Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

HyperBus™ Specification

Low Signal Count, High Performance DDR Bus

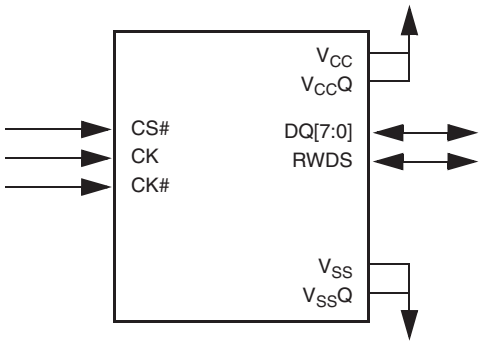
Distinctive Characteristics

General-Purpose Master-Slave Bus

- Connecting Host System Processor / ASIC with Memory and Peripherals
- HyperFlash™ NOR non-volatile memory
- HyperRAM Self-refresh DRAM

HyperBus 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 400MB/s
- Double-Data Rate (DDR) — two data transfers per clock
- 200-MHz clock rate (400 MB/s) at 1.8V/3.0V V_{CC}
- Sequential burst transactions
- Configurable burst characteristics
  - Wrapped burst length options:
    - 128 Bytes (64 clocks)
    - 64 Bytes (32 clocks)
    - 32 Bytes (16 clocks)
    - 16 Bytes (8 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 options
  - 24-Ball FBGA footprint for single or multi-chip packages

Contents

1. General Description 3

2. HyperBus Signal Descriptions 7
2.1 Input/Output Summary 7

3. HyperBus Protocol 8
3.1 Command/Address Bit Assignments 9
3.2 Read Transactions 12
3.3 Write Transactions with Initial Latency 14
3.4 Write Transactions without Initial Latency 16

4. Memory Space 17

5. Register Space 17
5.1 Device Identification Registers 17
5.2 Configuration Registers 18

6. HyperBus Connection Descriptions 22
6.1 Other Connectors Summary 22
6.2 Device Connection Diagrams 23
6.3 HyperBus Block Diagram 24

7. Interface States 25
7.1 Power Conservation Modes 26

8. Electrical Specifications 27
8.1 Absolute Maximum Ratings 27
8.2 Latchup Characteristics 27
8.3 Operating Ranges 28
8.4 DC Characteristics 28
8.5 Power-On Reset 29
8.6 Power Down 31
8.7 Hardware Reset 31

9. Timing Specifications 32
9.1 Key to Switching Waveforms 32
9.2 AC Test Conditions 32
9.3 AC Characteristics 33

10. Physical Interface 40
10.1 FBGA 24-Ball 5 x 5 Array Footprint 40
10.2 Multi-Chip Packages 40
10.3 Physical Diagrams 41

11. Document History Page 43

1. General Description

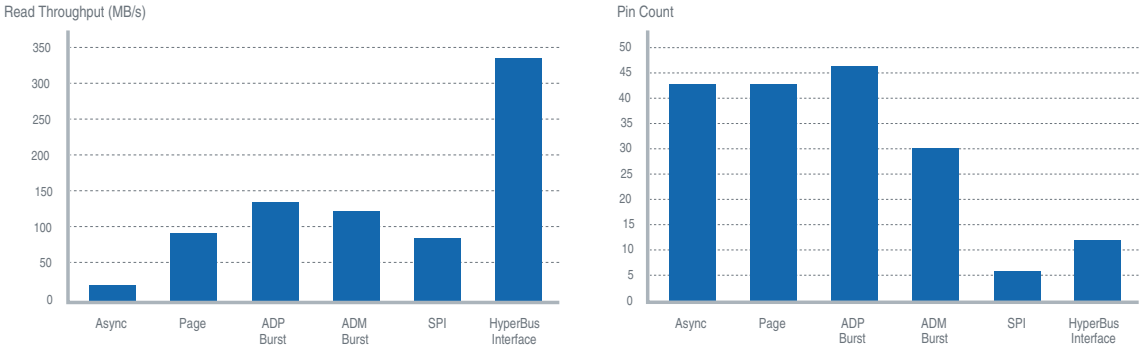
HyperBus™ products use the high-performance HyperBus to connect between a host system master and one or more slave interfaces. HyperBus is used to connect microprocessor, microcontroller, or ASIC devices with random access NOR flash memory, RAM, or peripheral devices.

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.

Parallel flash and PSRAM have long been the standard for simple interface, high performance, random access memory, used for embedded system code execution and data storage. However, parallel interface memory requires a high signal count with separate control, address, and data connections that involve 45 or more signals. There have been parallel interface variations that reduce the signal count by multiplexing some address and data signals yet, may still involve 20 or more signals. These high signal counts provide high data throughput but, at the cost of many connectors on the host system processor or ASIC and, higher cost multi-layer printed circuit boards (PCB) that suffer from signal routing congestion.

Many systems have moved to serial interface memories to reduce the number of signals needed for the connection to memory and to free up host system connections for use by other features or to reduce package and multi-layer PCB cost. However, serial memories generally have lower data throughput or longer random access time, which may relegate serial memories to the role of mainly transferring code and data to DRAM memory for random access and high throughput. But, such shadowing architectures cause a duplication of memory space that means paying twice for memory, once for non-volatile serial memory and again for additional DRAM to access the code and data during system operation.

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.

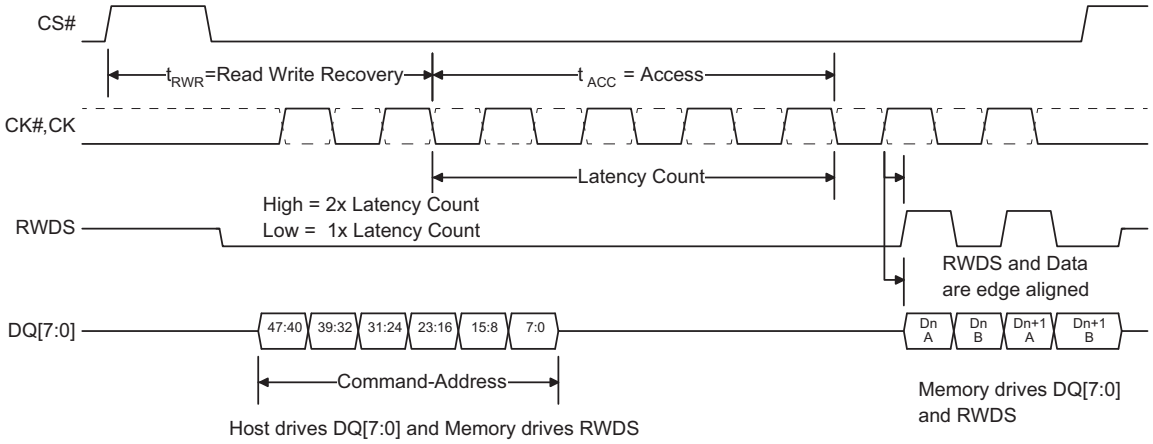
HyperBus has a single master (host system) interface and one or more slave interface devices.

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 1.1 Read Transaction, Single Initial Latency Count



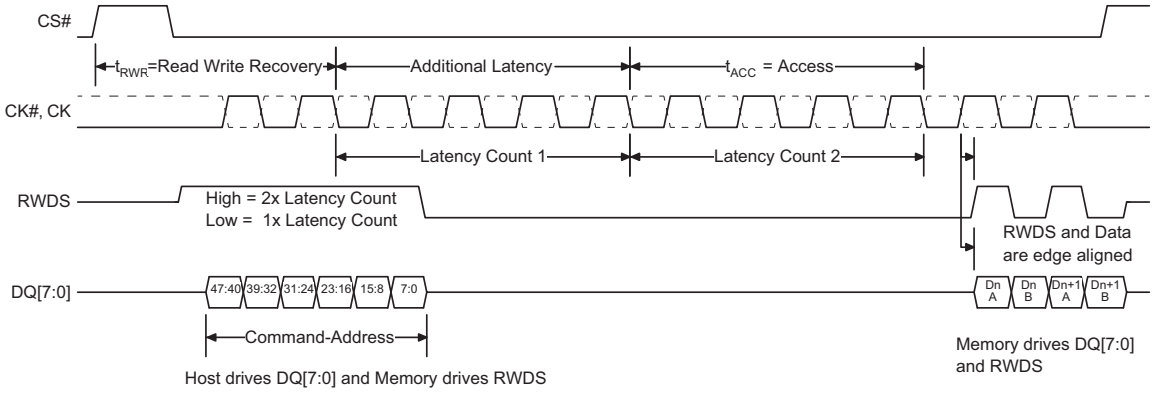
The Read/Write Data Strobe (RWDS) is a bidirectional signal that indicates:

- 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.

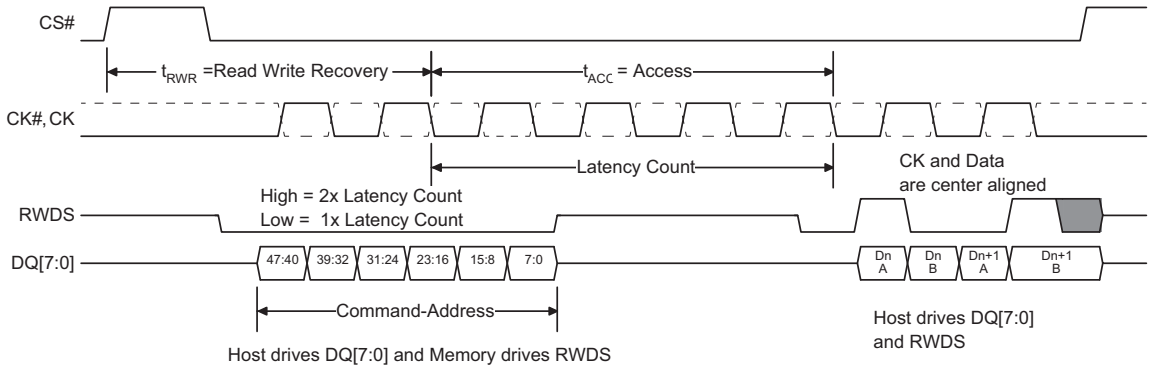
During read data transfers, RWDS is a read data strobe with data values edge aligned with the transitions of RWDS.

Figure 1.2 Read Transaction, Additional Latency Count



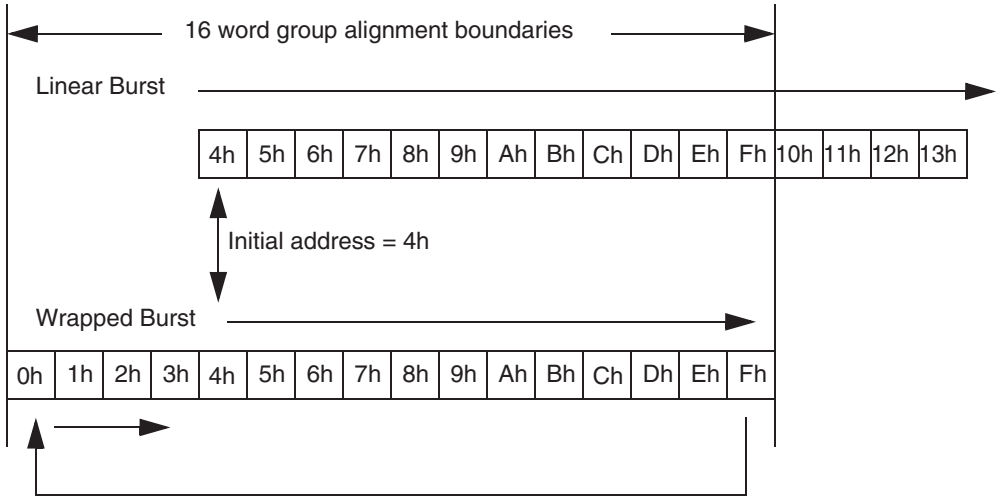
During write data transfers, RWDS indicates whether each data byte transfer is masked with RWDS High (invalid and prevented from changing the byte location in a memory) or not masked with RWDS Low (valid and written to a memory). Data masking may be used by the host to byte align write data within a 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 1.3 Write Transaction, Single Initial Latency Count



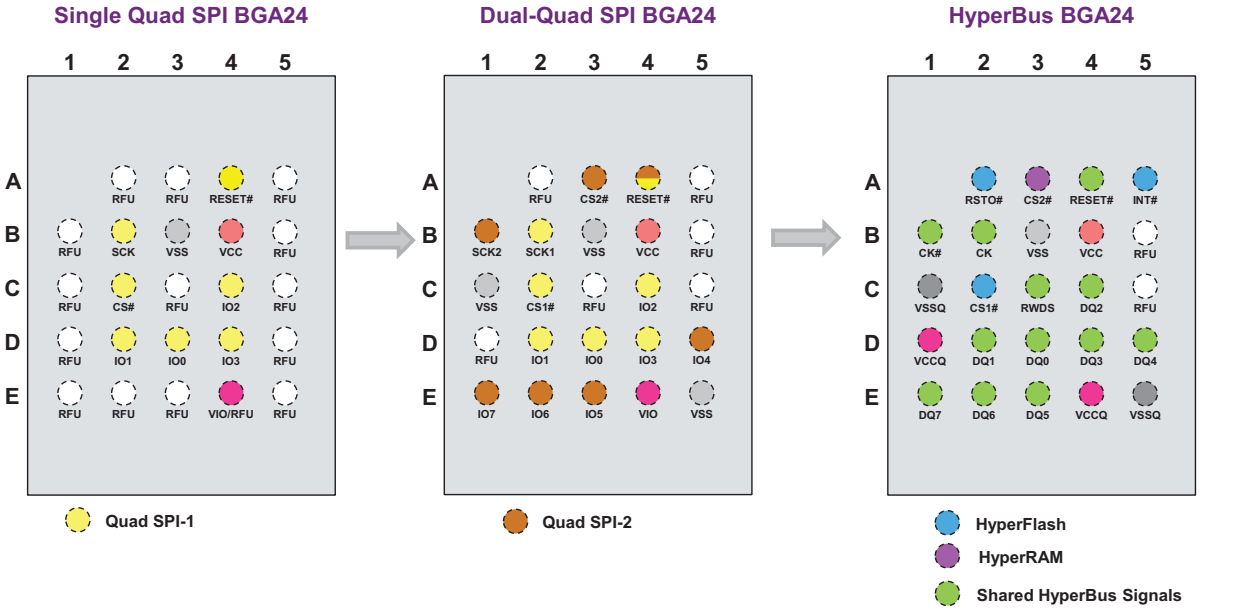
Read and write transactions are burst oriented, transferring the next sequential word during each clock cycle. Each individual read or write transaction can use either a wrapped or linear burst sequence.

Figure 1.4 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. Because each transaction command selects the type of burst sequence for that transaction, wrapped and linear bursts transactions can be dynamically intermixed as needed.

HyperBus devices are provided with an FBGA package footprint similar to that used by existing Quad and Dual Quad Serial Peripheral Interface (QSPI and DQSPI) memories. A single PCB footprint layout can be used for single QSPI, DQSPI, or HyperBus devices. This enables a wide range of system performance options on the same set of signal lines, in the same small package footprint. This footprint provides two chip select inputs that allow a single package footprint to hold two HyperBus devices within the same Multiple-Chip-Package (MCP).



Ball A3 is used as a second die chip select in a dual die package for added density

HyperBus Specification

2. HyperBus Signal Descriptions

2.1 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 2.1 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. **Single Ended Clock.** CK# is not used on 3.0V devices, only a single ended CK is used. 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 R WDS. Slave input during data transfer in write transactions to function as a data mask. (High = additional latency, Low = no additional latency).

Table 2.2 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.

3. 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, that is, 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.

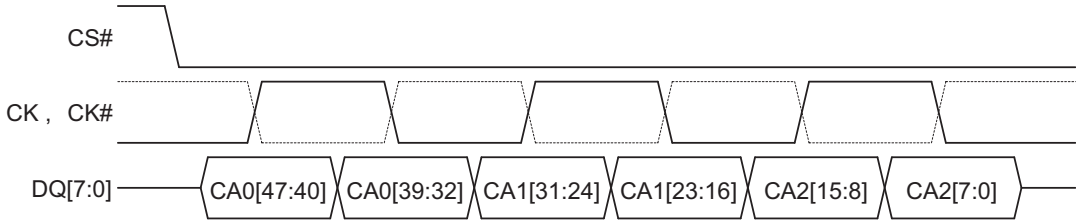
Data is always transferred as 16-bit values with the first eight bits (byte A) transferred on a High going CK (write data) or following a RWDS rising edge (read data) and the second eight bits (byte B) is transferred on the Low going CK edge or following the falling RWDS edge.

After 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.

3.1 Command/Address Bit Assignments

Figure 3.1 Command-Address Sequence



Notes:

1. *The 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 3.1 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]

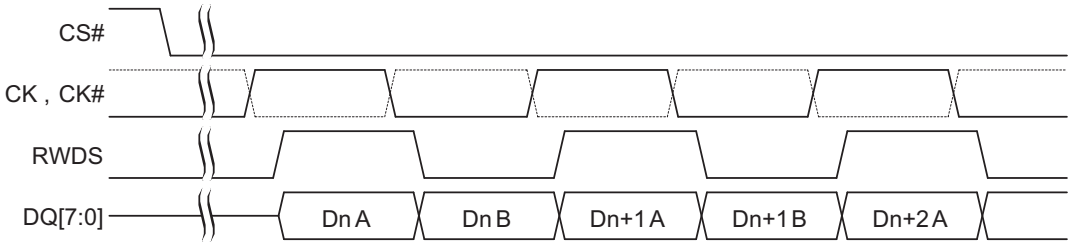
Table 3.2 Command/Address Bit Assignments

CA Bit#	Bit Name	Bit Function
47	R/W#	Identifies the transaction as a read or write. R/W#=1 indicates a Read transaction R/W#=0 indicates a Write transaction
46	Address Space (AS)	Indicates whether the read or write transaction accesses the memory or register space. AS=0 indicates memory space AS=1 indicates the register space The register space is used to access device ID and Configuration registers.
45	Burst Type	Indicates whether the burst will be linear or wrapped. Burst Type=0 indicates wrapped burst 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 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. 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 and 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 8 Gbyte (64 Gbit) 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 Terabyte address space

Figure 3.2 Data Placement During a Read Transaction



Notes:

1. *Figure 3.2 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.*

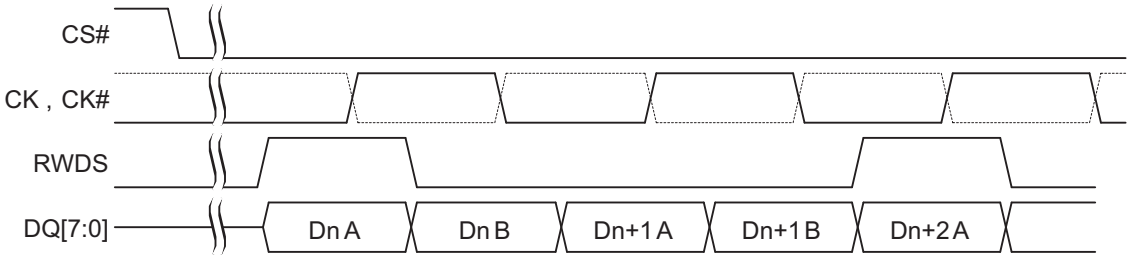
Table 3.3 Data Bit Placement During Read or Write Transaction (Sheet 1 of 2)

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: *The first byte of each word read or written is the "A" byte and the second is the "B" byte.* *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.* 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 3.3 Data Bit Placement During Read or Write Transaction (Sheet 2 of 2)

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: 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. 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. 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 3.3 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.*

3.2 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

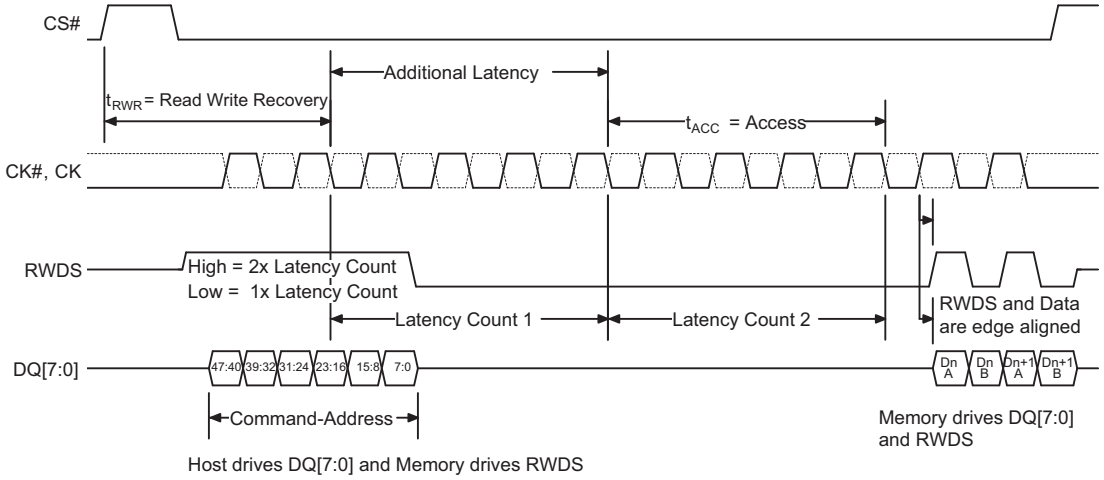
clocks have been completed the memory starts to simultaneously transition the Read-Write Data Strobe (RWDS) and output the target data.

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, 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.

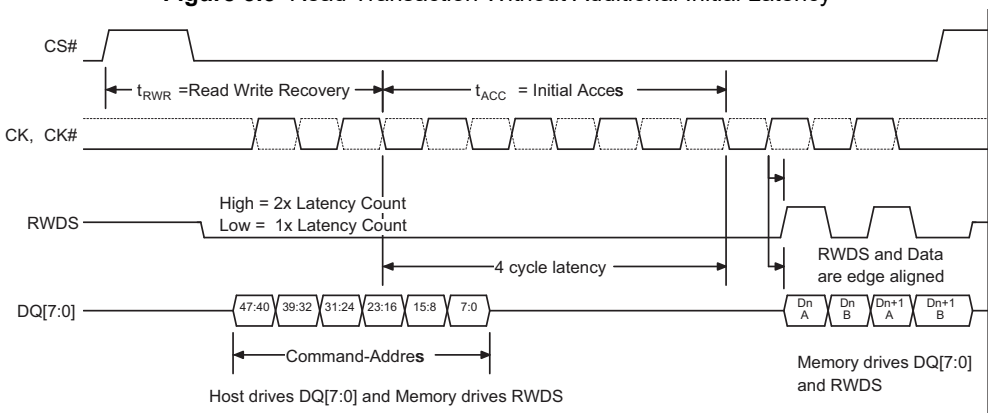
Figure 3.4 Read Transaction with Additional Initial Latency



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 3.5 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.*

3.3 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 transferred.

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.

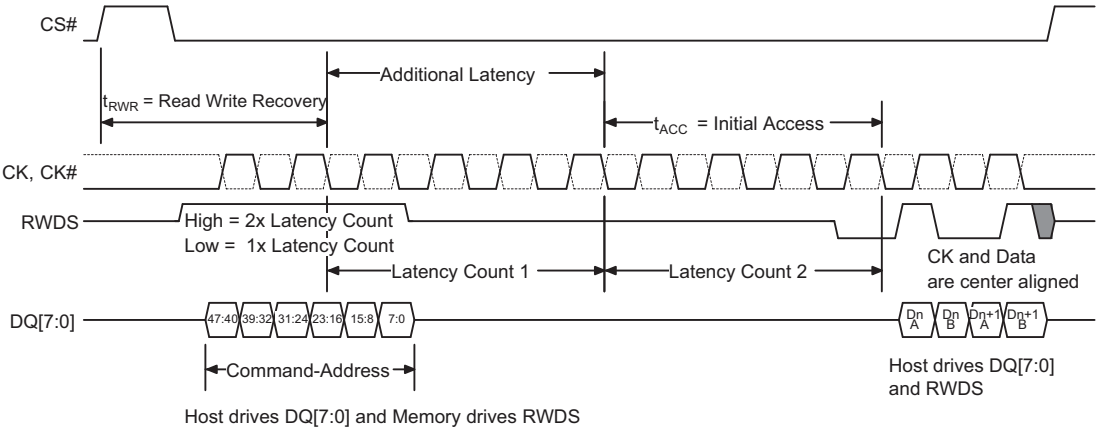
Data will continue to be transferred as long as the HyperBus master continues to transition the clock while CS# is Low. Legacy format wrapped bursts will continue to wrap within the burst length. Hybrid wrap will wrap once then switch to linear burst starting at the next wrap boundary. Linear burst accepts 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.

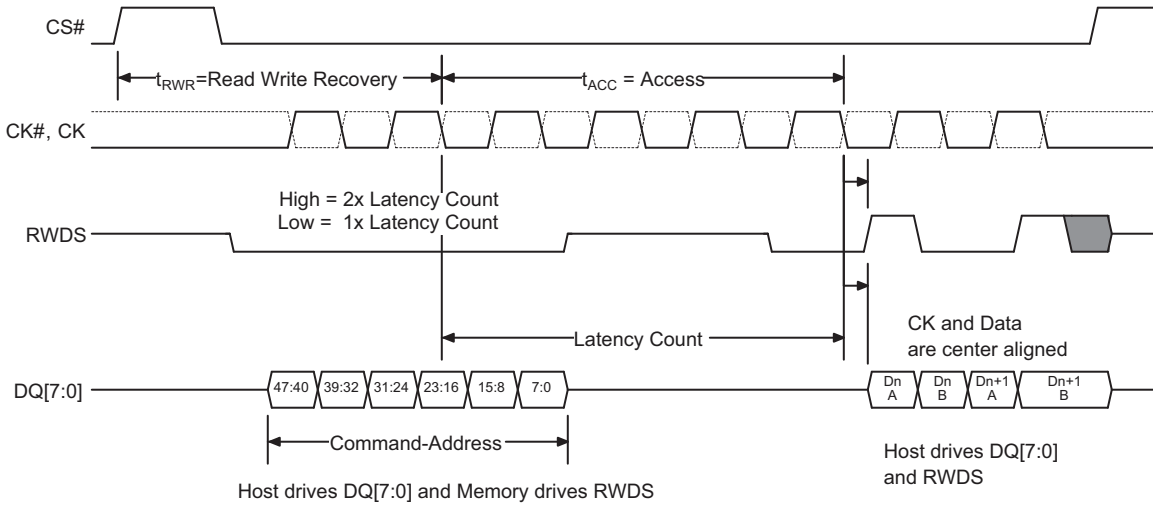
Figure 3.6 Write Transaction with Additional Initial Latency



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. *In this example, RWDS indicates that additional initial latency cycles are required.*
5. *At the end of Command-Address cycles the memory stops driving RWDS to allow the host HyperBus master to begin driving RWDS. The master must drive RWDS to a valid Low before the end of the initial latency to provide a data mask preamble period to the slave.*
6. *During data transfer, RWDS is driven by the host to indicate which bytes of data should be either masked or loaded into the array.*
7. *The figure shows RWDS masking byte A0 and byte B1 to perform an unaligned word write to bytes B0 and A1.*

Figure 3.7 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. *In this example, RWDS indicates that there is no additional latency required.*
3. *At the end of Command-Address cycles the memory stops driving RWDS to allow the host HyperBus master to begin driving RWDS. The master must drive RWDS to a valid Low before the end of the initial latency to provide a data mask preamble period to the slave.*
4. *During data transfer, RWDS is driven by the host to indicate which bytes of data should be either masked or loaded into the array.*
5. *The figure shows RWDS masking byte A0 and byte B1 to perform an unaligned word write to bytes B0 and A1.*

3.4 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, that is, 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#.

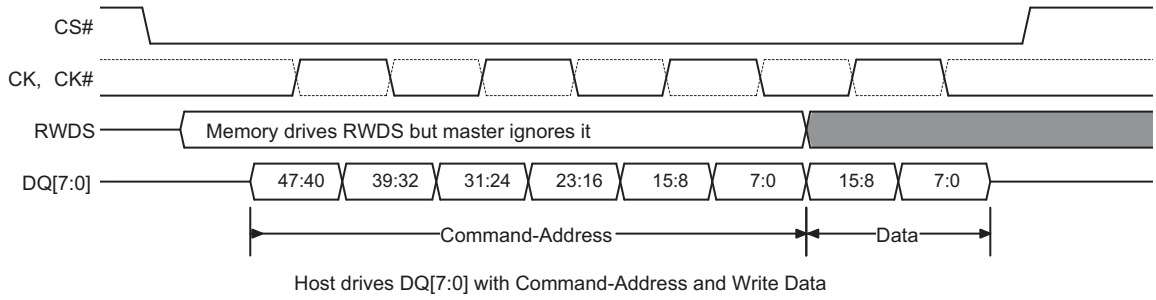
The first byte of data in each word is presented on the rising edge of CK and the second byte is presented on the falling edge of CK. Write data is center aligned with the clock inputs. Write transfers can be ended at any time by bringing CS# High when clock is idle. The clock is not required to be free-running.

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 3.8 Write Operation without Initial Latency



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

HyperBus Device Software Interface

4. 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.

5. 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.

5.1 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 Cypress 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 5.1 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) 0001 - Cypress 0010 to 1111 - Reserved

.

Table 5.2 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 0001 to 1101 - Reserved 1110 - HyperFlash 1111 - Reserved

5.2 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 5.3 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. 1 - Normal operation (default)
14-12	Drive Strength	000 - Device dependent impedance (default) 001 - Device dependent impedance 010 - Device dependent impedance 011 - Device dependent impedance 100 - Device dependent impedance 101 - Device dependent impedance 110 - Device dependent impedance 111 - Device dependent impedance
11-8	Reserved	1 - Reserved (default) Reserved for Future Use. When writing this register, these bits should be set to 1 for future compatibility.

Table 5.3 Example Configuration Register Bit Assignments (Sheet 2 of 2)

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

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

5.2.1 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_{DPDOUT} 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.

5.2.2 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.

5.2.3 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 3.4, *Read Transaction with Additional Initial Latency* on page 13 and Figure 3.6, *Write Transaction with Additional Initial Latency* on page 15. 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.

5.2.4 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.

5.2.5 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.

5.2.6 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 5.4 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 (wrap complete, now linear beyond the end of the initial 128 byte wrap group) 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, (wrap complete, now linear beyond the end of the initial 64 byte wrap group) 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, (wrap complete, now linear beyond the end of the initial 16 byte wrap group) 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, ...

5.2.7 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

6. HyperBus Connection Descriptions

For a description of HyperBus mandatory and optional signals refer to Section 2.1, *Input/Output Summary* on page 7. Following is the description for other package connectors.

6.1 Other Connectors Summary

Table 6.1 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 Cypress. The signal/ball is connected internally. The signal/ball must be left open on the PCB.

6.2 Device Connection Diagrams

Figure 6.1 HyperBus Master with Mandatory and Optional Signals

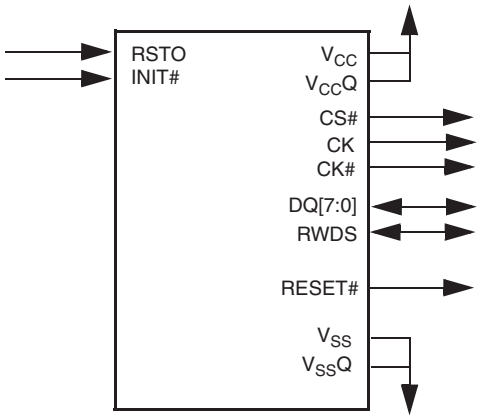
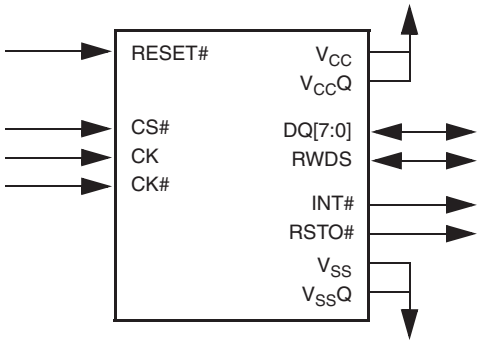
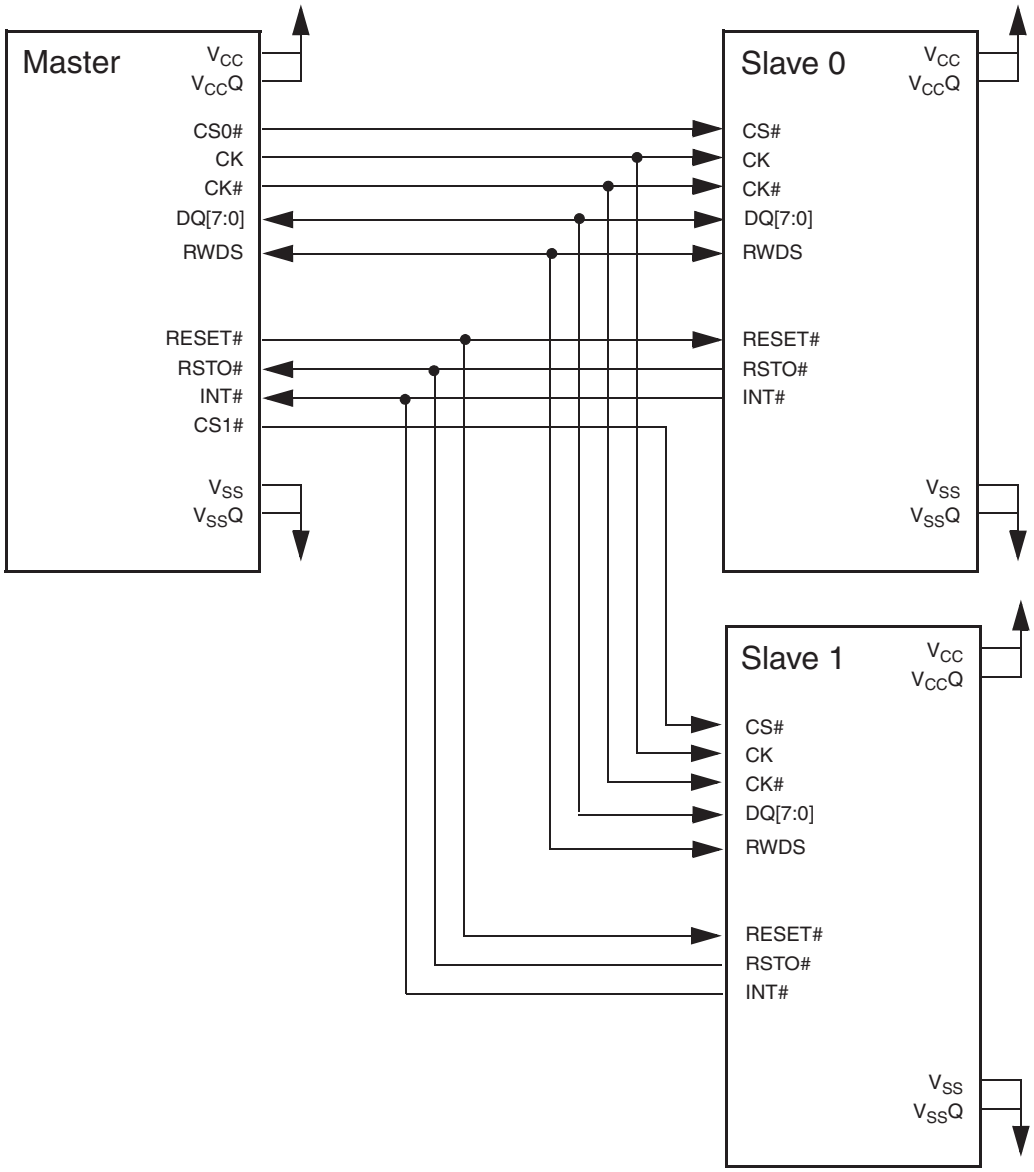


Figure 6.2 HyperBus Slave with Mandatory and Optional Signals



6.3 HyperBus Block Diagram

Figure 6.3 HyperBus Connections, Including Optional Signals



7. Interface States

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

Table 7.1 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 7.1.2, Active Clock Stop on page 26. DPD is described in Section 7.1.3, Deep Power-Down on page 26.*

7.1 Power Conservation Modes

7.1.1 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.

7.1.2 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.

7.1.3 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_{DPDOUT} time. Support for the DPD mode feature is slave device dependent. Refer to the individual device specifications for additional information.

8. Electrical Specifications

8.1 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 8.1. 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 8.2.*
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.*

8.1.1 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 8.1 Maximum Negative Overshoot Waveform

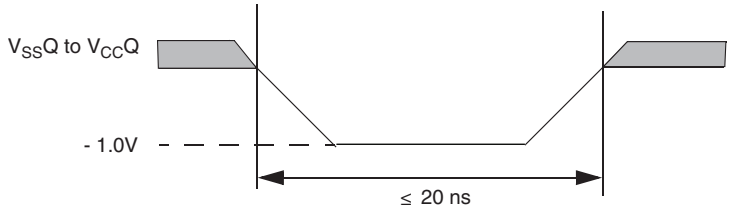
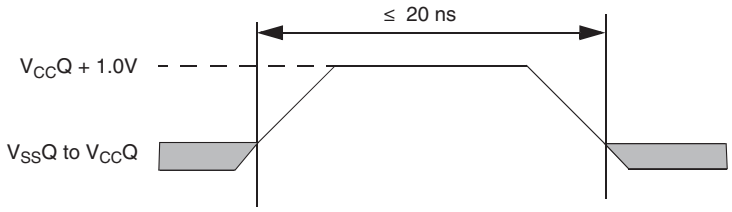


Figure 8.2 Maximum Positive Overshoot Waveform



8.2 Latchup Characteristics

Table 8.1 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

Table 8.1 Latchup Specification

Description	Min	Max	Unit
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}.

8.3 Operating Ranges

Operating ranges define those limits between which the functionality of a device is guaranteed. The operating range is device specific. Consult the device data sheet Ordering Part Number valid combinations to know which operating ranges are supported by a particular device.

8.3.1 Temperature Ranges

Parameter	Symbol	Conditions	Spec		Unit
			Min	Max	
Ambient Temperature	T_A	Industrial Devices	–40	+85	°C
		Industrial Plus Devices	–40	+105	
		Extended Devices	–40	+125	
		Hot Devices	–40	+145	

8.3.2 1.8V Power Supply Voltages

V_{CC} and V_{CCQ} 1.70 V to 2.00V

8.3.3 3.0V Power Supply Voltages

V_{CC} and V_{CCQ} 2.70V to 3.60V

8.4 DC Characteristics

Below are characteristics common to all HyperBus devices. Many other characteristics are device dependent. Refer to the individual device data sheets for additional information.

Table 8.2 DC Characteristics (CMOS Compatible)

Parameter	Description	Test Conditions	Min	Typ	Max	Unit
I_{LI}	Input Leakage Current	V_{IN} = V_{SS} to V_{CC}, V_{CC} = V_{CC} max	—	—	±2.00	µA
I_{LO}	Output Leakage Current	V_{OUT} = V_{SS} to V_{CC}, V_{CC} = V_{CC} max	—	—	±1.00	µA
V_{IL}	Input Low Voltage		-0.50	—	0.30 x V_{CC}	V
V_{IH}	Input High Voltage		0.70 x V_{CC}	—	V_{CC} + 0.30	V
V_{OL}	Output Low Voltage	I_{OL} = 100 µA for DQ[7:0]	—	—	0.15 x V_{CC}	V
V_{OH}	Output High Voltage	I_{OH} = 100 µA for DQ[7:0]	—	0.85 x V_{CC}	—	V

8.4.1 Capacitance Characteristics

These characteristics are device dependent, refer to the individual device data sheets for additional information.

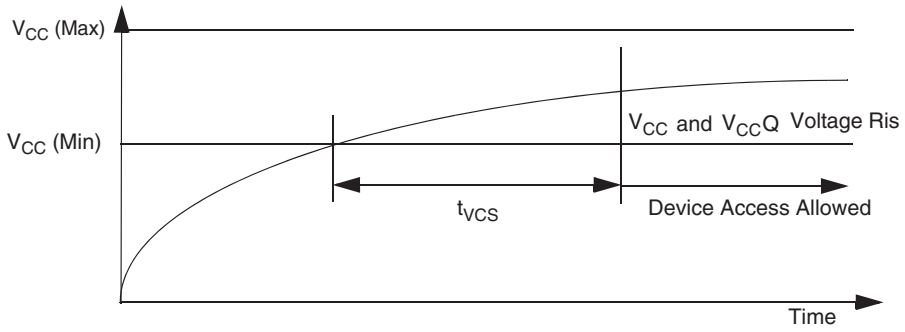
8.5 Power-On Reset

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 8.3 Power On Reset Timing



Note:
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.

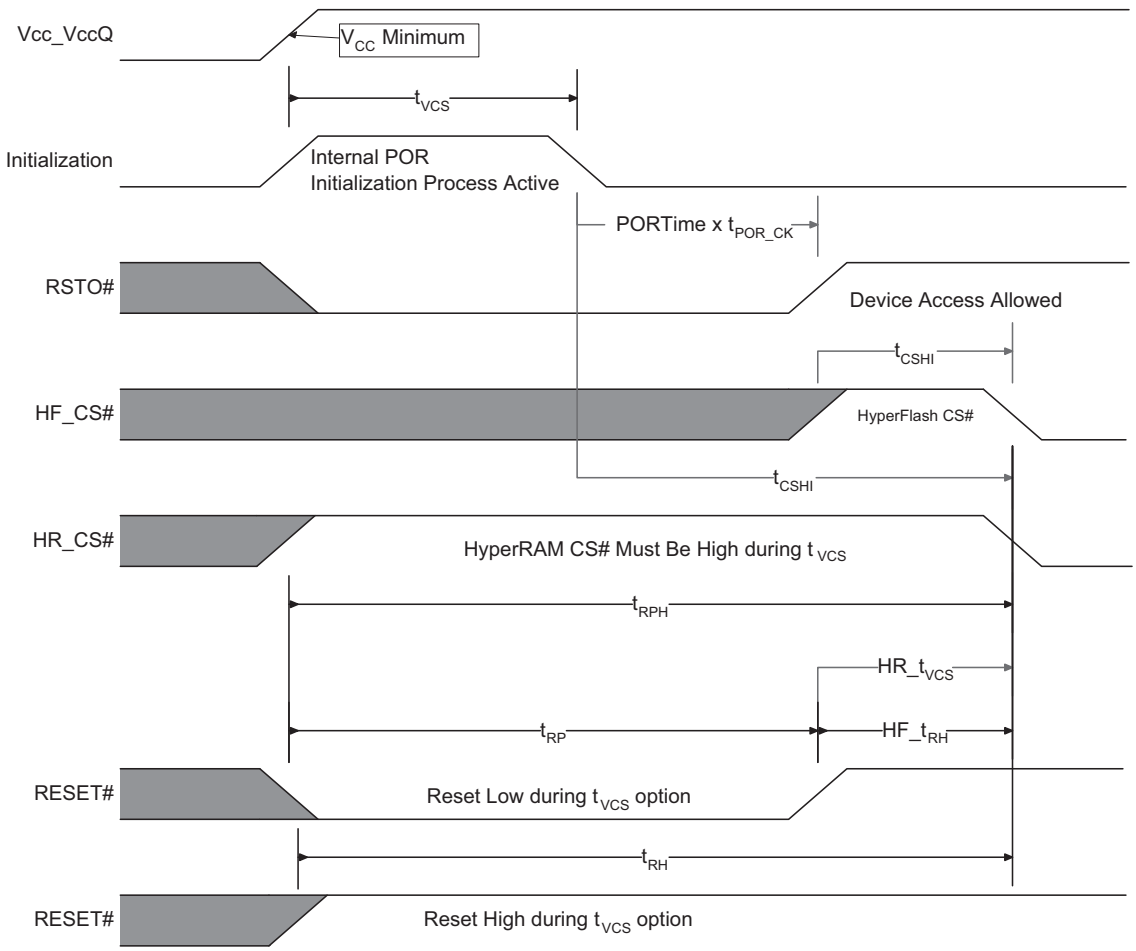
HyperFlash also provides a Reset Output (RSTO#) that the host system may use to drive all host system hardware Reset inputs, including the RESET# input of the HyperFlash driving RSTO#. In this case, the HyperFlash memory will be ready for access following the rise of RSTO# and after the HyperFlash t_{RH} 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 8.4 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 8.3 Power On Reset Parameters

Parameter	Description	Min	Max	Unit
Vcc	1.8V Vcc Power Supply	1.70	2.00	V
Vcc	3V Vcc Power Supply	2.70	3.60	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	400	—	ns
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)	200	—	ns
t_{CSHI}	Chip Select High Between Operations	10	—	ns

Refer to the individual device data sheets for additional information.

8.6 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}). When V_{CC} is below V_{LKO}, a HyperFlash memory array is protected against a program or erase operation. This ensures that no spurious alteration of the flash memory content can occur during power transition. 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.

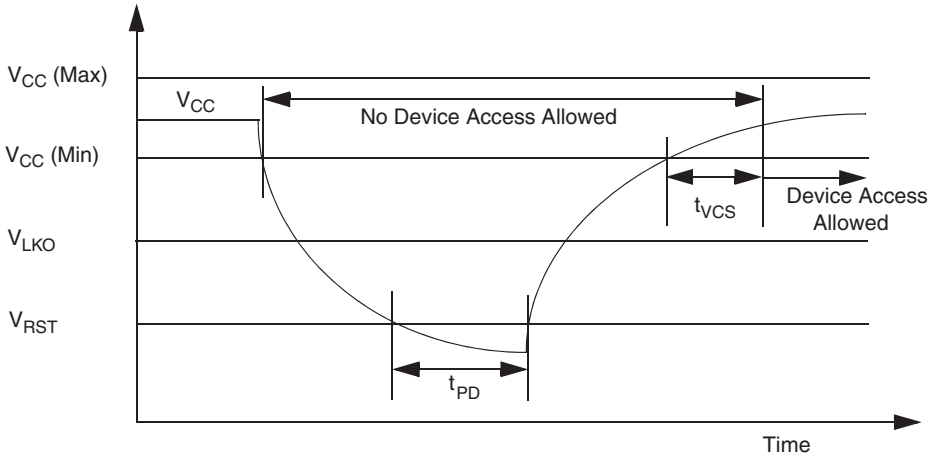
V_{CC} must always be greater than or equal to $V_{CC}Q$ ($V_{CC} \geq V_{CC}Q$).

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 8.5.

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 8.5 Power Down or Voltage Drop



8.7 Hardware Reset

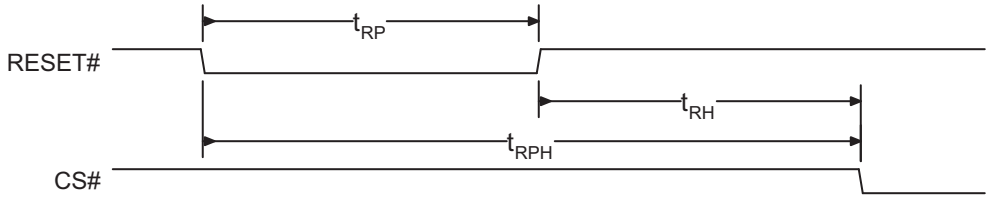
The RESET# input provides a hardware method of returning the device to the standby state.

During hardware reset the HyperBus device will draw ICC5 current. When RESET# continues to be held at V_{SS}, the device draws CMOS standby current (I_{CC4}). While RESET# is Low, bus transactions are not allowed.

A Hardware Reset will cause configuration registers to return to their default values, and can be used to force the device to exit the Deep Power-Down state.

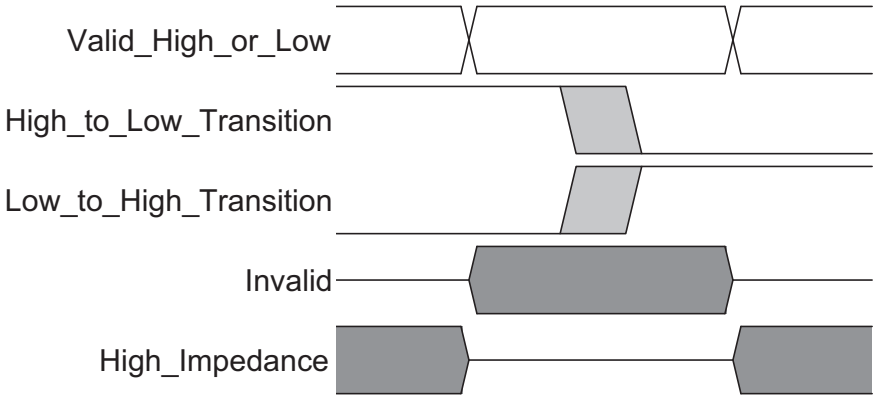
The values for the hardware reset parameters are device dependent. Refer to the individual device data sheets for additional information.

Figure 8.6 Hardware Reset Timing Diagram



9. Timing Specifications

9.1 Key to Switching Waveforms



9.2 AC Test Conditions

Figure 9.1 Test Setup

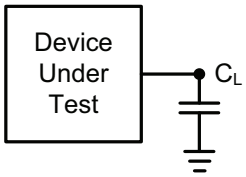


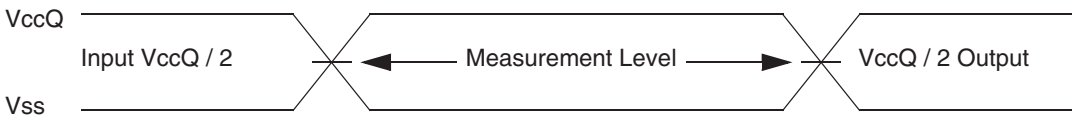
Table 9.1 Test Specification

Parameter	200 MHz	166 MHz	133 MHz	100 MHz	Units
Output load capacitance, C_L	15	20	20	20	pF
Minimum input rise and fall slew rates (Note 1)	2.0	2.0	2.0	2.0	V/ns
Input pulse levels	0.0–V_{CCQ}	0.0–V_{CCQ}	0.0–V_{CCQ}	0.0–V_{CCQ}	V
Input timing measurement reference levels	$V_{CCQ}/2$	$V_{CCQ}/2$	$V_{CCQ}/2$	$V_{CCQ}/2$	V
Output timing measurement reference levels	$V_{CCQ}/2$	$V_{CCQ}/2$	$V_{CCQ}/2$	$V_{CCQ}/2$	V

Notes:

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 9.2 Input Waveforms and Measurement Levels



Note:

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

9.3 AC Characteristics

9.3.1 CLK Characteristics

Figure 9.3 Clock Characteristics

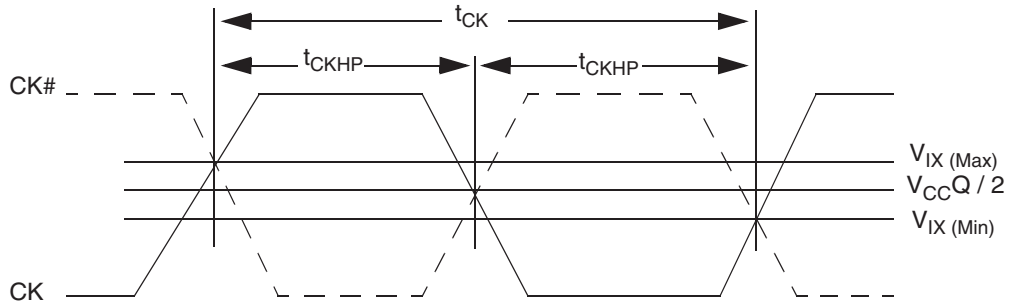


Table 9.2 Clock Timings

Parameter	Symbol	200 MHz		166 MHz		133 MHz		100 MHz		50 MHz (2)		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
CK Period	t_{CK}	5.00	–	6.00	–	7.50	–	10.00	–	20.00	–	ns
CK Half Period - Duty Cycle	t_{CKHP}	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	t_{CK}
CK Half Period at Frequency Min = 0.45 t_{CK} Min Max = 0.55 t_{CK} Min	t_{CKHP}	2.30	2.70	2.70	3.30	3.38	4.13	4.50	5.50	9.00	11.00	ns

Notes:

1. *Clock jitter of ±5 percent 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 9.3 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 $V_{CC}Q$.*

9.3.2 Read Transaction Diagrams

Figure 9.4 Read Timing Diagram – No Additional Latency Required

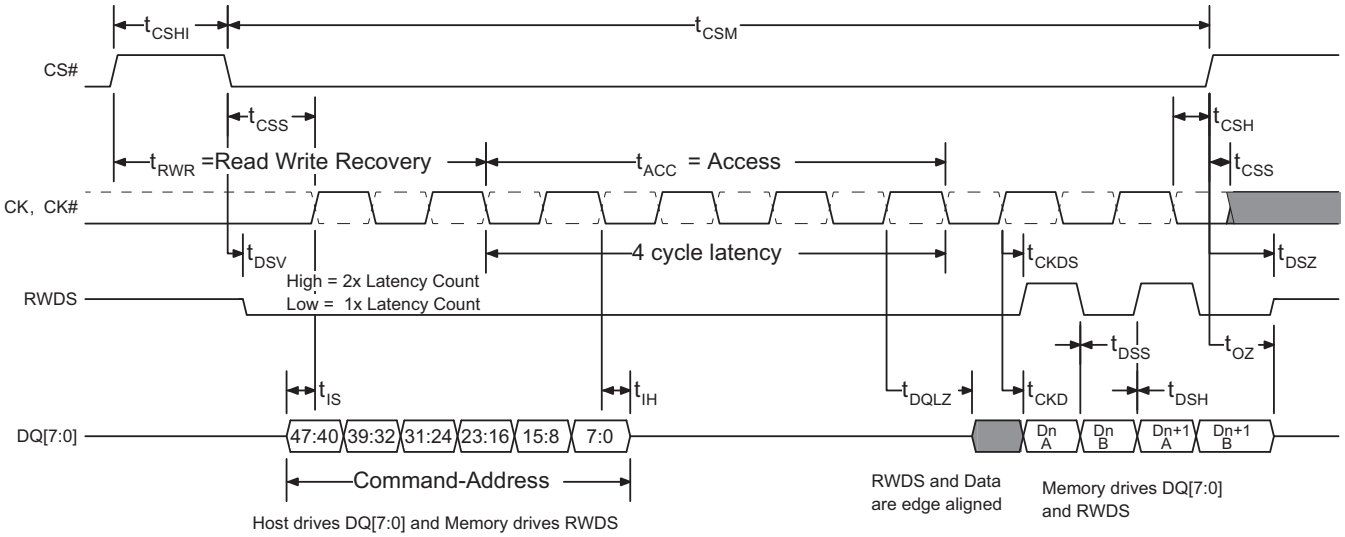
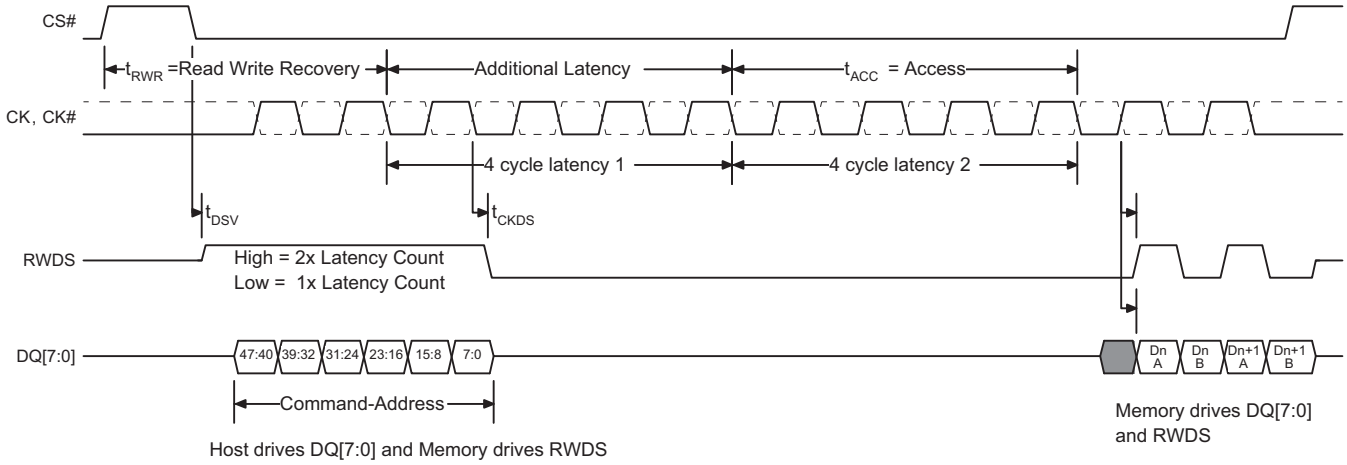


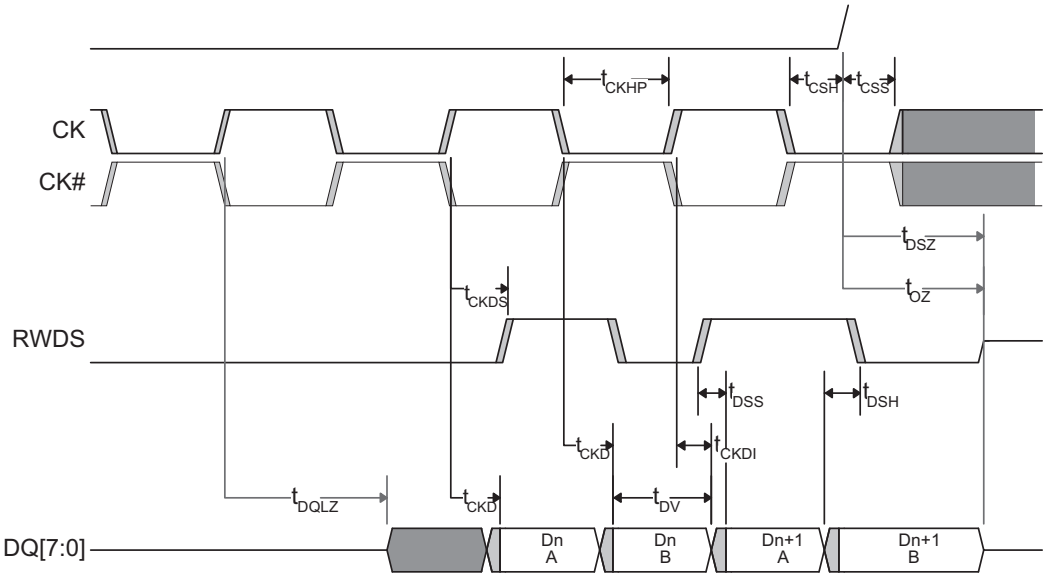
Figure 9.5 Read Timing Diagram – With Additional Latency



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 9.6 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 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.*

9.3.3 Read AC Parameters

The HyperBus read transaction timing parameters vary by device, voltage range, and temperature range. Refer to the individual device data sheets for exact timing information.

However, the following common parameter values may be used for HyperBus operation compatible with all HyperBus devices. These are the more conservative values from either HyperFlash or HyperRAM to provide timing parameters required when both devices are on the same HyperBus.

Table 9.4 HyperBus 1.8V/3.0V Device Common Read Timing Parameters

Parameter	Symbol	200 MHz		166 MHz		133 MHz		100 MHz		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Chip Select High Between Transactions	t_{CSHI}	5.00	–	6.00	–	7.50	–	10.00	–	ns
HyperRAM Read-Write Recovery Time	t_{RWR}	35.00	–	36.00	-	37.50	-	40.00	–	ns
Chip Select Setup to next CK Rising Edge	t_{CSS}	2.00	–	3.00	–	3.00	–	3.00	–	ns
Data Strobe Valid	t_{DSV}	–	5.00	–	12.00	–	12.00	–	12.00	ns
Input Setup	t_{IS}	0.40	–	0.60	–	0.80	–	1.00	–	ns
Input Hold	t_{IH}	0.40	–	0.60	–	0.80	–	1.00	–	ns
HyperFlash Read Initial Access Time	t_{ACC}	–	80.00	–	96.00	–	96.00	–	96.00	ns
HyperRAM Read Initial Access Time		–	35.00	–	36.00		37.50		40.00	ns
Clock to DQs Low Z	t_{DQLZ}	0	–	0	–	0	–	0	–	ns
CK transition to DQ Valid	t_{CKD}	1.20	5.00	1.00	5.50	1.00	5.50	1.00	5.50	ns
CK transition to DQ Invalid	t_{CKDI}	0.40	4.20	0	4.60	0	4.50	0	4.30	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}	1.45	–	1.70	–	2.37	–	3.30	–	ns
CK transition to RWDS valid	t_{CKDS}	1.20	5.00	1.00	5.50	1.00	5.50	1.00	5.50	ns
RWDS transition to DQ Valid	t_{DSS}	–	+0.40	-0.45	+0.45	-0.60	+0.60	-0.80	+0.80	ns
RWDS transition to DQ Invalid	t_{DSH}	–	+0.40	-0.45	+0.45	-0.60	+0.60	-0.80	+0.80	ns
Chip Select Hold After CK Falling Edge	t_{CSH}	0	–	0	–	0	–	0	–	ns
Chip Select Inactive to RWDS High-Z	t_{DSZ}	–	5.00	–	6.00	–	6.00	–	6.00	ns
Chip Select Inactive to DQ High-Z	t_{OZ}	–	5.00	–	6.00	–	6.00	–	6.00	ns
HyperRAM Chip Select Maximum Low Time - Industrial Temperature	t_{CSM}	–	4.00	-	4.00	-	4.00	-	4.00	µs
HyperRAM Chip Select Maximum Low Time - Industrial Plus Temperature		–	1.00	-	1.00	-	1.00	-	1.00	µs

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.*

9.3.4 Write Transaction Diagrams

Figure 9.7 Write Timing Diagram – No Additional Latency

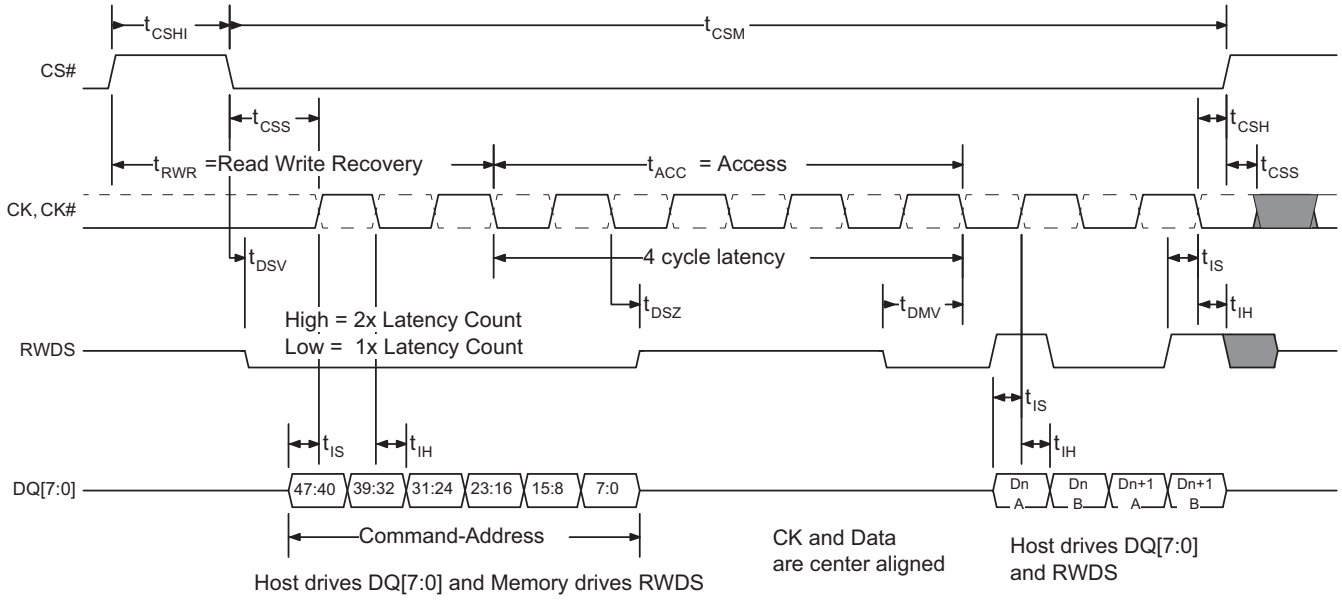
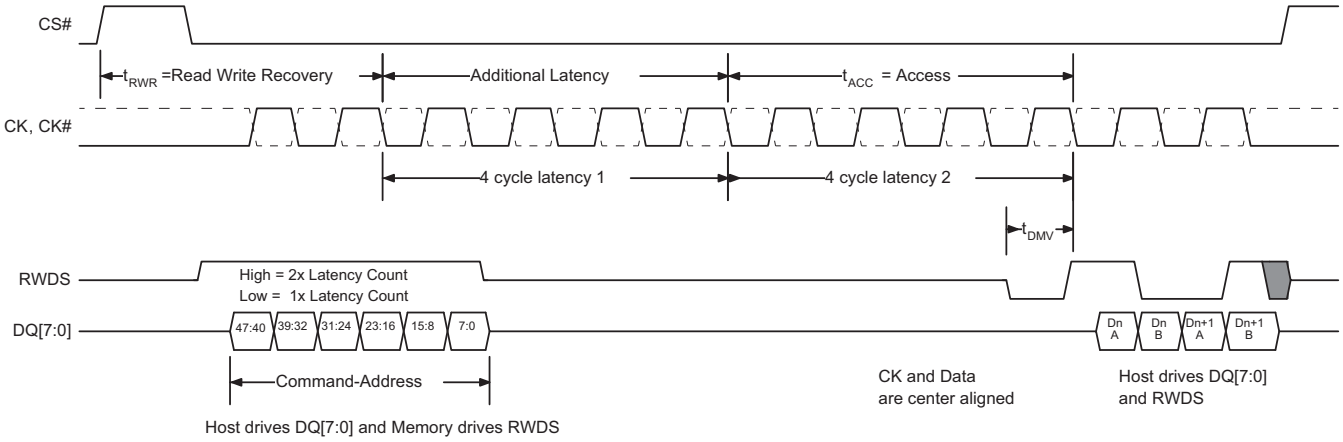


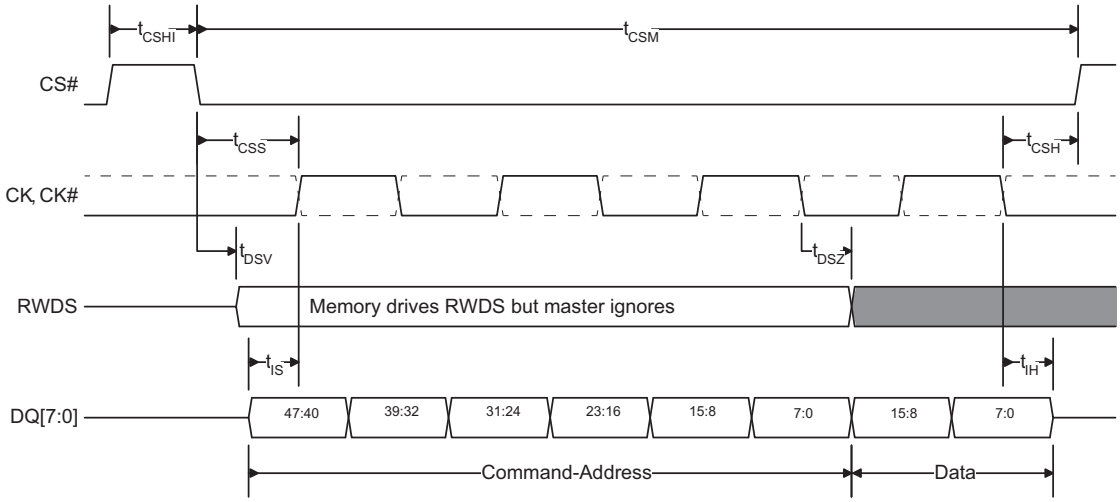
Figure 9.8 Write Timing Diagram – With Additional Latency



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. *At the end of Command-Address cycles the memory stops driving RWDS to allow the host HyperBus master to begin driving RWDS. The master must drive RWDS to a valid Low before the end of the initial latency to provide a data mask preamble period to the slave. This can be done during the last cycle of the initial latency.*
5. *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.*
6. *CK# is only used on the 1.8V device and is shown as a dashed waveform.*
7. *The 3V device uses a single ended CK clock input.*

Figure 9.9 Write Operation without Initial Latency



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, that is, 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.*

9.3.5 Write AC Parameters

The HyperBus write transaction timing parameters vary by device, voltage range, and temperature range. Refer to the individual device data sheets for exact timing information.

However, the following common parameter values may be used for HyperBus operation compatible with all HyperBus devices. These are the more conservative values from either HyperFlash or HyperRAM to provide timing parameters required when both devices are on the same HyperBus.

Table 9.5 HyperBus 1.8V/3.0V Device Common Write Timing Parameters

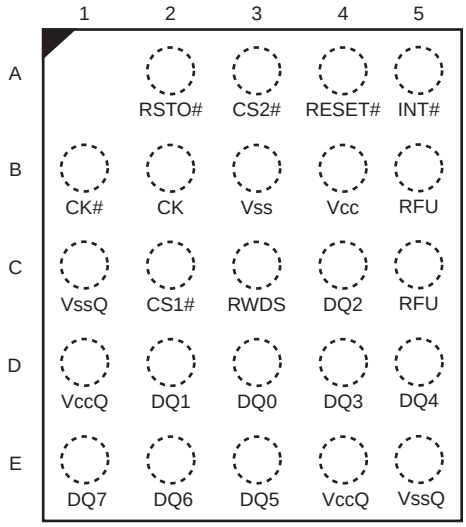
Parameter	Symbol	200 MHz		166 MHz		133 MHz		100 MHz		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Chip Select High Between Transactions	t_{CSHI}	5.00	–	6.00	–	7.50	–	10.00	–	ns
HyperRAM Read-Write Recovery Time	t_{RWR}	35.00	–	36.00	–	37.50	–	40.00	–	ns
Chip Select Setup to next CK Rising Edge	t_{CSS}	2.00	–	3.00	–	3.00	–	3.00	–	ns
Data Strobe Valid	t_{DSV}	–	5.00	–	12.00	–	12.00	–	12.00	ns
Data Mask Valid (RWDS setup to end of initial latency)	t_{DMV}	0	–	0	–	0	–	0	–	ns
Input Setup	t_{IS}	0.40	–	0.60	–	0.80	–	1.00	–	ns
Input Hold	t_{IH}	0.40	–	0.60	–	0.80	–	1.00	–	ns
HyperRAM Memory Space Write Initial Access Time	t_{ACC}	–	35.00	–	36.00		37.50		40.00	ns
Chip Select Hold After CK Falling Edge	t_{CSH}	0	–	0	–	0	–	0	–	ns
Chip Select Inactive or clock to RWDS High-Z	t_{DSZ}	–	5.00	–	6.00	–	6.00	–	6.00	ns
HyperRAM Chip Select Maximum Low Time - Industrial Temperature	t_{CSM}	–	4.00	-	4.00	-	4.00	-	4.00	µs
HyperRAM Chip Select Maximum Low Time - Industrial Plus Temperature		–	1.00	-	1.00	-	1.00	-	1.00	µs

10. Physical Interface

10.1 FBGA 24-Ball 5 x 5 Array Footprint

HyperBus devices are provided in Fortified Ball Grid Array (FBGA), 1 mm pitch, 24-ball, 5 x 5 ball array footprint, with 6mm x 8mm body. The package height is device dependent and may be either 1 mm or 1.2 mm. Consult the device data sheet Ordering Part Number valid combinations section for the package in use.

Figure 10.1 24-Ball FBGA, 6 x 8 mm, 5x5 Ball Footprint, Top View



Notes:

1. *B1 is assigned to CK# on the 1.8V device.*
2. *B1 is a RFU on the 3.0V device*
3. *CS1# and CS2# use is HyperBus device dependent. Discrete device packages use CS1# for HyperFlash CS# and CS2# for HyperRAM CS#. For information on Multi-Chip Package use of these signals see Multi-Chip Packages.*

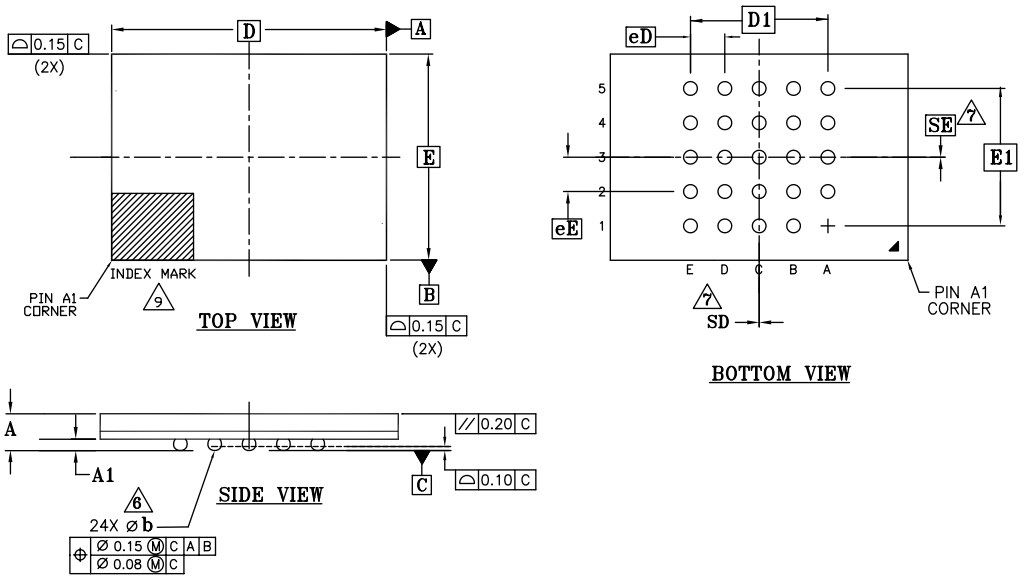
10.2 Multi-Chip Packages

The HyperRAM family S70K members are Multi-Chip Package (MCP) devices that may contain multiple HyperFlash, or multiple HyperRAM devices to increase the total memory density within a single package. MCP devices may also contain a combination of HyperFlash and HyperRAM devices.

CS1# is used for HyperFlash primary die CS#. CS2# is used for HyperRAM CS#. Multi-Chip Packages (MCP) for HyperFlash with HyperRAM therefore have separate CS1# and CS2# for HyperFlash and HyperRAM respectively. Other MCP packages may use CS#2 for a secondary HyperFlash die CS# or may use CS1# for a secondary HyperRAM die CS#.

10.3 Physical Diagrams

10.3.1 24-ball FBGA (8.0 × 6.0 × 1.2 mm) FAB024 Package Outline



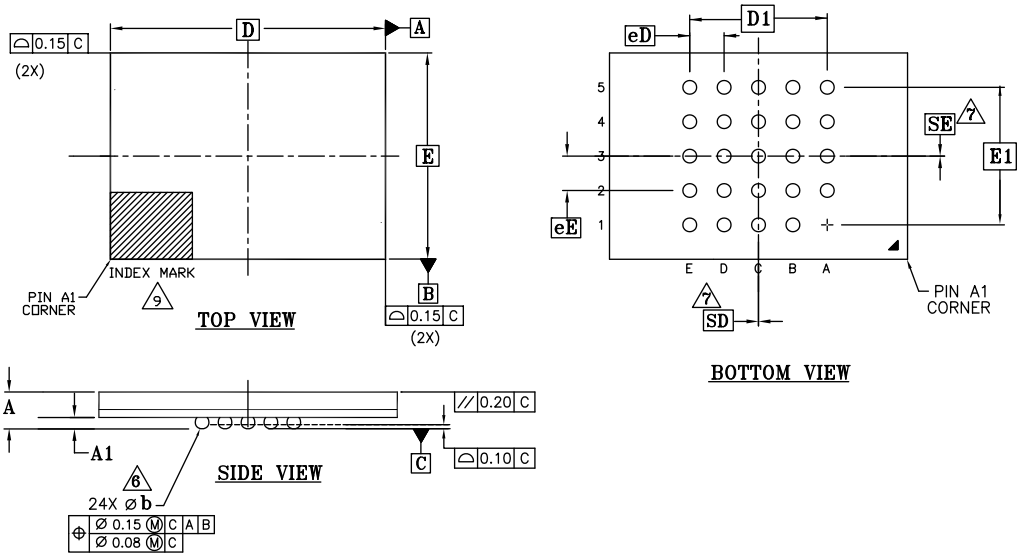
SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	-	-	1.20
A1	0.20	-	-
D	8.00 BSC		
E	6.00 BSC		
D1	4.00 BSC		
E1	4.00 BSC		
MD	5		
ME	5		
N	24		
Ø b	0.35	0.40	0.45
eE	1.00 BSC		
eD	1.00 BSC		
SD	0.00 BSC		
SE	0.00 BSC		

NOTES:

1. DIMENSIONING AND TOLERANCING METHODS PER ASME Y14.5M-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. BALL POSITION DESIGNATION PER JEP95, SECTION 3, SPP-020.
4. e REPRESENTS THE SOLDER BALL GRID PITCH.
5. SYMBOL "MD" IS THE BALL MATRIX SIZE IN THE "D" DIRECTION.
   SYMBOL "ME" IS THE BALL MATRIX SIZE IN THE "E" DIRECTION.
   N IS THE NUMBER OF POPULATED SOLDER BALL POSITIONS FOR MATRIX SIZE MD X ME.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
7. "SD" AND "SE" ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" OR "SE" = 0.
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" = eD/2 AND "SE" = eE/2.
8. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
9. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.

002-15534 **

10.3.2 24-ball BGA (8.0 × 6.0 × 1.0 mm) VAA024 Package Outline



SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	-	-	1.00
A1	0.20	-	-
D	8.00 BSC		
E	6.00 BSC		
D1	4.00 BSC		
E1	4.00 BSC		
MD	5		
ME	5		
N	24		
Ø b	0.35	0.40	0.45
eE	1.00 BSC		
eD	1.00 BSC		
SD	0.00 BSC		
SE	0.00 BSC		

NOTES:

1. DIMENSIONING AND TOLERANCING METHODS PER ASME Y14.5M-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. BALL POSITION DESIGNATION PER JEP95, SECTION 3, SPP-020.
4. "e" REPRESENTS THE SOLDER BALL GRID PITCH.
5. SYMBOL "MD" IS THE BALL MATRIX SIZE IN THE "D" DIRECTION.
   SYMBOL "ME" IS THE BALL MATRIX SIZE IN THE "E" DIRECTION.
   N IS THE NUMBER OF POPULATED SOLDER BALL POSITIONS FOR MATRIX SIZE MD X ME.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
7. "SD" AND "SE" ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW "SD" OR "SE" = 0.
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" = eD/2 AND "SE" = eE/2.
8. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
9. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.
10. JEDEC SPECIFICATION NO. REF: N/A

002-15550 *A

11. Document History Page

Document Title: HyperBus™ Specification Low Signal Count, High Performance DDR Bus Document Number: 001-99253				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	–	MAMC	06/11/2015	Initial release.
*A	–	MAMC	07/09/2015	General Description: Updated Single Quad, Dual-Quad, HyperBus footprint figure. Physical Interface: Replaced with 24-ball diagrams.
*B	4854227	MAMC	07/29/2015	Updated to Cypress template.
*C	5264296	SZZX	05/09/2016	Updated *Timing Specifications*: Updated *AC Characteristics*: Added 200 MHz frequency related information. Updated *Read AC Parameters*: Made 1.8 V and 3.0 V AC specifications to be identical. Updated *Write AC Parameters*: Made 1.8 V and 3.0 V AC specifications to be identical. Updated to new template. Completing Sunset Review.
*D	5348827	TOCU	07/13/2016	Updated *Timing Specifications*: Updated *AC Characteristics*: Updated *Read AC Parameters*: Updated Table 9.4 (Updated entire table).
*E	5441374	SZZX	9/19/2016	Updated *Timing Specifications*: Updated *AC Test Conditions*: Updated Table 9.1 (Updated entire table).
*F	5766300	AESATMP8	06/07/2017	Updated logo and Copyright.
*G	6295505	PRIT	08/30/2018	Obsolete document.
*H	6474592	PRIT	02/06/2019	Reactivated document. Updated *Physical Interface*: Updated *Physical Diagrams*: Updated *24-ball FBGA (8.0 × 6.0 × 1.2 mm) FAB024 Package Outline*: Removed existing spec. Added spec 002-15534 **. Updated *24-ball BGA (8.0 × 6.0 × 1.0 mm) VAA024 Package Outline*: Removed existing spec. Added spec 002-15550 *A. Updated to new template.

Sales, Solutions, and Legal Information

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

© Cypress Semiconductor Corporation, 2015–2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.