (edge aligned). The lower order byte B (Word[7:0]) is transferred between the falling and rising edges of RWDS.<br><br>During a write, the upper order byte A (Word[15:8]) is transferred on the CK rising edge and the lower order byte B (Word[7:0]) is transferred on the CK falling edge.<br>So, register space is always read and written in Big-endian order because registers have device dependent fixed bit location and meaning definitions. |
| | | | 14 | 6 | |
| | | | 13 | 5 | |
| | | | 12 | 4 | |
| | | | 11 | 3 | |
| | | | 10 | 2 | |
| | | | 9 | 1 | |
| | | | 8 | 0 | |
| | | B | 7 | 7 | |
| | | | 6 | 6 | |
| | | | 5 | 5 | |
| | | | 4 | 4 | |
| | | | 3 | 3 | |
| | | | 2 | 2 | |
| | | | 1 | 1 | |
| | | | 0 | 0 | |

**Figure 11.3** Data Placement During a Write Transaction




***Notes:***

1. *Figure shows a portion of a Write transaction on the HyperBus.*
2. *Data is "center aligned" with the clock during a Write transaction.*
3. *RWDS functions as a data mask during write data transfers with initial latency. Masking of the first and last byte is shown to illustrate an unaligned 3 byte write of data.*
4. *RWDS is not driven by the master during write data transfers with zero initial latency. Full data words are always written in this case. RWDS may be driven low or left High-Z by the slave in this case.*

## 11.2 Read Transactions

The HyperBus master begins a transaction by driving CS# Low while clock is idle. Then the clock begins toggling while Command-Address CA words are transfered.

In CA0, CA[47] = 1 indicates that a Read transaction is to be performed. CA[46] = 0 indicates the memory space is being read or CA[46] = 0 indicates the register space is being read. CA[45] indicates the burst type (wrapped or linear). Read transactions can begin the internal array access as soon as the row and upper column address has been presented in CA0 and CA1 (CA[47:16]). CA2 (CA(15:0]) identifies the target Word address within the chosen row. However, some HyperBus devices may require a minimum time between the end of a prior transaction and the start of a new access. This time is referred to as Read-Write-Recovery time ($t_{RWR}$). The master interface must start driving CS# Low only at a time when the CA1 transfer will complete after $t_{RWR}$ is satisfied.

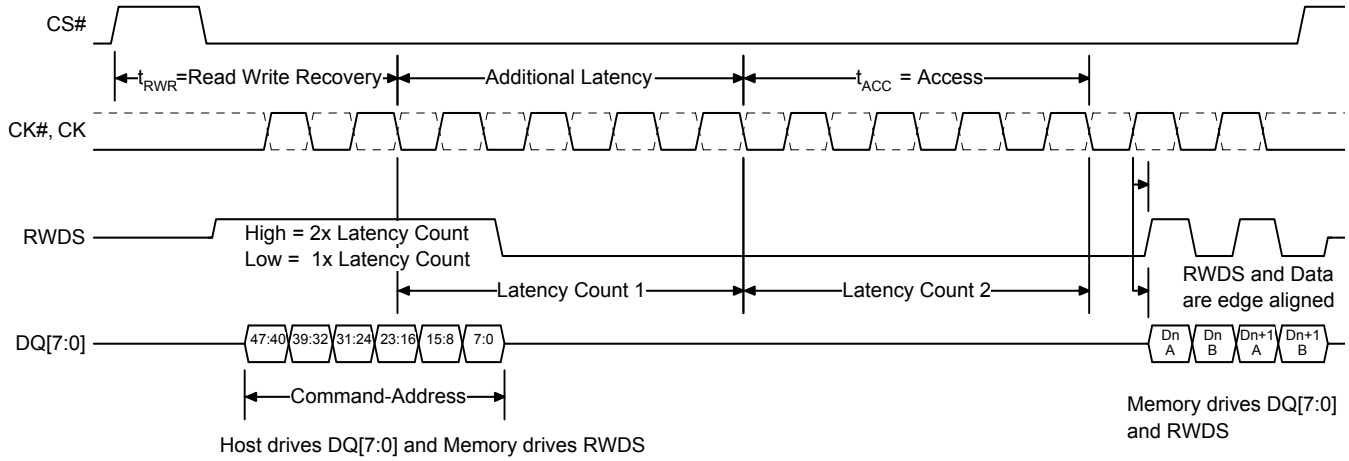
The HyperBus master then continues clocking for a number of cycles defined by the latency count setting in a configuration register. The initial latency count required for a particular clock frequency is device dependent - refer to the device data sheet to determine the correct configuration register and initial latency setting for the desired operating frequency. If RWDS is Low during the CA cycles, one latency count is inserted. If RWDS is High during the CA cycles, an additional latency count is inserted. Once these latency clocks have been completed the memory starts to simultaneously transition the Read-Write Data Strobe (RWDS) and output the target data.

New data is output edge aligned with every transition of RWDS. Data will continue to be output as long as the host continues to transition the clock while CS# is Low. However, the HyperBus slave may stop RWDS transitions with RWDS Low, between the delivery of words, in order to insert latency between words when crossing certain memory array boundaries or for other reasons. The slave may also hold the RWDS Low for an extended period of 32 or more clocks as an indication of an error condition that requires the read transaction be discontinued by the master. The use of RWDS for insertion of latency between word transfers or to indicate errors is device dependent, refer to individual device data sheets for additional information on if or when this latency insertion method may be used by a HyperBus device.

Wrapped bursts will continue to wrap within the burst length and linear burst will output data in a sequential manner across row boundaries. When a linear burst read reaches the last address in the array, continuing the burst beyond the last address will provide undefined data. Read transfers can be ended at any time by bringing CS# High when the clock is idle.

The clock is not required to be free-running. The clock may remain idle while CS# is high.

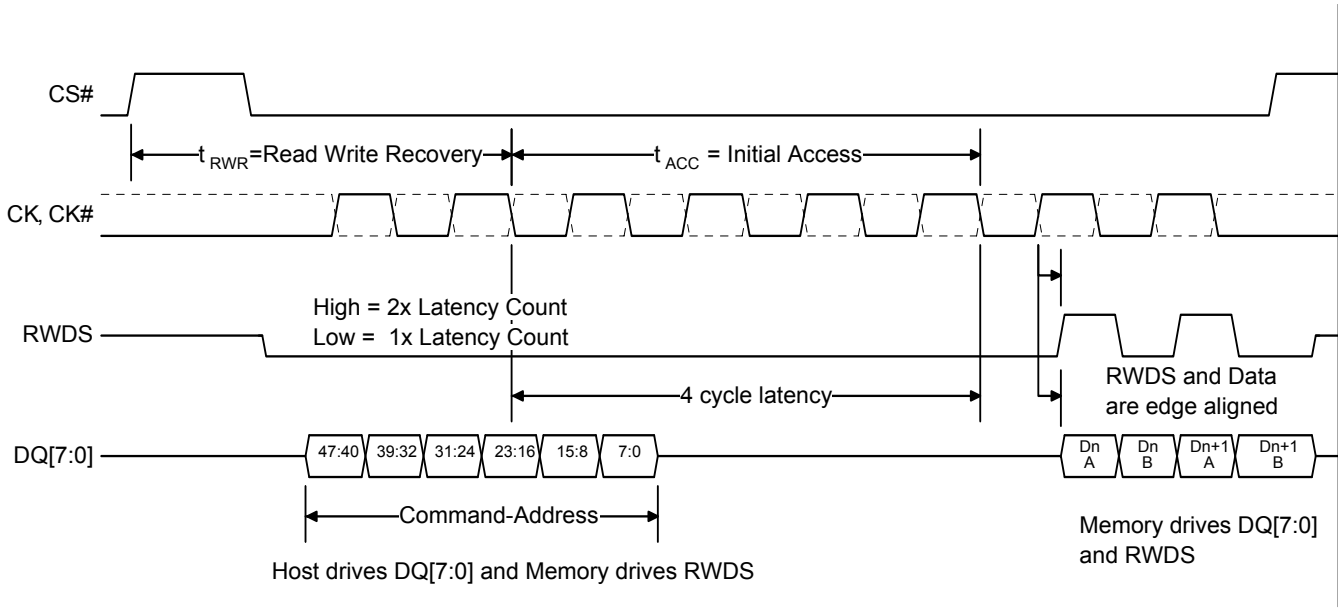
**Figure 11.4** Read Transaction with Additional Initial Latency




***Notes:***

1. *Transactions are initiated with CS# falling while CK=Low and CK#=High.*
2. *CS# must return High before a new transaction is initiated.*
3. *CK# is the complement of the CK signal. 3V devices use a single ended clock (CK only), CK# is used with CK on1.8V devices to provide a differential clock. CK# of a differential clock is shown as a dashed line waveform.*
4. *Read access array starts once CA[23:16] is captured.*
5. *The read latency is defined by the initial latency value in a configuration register.*
6. *In this read transaction example the initial latency count was set to four clocks.*
7. *In this read transaction a RWDS High indication during CA delays output of target data by an additional four clocks.*
8. *The memory device drives RWDS during read transactions.*

**Figure 11.5** Read Transaction Without Additional Initial Latency




***Note:***

1. *RWDS is Low during the CA cycles. In this Read Transaction there is a single initial latency count for read data access because, this read transaction does not begin at a time when additional latency is required by the slave.*

## 11.3 Write Transactions with Initial Latency

The HyperBus master begins a transaction by driving CS# Low while clock is idle. Then the clock begins toggling while Command-Address CA words are transfered.

In CA0, CA[47] = 0 indicates that a Write transaction is to be performed. CA[46] = 0 indicates the memory space is being written. CA[45] indicates the burst type (wrapped or linear). Write transactions can begin the internal array access as soon as the row and upper column address has been presented in CA0 and CA1 (CA[47:16]). CA2 (CA(15:0]) identifies the target word address within the chosen row. However, some HyperBus devices may require a minimum time between the end of a prior transaction and the start of a new access. This time is referred to as Read-Write-Recovery time ($t_{RWR}$). The master interface must start driving CS# Low only at a time when the CA1 transfer will complete after $t_{RWR}$ is satisfied.

The HyperBus master then continues clocking for a number of cycles defined by the latency count setting in a configuration register. The initial latency count required for a particular clock frequency is device dependent - refer to the device data sheet to determine the correct configuration register and initial latency setting for the desired operating frequency. If RWDS is Low during the CA cycles, one latency count is inserted. If RWDS is High during the CA cycles, an additional latency count is inserted. The use for the additional initial latency is device dependent - refer to the device data sheet for more information.

Once these latency clocks have been completed the HyperBus master starts to output the target data. Write data is center aligned with the clock edges. The first byte of data in each word is captured by the memory on the rising edge of CK and the second byte is captured on the falling edge of CK.

During the CA clock cycles, RWDS is driven by the memory.

During the write data transfers, RWDS is driven by the host master interface as a data mask. When data is being written and RWDS is High the byte will be masked and the array will not be altered. When data is being written and RWDS is Low the data will be placed into the array. Because the master is driving RWDS during write data transfers, neither the master nor the slave device are able to indicate a need for latency within the data transfer portion of a write transaction. The slave must be able to accept a continuous burst of write data or require a limit on the length of a write data burst that the slave can accept. The acceptable write data burst length is device dependent - refer to the device data sheet for more information on any burst length limitation.

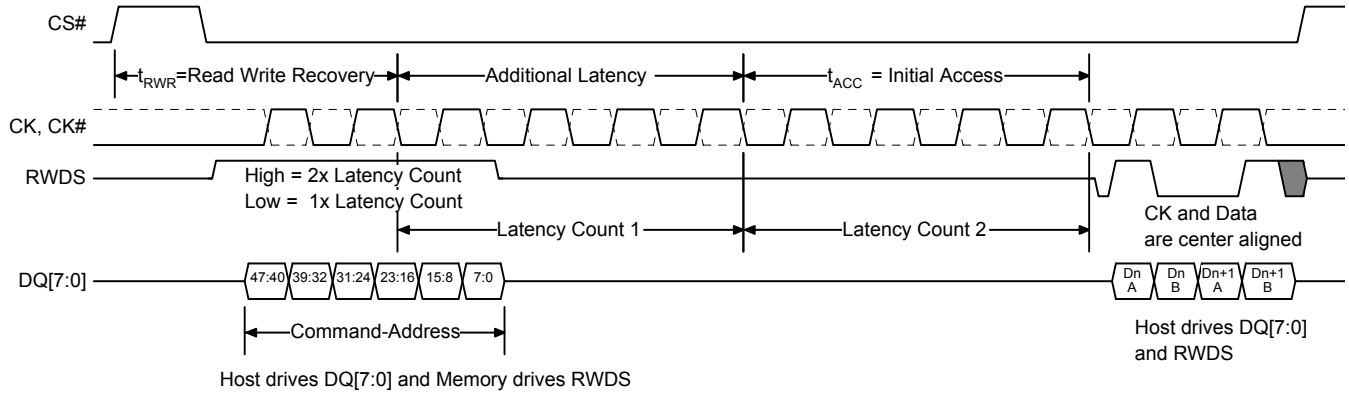
Data will continue to be transferred as long as the HyperBus master continues to transition the clock while CS# is Low. Legacy format wrapped bursts will continue to wrap within the burst length. Hybrid wrap will wrap once then switch to linear burst starting at the next wrap boundary. Linear burst accepts data in a sequential manner across page boundaries. Write transfers can be ended at any time by bringing CS# High when the clock is idle.

Some HyperBus devices do not support Legacy or Hybrid wrapped write transactions. This is device dependent behavior - refer to the device data sheet to determine if the device supports wrapped write transactions.

When a linear burst write reaches the last address in the memory array space, continuing the burst has device dependent results - refer to the device data sheet for more information.

The clock is not required to be free-running. The clock may remain idle while CS# is high.

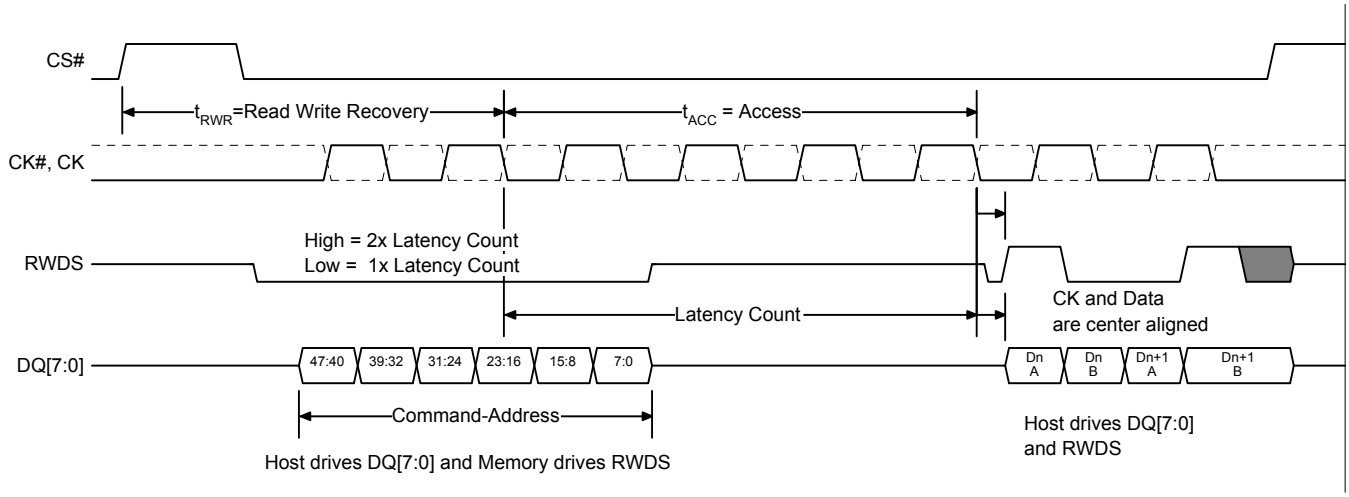
**Figure 11.6** Write Transaction with Additional Initial Latency




***Notes:***

1. *Transactions must be initiated with CK=Low and CK#=High.*
2. *CS# must return High before a new transaction is initiated.*
3. *During Command-Address, RWDS is driven by the memory and indicates whether additional latency cycles are required.*
4. *RWDS indicates that additional initial latency cycles are required.*
5. *During data transfer, RWDS is driven by the host to indicate which bytes of data should be either masked or loaded into the array.*
6. *The figure shows RWDS masking byte A0 and byte B1 to perform an unaligned word write to bytes B0 and A1.*

**Figure 11.7** Write Transaction Without Additional Initial Latency




***Notes:***

1. *During Command-Address, RWDS is driven by the memory and indicates whether additional latency cycles are required.*
2. *RWDS indicates that there is no additional latency required.*
3. *During data transfer, RWDS is driven by the host to indicate which bytes of data should be either masked or loaded into the array.*
4. *The figure shows RWDS masking byte A0 and byte B1 to perform an unaligned word write to bytes B0 and A1.*

## 11.4 Write Transactions without Initial Latency

A Write transaction starts with the first three clock cycles providing the Command/Address information indicating the transaction characteristics. CA0 may indicate that a Write transaction is to be performed and also indicates the address space and burst type (wrapped or linear).

Some slave devices may require write transactions with zero latency between the CA cycles and following write data transfers. Writes with zero initial latency, do not have a turn around period for RWDS. The slave device will always drive RWDS during the Command-Address period to indicate whether extended latency is required for a transaction that has initial latency. However, the RWDS is driven before the slave device has received the first byte of CA i.e. before the slave knows whether the transaction is a read or write, to memory space or register space. In the case of a write with zero latency, the RWDS state during the CA period does not affect the initial latency of zero. Since master write data immediately follows the Command-Address period in this case, the slave may continue to drive RWDS Low or may take RWDS to High-Z during write data transfer. The master must not drive RWDS during Writes with zero latency. Writes with zero latency do not use RWDS as a data mask function. All bytes of write data are written (full word writes). Writes without initial latency are generally used for register space writes but the requirement for writes with zero latency is slave device dependent. Writes with zero latency may be required for memory space or register space or neither, depending on the slave device capability.

HyperBus master interfaces must provide a configuration setting to support zero initial latency in write transactions for either or both Memory Space and Register Space as required by the device being selected by a CS#.

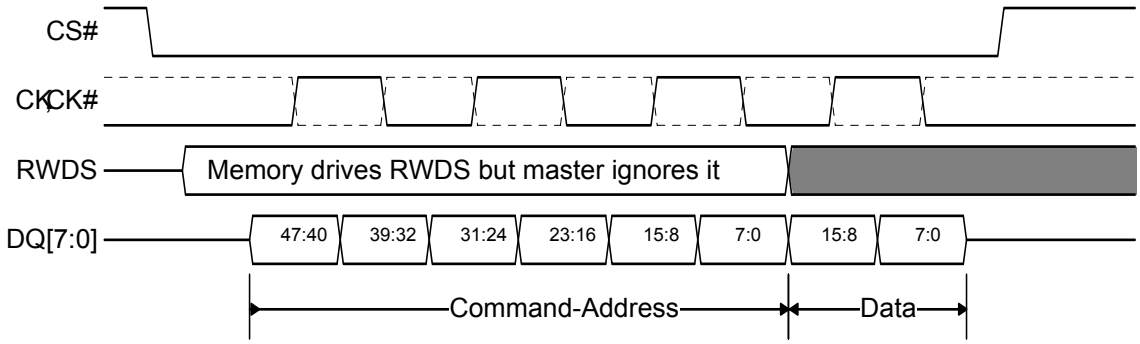
The first byte of data in each word is presented on the rising edge of CK and the second byte is presented on the falling edge of CK. Write data is center aligned with the clock inputs. Write transfers can be ended at any time by bringing CS# High when clock is idle. The clock is not required to be free-running.

Because the slave is not driving RWDS as a flow control indication during write data transfers, the slave device is not able to indicate a need for latency within the data transfer portion of a write transaction. The slave must be able to accept a continuous burst of write data or the master must limit the length of a write data burst to that which the slave can accept. This is device dependent behavior - refer to the device data sheet for more information on any burst length limitation.

Legacy format wrapped bursts will continue to wrap within the burst length. Hybrid wrap will wrap once then switch to linear burst starting at the next wrap boundary. Linear burst accepts data in a sequential manner across page boundaries. Some HyperBus devices do not support wrapped write transactions. This is device dependent behavior - refer to the device data sheet to determine if the device supports wrapped write transactions.

When a linear burst write reaches the last address in the memory array space, continuing the burst has device dependent results - refer to the device data sheet for more information.

**Figure 11.8** Write Operation without Initial Latency




Host drives DQ[7:0] with Command-Address and Write Data

# HyperBus Device Software Interface

## 12. Memory Space

When CA[46] is 0 a read or write transaction accesses the memory array. The Memory Space address map is device dependent - refer to the device data sheet to determine the device memory space address map.

## 13. Register Space

When CA[46] is 1 a read or write transaction accesses the Register Space. The Register Space map is device dependent - refer to the device data sheet to determine the complete device register space address map.

### 13.1 Device Identification Registers

There are HyperBus device information words (registers) that can be read to identify the device manufacturer, the device type, the data capacity if the device is a memory, and other device specific feature parameters.

HyperFlash devices are software backward compatible with the ISSI parallel interface GL family flash memories. HyperFlash devices require a command sequence to make the device identification (ID) information visible. This device ID command sequence was referred as the Autoselect Address Space Overlay (ASO) in GL family memories. When the ID command sequence is written to the HyperFlash device, the ID information replaces (overlays) one sector (erase block) of the memory array space. The base address of the block where the ID information appears is set by an address in the in the command sequence. HyperFlash memories do not require selection of the HyperBus Register Address space when writing commands or reading register information.

The CA[46] bit is ignored by HyperFlash devices. HyperFlash memories will respond to the device ID command sequence when it is written to either the Memory Space or the Register Space. The device ID information starts at location zero of the selected block of the Memory / Register space.

HyperRAM and other HyperBus peripherals are required to implement the Register Space and place the device ID information starting at location zero in the Register Space. Upper address bits not needed for selecting portions of the Register Space are don't care, such that the implemented portion of each device Register Space aliases within (is repeated throughout) the Register Space.

The recommended method for reading the device ID information is to write the HyperFlash ID command sequence to the Register Space of a device (with chip select low for the desired device) with the upper address bits set to zero. This will cause HyperFlash devices to display their ID information at the base address of the selected block, in what appears to the host system HyperBus master interface to be Register Space.

HyperRAM or other HyperBus peripherals are required to ignore the HyperFlash command sequences and simply always display their ID information starting at location zero in Register Space. By following this recommendation, each device on HyperBus will display its ID starting at location zero in Register Space.

The structure of the first two words of device ID is consistent for all HyperBus devices and provides information on the device selected when CS# is low. Information fields in these first two words identify:

- Manufacturer
- Device Type

Other fields in these first two words and the remainder of the Register Space are device dependent - refer to the device data sheet for additional information on the contents of the device ID and other Register Space locations.

.

**Table 13.1** Word Address 0 ID Register Bit Assignments

| Bits | Function | Settings (Binary) |
|---|---|---|
| 15-4 | Device Dependent | Refer to the device data sheet for more information. |
| 3-0 | Manufacturer | 0000 - Reserved<br>0011 - ISSI<br>0010 to 1111 - Reserved |

.

**Table 13.2** Word Address 1 ID Register Bit Assignments

| Bits | Function | Settings (Binary) |
|---|---|---|
| 15-4 | Device Dependent | Refer to the device data sheet for more information. |
| 3-0 | Device Type | 0000 - HyperRAM<br>0001 to 1101 - Reserved<br>1110 - HyperFlash<br>1111 - Reserved |

## 13.2 Configuration Registers

Configuration register default values are loaded upon power-up or hardware reset. The configuration registers controlling the HyperBus interface behavior are volatile and can be written at any time the device is in standby state.

The location in Register Space and structure of the register contents are device dependent - refer the device data sheet for more information.

All HyperBus devices have configuration register fields in Register Space that define:

- Deep Power Down (DPD) operation mode
- Output Drive Strength
- Initial Latency Count (clock cycle count from the capture of CA1 to the beginning of data transfer)
- Fixed Initial Latency Option (device dependent effect)
- Hybrid Wrap Option (one wrapped access then switch to linear burst)
- Wrapped burst length and alignment (options supported are device dependent)

.

**Table 13.3** Example Configuration Register Bit Assignments (Sheet 1 of 2)

| CR0 Bit | Function | Settings (Binary) |
|---|---|---|
| 15 | Deep Power Down Enable | 0 - Writing 0 to CR[15] causes the device to enter Deep Power Down.<br>1 - Normal operation (default) |
| 14-12 | Drive Strength | 000 - Device dependent impedance (default)<br>001 - Device dependent impedance<br>010 - Device dependent impedance<br>011 - Device dependent impedance<br>100 - Device dependent impedance<br>101 - Device dependent impedance<br>110 - Device dependent impedance<br>111 - Device dependent impedance |

**Table 13.3** Example Configuration Register Bit Assignments (Sheet 2 of 2)

| CR0 Bit | Function | Settings (Binary) |
|---|---|---|
| 7-4 | Initial Latency | 0000 - 5 Clocks<br>0001 - 6 Clocks<br>0010 - 7 Clocks<br>0011 - 8 Clocks<br>0100 - 9 Clocks<br>0101 - 10 Clocks<br>0110 - 11 Clocks<br>0111 - 12 Clocks<br>1000 - 13 Clocks<br>1001 - 14 Clocks<br>1010 - 15 Clocks<br>1011 - 16 Clocks<br>1100 - Reserved<br>1101 - Reserved<br>1110 - 3 Clock Latency<br>1111 - 4 Clock Latency<br>*The number of initial latency options implemented and the POR or reset default initial value is device dependent.* |
| 3 | Fixed Latency Enable | 0 - Variable Initial Latency<br>1 - Fixed Initial Latency (default)<br>Effect of this bit is device dependent. |
| 2 | Hybrid Burst Enable | 0: Wrapped burst sequences to follow hybrid burst sequencing<br>1: Wrapped burst sequences in legacy wrapped burst manner (default) |
| 1-0 | Burst Length | 00 - 128 bytes<br>01 - 64 bytes<br>10- 16 bytes<br>11 - 32 bytes (default)<br>The options supported are device dependent. |

***Note:***
*1. For device dependent settings, refer to the device data sheet for more information.*

#### 13.2.0.1 Deep Power Down

When the device is not needed for system operation, it may be placed in a very low power consuming mode called Deep Power Down (DPD), by writing 0 to CR0[15]. When CR0[15] is cleared to 0, the device immediately enters the DPD mode. A write to the device setting CR0[15] to 1, POR, or reset will cause the device to exit DPD mode. Returning to Standby mode requires $t_{DPD}$ time. The power consumption and any side effects of DPD mode are device dependent - refer to the device data sheet for additional details. DPD mode support is not required for HyperBus devices, some devices do not implement a DPD mode.

#### 13.2.0.2 Drive Strength

DQ signal line loading, length, and impedance vary depending on each system design. Configuration register bits CR0[14:12] provide a means to adjust the DQ[7:0] signal output impedance to customize the DQ signal impedance to the system to minimize high speed signal behaviors such as overshoot, undershoot, and ringing. The default POR or reset configuration value is 000b to select the mid point of the available output impedance options. The impedance values for each Drive Strength code is device dependent - refer to the device data sheet for additional details.

#### 13.2.0.3 Initial Latency

Memory Space read and write transactions or Register Space read transactions require some initial latency to open the row selected by the Command/Address. This initial latency is $t_{ACC}$ as shown in Figure 11.4, *Read Transaction with Additional Initial Latency* on page 45 and Figure 11.6, *Write Transaction with Additional Initial Latency* on page 48. The number of latency clocks needed to satisfy $t_{ACC}$ depends on the HyperBus frequency and can vary from 3 to 16 clocks. The value in CR0[7:4] selects the number of clocks for initial latency. The number of initial latency options implemented and the POR or reset default initial value is device dependent. The default value is selected to allow for operation up to a maximum frequency of the device prior to the host system setting a lower an initial latency value that may be more optimal for the system.

In the event additional latency is required at the time a read or write transaction begins, the RWDS signal goes high during the Command/Address to indicate that an additional initial latency count is being inserted to allow an internal operation to complete before opening the selected row.

Write transactions to memory or register space may be defined by a device to always have zero initial latency. The use of zero initial latency writes are device dependent. RWDS is always driven by the HyperBus slave during the Command/Address period with an indication of the initial latency requirement. This indication is intended for write transactions that support initial latency and is always provided because RWDS is driven before CA0 is received to indicate whether the transaction is a read or write and to memory or register space. RWDS is ignored by the HyperBus master when a write transaction target address space is designated by a device as always having zero latency.

#### 13.2.0.4 Fixed Latency

A configuration register option bit CR0[3] is provided to make all read or write transactions with initial latency require the same initial latency by always driving RWDS low or high during the Command/Address to indicate whether one or two initial latency counts are required. This fixed initial latency is independent of any need for additional initial latency, it simply provides a fixed (deterministic) initial latency for all of these transaction types. The fixed latency option may simplify the design of some HyperBus memory controllers or ensure deterministic transaction performance. Fixed latency is the default POR or reset configuration. The system may clear this configuration bit to disable fixed latency and allow variable initial latency with RWDS driven high only when additional latency is required.

Implementation of the fixed initial latency feature is optional and device dependent. Some HyperBus devices do not implement the feature and always drive RWDS low during CA cycles to use a single fixed initial latency count with transactions that have initial latency. Some HyperBus devices that implement the fixed latency feature will use the fixed latency setting to always drive RWDS high during CA cycles to always require two latency counts.

#### 13.2.0.5 Wrapped Burst

A wrapped burst transaction accesses memory within a group of words aligned on a word boundary matching the length of the configured group. Wrapped access groups can be configured as 16, 32, 64 or 128 bytes alignment and length. However, the number of wrap configuration options supported is device dependent.

During wrapped transactions, access starts at the Command/Address selected location within the group, continues to the end of the configured word group aligned boundary, then wraps around to the beginning location in the group, then continues back to the starting location. Wrapped bursts are generally used for critical word first instruction or data cache line fill read accesses.

#### 13.2.0.6 Hybrid Burst

The beginning of a hybrid burst will wrap within the target address wrapped burst group length before continuing to the next half-page of data beyond the end of the wrap group. Continued access is in linear burst order until the transfer is ended by returning CS# high. This hybrid of a wrapped burst followed by a linear burst starting at the beginning of the next half-page, allows multiple sequential address cache lines to be filled in a single access. The first cache line is filled starting at the critical word. Then the next sequential cache line in memory can be read in to the cache while the first line is being processed. Hybrid burst support is device dependent.

.

**Table 13.4** Example Wrapped Burst Sequences

| Burst Selection CA[45] | Burst Type | Wrap Boundary (Bytes) | Start Address (Hex) | Address Sequence (Hex) (Words) |
|---|---|---|---|---|
| 0 | Hybrid 128 | 128 Wrap once then Linear | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 00, 01, 02<br>(wrap complete, now linear beyond the end of the initial 128 byte wrap group)<br>40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, ... |
| 0 | Hybrid 64 | 64 Wrap once then Linear | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 00, 01, 02,<br>(wrap complete, now linear beyond the end of the initial 64 byte wrap group)<br>20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, ... |
| 0 | Hybrid 32 | 32 Wrap once then Linear | XXXXXX0A | 0A, 0B, 0C, 0D, 0E, 0F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, ... |
| 0 | Hybrid 16 | 16 Wrap once then Linear | XXXXXX02 | 02, 03, 04, 05, 06, 07, 00, 01,<br>(wrap complete, now linear beyond the end of the initial 16 byte wrap group)<br>08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, ... |
| 0 | Wrap 128 | 128 | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 00, 01, 02, ... |
| 0 | Wrap 64 | 64 | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 00, 01, 02, ... |
| 0 | Wrap 32 | 32 | XXXXXX0A | 0A, 0B, 0C, 0D, 0E, 0F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, ... |
| 0 | Wrap 16 | 16 | XXXXXX02 | 02, 03, 04, 05, 06, 07, 00, 01, ... |
| 1 | Linear | Linear Burst | XXXXXX03 | 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, ... |

### 13.2.1 Additional Identification or Configuration Registers

Any additional identification or configuration registers are device dependent - refer to the device data sheet for more information.

# HyperBus Device Hardware Interface

## 14. HyperBus Connection Descriptions

For a description of HyperBus mandatory and optional signals refer to Section 10.1, *Input/Output Summary* on page 39. Following is the description for other package connectors.

### 14.1 Other Connectors Summary

**Table 16.1** Other Connectors Summary

| Symbol | Type | Description |
|---|---|---|
| $V_{CC}$ | Power Supply | **Core Power** |
| $V_{CC}Q$ | Power Supply | **Input/Output Power** |
| $V_{SS}$ | Power Supply | **Core Ground** |
| $V_{SS}Q$ | Power Supply | **Input/Output Ground** |
| NC | No Connect | Not Connected internally. The signal/ball location may be used in Printed Circuit Board (PCB) as part of a routing channel. |
| RFU | No Connect | Reserved for Future Use. May or may not be connected internally, the signal/ball location should be left unconnected and unused by PCB routing channel for future compatibility. The signal/ball may be used by a signal in the future. |
| DNU | Reserved | Do Not Use. Reserved for use by ISSI. The signal/ball is connected internally. The signal/ball must be left open on the PCB. |

## 14.2 Device Connection Diagrams

**Figure 14.1** HyperBus Master with Mandatory and Optional Signals

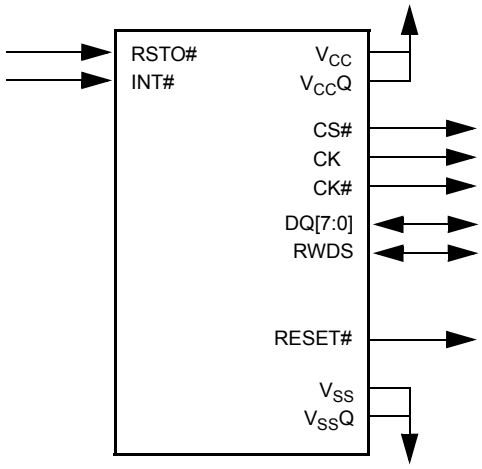
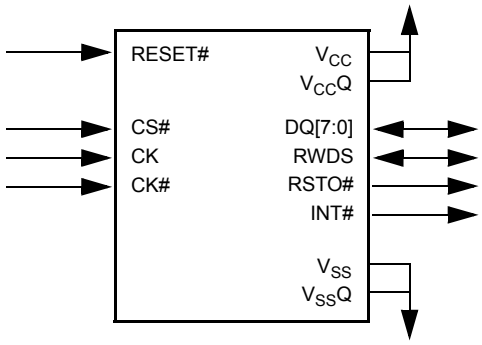
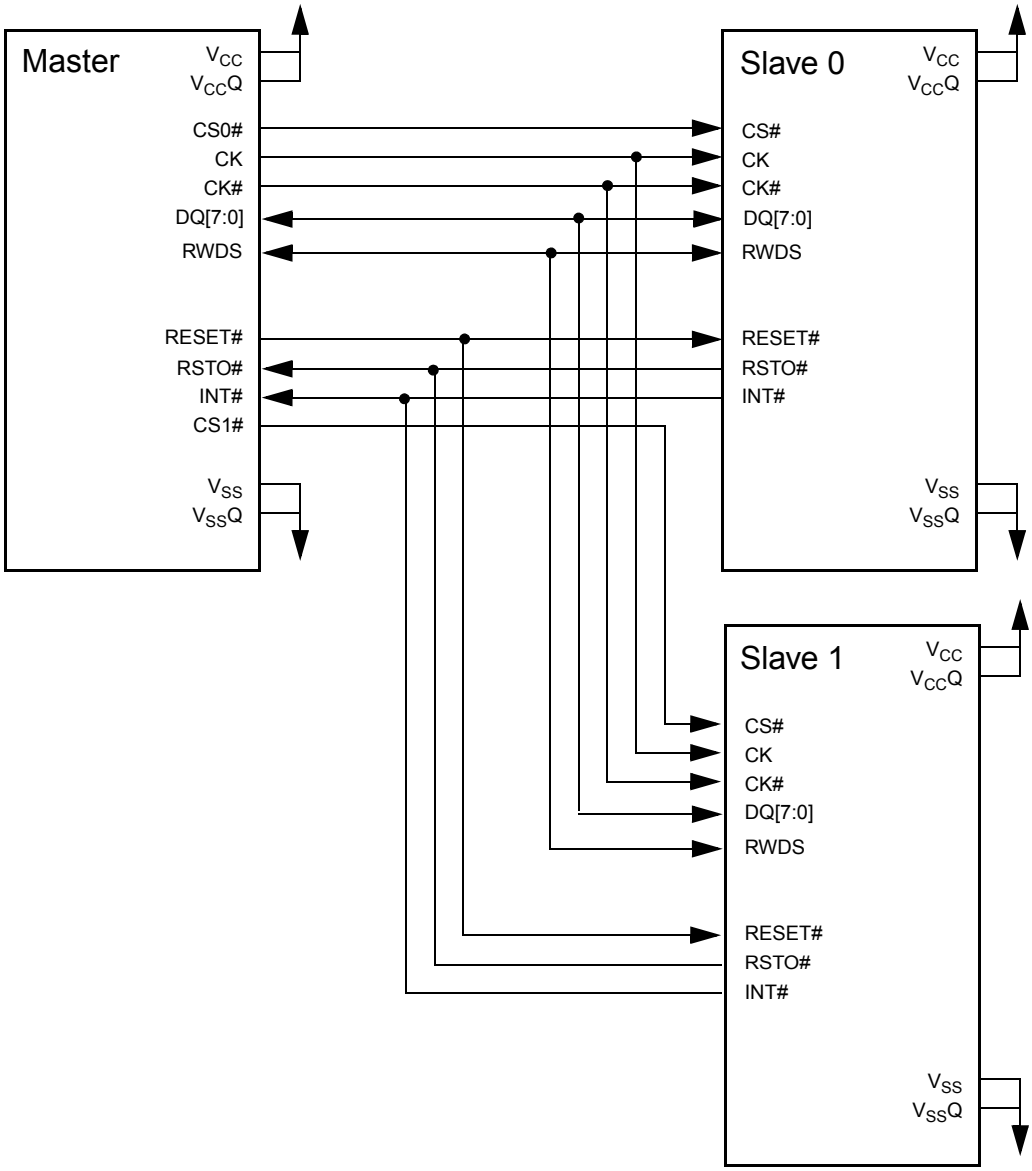



**Figure 14.2** HyperBus Slave with Mandatory and Optional Signals



## 14.3 HyperBus Block Diagram

**Figure 14.3** HyperBus Connections, Including Optional Signals



# 15. Interface States

The Interface States table describes the required value of each signal for each interface state.

**Table 15.1** Interface States

| Interface State | $V_{CC}$ / $V_{CC}Q$ | CS# | CK, CK# | D7-D0 | RWDS | RESET# | RSTO# | INT# |
|---|---|---|---|---|---|---|---|---|
| Power-Off with Hardware Data Protection (Flash memory) | < $V_{LKO}$ | X | X | High-Z | High-Z | X | High-Z | High-Z |
| Power-On (Cold) Reset | ≥ $V_{CC}$ / $V_{CC}Q$ min | X | X | High-Z | High-Z | X | L | High-Z |
| Hardware (Warm) Reset | ≥ $V_{CC}$ / $V_{CC}Q$ min | X | X | High-Z | High-Z | L | L | High-Z |
| Interface Standby | ≥ $V_{CC}$ / $V_{CC}Q$ min | H | X | High-Z | High-Z | H | High-Z | High-Z |
| Command-Address | ≥ $V_{CC}$ / $V_{CC}Q$ min | L | T | Master Output Valid | X | H | High-Z | High-Z |
| Read Initial Access Latency (data bus turn around period) | ≥ $V_{CC}$ / $V_{CC}Q$ min | L | T | High-Z | L | H | High-Z | High-Z |
| Write Initial Access Latency (RWDS turn around period) | ≥ $V_{CC}$ / $V_{CC}Q$ min | L | T | High-Z | High-Z | H | High-Z | High-Z |
| Read data transfer | ≥ $V_{CC}$ / $V_{CC}Q$ min | L | T | Slave Output Valid | Slave Output Valid X or T | H | High-Z | High-Z |
| Write data transfer with Initial Latency | ≥ $V_{CC}$ / $V_{CC}Q$ min | L | T | Master Output Valid | Master Output Valid X or T | H | High-Z | High-Z |
| Write data transfer without Initial Latency (2) | ≥ $V_{CC}$ / $V_{CC}Q$ min | L | T | Master Output Valid | Slave Output L or High-Z | H | High-Z | High-Z |
| Interrupt (3) | ≥ $V_{CC}$ / $V_{CC}Q$ min | X | X or T | X | X | H | High-Z | L |
| Active Clock Stop (4) | ≥ $V_{CC}$ / $V_{CC}Q$ min | L | Idle | Master or Slave Output Valid or High-Z | X | H | High-Z | High-Z |
| Deep Power Down(4) | ≥ $V_{CC}$ / $V_{CC}Q$ min | H | X or T | Slave Output High-Z | High-Z | H | High-Z | High-Z |

***Legend***

*L = $V_{IL}$*

*H = $V_{IH}$*

*X = either $V_{IL}$ or $V_{IH}$*

*L/H = rising edge*

*H/L = falling edge*

*T = Toggling during information transfer*

*Idle = CK is low and CK# is high.*

*Valid = all bus signals have stable L or H level*

***Notes:***

1. *When the RSTO# or INT# open-drain outputs are High-Z the pull-up resistance provided by the master or an external resistor will pull the signal to High.*
2. *Writes without initial latency (with zero initial latency), do not have a turn around period for RWDS. The slave device will always drive RWDS during the Command-Address period to indicate whether extended latency is required. Since master write data immediately follows the Command-Address period the slave may continue to drive RWDS Low or may take RWDS to High-Z. The master must not drive RWDS during Writes with zero latency. Writes with zero latency do not use RWDS as a data mask function. All bytes of write data are written (full word writes). Writes without initial latency are generally used for register space writes but the requirement for writes with zero latency is slave device dependent. Writes with zero latency may be required for memory space or register space or neither, depending on the slave device capability.*
3. *The Interrupt state may be concurrent with other states. The Interrupt signal assertion is independent of other bus states.*
4. *Active Clock Stop is described in Section 15.1.2, Active Clock Stop on page 59. DPD is described in Section 15.1.3, Deep Power Down on page 59.*

## 15.1 Power Conservation Modes

### 15.1.1 Interface Standby

Standby is the default, low power, state for the interface while the device is not selected by the host for data transfer (CS#= High). All inputs, and outputs other than CS# and RESET# are ignored in this state.

### 15.1.2 Active Clock Stop

Active Clock Stop is a read / write operation where the clock has not transitioned for an extended period, without CS# going High and the device internal logic has gone into Standby Mode to conserve power. Slave read output data or master command, address, or write output data is latched and remains actively driven and valid during Active Clock Stop during transaction periods where the master or slave would normally drive their output. During latency (bus turnaround) periods the data lines are High-Z. Support for the Active Clock Stop feature is slave device dependent.

### 15.1.3 Deep Power Down

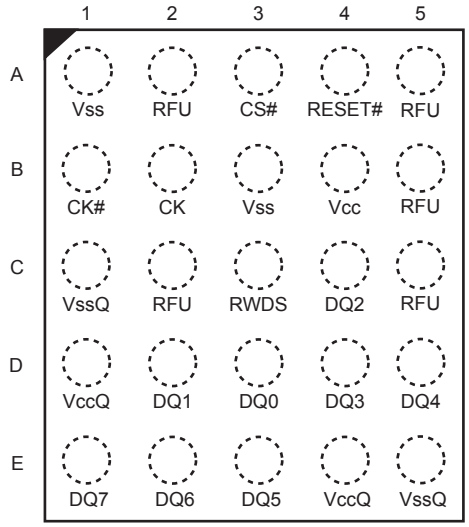
In the Deep Power Down (DPD) mode, current consumption is driven to the lowest possible level ($I_{DPD}$). DPD mode is entered by writing a 0 to CR0[15]. The device immediately reduces power. A write to the device, POR, or hardware reset will cause the device to exit DPD mode. Returning to Standby mode requires $t_{DPD}$ time. Support for the DPD mode feature is slave device dependent.

# 16. Physical Interface

## 16.1 Connection Diagram

### 16.1.1 FBGA 25-Ball, 5x5 Array Footprint

**Figure 16.1** 25-ball BGA, 5x5 Ball Footprint (FAE025), Top View

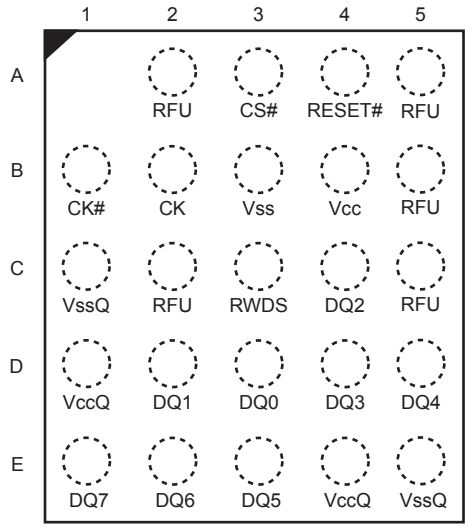



***Notes:***

1. *B1 is a RFU on the 3.0V device and is assigned to CK# on the 1.8V device.*
2. *B5 and C5 are Reserved for Future Use (RFU), internally connected, and are recommended to be connected to VSS to minimize noise. However, signals on these balls are ignored and the balls may be left floating or used in a PCB routing channels if necessary.*
3. *C2 is an RFU and must be floating or be pulled high.*

### 16.1.2 FBGA 24-Ball, 5x5 Array Footprint

**Figure 16.1** 24-ball BGA, 5x5 Ball Footprint (FAE025), Top View

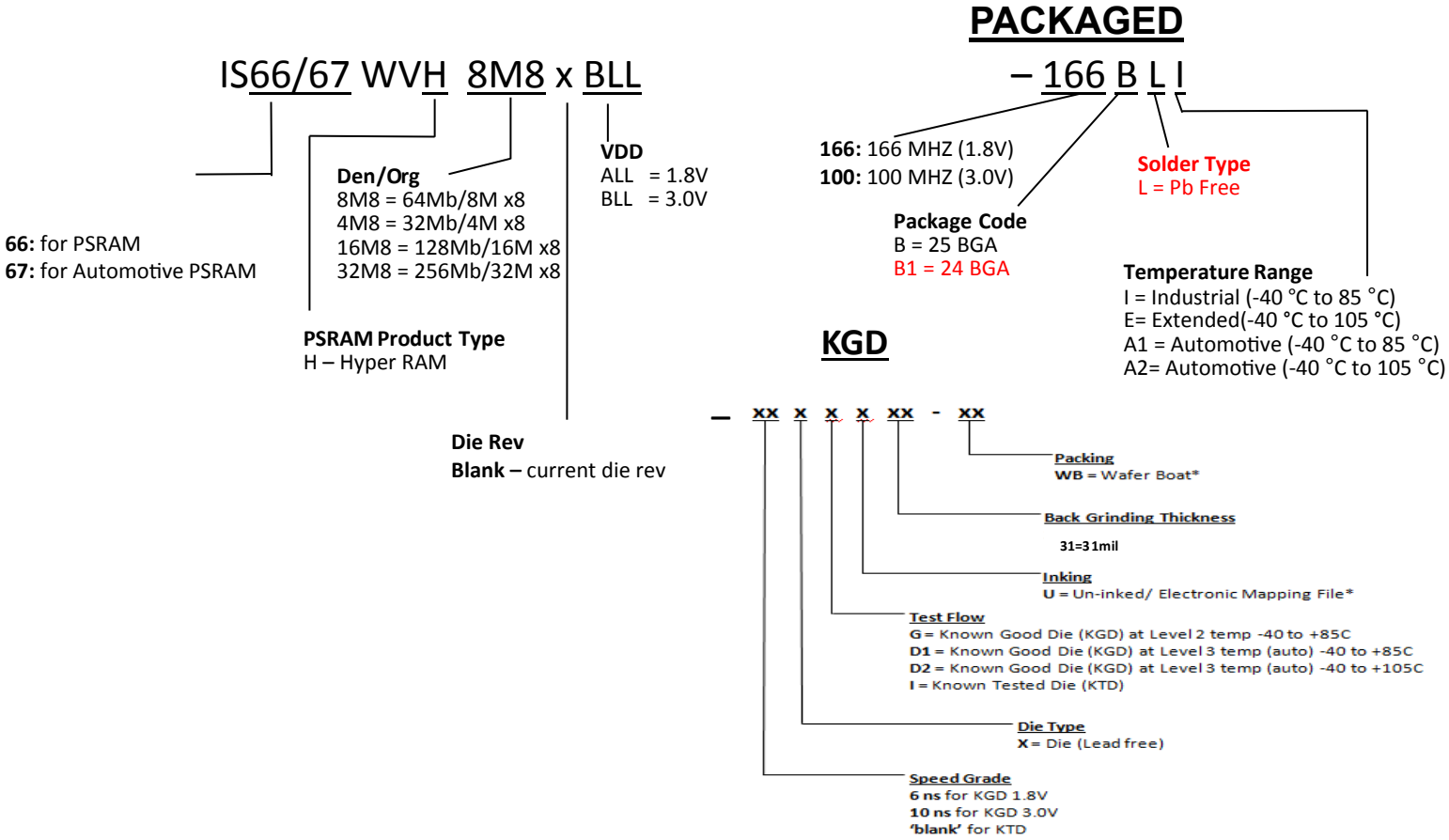



***Notes:***

1. *B1 is a RFU on the 3.0V device and is assigned to CK# on the 1.8V device.*
2. *B5 and C5 are Reserved for Future Use (RFU), internally connected, and are recommended to be connected to VSS to minimize noise. However, signals on these balls are ignored and the balls may be left floating or used in a PCB routing channels if necessary.*
3. *C2 is an RFU and must be floating or be pulled high.*

# 17. ORDERING RULE & INFORMATION

## 17.1 ORDERING RULE



## 17.2 ORDERING INFORMATION

### Industrial Temperature Range: (-40°C to +85°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 8Mx8 | 166 | IS66WVH8M8ALL-166BLI | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS66WVH8M8ALL-166B1LI | 24-ball TFBGA, Pb Free, 5x5 Array |
| | 133 | IS66WVH8M8ALL-133BLI | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS66WVH8M8ALL-133B1LI | 24-ball TFBGA, Pb Free, 5x5 Array |

### Extended Temperature Range: (-40°C to +105°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 8Mx8 | 166 | IS66WVH8M8ALL-166BLE | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS66WVH8M8ALL-166B1LE | 24-ball TFBGA, Pb Free, 5x5 Array |
| | 133 | IS66WVH8M8ALL-133BLE | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS66WVH8M8ALL-133B1LE | 24-ball TFBGA, Pb Free, 5x5 Array |

### Automotive A1 Temperature Range: (-40°C to +85°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 8Mx8 | 166 | IS67WVH8M8ALL-166BLA1 | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS67WVH8M8ALL-166B1LA1 | 24-ball TFBGA, Pb Free, 5x5 Array |
| | 133 | IS67WVH8M8ALL-133BLA1 | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS67WVH8M8ALL-133B1LA1 | 24-ball TFBGA, Pb Free, 5x5 Array |

### Automotive A2 Temperature Range: (-40°C to +105°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 8Mx8 | 166 | IS67WVH8M8ALL-166BLA2 | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS67WVH8M8ALL-166B1LA2 | 24-ball TFBGA, Pb Free, 5x5 Array |
| | 133 | IS67WVH8M8ALL-133BLA2 | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS67WVH8M8ALL-133B1LA2 | 24-ball TFBGA, Pb Free, 5x5 Array |

## Ordering Information :
## BLL : VDD 2.7V~3.6V, VDDQ 2.7V~3.6V

### Industrial Temperature Range: (-40°C to +85°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 8Mx8 | 100 | IS66WVH8M8BLL-100BLI | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS66WVH8M8BLL-100B1LI | 24-ball TFBGA, Pb Free, 5x5 Array |

### Extended Temperature Range: (-40°C to +105°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 8Mx8 | 100 | IS66WVH8M8BLL-100BLE | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS66WVH8M8BLL-100B1LE | 24-ball TFBGA, Pb Free, 5x5 Array |

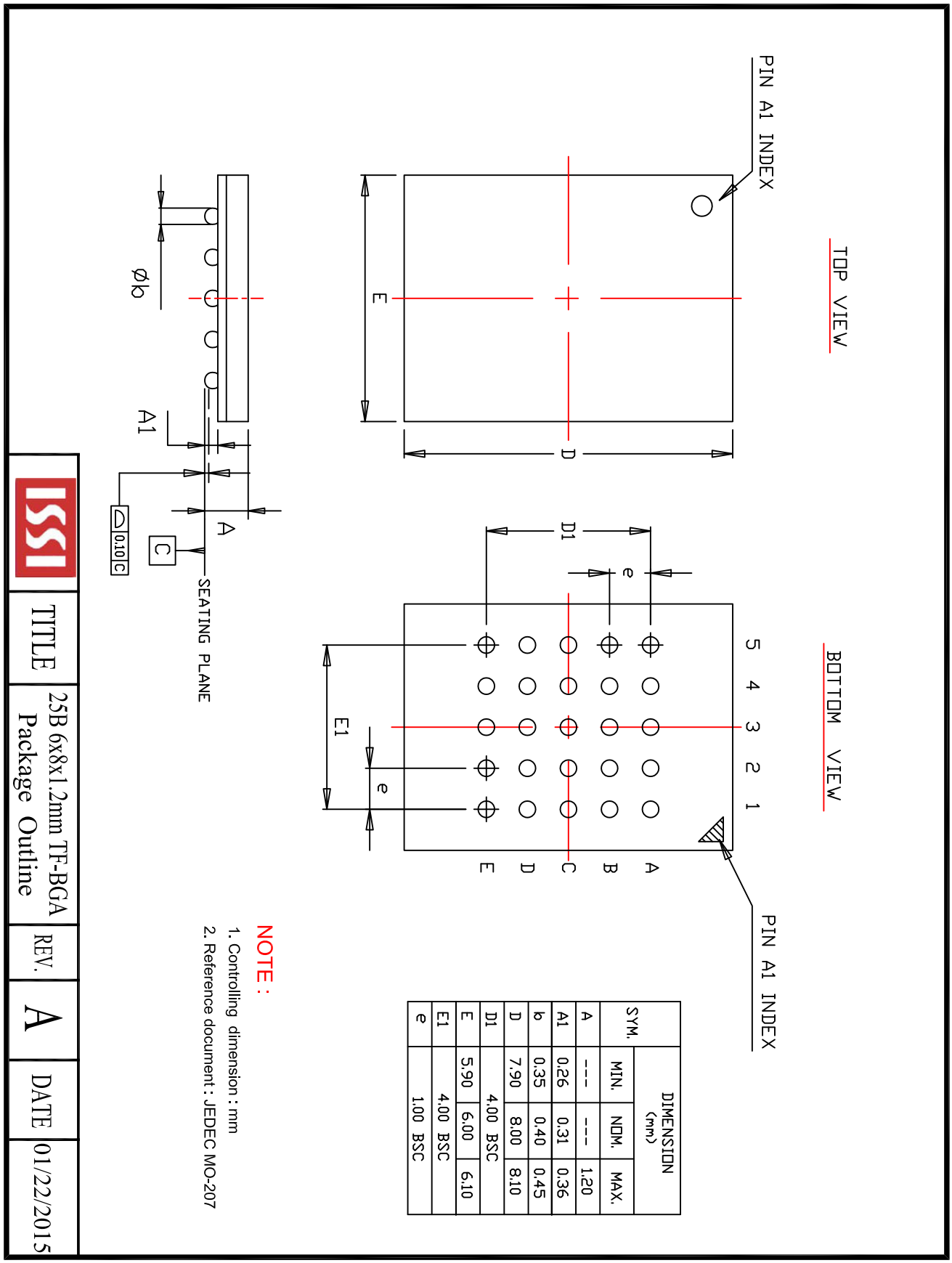
### Automotive A1 Temperature Range: (-40°C to +85°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 16Mx8 | 100 | IS67WVH8M8BLL-100BLA1 | 25-ball TFBGA, Pb Free, 5x5 Array |
| | | IS67WVH8M8BLL-100B1LA1 | 24-ball TFBGA, Pb Free, 5x5 Array |

### Automotive A2 Temperature Range: (-40°C to +105°C)

| Config. | Frequency (MHz) | Order Part No. | Package |
|---|---|---|---|
| 16Mx8 | 100 | IS67WVH16M8BLL-100BLA2 | 25-ball TFBGA, Pb Free, 5x5 Array |
| 16Mx8 | 100 | IS67WVH16M8BLL-100B1LA2 | 24-ball TFBGA, Pb Free, 5x5 Array |

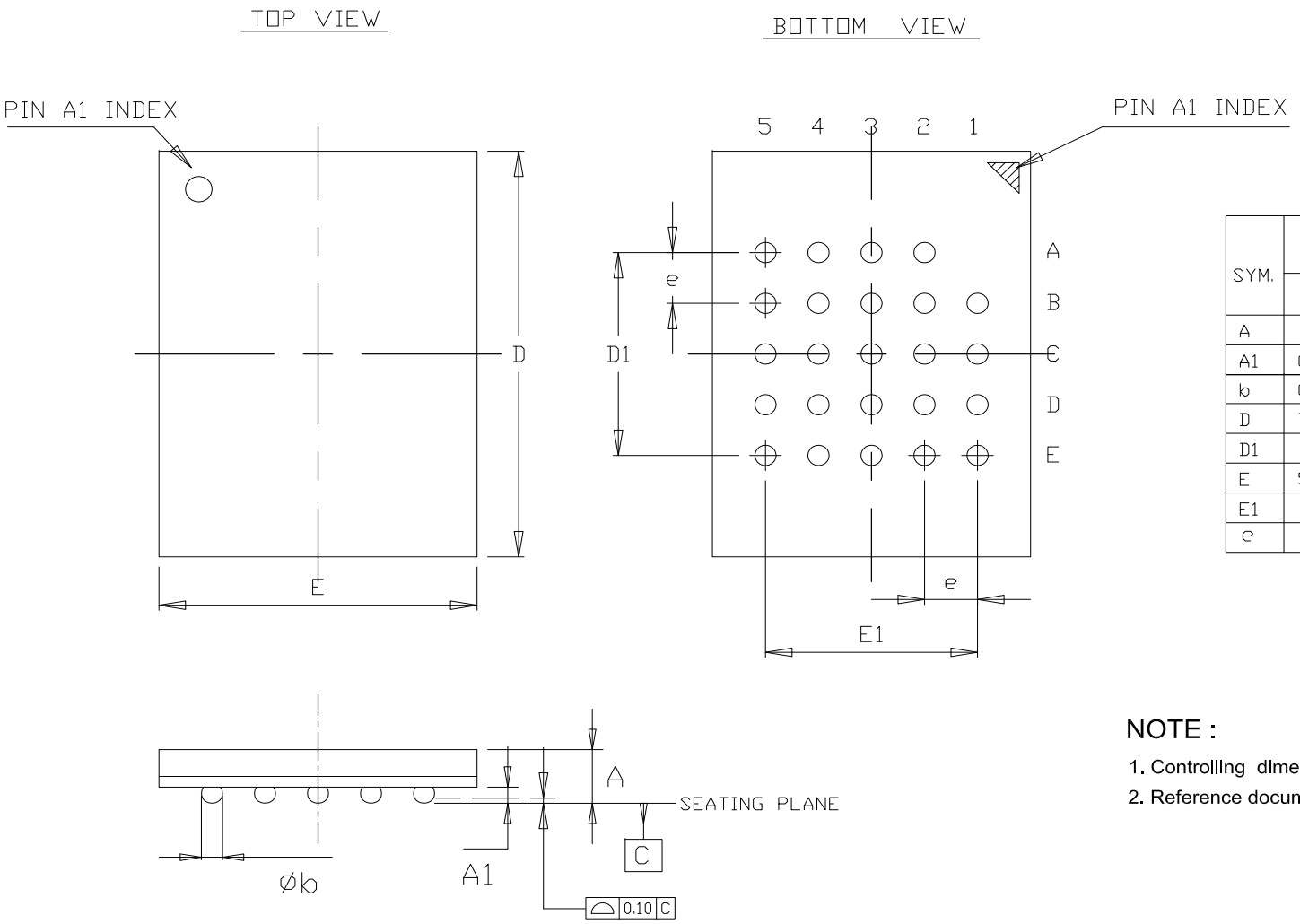
## Package Outline Drawing:
## 25B 6x8x1.2mm TF-BGA




| DIMENSION (mm) | | | |
|---|---|---|---|
| SYM. | MIN. | NOM. | MAX. |
| A | --- | --- | 1.20 |
| A1 | 0.26 | 0.31 | 0.36 |
| b | 0.35 | 0.40 | 0.45 |
| D | 7.90 | 8.00 | 8.10 |
| D1 | | 4.00 BSC | |
| E | 5.90 | 6.00 | 6.10 |
| E1 | | 4.00 BSC | |
| e | | 1.00 BSC | |

NOTE :
1. Controlling dimension : mm
2. Reference document : JEDEC MO-207

| | |
|---|---|
| TITLE | 25B 6x8x1.2mm TF-BGA Package Outline |
| REV. | A |
| DATE | 01/22/2015 |




| SYM. | DIMENSION (mm) | | |
|---|---|---|---|
| | MIN. | NOM. | MAX. |
| A | --- | --- | 1.20 |
| A1 | 0.26 | 0.31 | 0.36 |
| b | 0.35 | 0.40 | 0.45 |
| D | 7.90 | 8.00 | 8.10 |
| D1 | 4.00 BSC | | |
| E | 5.90 | 6.00 | 6.10 |
| E1 | 4.00 BSC | | |
| e | 1.00 BSC | | |

NOTE :

1. Controlling dimension : mm
2. Reference document : JEDEC MO-207

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ISSI | TITLE | 24B 6x8x1.2mm TF-BGA Package Outline | REV. | A | DATE | 11/20/2014 |



Integrated Silicon Solution, Inc.- www.issi.com
Rev.00C
05/01/2015