

# **16-Megabit Synchronous DRAM**

*Technical  
Reference*

# ***16-Megabit Synchronous DRAM Technical Reference***

## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

**TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.**

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

## Read This First

---

---

### ***About This Manual***

The purpose of this technical reference is to give you detailed operational information about Texas Instruments synchronous DRAMs. It is intended to be used with the specific device data sheets (listed below), which contain the electrical characteristics, operating requirements and conditions, as well as timing and switching information.

### ***Notational Conventions***

This document contains the following conventions:

- The TMS626402 and TMS626802 SDRAM devices are both included by use of the term TMS626x02.
- The TMS626x02 devices are available as TMS626x02-10, TMS626x02-12, and TMS626x02-15. The -xx portion is the device speed indicator. Throughout this book, the device speed indicator is used as an abbreviation of the full device name and applies to both the TMS626402 and the TMS626802 (e.g., the -15 specification allows read latency of one, two, or three cycles to be used).

### ***Related Documentation From Texas Instruments***

To obtain a copy of these TI documents, call the Texas Instruments Literature Response Center at (800) 477–8924. When ordering, please identify the document by its title and literature number.

***TMS626402 Synchronous DRAM Data Sheet*** (literature number SMOS642) describes the 2M-word × 4-bit × 2-bank SDRAM.

**TMS626802 Synchronous DRAM Data Sheet** (literature number SMOS182)  
describes the 1M-word × 8-bit × 2-bank SDRAM.

***Other Related Documentation***

***Configurations for Solid State Memories***, JEDEC Standard No. 21-C,  
Release 4, Electronic Industries Association, November 1993.

# Contents

---

---

---

<b>1</b>	<b>Introduction</b>	<b>1-1</b>
1.1	Synchronous DRAM	1-2
1.2	Functional Comparison: SDRAM Versus DRAM	1-2
<b>2</b>	<b>General Description of the 16M-Bit SDRAM</b>	<b>2-1</b>
2.1	Overview of the SDRAM	2-2
2.2	Burst Sequence	2-3
2.3	Latency	2-4
2.4	Two-Bank Operation	2-5
2.4.1	Two-Bank Row-Access Operation	2-5
2.4.2	Two-Bank Column-Access Operation	2-5
2.5	Bank Deactivation (Precharge)	2-6
2.6	Chip Select	2-6
2.7	Data/Output Mask	2-7
2.8	CLK Suspend/Power-Down Mode	2-7
2.9	Mode-Register Set	2-8
2.10	Refresh	2-9
2.10.1	$\overline{\text{CAS}}$ -Before- $\overline{\text{RAS}}$ (CBR) Refresh	2-9
2.10.2	Self Refresh	2-9
2.11	Interrupted Bursts	2-10
2.12	Design Comparison: JEDEC-Standard Versus TI's SDRAM	2-10
<b>3</b>	<b>Detailed Operations on the 16M-Bit SDRAM</b>	<b>3-1</b>
3.1	Power-Up Sequence	3-2
3.2	Seamless Read With Bank Interleaving	3-4
3.3	Read With Bank Interleaving – Random Row Address, Autodeactivate	3-9
3.4	Write With Bank Interleaving	3-13
3.5	Write With Bank Interleaving – Random Row Address, Autodeactivate	3-17
3.6	Read Burst Interrupted by a READ Command	3-21
3.7	Read Burst Interrupted by a WRT Command	3-23
3.8	Read Burst Interrupted by a STOP Command	3-26
3.9	Read Burst Interrupted by a DEAC/DCAB Command	3-28
3.10	Write Burst Interrupted by a READ Command	3-30
3.11	Write Burst Interrupted by a WRT Command	3-32
3.12	Write Burst Interrupted by a STOP Command	3-34
3.13	Write Burst Interrupted by a DEAC/DCAB Command	3-36

3.14	Clock Suspend .....	3-38
3.15	Power-Down-Mode Entry — Both Banks Active .....	3-42
3.16	Power-Down-Mode Entry — Both Banks Deactivated .....	3-44
3.17	Self-Refresh Operation .....	3-46
3.18	Data-Masking Operation .....	3-48
<b>A</b>	<b>Glossary .....</b>	<b>A-1</b>

# Figures

---

---

2-1	Block Diagram	2-2
2-2	Mode-Register Programming	2-8
3-1	Power-Up Sequence	3-3
3-2	Seamless Read With Bank Interleaving	3-7
3-3	Read With Bank Interleaving – Random Row Address, Autodeactivate	3-11
3-4	Write With Bank Interleaving	3-15
3-5	Write With Bank Interleaving – Random Row Address, Autodeactivate	3-19
3-6	Read Burst Interrupted by a READ Command	3-22
3-7	Read Burst Interrupted by a WRT in the Same Bank	3-24
3-8	Read Burst Interrupted by a WRT in a Different Bank	3-25
3-9	Read Burst Interrupted by a STOP Command	3-27
3-10	Read Burst Interrupted by a DEAC Command	3-29
3-11	Write Burst Interrupted by a READ Command	3-31
3-12	Write Burst Interrupted by a WRT Command	3-33
3-13	Write Burst Interrupted by a STOP Command	3-35
3-14	Write Burst Interrupted by a DEAC Command	3-37
3-15	Clock Suspend During a Write Cycle	3-40
3-16	Clock Suspend During a Read Cycle	3-41
3-17	Power-Down-Entry Operation, Both Banks Active	3-43
3-18	Power-Down-Entry Operation, Both Ranks Deactivated	3-45
3-19	Self Refresh	3-47
3-20	Data-Masking Operation	3-49

# Tables

---

---

1-1	SDRAM Versus DRAM Functional Differences	1-2
1-2	Raw Data Throughput	1-3
2-1	Bit Burst Sequences	2-3
2-2	Read Latencies for Different Operating Frequencies and Clock Speeds	2-4
2-3	Comparison of JEDEC-Standard Features With TI SDRAM	2-10
2-4	Comparison of Prefetch and Pipeline Versus JEDEC	2-11



# Introduction

---

---

This chapter introduces the SDRAM (synchronous dynamic random-access memory) and compares it with the standard DRAM.

Topics covered are:

<b>Topic</b>	<b>Page</b>
1.1 Synchronous DRAM .....	1-2
1.2 Functional Comparison: SDRAM Versus DRAM .....	1-2

## 1.1 Synchronous DRAM

As the speed of processors and buses continues to increase, the speed of the standard DRAM (dynamic random-access memory) becomes increasingly inadequate. In order to improve the overall system performance, the DRAM operations have been synchronized to the system clock, creating a synchronous DRAM (SDRAM). This book, in conjunction with the Texas Instruments TMS626x02<sup>1</sup> 16M-bit SDRAM data sheets, will help you use SDRAMs efficiently.

## 1.2 Functional Comparison: SDRAM Versus DRAM

There are some major differences between the SDRAM and the DRAM that make control of the SDRAM easier than control of DRAMs. Also, there are functional differences, some of which are summarized in Table 1–1.

*Table 1–1. SDRAM Versus DRAM Functional Differences*

<b>DRAM</b>	<b>SDRAM</b>
No system clock	Runs off system clock
Level $\overline{\text{RAS}}$ control	Pulsed $\overline{\text{RAS}}$ control
One-bank operation	Two banks for on-chip interleaving
1 transfer per column address	Burst of 1, 2, 4, or 8 transfers per column address
Read latency is nonprogrammable	Read latency is programmable

A DRAM is an asynchronous device. Systems using DRAMs are required to incorporate wait states that match the performance specifications of the DRAM. Timing of commands is dependent upon the speed of the DRAM and not necessarily upon system speeds. To facilitate better interaction between the SDRAM and the rest of the system, all commands are referenced to the system clock; therefore, the wait-state times needed to match the asynchronous DRAM timing with the system clock are avoided.

To keep a DRAM row active, it is necessary to provide circuitry to hold  $\overline{\text{RAS}}$  low. Precharge of the row is initiated by bringing  $\overline{\text{RAS}}$  high. The SDRAM accepts commands on the rising edge of the system clock. To activate a row,  $\overline{\text{RAS}}$  needs to be held low only for the setup and hold times relative to that clock edge. To deactivate the row, a deactivate command is given on a rising clock edge. The deactivate command initiates the precharge of the row. Having the pulsed  $\overline{\text{RAS}}$  allows the SDRAM to have two independent memory banks on the device with only one  $\overline{\text{RAS}}$  terminal.

<sup>1</sup> TMS626x02 is compatible with the low-voltage TTL (LVTTTL) interface.

The two independent banks allows each SDRAM to have two different rows active at the same time. This allows the reading or writing of data to one bank while the other is being readied. The delay normally associated with precharging and activating a row can be hidden by interleaving the bank accesses.

The SDRAM can achieve a greater data throughput than a DRAM as shown in Table 1–2. One reason for this is the bursting capability of the SDRAM. The SDRAM can burst a series of addresses (1, 2, 4, or 8 addresses) based on a given starting address. The length of the burst can be programmed by the system. Up to eight consecutive addresses can be fetched internally before another column address has to be sent, unlike conventional DRAMs, which require that each address fetch be initiated by a column address sent to the chip. This allows consecutive addresses to be transferred sequentially in a very short period of time after the initial column address has been sent. This means that any time data is being transferred in sequential blocks, the SDRAM performs at a much faster rate than a conventional DRAM. The SDRAM also uses a wider internal data bus to achieve the greater data rate.

The amount of time needed to have valid data output from a DRAM depends on when the column-address command is given. This causes the system control to be more complicated and can cause delay to accommodate the various timings. The SDRAM has the delay between column entry and data valid programmed by the system; this is known as the read latency. Read latency is programmed to be from one to three clock cycles. The read latency used for a given system depends on the frequency of the system clock.

*Table 1–2. Raw Data Throughput*

Bytes Per Transfer	SDRAM Time Delay (ns)	DRAM Time Delays (ns)
1	60	60
2	70	100
4	90	180
8	130	340
16	210	660
32	370	1300
64	690	2580
128	1330	5140

**Note:** Assumes 1 byte transfer per chip per cycle at 100 MHz



# General Description of the 16M-Bit SDRAM

---

---

This chapter contains an overview of the SDRAM and its functions.

Topics covered are:

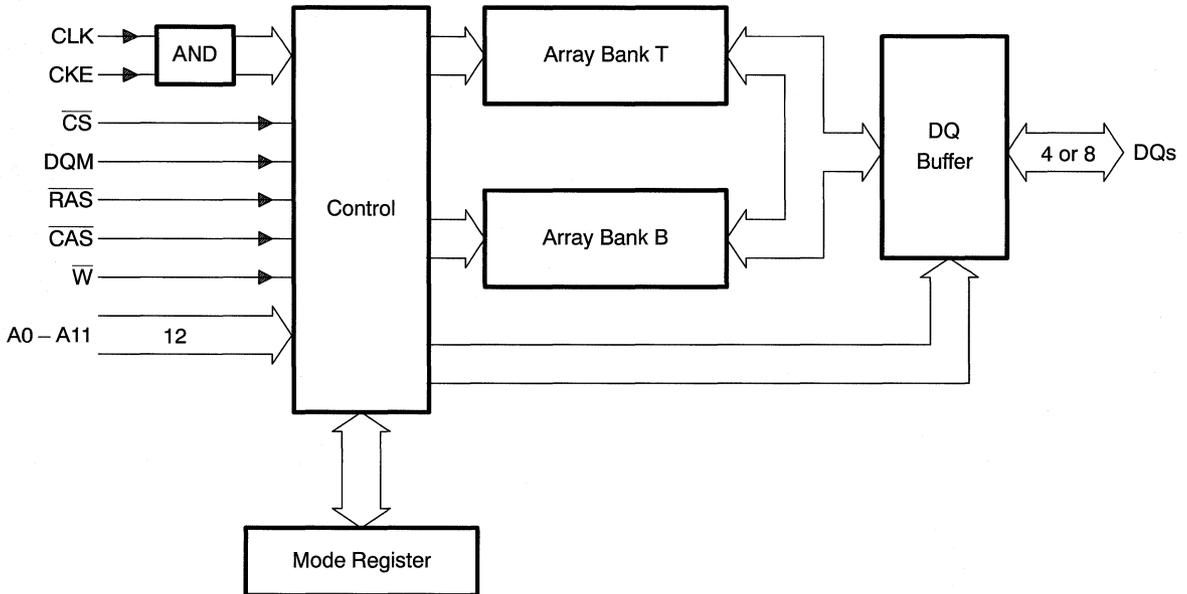
<b>Topic</b>	<b>Page</b>
2.1 Overview of the SDRAM .....	2-2
2.2 Burst Sequence .....	2-3
2.3 Latency .....	2-4
2.4 Two-Bank Operation .....	2-5
2.5 Bank Deactivation (Precharge) .....	2-6
2.6 Chip Select .....	2-6
2.7 Data/Output Mask .....	2-7
2.8 CLK Suspend/Power-Down Mode .....	2-7
2.9 Mode-Register Set .....	2-8
2.10 Refresh .....	2-9
2.11 Interrupted Bursts .....	2-10
2.12 Design Comparison: JEDEC-Standard Versus TI's SDRAM .....	2-10

## 2.1 Overview of the SDRAM

The Texas Instruments SDRAM devices are high-speed 16777216-bit synchronous dynamic random-access memories. All inputs and outputs are synchronized with the system clock input (CLK) to simplify system design with high-speed microprocessors. All inputs of the SDRAM are latched on the rising edge of CLK. The SDRAM outputs are also referenced to the rising edge of CLK.

The SDRAM array is divided into two banks, which are accessed independently. Either bank must be activated before it can be accessed (read from or written to). Refresh cycles refresh both banks alternately. Figure 2-1 is the functional block diagram for the SDRAM.

Figure 2-1. Block Diagram



The following basic commands or functions control most operations of the SDRAM:

- Row-address entry/bank activate
- Column-address entry/write operation
- Column-address entry/read operation
- Bank deactivate
- $\overline{CAS}$ -before- $\overline{RAS}$  refresh
- Self refresh

The operation of the SDRAM can be controlled by three additional methods:

- $\overline{CS}$  to select/deselect the chip
- DQM to enable/mask the DQ signals on a cycle-by-cycle basis
- CKE to suspend (or gate) the CLK input

## 2.2 Burst Sequence

All data for the SDRAM is written or read in burst fashion. Given a single starting address, the SDRAM internally accesses a sequence of locations based on that starting address. Some of the subsequent accesses may be at preceding column addresses and some may be at succeeding column addresses, depending on the starting address entered. The sequence can be programmed to follow either a serial or an interleaved burst sequence. The length of the burst sequence is user-programmable to be any of one, two, four, or eight accesses.

Table 2–1. Bit Burst Sequences

Burst Type	Internal Column Address A1 A0							
	Decimal				Binary			
	Start	2nd	3rd	4th	Start	2nd	3rd	4th
Serial	0	1	2	3	00	01	10	11
	1	2	3	0	01	10	11	00
	2	3	0	1	10	11	00	01
	3	0	1	2	11	00	01	10
Interleaved	0	1	2	3	00	01	10	11
	1	0	3	2	01	00	11	10
	2	3	0	1	10	11	00	01
	3	2	1	0	11	10	01	00

## 2.3 Latency

The delay between the read command (READ) and the first output burst is known as the read latency (also referred to as  $\overline{\text{CAS}}$  latency). The first data output cycle of a read burst can be programmed to occur one, two, or three CLK cycles after the read command. Minimum read latency is based on the particular maximum frequency rating of the SDRAM. This maximum frequency rating is provided in the data sheet as the minimum clock period,  $t_{\text{CK}}$ . Table 2–2 summarizes read-latency requirements for the three speeds of the TI 16M-bit SDRAMs.

Table 2–2. Read Latencies for Different Operating Frequencies and Clock Speeds

Read Latency (Cycles)	Device Speed Indicator					
	-10		-12		-15	
	Speed (ns)	Frequency (MHz)	Speed (ns)	Frequency (MHz)	Speed (ns)	Frequency (MHz)
1	30	≤33	35	≤28.5	40	≤25
2	15	≤66	17.5	≤57	20	≤50
3	10	≤100	12.5	≤80	15	≤66

For example, a -15 device can operate over three ranges of frequencies (≤25 MHz, 25–50 MHz, and 50–66 MHz), but the allowable read latency in each case is different. At frequencies of 25 MHz and below, the -15 specification allows a read latency of one, two, or three cycles to be used. At frequencies between 25 MHz and 50 MHz, the read latency can be either two or three cycles. Finally, at frequencies between 50 MHz and 66 MHz, the read latency must be three cycles. The programmable read-latency feature is provided to allow efficient use of the SDRAM over a wide range of clock frequencies.

There is no latency for data-in cycles (write latency). The first data-in cycle of a write burst is entered at the same rising edge of CLK on which the write command (WRT) is entered. Note that the write latency is fixed and is not determined by the mode register contents.

## 2.4 Two-Bank Operation

The SDRAM contains two independent array banks that can be accessed individually or in an interleaved, or seamless, fashion. Each bank can have one row activated at any given time. This is achieved by executing a bank-activate command (ACTV) with the bank selected by the state of address terminal A11 and the specific row selected by the state of address terminals A0–10. Each bank must be deactivated before it can be activated again with a new row address. A bank can be deactivated automatically using the READ-P or WRT-P commands, or you can use the DEAC command during a READ or WRT operation. Both banks can be deactivated simultaneously by use of the DCAB command.

The availability of two banks allows enhanced performance and a wider variety of possible combinations and methods of data access to choose from, based on the system needs.

### 2.4.1 Two-Bank Row-Access Operation

The two-bank design allows you to access information on random rows at a higher rate of operation than is possible with a standard DRAM. Accomplish this by activating one bank with a row address as described previously, then, while the data stream is being accessed to/from the bank, activate the second bank with another row address. When the data stream to/from the first bank is complete, the data stream to/from the second bank commences without interruption. After the second bank is activated, you can deactivate the first bank to allow the entry of new row address for the next round of accesses. Operation can continue in this interleaved “ping-pong” fashion.

### 2.4.2 Two-Bank Column-Access Operation

The availability of two banks also allows you to access data between banks from random starting columns at a higher rate of operation. After activating each bank with an ACTV command, use A11 to alternate READ or WRT commands between the banks to provide gapless accesses at the CLK frequency, provided all specified timing requirements are met.

## 2.5 Bank Deactivation (Precharge)

Both banks can be simultaneously deactivated, or placed in precharge, by use of the DCAB command. A single bank can be deactivated by use of the DEAC command. The DEAC command and the DCAB command are differentiated from one another by the use of the A10 terminal. When A10 is held high during a deactivate command, the DCAB command is executed and both banks are deactivated. When A10 is held low, the DEAC command is executed and the state of A11 determines which bank is deactivated. Each bank can also be deactivated automatically by using the autodeactivate read (READ-P) and autodeactivate write (WRT-P) operations. The READ-P and WRT-P commands are distinguished from the standard read and write operations by the state of A10. Autodeactivate commands are selected when A10 is held high at the entry of a read or write operation.

## 2.6 Chip Select

The SDRAM features a chip-select input,  $\overline{CS}$ , that can be used to select or deselect the SDRAM for command entry. This provides a means for using the SDRAM in memory systems that require multiple memory device decoding. Hold the  $\overline{CS}$  input high on the rising edge of CLK to deselect the device. This is a DESL command and affects only the  $\overline{RAS}$ ,  $\overline{CAS}$ , and  $\overline{W}$  inputs. The device remains in its present state until  $\overline{CS}$  is brought low and a valid command is input. Use of  $\overline{CS}$  does not affect an access burst that is in progress. The DESL command is equivalent to the NOOP command and the two can be used interchangeably. The device can be selected or deselected on a cycle-by-cycle basis.

## 2.7 Data/Output Mask

Masking of individual data cycles within a burst sequence is accomplished by use of the MASK command. During a write burst, if DQM is held high on the rising edge of CLK, then the incident (referenced to the same rising edge of CLK) input on the DQs is ignored. For a read burst, if DQM is held high on the rising edge of CLK, the output data on the DQs is referenced to the second rising edge of CLK and is placed in the high-impedance state. Therefore, the application of DQM to data-output cycles (read bursts) involves a latency of two CLK cycles, while the application of DQM to data-in cycles (write bursts) has no latency. Also, the MASK command (or its opposite, the ENBL command) is performed on a cycle-by-cycle basis, allowing you to gate any individual data cycle, or multiple cycles, within either a read or a write burst sequence.

## 2.8 CLK Suspend/Power-Down Mode

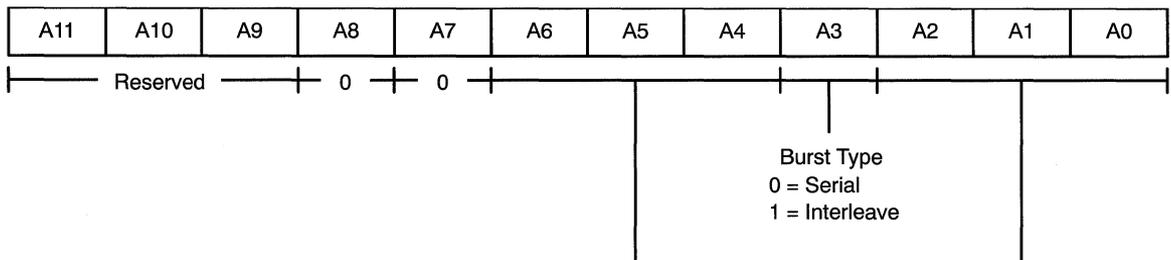
For normal device operation, CKE should be held high to enable CLK. If CKE is brought low during the execution of a read or write operation, the state of the DQ bus occurring at the immediate next rising edge of CLK is frozen and no further inputs are accepted until CKE is returned high. This is known as a CLK-suspend operation and it is executed by a HOLD command. The device resumes operation from the point at which it was placed in suspension, beginning with the second rising edge of CLK after CKE is returned high.

The device enters power-down mode if CKE is brought low when no read or write command is in progress (PDE command). If both banks are deactivated when power-down mode is entered, the power consumption is reduced to the minimum. Power-down mode can be used during row-active periods or  $\overline{\text{CAS}}$ -before- $\overline{\text{RAS}}$  refreshes to reduce input-buffer power. After power-down mode is entered, no further inputs are accepted until CKE is returned high. When power-down mode is exited, new commands can be entered on the first CLK edge after CKE is returned high, provided that the setup time for CKE ( $t_{\text{CESP}}$ ) is satisfied. If  $t_{\text{CESP}} > t_{\text{CK}}$ , then NOOP or DESL commands must be entered until  $t_{\text{CESP}}$  is met. CLK must be active and stable (if CLK was turned off for power-down) before CKE is returned high.

## 2.9 Mode-Register Set

The SDRAM contains a mode register, which you program with the read latency, the burst type, and the burst length. Accomplish this by executing an MRS command. The read latency, burst length, and burst type are encoded on the address lines A0–A8 during execution of the MRS command. A9–A11 are reserved for future use. Logic 0s should always be entered on A7 and A8, but A9–A11 are don't-care entries for the SDRAM. The encoding for all the latency, length, and type combinations supported by the Texas Instruments SDRAM are shown in Figure 2–2.

Figure 2–2. Mode-Register Programming



Register Bits <sup>†</sup>			Read Latency <sup>‡</sup>
A6	A5	A4	
0	0	1	1
0	1	0	2
0	1	1	3

<sup>†</sup> All other combinations are reserved.

<sup>‡</sup> Refer to timing requirements for minimum valid read latencies based on maximum frequency rating.

Register Bits <sup>§</sup>			Burst Length
A2	A1	A0	
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8

<sup>§</sup> All other combinations are reserved.

## 2.10 Refresh

The SDRAM, as the name implies, is still a dynamic RAM and must be refreshed, just as all other dynamic RAMs must be. All rows of the SDRAM must be refreshed at intervals not exceeding the refresh-period specification,  $t_{REF}$ , to assure data retention. The SDRAM offers two commands to accomplish this task: REFR and SLFR. In addition to these commands, refresh can be accomplished manually by performing an ACTV operation in each of the 4096 rows within the refresh period. Because the SDRAM is divided into two independent and equal banks, 2048 of the read or write operations must be performed in each bank. The use of the REFR and SLFR commands is discussed in more detail in following subsections.

### 2.10.1 $\overline{CAS}$ -Before- $\overline{RAS}$ (CBR) Refresh

Before performing a  $\overline{CAS}$ -before- $\overline{RAS}$  refresh, both banks must be deactivated. The refresh address is generated internally such that after 4096 REFR commands, both banks of the SDRAM have been refreshed. The external address and bank-select (A11) inputs are ignored. The execution of a REFR command automatically deactivates both banks upon completion of the internal CBR cycle. This allows consecutive REFR-only commands to be executed without any intervening DEAC commands. The REFR commands do not necessarily have to be consecutive, but 4096 of them must be completed before the refresh interval  $t_{REF}$  expires.

### 2.10.2 Self Refresh

To enter self refresh, you must deactivate both banks of the SDRAM. Following this, execute a SLFR command. For proper entry of the SLFR command, bring CKE low for the same rising edge of CLK that  $\overline{RAS}$  and  $\overline{CAS}$  are brought low and  $\overline{W}$  is brought high. The CKE input must remain low for the device to stay in the self-refresh mode. In the self-refresh mode, all refresh signals are generated internally, and all inputs except CKE are ignored. Power consumption is reduced to a minimum and data is retained by the device for an indefinite period as long as power is maintained. To exit the self-refresh mode, return CKE high. You can issue new commands after waiting for the time interval  $t_{RC}$ . If CLK is made inactive during self refresh, it must be returned to an active and stable condition before CKE is brought high to exit self refresh.

## 2.11 Interrupted Bursts

A read or write operation can be interrupted before the burst sequence has been completed with no adverse performance by entering certain superseding commands, provided that all timing requirements are met. The command interrupting either a read or a write burst must be entered only on an even number of cycles from the initial burst command as determined by the specification nCCD. An additional restriction on burst interruptions is that the interruption of autodeactivate read and write operations is not supported.

## 2.12 Design Comparison: JEDEC-Standard Versus TI's SDRAM

The flexibility of JEDEC Standard No. 21-C for SDRAMs causes some issues to arise for consideration.

The first issue focuses on the fact that the JEDEC standard defines the minimum functionality required, which allows the individual SDRAM vendors to differentiate their products by adding additional functions that can provide a competitive advantage.

The second issue is that there are two ways to implement the SDRAM: one is to use a pipeline approach to access the array, and the other is to use a prefetch approach. In the pipeline implementation, an array access is performed for each bit of data in a read or write burst. In the prefetch implementation, two or more bits of data are retrieved per array access in a read or write burst. Both implementations result in compliance with the JEDEC specification for an SDRAM; however, there are several device specifications that are implementation dependent.

The minimum feature set and the implementation-dependent parameters are summarized in Table 2–3 and Table 2–4.

*Table 2–3. Comparison of JEDEC-Standard Features With TI SDRAM Features*

Features	Texas Instruments	JEDEC
Burst stop (STOP command)	Yes	(optional)
<b>Burst Lengths</b>		
1	Yes	(optional)
2	Yes	(optional)
4	Yes	Yes
8	Yes	Yes
Full page	No	(optional)

Table 2–4. Comparison of Prefetch and Pipeline Versus JEDEC

Feature	Burst Length	Latency	Prefetch (TI SDRAM)	Pipeline <sup>†</sup>	JEDEC
<b>Column-to-Column Address Delay</b>					
nCCD	n/a	n/a	2n	1n	2n
<b>Precharge Timing</b>					
nEP	1	1	1	0	n/a
		2	0	-1	n/a
		3	-1	-1	n/a
	>1	1	0	0	0
		2	-1	-1	-1
		3	-2	-1	-1
t <sub>APR</sub>	1	1	t <sub>RP</sub> + t <sub>CK</sub>	t <sub>RP</sub>	n/a
		2	t <sub>RP</sub>	t <sub>RP</sub> - t <sub>CK</sub>	n/a
		3	t <sub>RP</sub> - t <sub>CK</sub>	t <sub>RP</sub> - t <sub>CK</sub>	n/a
	>1	1	t <sub>RP</sub>	t <sub>RP</sub>	t <sub>RP</sub>
		2	t <sub>RP</sub> - t <sub>CK</sub>	t <sub>RP</sub> - t <sub>CK</sub>	t <sub>RP</sub> - t <sub>CK</sub>
		3	t <sub>RP</sub> - 2t <sub>CK</sub>	t <sub>RP</sub> - t <sub>CK</sub>	t <sub>RP</sub> - t <sub>CK</sub>
t <sub>RWL</sub>	1	n/a	x <sup>‡</sup> + t <sub>CK</sub>	x <sup>‡</sup>	n/a
	>1	n/a	x <sup>‡</sup>	x <sup>‡</sup>	x <sup>‡</sup>
t <sub>APW</sub>	1	n/a	y <sup>‡</sup> + t <sub>CK</sub>	y <sup>‡</sup>	n/a
	>1	n/a	y <sup>‡</sup>	y <sup>‡</sup>	y <sup>‡</sup>

<sup>†</sup> These values are based on a small-sample comparison of competitors' specifications. They are not assured by TI.

<sup>‡</sup> x and y values are dependent on the speed version of product selected. Refer to the data sheets for actual values.



# Detailed Operations on the 16M-Bit SDRAM

This chapter describes specific operations on the 16M-bit SDRAM. Each section discusses a different operation, but all operations use the following assumptions:

- SDRAM device speed indicator = -15 (15-ns CLK cycle time)
- Read latency = 3
- System clock frequency = 66 MHz
- Burst length = 4

Topics covered are:

<b>Topic</b>	<b>Page</b>
3.1 Power-Up Sequence .....	3-2
3.2 Seamless Read With Bank Interleaving .....	3-4
3.3 Read With Bank Interleaving — Random Row Address, Autodeactivate .....	3-9
3.4 Write With Bank Interleaving .....	3-13
3.5 Write With Bank Interleaving — Random Row Address, Autodeactivate .....	3-17
3.6 Read Burst Interrupted by a READ Command .....	3-21
3.7 Read Burst Interrupted by a WRT Command .....	3-23
3.8 Read Burst Interrupted by a STOP Command .....	3-26
3.9 Read Burst Interrupted by a DEAC/DCAB Command .....	3-28
3.10 Write Burst Interrupted by a READ Command .....	3-30
3.11 Write Burst Interrupted by a WRT Command .....	3-32
3.12 Write Burst Interrupted by a STOP Command .....	3-34
3.13 Write Burst Interrupted by a DEAC/DCAB Command .....	3-36
3.14 Clock Suspend .....	3-38
3.15 Power-Down-Mode Entry — Both Banks Active .....	3-42
3.16 Power-Down-Mode Entry — Both Banks Deactivated .....	3-44
3.17 Self-Refresh Operation .....	3-46
3.18 Data-Masking Operation .....	3-48

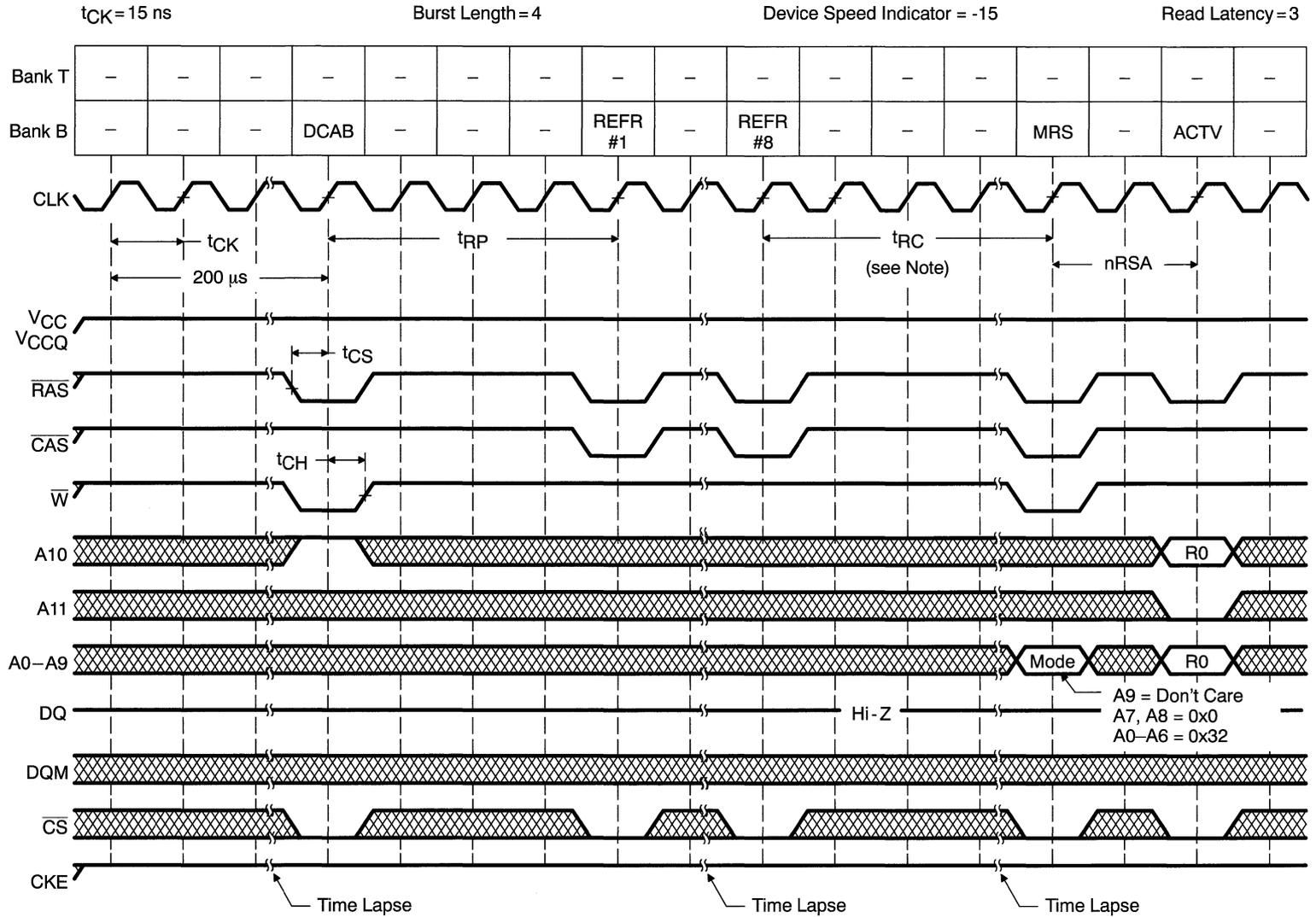
### 3.1 Power-Up Sequence

The power-up sequence ensures that all the internal logic circuits for the SDRAM are initialized to the proper state. After the device is powered up to the full  $V_{CC}$  level, all device inputs must be held stable at a valid logic 0 or logic 1 level for 200  $\mu$ s. The inputs should be set such that NOOP commands are entered during the power-up period. After waiting for 200  $\mu$ s, both banks must be deactivated. In the example shown in Figure 3–1, a DCAB command is used, although two DEAC commands, one for each bank, are also acceptable. Next, eight REFR commands must be performed. The timing specification  $t_{RP}$  must be satisfied before the first REFR command. For the -15 device, the clock-cycle equivalent of  $t_{RP}$  is calculated by:

$$t_{RP} \div t_{CK} = 50 \text{ ns} \div 15 \text{ ns} = 3.3, \text{ which rounds up to 4 cycles}$$

This calculation of 4 cycles agrees with Figure 3–1. The timing specification that determines the number of cycles that must occur between each successive REFR command is proportional to  $t_{RC}$ . From the data sheet,  $t_{RC}$  is 130 ns. A calculation similar to that used for  $t_{RP}$  yields a result of 9 cycles. After the last REFR command and its associated cycle time have elapsed, the SDRAM is ready to be programmed. This is accomplished by setting the mode register for the desired read latency, burst length, and burst sequence. In the example, the read latency is 3, the burst length is 4, and the burst sequence is serial. The correct address to be entered on inputs A0–A7 is 0x32 (hexadecimal). The MRS command with this address programs the SDRAM to operate in the desired mode. The final step in the power-up sequence is to wait for a time given by the specification  $n_{RSA}$ . This specification determines the minimum number of cycles that must elapse between entry of the MRS command and entry of any other valid command. Once this number of cycles has elapsed, the SDRAM is ready for operation.

Figure 3-1. Power-Up Sequence



## 3.2 Seamless Read With Bank Interleaving

Figure 3–2 illustrates a seamless read with two banks. It assumes that both banks are deactivated at the beginning of the time interval shown. The first step is to activate row R0 in bank B. This is accomplished by performing an ACTV command. According to the data sheet, a time interval  $t_{\text{RCD}}$  is required before a READ command can be executed. Given a clock period of 15 ns and  $t_{\text{RCD}} = 40$  ns, it follows that a READ command must wait at least until the third cycle after the ACTV command before execution:

$$\text{number of cycles (in whole numbers)} = t_{\text{RCD}} \div t_{\text{CK}}$$

$$\text{number of cycles} = 40 \div 15 = 2.67$$

$$\text{number of cycles} = 3$$

Data is available on the third cycle after the READ command. Data is valid after a time  $t_{\text{AC}}$  measured from the rising clock edge previous to the data-out cycle. After the read latency, the SDRAM outputs data from a number of locations equal to the burst length. In this example, it is four consecutive locations.

Additional READ commands can be executed before the burst is completed. In order to achieve seamless data in single-bank operations, the next READ command must be executed “burst” number of cycles after the previous READ command cycle. For example, read commands with burst length = 8 should be 8 cycles apart for seamless operation. READ commands can be executed in this manner to achieve seamless data within the limits of  $t_{\text{RAS max}}$ .

In this particular example, the objective is to achieve seamless data using both banks instead of just one. In order for this to occur, you must activate bank T. With the SDRAM, it is possible to have a row in both banks active at the same time. However, data from only one bank can be output at any given time because of the shared data bus. Commands are directed to only one bank at a time using A11. Because you want data to begin on bank T as soon as the data on bank B is finished, a READ command must be issued “burst” cycles (four cycles) after the previous READ command cycle on bank B. In order to issue a READ command on the correct cycle, an ACTV command must be issued at least a time  $t_{\text{RCD}}$  previous to that READ command cycle on bank T. This is possible because no other command is being issued on that cycle. Therefore, seamless data is achieved using two banks.

In this example, bank B is deactivated to demonstrate the issues surrounding the DEAC command, not because it is necessary. Deactivating bank B also illustrates that it is possible to change rows and still maintain seamless data when using both banks. If you wish to get data from a different row in the same

bank, then you must wait a specified number of cycles, which the following calculation shows:

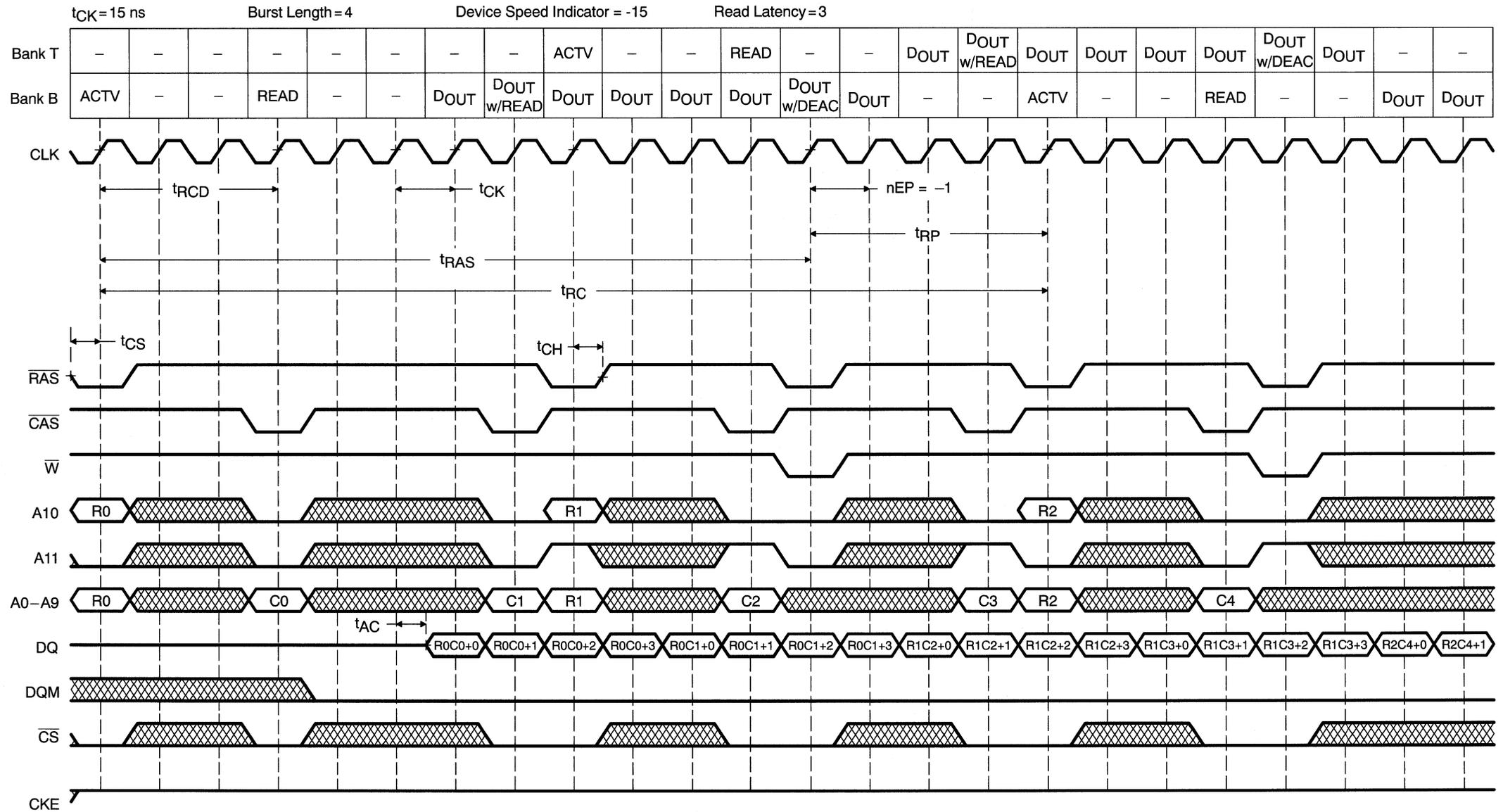
$$\text{number of cycles delayed} = (t_{RP} \div t_{CK}) + (t_{RCD} \div t_{CK}) + \text{read latency}$$

where the quotients of the two divisions are rounded up to the next whole number.

In this particular example,  $t_{RP}$  is equivalent to four cycles,  $t_{RCD}$  is equivalent to three cycles, and the read latency is three cycles. This means the delay would be ten cycles after the DEAC command was issued. For this reason, it is desirable to issue the DEAC command as soon as possible. The DEAC command is restricted by the parameter  $nEP$ . For this example, the maximum value allowed for  $nEP$  is  $-2$  cycles; however, at  $-2$  cycles, a READ command is being issued in bank T. Therefore, the earliest the DEAC command can be issued is one cycle before the last data-out cycle as shown in Figure 3-2. In order to achieve seamless data with the specified conditions, two or more bursts must be completed per row address. This is because of the delay required from the DEAC command to the first data output as discussed previously. If the burst length were set to eight, only one burst would be required per row.



Figure 3–2. Seamless Read With Bank Interleaving



- Notes:**
- 1) Timing relationships shown are for bank B operations.
  - 2) Seamless for two or more bursts per row address



### 3.3 Read With Bank Interleaving – Random Row Address, Autodeactivate

Figure 3–3 illustrates how the autodeactivate feature is used to maximize the data bandwidth of the SDRAM in situations when the row-address characteristics of an application are random. This allows you to automatically deactivate the current row after the data has been output. In this example, the READ-P command is entered for the first access in each row of each bank.

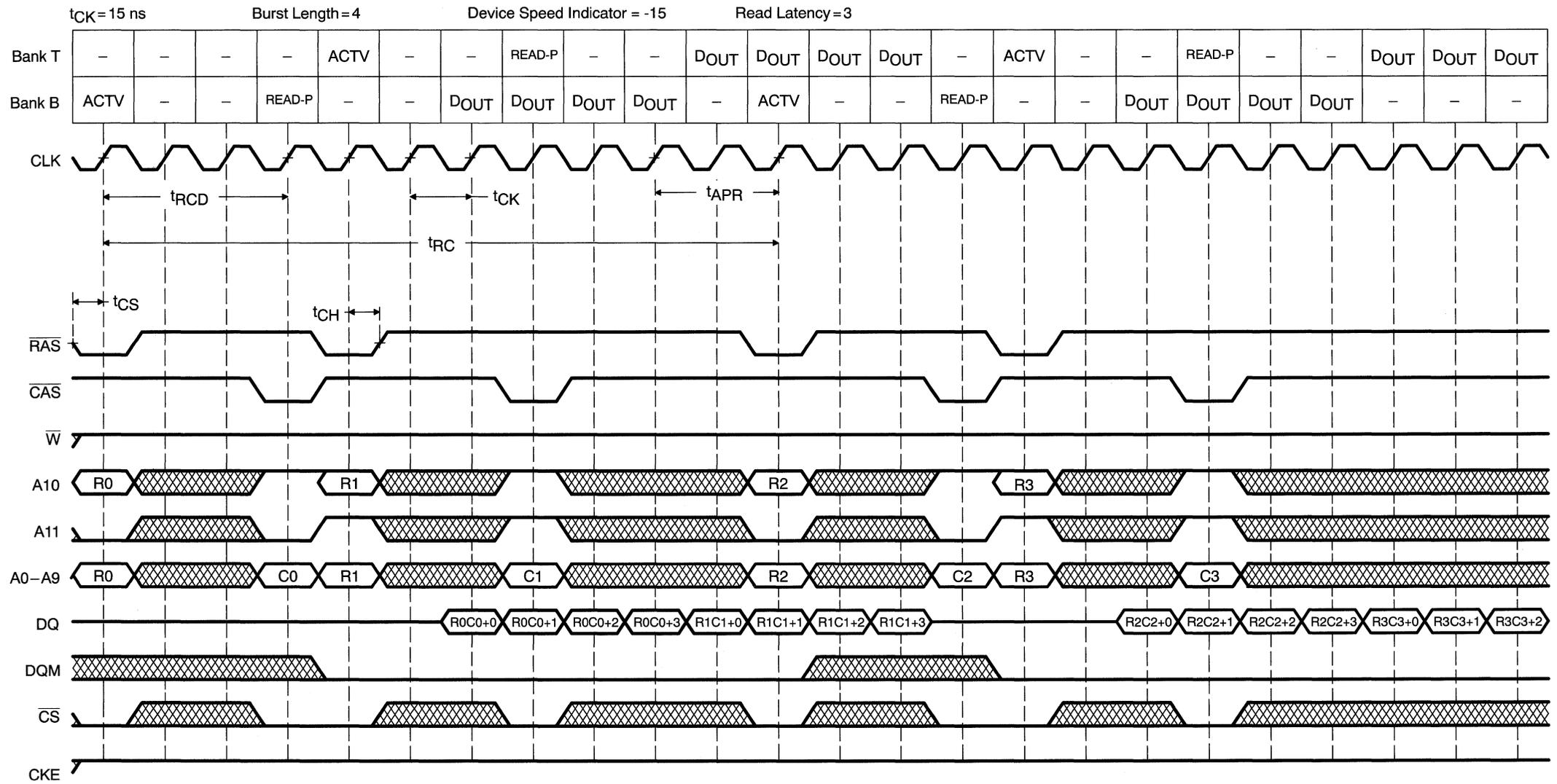
The considerations for the placement of the ACTV and READ-P commands are the same as in Figure 3–2. The minimum cycle time for one bank is determined by the specification  $t_{APR}$ , which is a combination of the nonautodeactivate specifications  $nEP$  and  $t_{RP}$ . The number of cycles for  $t_{APR}$  is calculated as follows:

$$t_{APR} = t_{RP} + (nEP \times t_{CK}) = 50 \text{ ns} + (-2 \times 15 \text{ ns}) = 20 \text{ ns}, \text{ or } 2 \text{ cycles}$$

In the previous case, the minimum time from the last data out to the entry of the DEAC command was limited to  $-1$  cycles because of a conflict with the entry of the READ command in the opposite bank. In this example, the conflict is eliminated with the use of the autodeactivate function, and the cycle time for a read operation is reduced by one clock cycle. The penalty for changing rows after only one access is three null cycles for every eight data access cycles.



Figure 3–3. Read With Bank Interleaving – Random Row Address, Autodeactivate



Note: Timing relationships shown are for bank B operations.



### 3.4 Write With Bank Interleaving

In using the bank interleaving feature with the WRT command, the timing considerations for bank activation are the same as for the read operation (Section 3.2 on page C-4). One difference between the read and write operations is in the latency associated with each operation. Remember that there is a delay from the time the READ command is entered to the time that data is available at the SDRAM output. This is not the case for a write operation. The applicable specification that determines the time from a WRT command to the time when data is first accepted is  $nWCD$ , which is 0. This means that the first bit of data is presented at the SDRAM inputs at the same time that the write command is entered. This is illustrated in Figure 3-4.

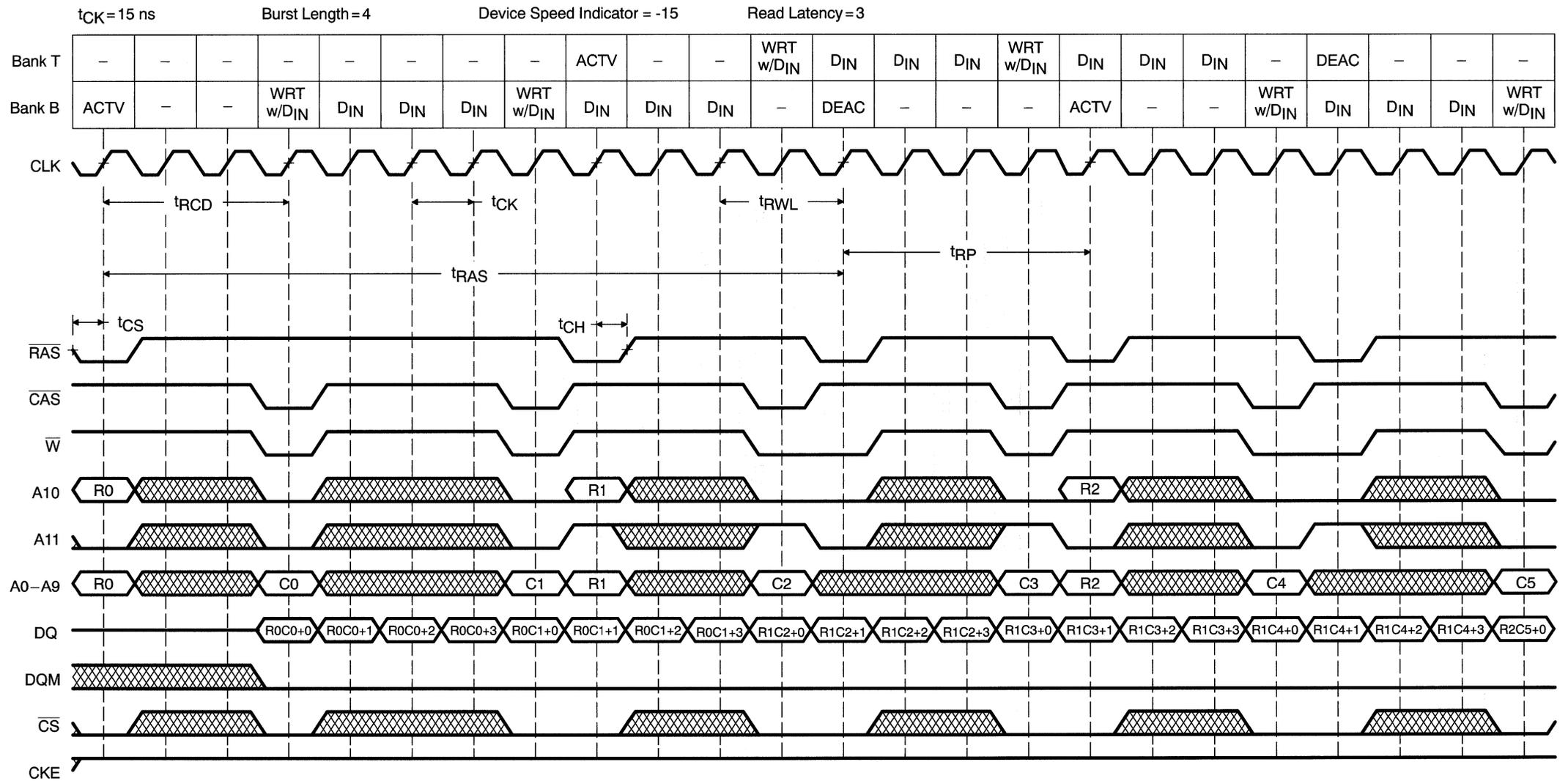
Another timing parameter that is specific to the write operation is  $t_{RWL}$ . This parameter defines the minimum time required between the last data in and entry of a bank-deactivation command. The number of cycles for  $t_{RWL}$  in this example is calculated as follows:

$$\text{number of cycles} = t_{RWL} \div t_{CK} = 30 \text{ ns} \div 15 \text{ ns}, \text{ or } 2 \text{ cycles}$$

Remember from Section 3.2 on page C-4 that a bank can be deactivated before the last data has been output by the SDRAM. This is not the case for write operations. You must wait until the final bit of data has been entered plus two additional cycles before deactivating a bank. Thus, read operations are more effective than write operations at hiding the latency associated with the row-deactivate operation.



Figure 3-4. Write With Bank Interleaving



Note: Timing relationships shown are for bank B operations.



### 3.5 Write With Bank Interleaving – Random Row Address, Autodeactivate

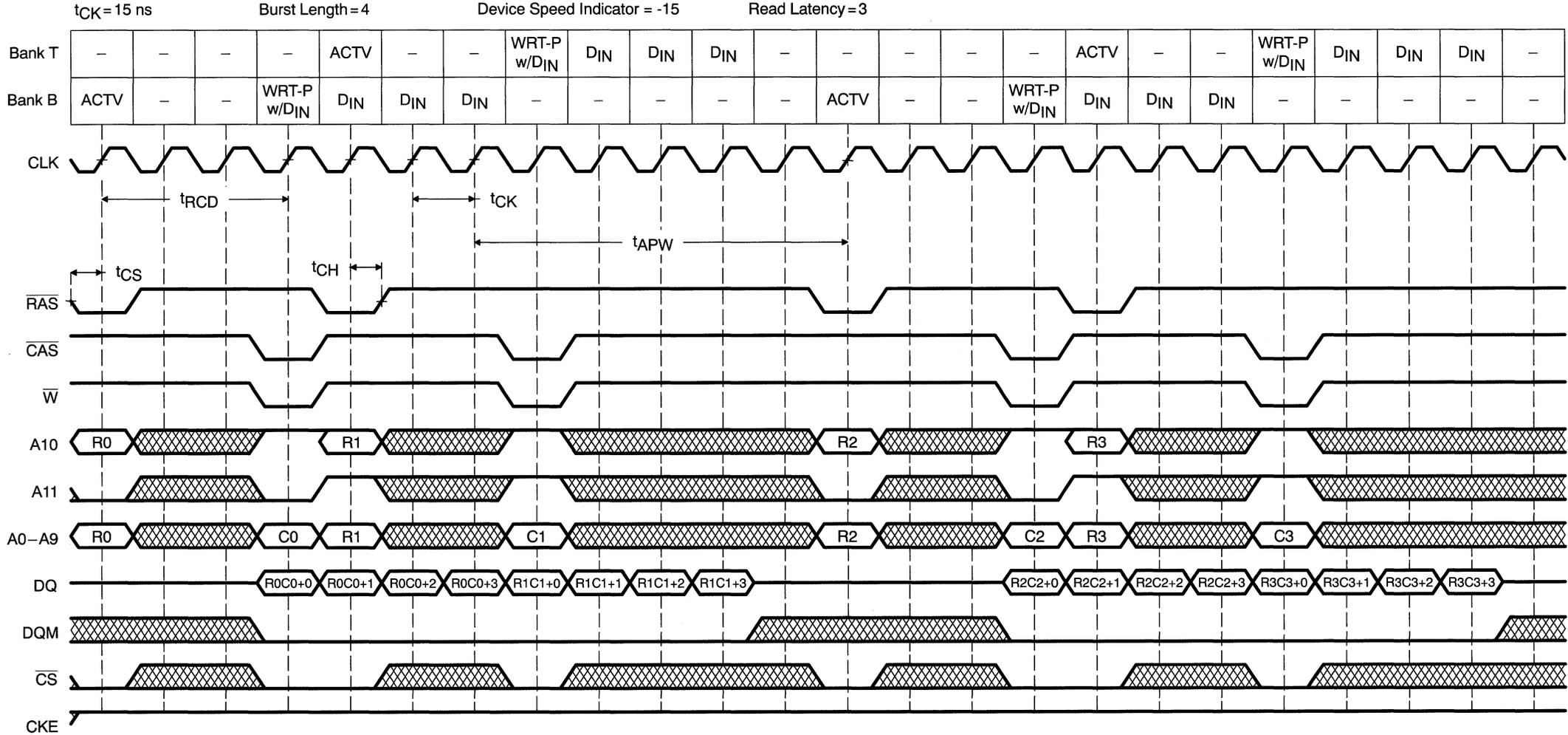
Similar to a read operation with bank interleaving and autodeactivate (Section 3.3 on page C-9), the write-with-autodeactivate feature can eliminate conflict between the DEAC command in one bank and other commands in the opposite bank. As Figure 3–5 shows, the placement of the ACTV and WRT-P commands is the same as in the previous examples. The new specification that must be considered when developing a write cycle that uses the WRT-P command is  $t_{APW}$ , the minimum time from the last data in to the activation of the next row in that bank. This specification is best understood by realizing that it is based on combining the two relevant specifications that were discussed in the previous section,  $t_{RWL}$  and  $t_{RP}$ . This is precisely how the value for  $t_{APW}$  was derived. For our example, the minimum number of cycles for  $t_{APW}$  is calculated as follows:

$$\begin{aligned}
 t_{APW} \text{ in number of cycles} &= (t_{RWL} + t_{RP}) \div t_{CK} \\
 &= (30 \text{ ns} + 50 \text{ ns}) \div 15 \text{ ns} \\
 &= 6 \text{ cycles} \\
 &\quad \text{(rounded up to next whole number )}
 \end{aligned}$$

The benefit of using the autodeactivate command can be seen by observing that the minimum number of cycles for  $t_{RWL}$  and for  $t_{RP}$  may be less than the number of cycles when calculated independently (for some speed configuration); thus, the use of the WRT-P command can save one cycle in the total write cycle time for one bank. Even without the capability to precharge before the completion of a write burst, the data-in throughput is the same as in the read case in Section 3.2 on page C-4. This is because the write operation does not incur any latency from command entry to data-in entry, whereas the read operation does.



Figure 3–5. Write With Bank Interleaving – Random Row Address, Autodeactivate



Note: Timing relationships shown are for bank B operations.



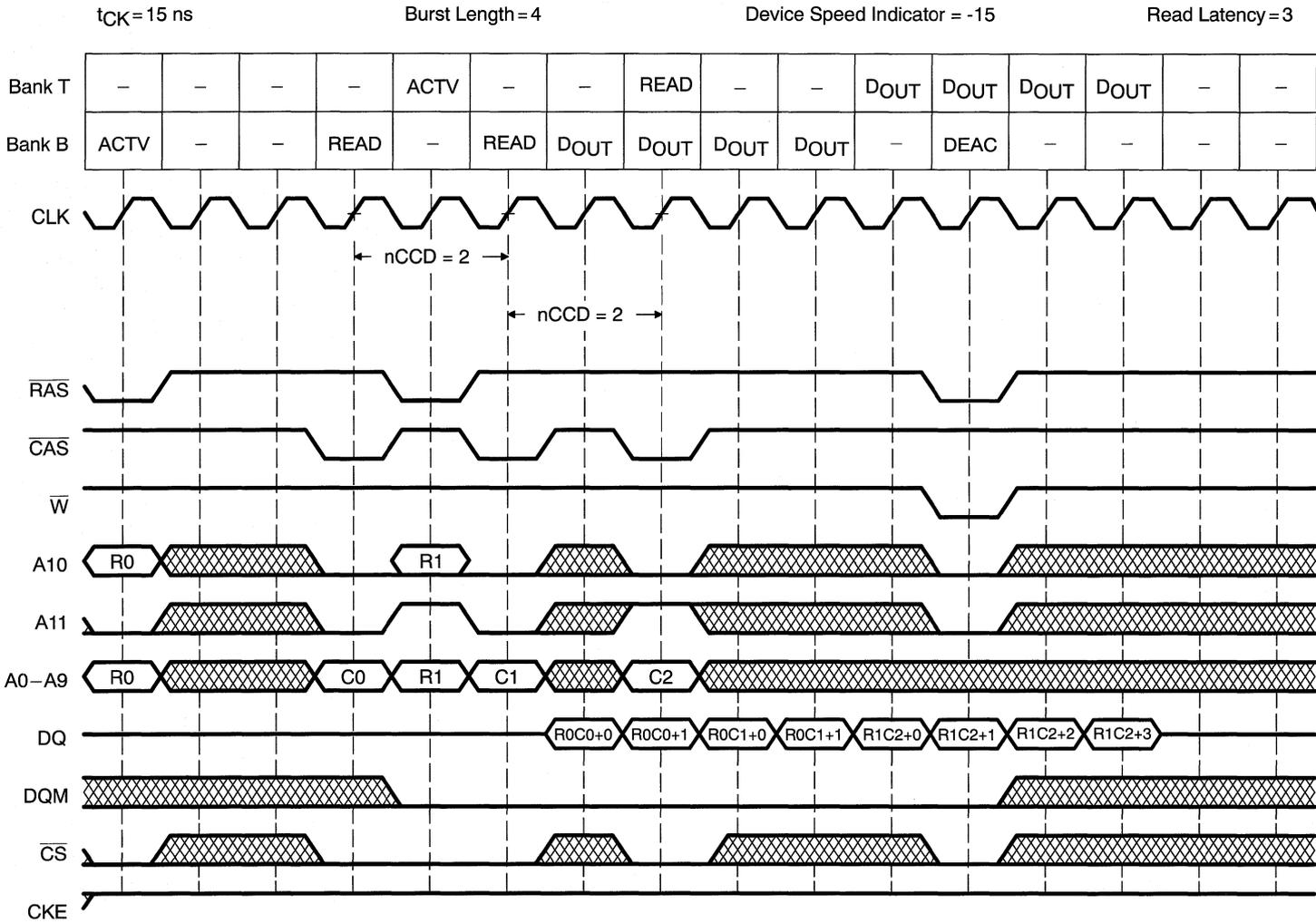
### 3.6 Read Burst Interrupted by a READ Command

The interruption of read bursts is permitted as described in Section 2.11, *Interrupted Bursts*, page 2-10. There are two restrictions on interrupting a read burst. The first is that the interrupting command must be entered an even number of cycles following the entry of the read. The second restriction is that the interruption of a READ-P command is not allowed.

In the interruption of a read operation by another read operation, an interesting issue is the distinction between an interrupting read and a normal read. As discussed in Section 3.2, page C-4, on seamless read operations, a new READ command begins a normal read operation if it is entered “burst” cycles or more after the previous READ command. If, however, the new READ command is entered before “burst” cycles are complete, it interrupts the read operation in progress. In our example shown in Figure 3–6, the only valid cycle in which an interrupting read command can be entered is for  $nCCD = 2$  cycles.

The first READ command is interrupted by another READ command in the same bank. The interrupted READ command continues normal execution until the read latency of the interrupting READ has been satisfied. With a read latency of three cycles, the first two portions of data are output by the first read. At this time, the interrupting read takes effect, but it is also interrupted by a READ command in the opposite bank. As in the previous case, only the first two portions of the second read operation are output. The third read operation interrupts the last two portions of the second read and outputs the full four portions without further interruption. As demonstrated by the second and third READ commands, the operation of the SDRAM during interrupted bursts is not dependent on the bank in which the interrupting command is entered.

Figure 3-6. Read Burst Interrupted by a READ Command



### 3.7 Read Burst Interrupted by a WRT Command

The READ command can also be interrupted by a write operation with two additional considerations. The first is that there is no latency associated with a write operation, so the read operation is immediately interrupted by the entry of a WRT or WRT-P command. The second consideration is the possibility of data contention. Because of the immediacy of the write operation, it is possible for the SDRAM to output data on one rising edge of the clock and, on the next rising edge of the clock, for the system to send a write operation. The resulting data contention causes erroneous data to be written to the SDRAM. This problem can be avoided by the use of the DQM input.

In Figure 3–7, the first read in bank B is interrupted after  $nCCD = 4$  cycles. The DQM input must be brought high  $nDOD + 1$  cycles prior to the entry of the interrupting write operation, as specified in Table 7 of the data sheet. With the DQM input at a high logic level, the outputs of the SDRAM are turned off. The first READ causes data to be available after three cycles, but because of the high state of the DQM two cycles prior to the  $D_{OUT}$ , no data is output. In Figure 3–8, a read operation is executed in bank T and is interrupted by a write in bank B after  $nCCD = 6$  cycles. This case more clearly demonstrates the need to avoid data contention. The first three bits of data are due to be clocked out on the three clock cycles before the interrupting write command is entered. The first two bits will not cause contention, but the third one can. To mask this data from being output, the DQM input must be high  $nDOD + 1$  cycles prior to the write operation. To mask the third and fourth data bits of the read burst, the DQM must be high two cycles prior to the third data bit output and held high for the next cycle to mask the fourth bit. This provides one cycle to clear the DQ bus and avoid any data contention.

Figure 3–7. Read Burst Interrupted by a WRT in the Same Bank

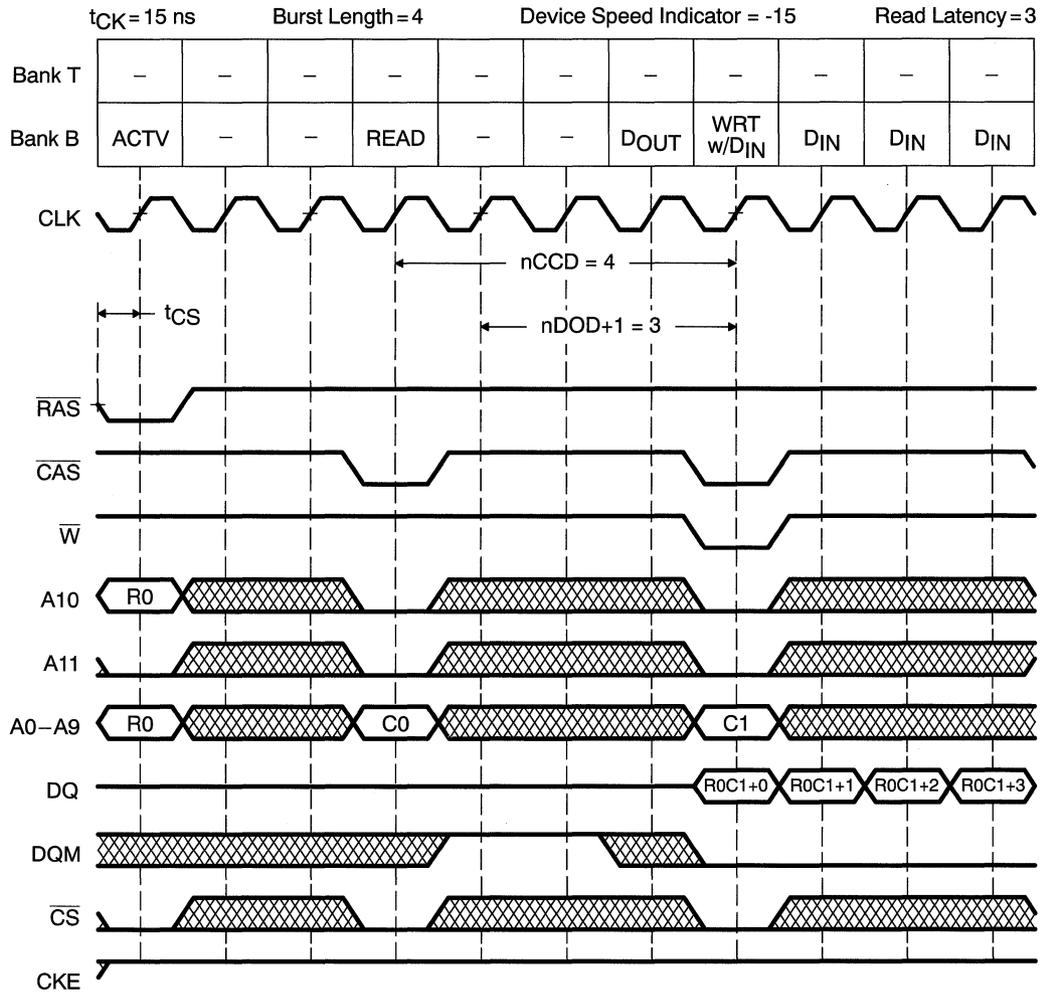
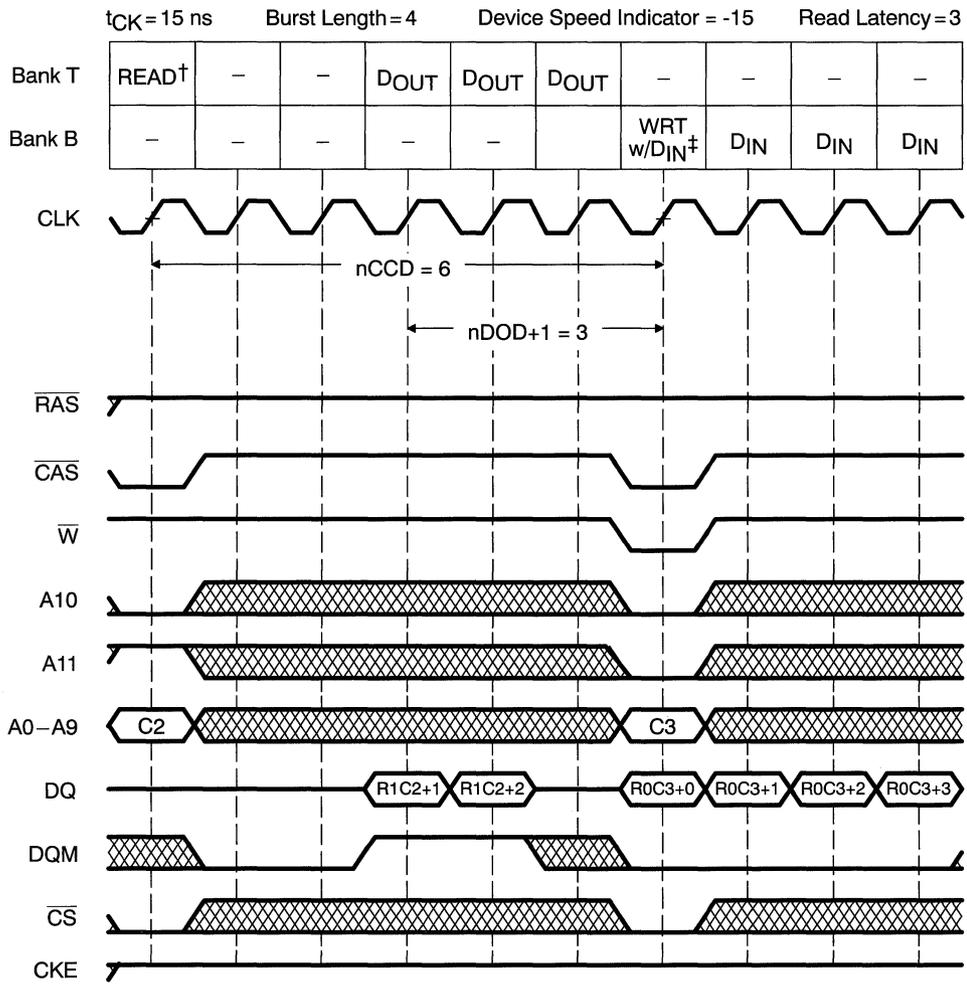


Figure 3–8. Read Burst Interrupted by a WRT in a Different Bank



### 3.8 Read Burst Interrupted by a STOP Command

To understand the effects of the interrupting STOP, it is helpful to refer to Table 7 in the data sheet.

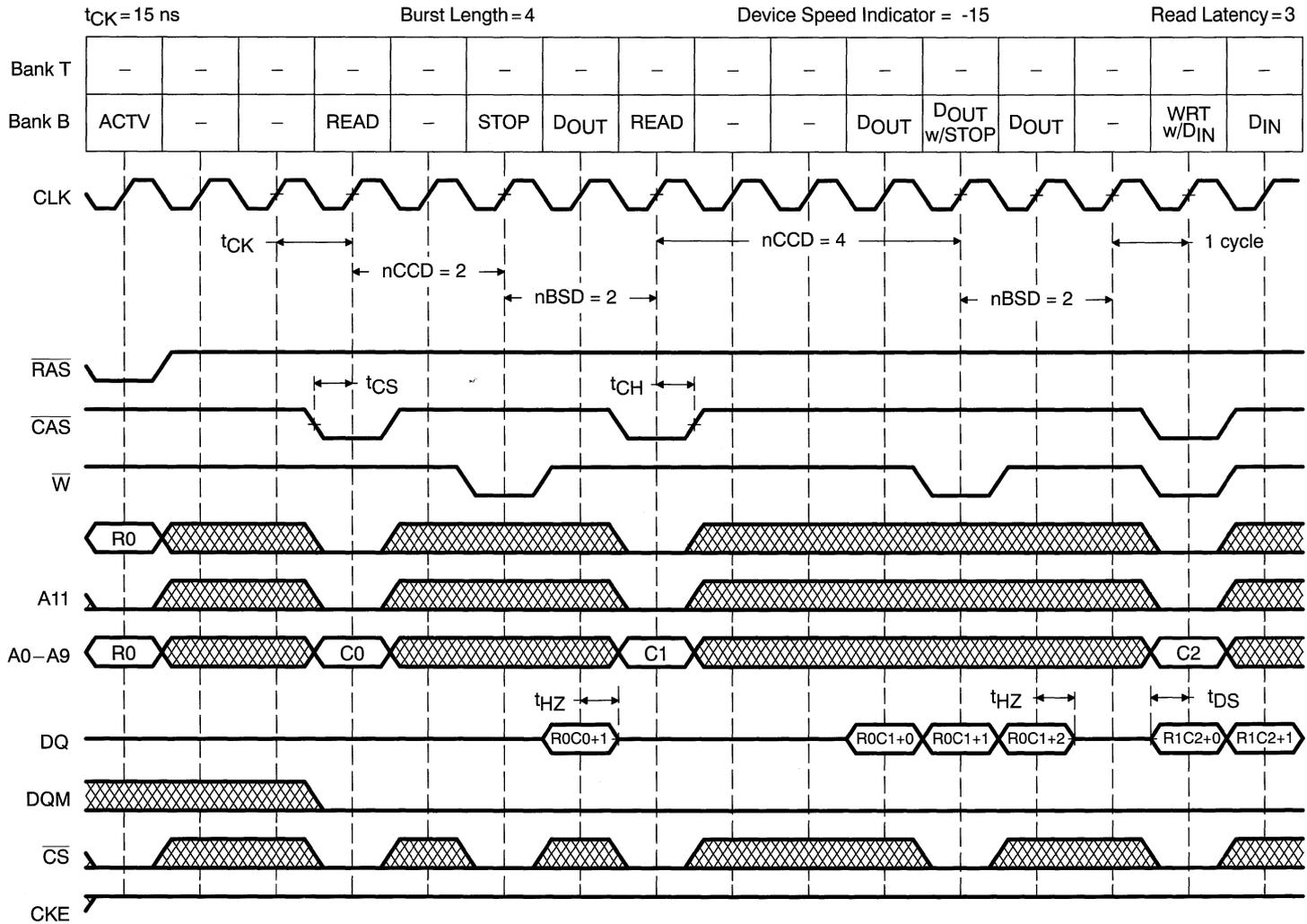
There are three points that determine the effect of the STOP command. The first is that there is a 2-cycle latency from the time the STOP command is entered to the time the SDRAM outputs are placed in the high-impedance state. The data-sheet specification that determines the time required to return the outputs to high impedance is  $t_{HZ}$ , which in this case is 11 ns. It is possible, then, that in certain situations the STOP command has no effect on the data being output by the current read operation. Therefore, the effect of the interrupting STOP command depends on where the command is entered. In Figure 3–9, the first STOP command is input on cycle  $nCCD = 2$  after the READ command. Thus, the first bit of data is output as normal, and the STOP command causes the SDRAM output to turn off on the next cycle. Only one bit of the 4-bit burst is output. The second STOP command is input on cycle  $nCCD = 4$ . The first three bits of data are output as normal and the fourth and final bit is stopped.

The second point is that the STOP command affects only the output of data. The row in the bank in which the STOP was executed remains active.

The last point involves when the next new command in that bank can be executed. The next read or write operation must wait for a minimum of two clock cycles after the entry of the STOP command. The new READ command following the first STOP command can be input after the minimum number of cycles because there is no data contention on the bus. This is not necessarily true for the new WRT command following the second STOP command. According to the specification, the output returns to the high-impedance state a maximum of 11 ns after the last rising clock edge, and the data to be written must be valid at least 2 ns before the next rising clock edge. For the example, this leaves 2 ns of margin to avoid data contention. Because of this, one additional delay cycle is shown to ensure that there is no bus contention due to the switching from a read to write cycle.

The 2-cycle delay between entry of the STOP command and the next READ or WRT command means that there is a penalty associated with using the STOP command. This is an important point to keep in mind when determining the best way to prevent the SDRAM from sending or receiving data in increments that are less than the programmed burst length. A MASK command may be more efficient in some cases. Please refer to Section 3.18 on page C-48 for a full description of data-masking features of the SDRAM.

Figure 3–9. Read Burst Interrupted by a STOP Command

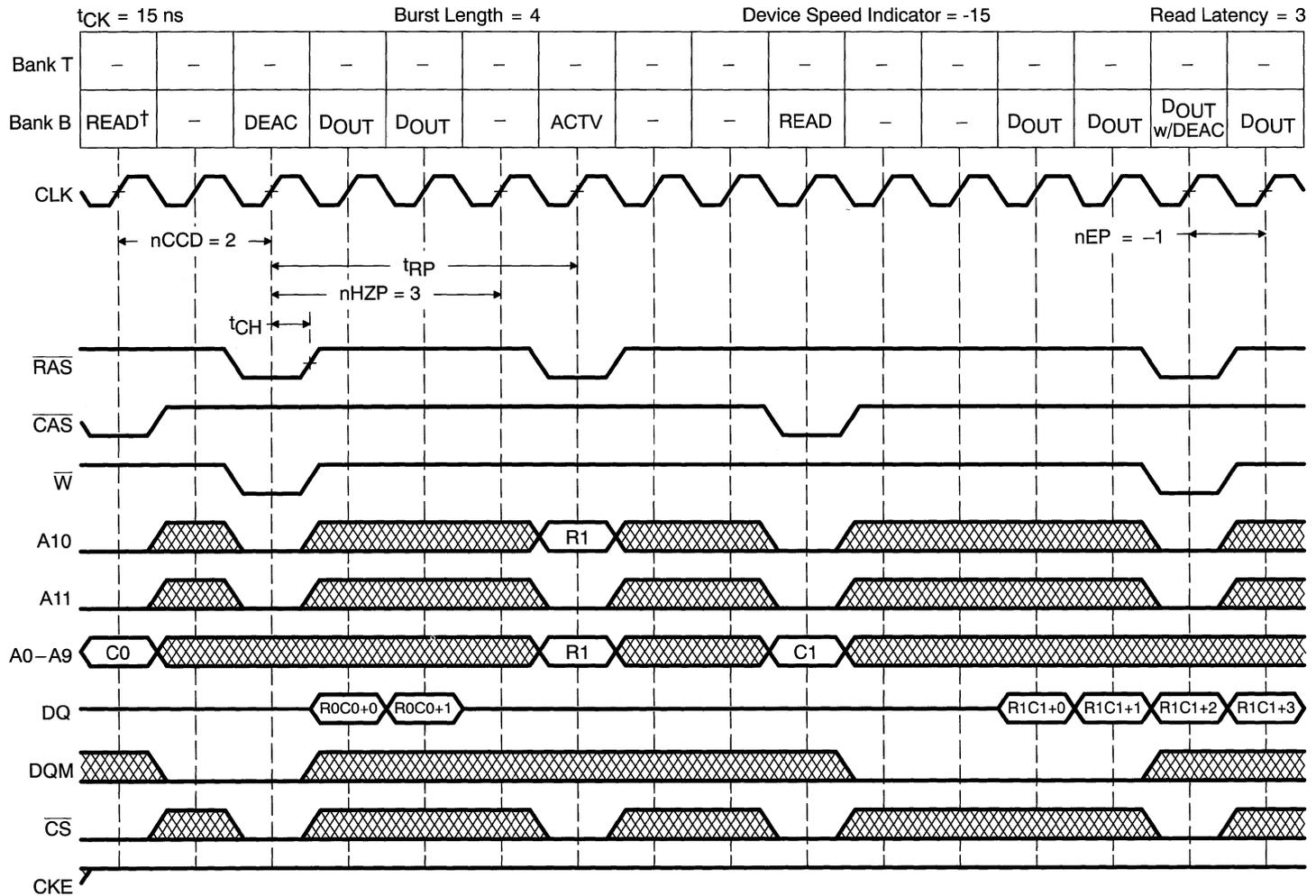


### 3.9 Read Burst Interrupted by a DEAC/DCAB Command

A read burst can be interrupted by a deactivate operation: either the DCAB command or the DEAC command can be used. A cycle-by-cycle analysis of the timing diagram in Figure 3–10 is used to study the relevant specifications. As with all interrupting commands, the  $2n$  rule specified by  $nCCD$  must be followed. In the first case, the interrupting command is DEAC and is entered on the second cycle following the READ command. Because  $nCCD$  is satisfied, the interrupt is a valid operation. The interrupting DEAC command takes effect after  $nHZP$  cycles have elapsed. The value of  $nHZP$  is always equal to the read latency; in this example, three cycles. The SDRAM outputs two bits of the 4-bit burst before the interrupting DEAC command takes effect. It is also possible for a deactivate operation to have no effect on the interrupted read operation. This would have been the case in the example if the DEAC command had been entered after  $nCCD = 4$  cycles. In either case, the selected bank is deactivated.

The second instance of a DEAC command is included to illustrate the relationship between  $nCCD$  and  $nEP$ . In this case, the DEAC command is entered  $nCCD = 5$  cycles after the READ command. While this does violate the specification for  $nCCD$ , we note that the DEAC also occurs  $nEP = -1$  cycles before the end of the read burst. The specification for  $nEP$  takes precedence over the specification for  $nCCD$ . Therefore, the DEAC command is not an interrupt, the read operation completes normally, and the selected bank is placed in the precharge state. The precharge time,  $t_{RP}$ , must be satisfied before the next command can be input. The effect of the interrupting DEAC (or DCAB) is specified in Table 7 of the data sheet.

Figure 3–10. Read Burst Interrupted by a DEAC Command



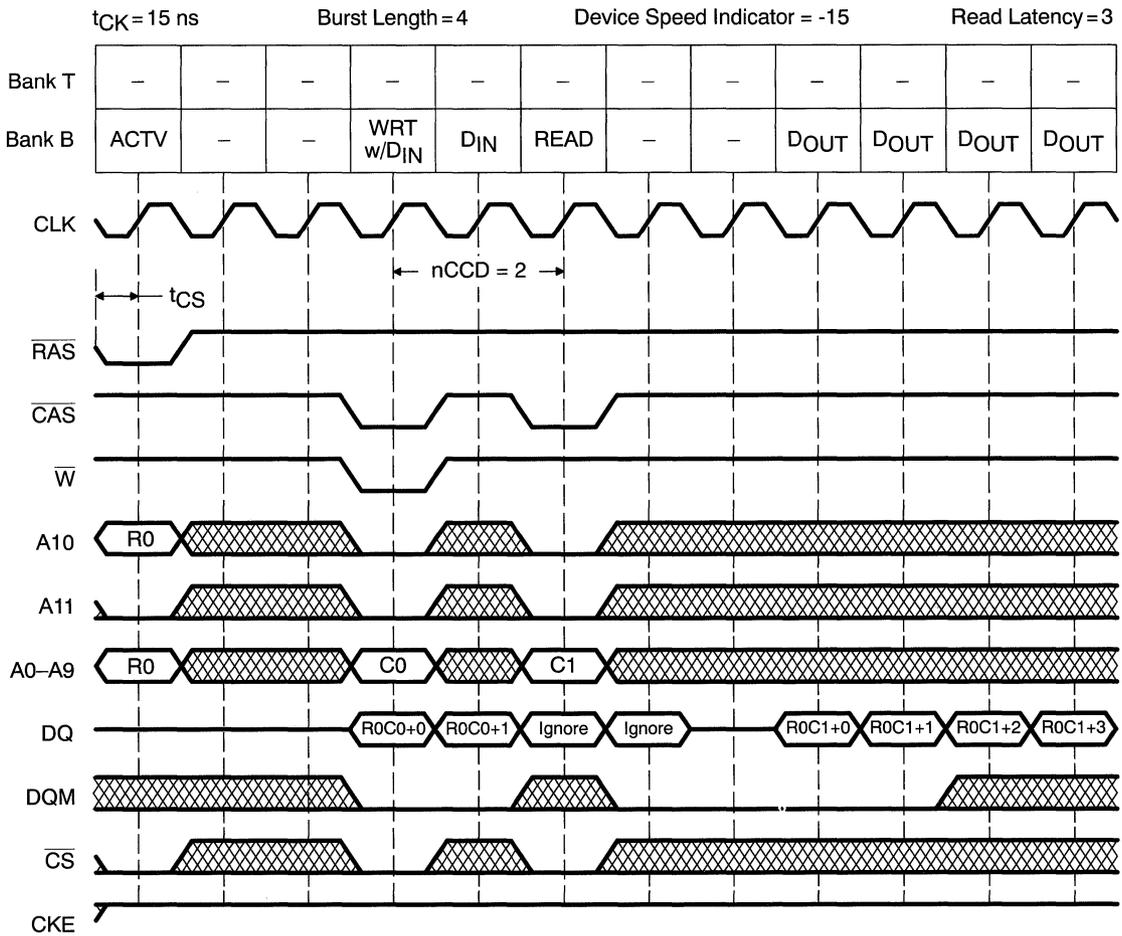
† A previous ACTV instruction activated row R0 in bank B.

### **3.10 Write Burst Interrupted by a READ Command**

The write operation can be interrupted by the same commands that are allowed to interrupt read operations. The same restrictions that apply to read interrupts also apply for write interrupts. These are that the interrupting command must be entered on an even number of clock cycles after the WRT command is entered, and that the WRT-P command cannot be interrupted.

The write operation can be interrupted by a read operation. This case is relatively straightforward to implement as shown in Figure 3–11. The write operation is interrupted after  $n_{CCD} = 2$  cycles by a READ command in bank B. As described in Table 8 of the data sheet, the data input on the previous cycle is written by the SDRAM. The READ command immediately supersedes the data input on the next cycle and no further data is accepted by the SDRAM. The read operation executes as described in Section 3.2 on page C-4.

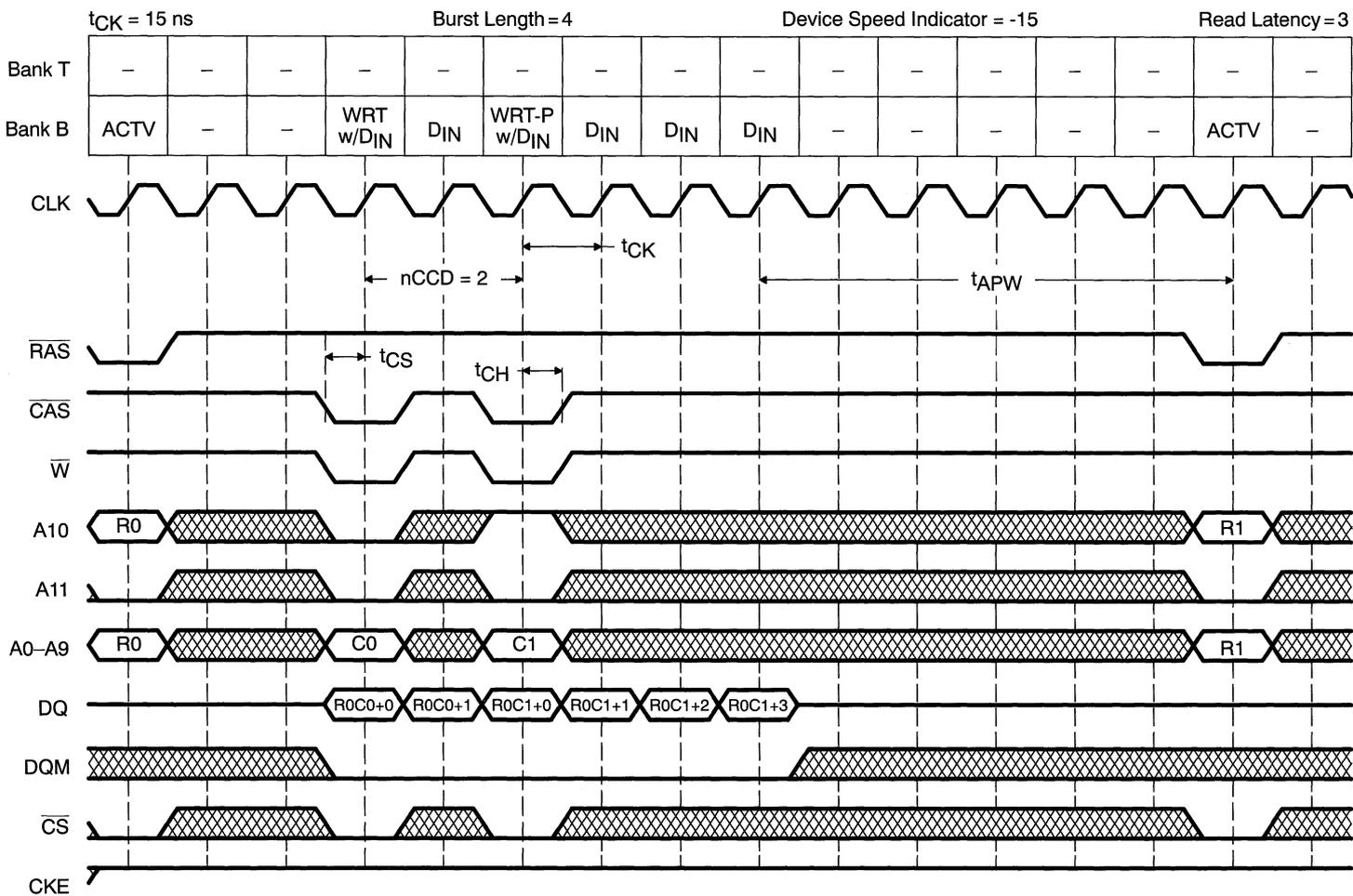
Figure 3–11. Write Burst Interrupted by a READ Command



### **3.11 Write Burst Interrupted by a WRT Command**

The write operation can also be interrupted by another write operation as shown in Figure 3–12. The WRT command is interrupted after  $nCCD = 2$  cycles by a WRT-P command. The interrupting write operation immediately supersedes the write that is in process. The result is that only the first two bits of the WRT command are written to the SDRAM. The next four data bits are written into the SDRAM at the column address specified at the time the interrupting WRT-P command was entered.

Figure 3–12. Write Burst Interrupted by a WRT Command

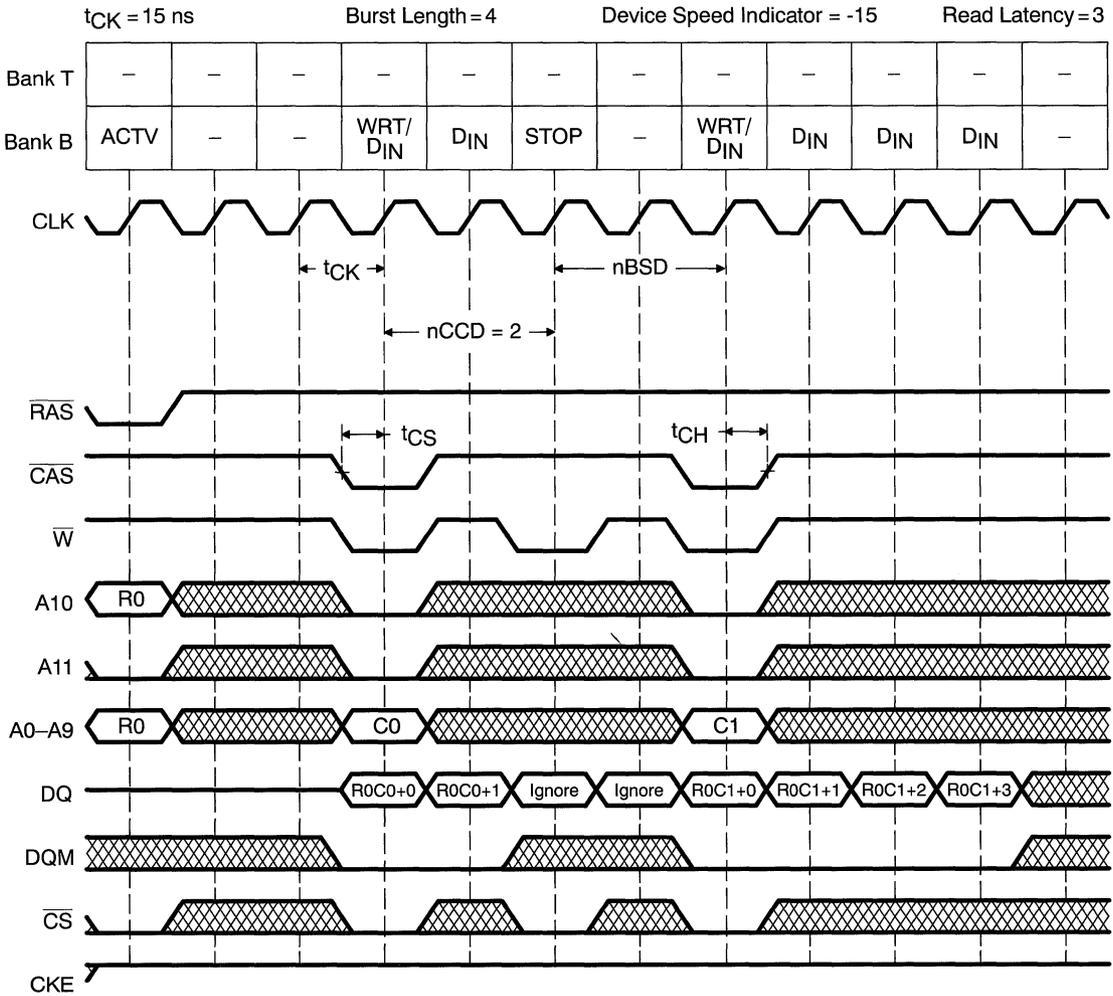


### **3.12 Write Burst Interrupted by a STOP Command**

The example in Figure 3–13 shows the result of a write operation that is interrupted by a STOP command. Unlike the case for a read operation interrupted by a STOP, the effect of the interrupting STOP command is immediate. The data that was to be written on the cycle interrupted by the STOP command is ignored. The SDRAM data inputs ignore all remaining data-in cycles of the write burst. In the example, the STOP command is entered  $nCCD = 2$  cycles after the WRT command. The first two data bits of the burst are written by the SDRAM normally. The last two data bits of the burst are ignored. Because the data is ignored as soon as the STOP command is recognized, it is possible for the system memory controller to withhold the last two data bits without jeopardizing the data integrity of the SDRAM.

The final consideration for the write interrupted by a STOP command is the same as the read case. The next command must not be entered until two clock cycles after the STOP command.

Figure 3–13. Write Burst Interrupted by a STOP Command



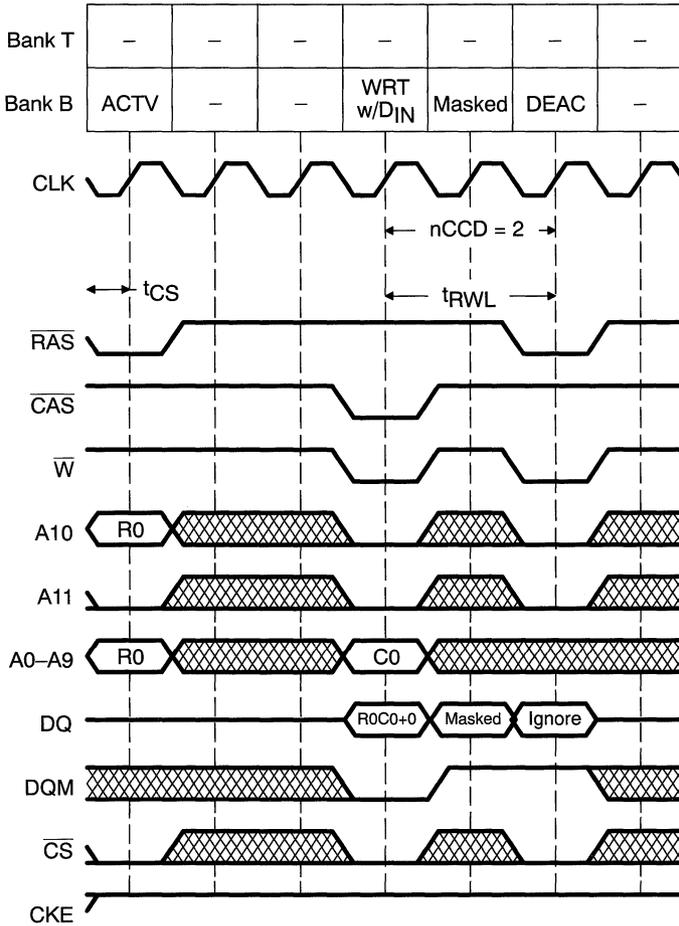
### 3.13 Write Burst Interrupted by a DEAC/DCAB Command

The final operation that can be used to interrupt a WRT command is a bank deactivate. Either of the two bank deactivate commands, DEAC or DCAB, can be used in this case. A DEAC command entered in bank T does not interrupt an operation in bank B. The DCAB command interrupts an operation ongoing in either bank.

The implementation of a write operation interrupted by a DEAC is complicated by the write-recovery specification,  $t_{RWL}$ . This is the specification that determines the time that must elapse from the last data-in cycle to the entry of a deactivate operation. As indicated in Table 7 of the data sheet, the DQM input must be used to mask the data inputs so this specification is not violated. The specific details are shown in Figure 3–14. The WRT command is interrupted after  $n_{CCD} = 2$  cycles by the DEAC command. The minimum number of cycles required to meet  $t_{RWL}$  in this case is two cycles. The DQM input must be brought high one cycle before the DEAC command is entered in order to satisfy the conditions for interrupting a write operation. The result is only one cycle of data is written to the SDRAM.

Figure 3–14. Write Burst Interrupted by a DEAC Command

$t_{CK} = 15 \text{ ns}$     Burst Length = 4    Device Speed Indicator = -15    Read Latency = 3



### 3.14 Clock Suspend

This section and the next discuss the clock gating features of the clock enable terminal, CKE. The two uses of CKE are clock suspension and reduced power consumption. In both cases, while CKE is held low, the SDRAM does not recognize any activity on CLK. The effect of bringing CKE low is dependent on the state of the SDRAM. The SDRAM is placed in clock-suspend mode if CKE is brought low while a read or write operation is in progress. If no read or write operation is in progress at the time CKE is brought low, the SDRAM enters the power-down mode.

There are two keys to understanding the effect of a clock-suspend operation that is accomplished by the entry of the HOLD command. The first is that there is a latency associated with the execution of a HOLD command. This latency is determined by the timing specification nCLE. This parameter defines the number of clock cycles that must elapse before the clock-suspend operation takes effect, and also the number of clock cycles that must elapse before a new command can be entered after the clock-suspend operation is exited. The value of nCLE is always one clock cycle. The second point is that the clock input is not recognized by the SDRAM while clock suspend is in effect. The use of the HOLD command is shown in Figure 3–15 for a write and in Figure 3–16 for a read.

The clock is suspended twice during the write cycle of Figure 3–15. This is done to illustrate the effect of varying the length of a clock-suspend operation. CKE is first brought low before the rising clock edge of the second data bit of the 4-bit burst. The clock-suspend operation is valid after nCLE cycles, provided that the specification for CKE setup time,  $t_{CES} = 2 \text{ ns}$ , is satisfied. Because there is a one-cycle latency from the time CKE is brought low to the time the SDRAM responds to the clock-suspend request, the second data bit is written to the SDRAM. The clock input is inhibited for as many cycles as CKE is held low. In this case, CKE is held low for two cycles, so the data presented to the SDRAM inputs on the two cycles after nCLE is satisfied is ignored. Because of the entry and exit latency associated with the HOLD command, the clock edges, and hence the data inputs, that are ignored do not correspond to the time CKE is held low. The offset is equal to nCLE. Therefore, the second clock suspend during the write operation is entered during the same cycle that the third data bit is written, and is effective during the next cycle. The fourth and final data bit is written on the following cycle. With a total of three clock-suspend cycles, the write operation for a burst length of 4 takes seven cycles to complete.

Figure 3–16 shows clock suspend having a similar effect on the read operation. The first clock suspend in the read cycle occurs during the same cycle that

the first cycle in the read latency is counted. The second and third cycles of the read latency are delayed for two cycles because CKE is held low. The effective access time from the start of the read operation is extended to five cycles. The first two bits of the read operation are output normally. However, CKE is held low for another two cycles, causing the third data bit to remain on the SDRAM outputs for as many cycles. The final data bit is output on the cycle after CKE is brought high.

This example also points out the different effect that CKE has on the read and write operations. In the write operation, the data that is to be input to the SDRAM on the first cycle after nCLE is satisfied is not written. However, the data that is output by the SDRAM on the first cycle after nCLE is satisfied is output and held. This is because of the difference in how the SDRAM handles the read and write operation. Recall the discussion of the read latency in Section 3.2 on page C-4: the data is actually output by the SDRAM one clock cycle before it is valid. In the example, the clock-suspend command does not take effect until after the SDRAM has already sent the third data bit out. The next two clock cycles are ignored so that the SDRAM does not advance the next data bit until nCLE cycles after CKE is brought high. Data is written to the SDRAM in the same clock cycle it is presented. Therefore, the clock suspend affects the data input immediately.

Figure 3–15. Clock Suspend During a Write Cycle

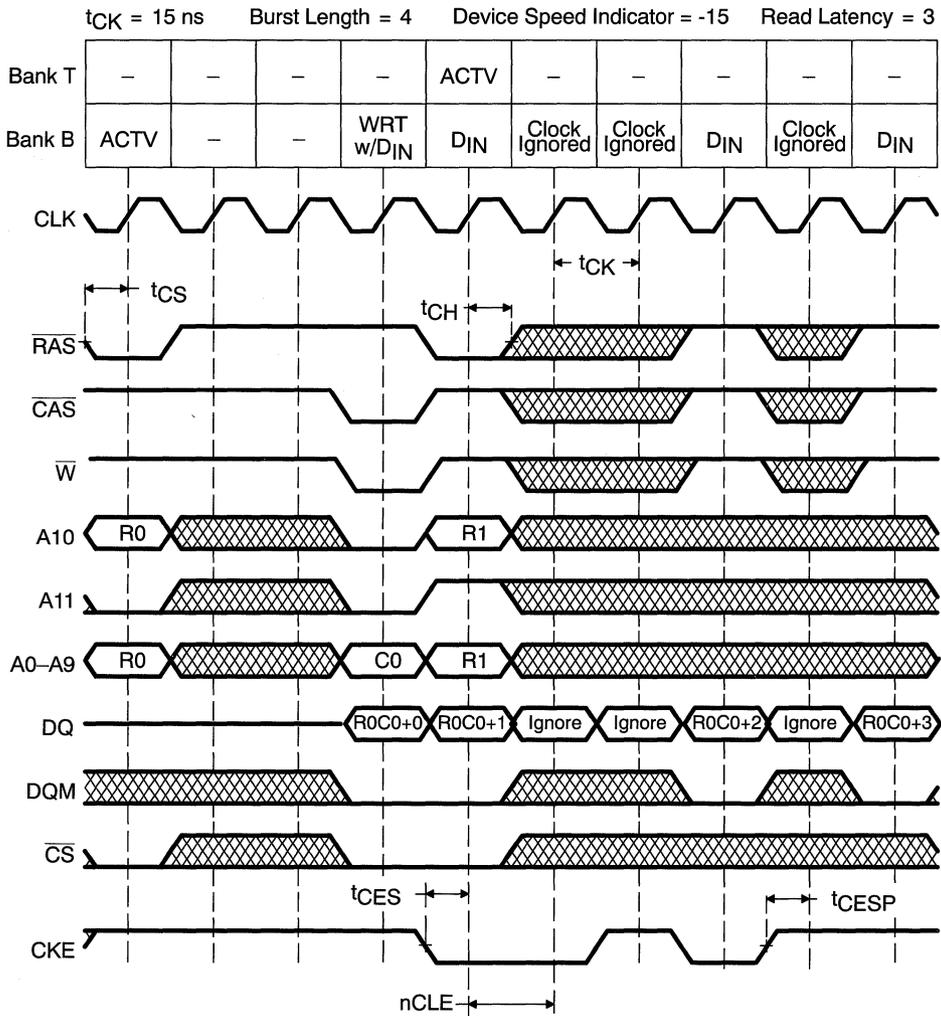
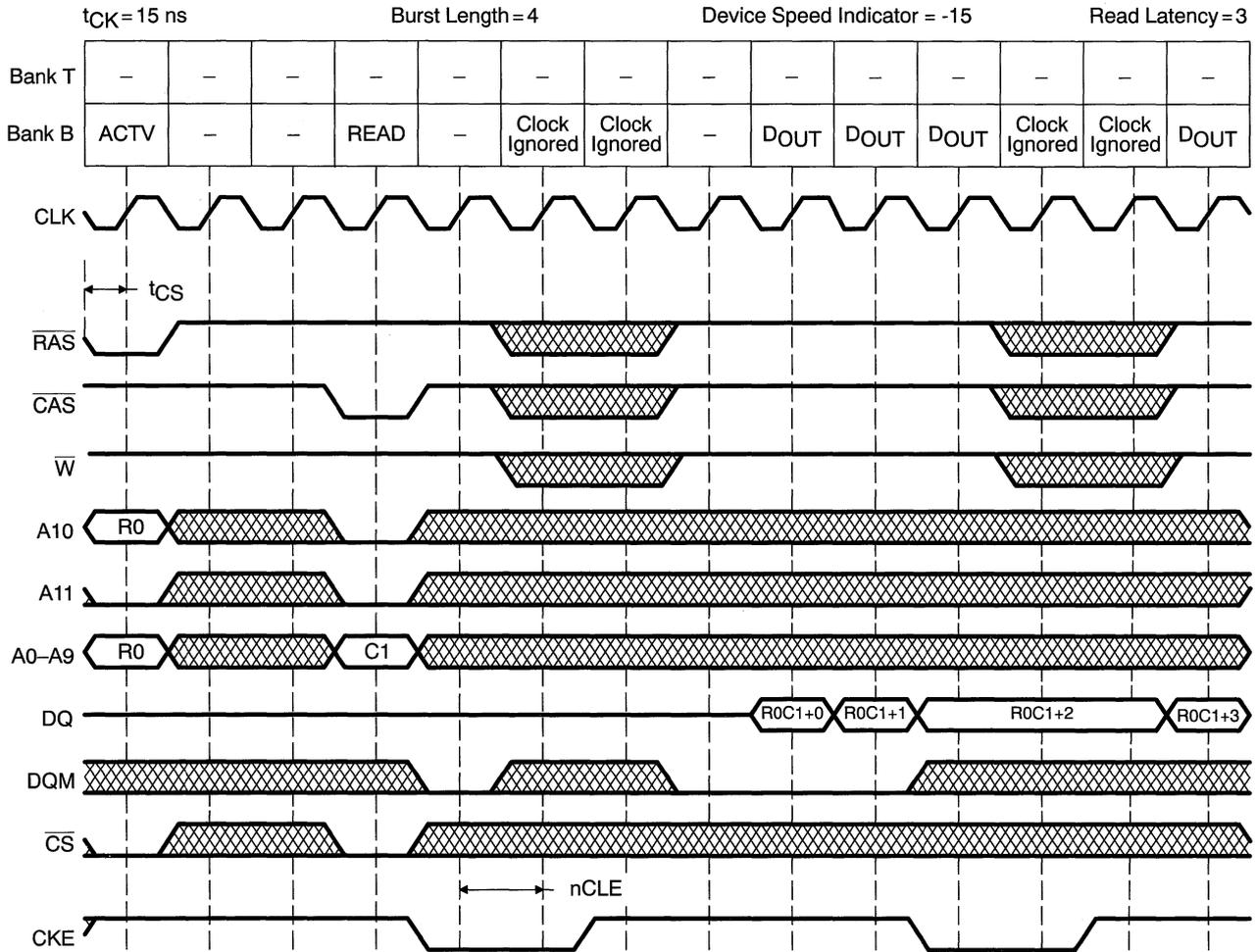


Figure 3–16. Clock Suspend During a Read Cycle

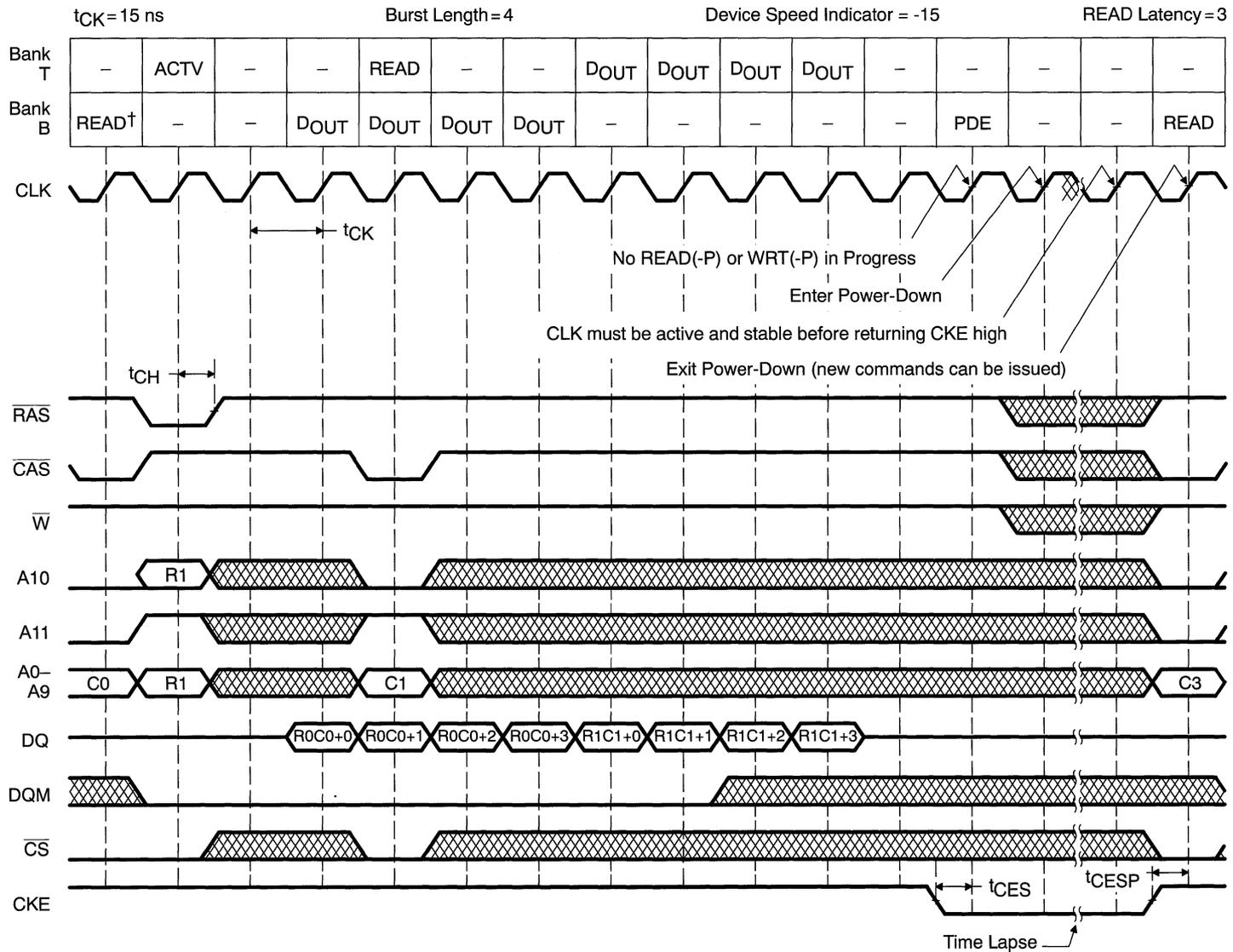


### 3.15 Power-Down-Mode Entry — Both Banks Active

The second clock-gating function of the SDRAM provides a reduced-power operating mode. This mode is entered when CKE is brought low when a bank is active and no read or write operations are in progress. The effect of holding CKE low in a power-down operation is similar to the clock-suspend case in its effect on the clock input buffer. In both cases, the external clock signal is ignored by the SDRAM. However, for a power-down operation, all of the input buffers of the SDRAM are turned off, including the clock input buffer, resulting in a significant power savings. The standby current consumed by the SDRAM is reduced by 14 mA when the device is placed in power-down mode.

The example in Figure 3–17 shows a power-down entry and exit sequence for the case when both banks of the SDRAM are active. The power-down mode is entered, as opposed to the clock-suspend mode, because the last data output in the read burst was completed on the cycle previous to the entry of the PDE command and the timing parameter  $t_{CES} = 2 \text{ ns}$  was satisfied. Like the clock-suspend case, the power-down mode takes one cycle before taking effect. Therefore, the clock must remain valid for at least one cycle after the PDE command has been entered. Once the SDRAM has entered power-down mode, the external clock can be stopped for the duration of the power-down period. However, if the clock is turned off, it must be restarted and stable before bringing CKE high. To exit the power-down mode, CKE must be brought high  $t_{CESP} = 12 \text{ ns}$  before the rising edge of the clock. If this time is met, a new command can be input to the SDRAM. In cases where the time  $t_{CESP}$  is not met, a new command can be input on the following rising edge of the clock.

Figure 3–17. Power-Down-Entry Operation, Both Banks Active

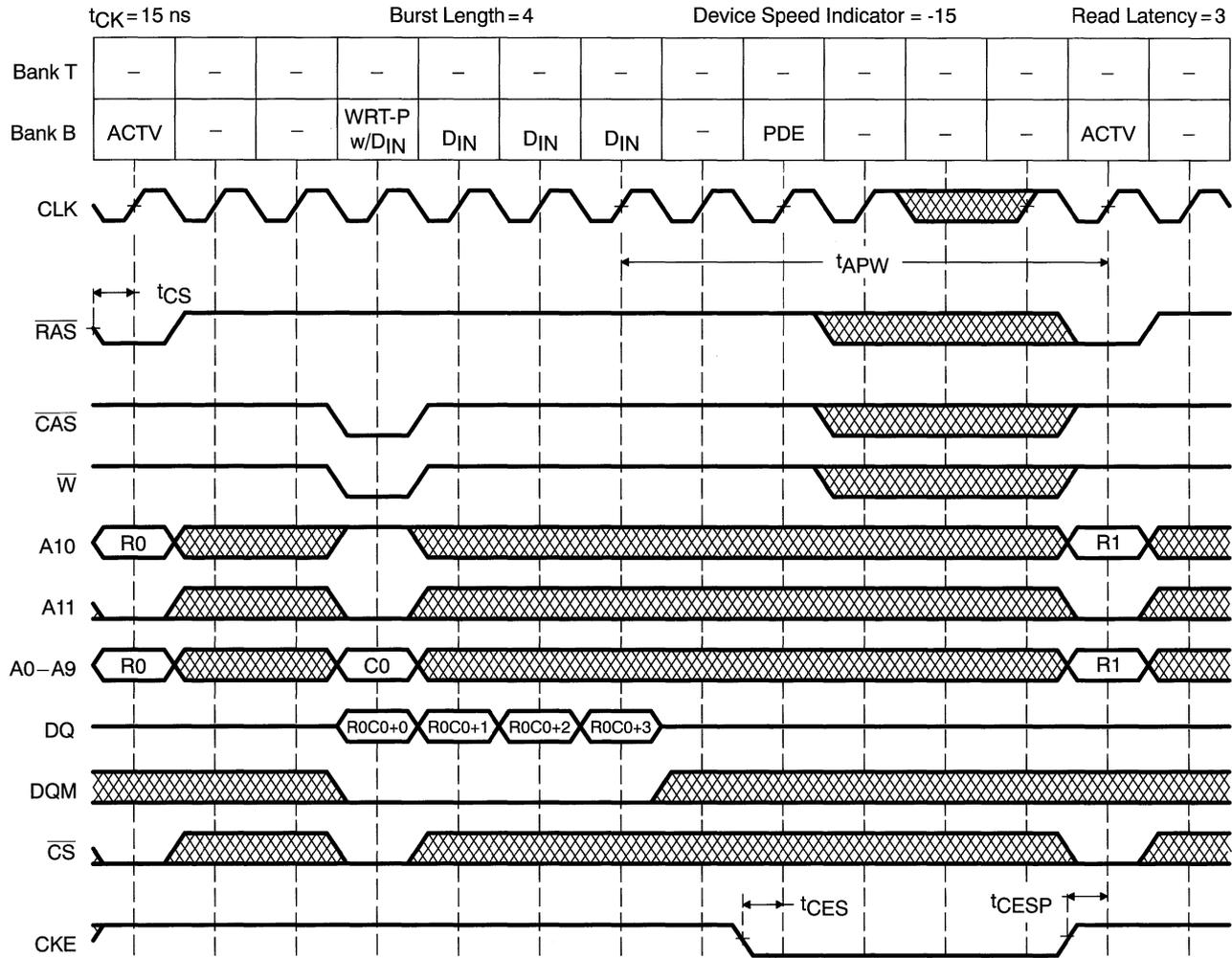


† A previous ACTV instruction activated row R0 in bank B.

### **3.16 Power-Down-Mode Entry — Both Banks Deactivated**

The last example of the SDRAM's power-down mode is for the case when both banks are inactive. As explained in the previous section, the reduction in power consumption is different depending on the state of the SDRAM. As shown in Figure 3–18, an autodeactivate write operation is used to deactivate bank B. It is assumed that bank T is already inactive. Also, from the description of the PDE command given in the previous section, there must be one NOOP or DESL cycle preceding a PDE command entry. Because of this one-cycle delay, the SDRAM has enough time to complete the write operation and turn off the internal row-address circuitry. Therefore, both banks are inactive at the time the power-down operation commences. The rest of the power-down operation continues as described previously.

Figure 3–18. Power-Down-Entry Operation, Both Ranks Deactivated



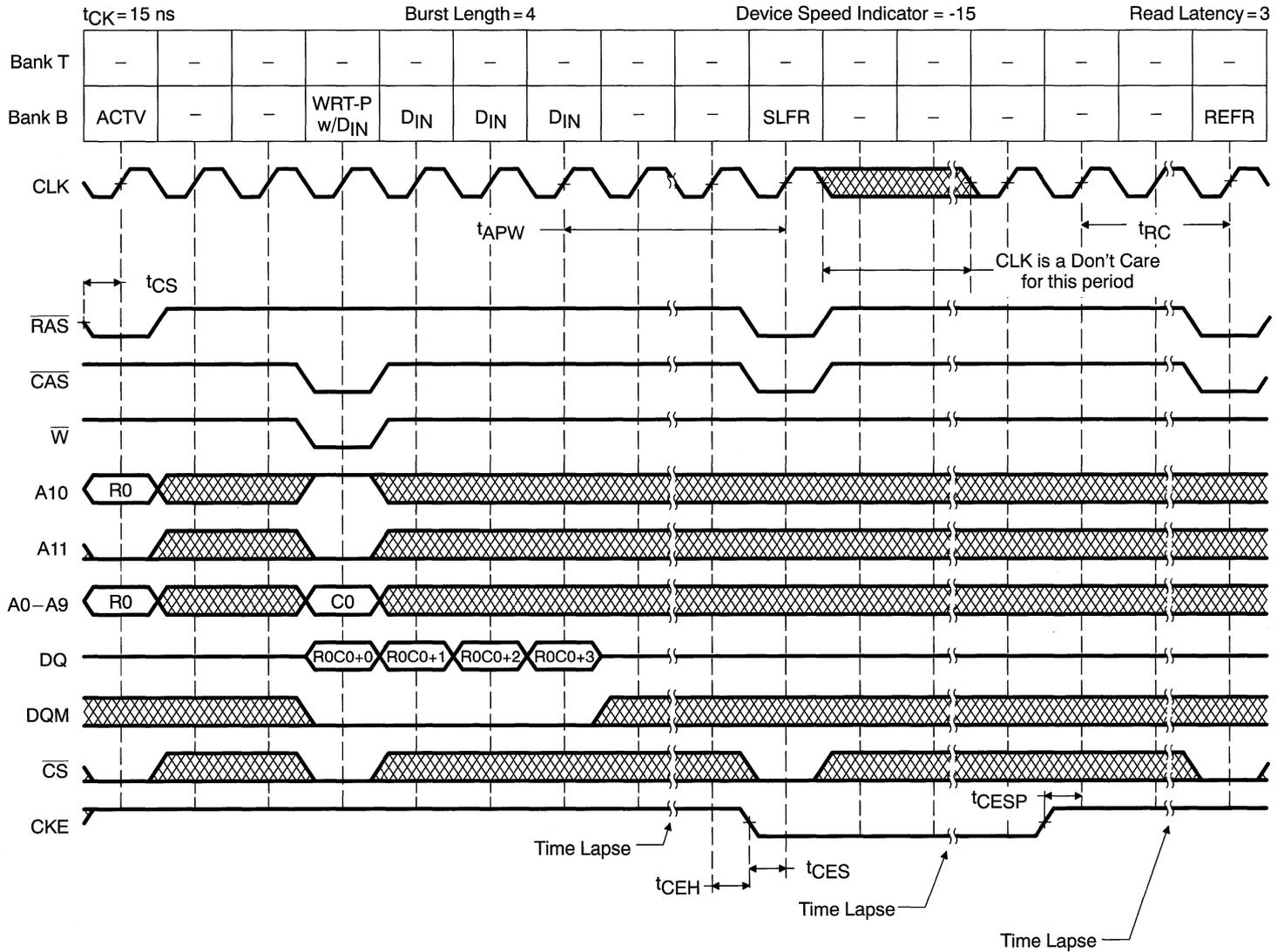
### 3.17 Self-Refresh Operation

In addition to the power-down feature, the self-refresh feature provides the capability to reduce the power consumption of the SDRAM during refresh operations. The conditions for proper execution of the SLFR command are similar to those of the power-down mode. Both modes of operation require that CKE is held low, but there are two additional constraints on the self-refresh operation. The first is that both banks must be deactivated and fully precharged prior to the SLFR command entry. The second constraint is that the first command after the self-refresh exit must wait for a time specified by  $t_{RC}$  before it can be entered. These constraints are illustrated in Figure 3–19.

In this example, bank T is considered to be fully precharged. To meet the first constraint as described in the previous paragraph, entry of the SLFR command must wait until the time  $t_{APW}$  has expired. A valid SLFR command is entered on the cycle shown, provided CKE is held high for a time determined by the hold-time specification  $t_{CEH}$  and is brought low in time to meet the setup-time specification  $t_{CES}$ . The device remains in the self-refresh mode as long as CKE is held low. As is the case during a power-down operation, CLK and all other inputs are ignored while the device is in the self-refresh mode. To exit the self-refresh mode, CKE must be brought high in time to meet the specification for  $t_{CESP}$ . If the system clock is off, it must be returned to an active and stable state prior to bringing CKE high. To meet the second constraint described in the previous paragraph, entry of the next command must wait for a time determined by the specification  $t_{RC}$ . After this time, a valid command may be entered.

Upon exiting self refresh, you must begin the normal refresh scheme immediately. If you are using a burst-refresh scheme, then 4096 REFR commands must be executed before continuing with normal device operations. If a distributed refresh scheme using CBR is employed (e.g., two rows every 32  $\mu$ s), then you must perform the first set of refreshes before continuing with normal device operation. This ensures that the SDRAM is fully refreshed.

Figure 3–19. Self Refresh



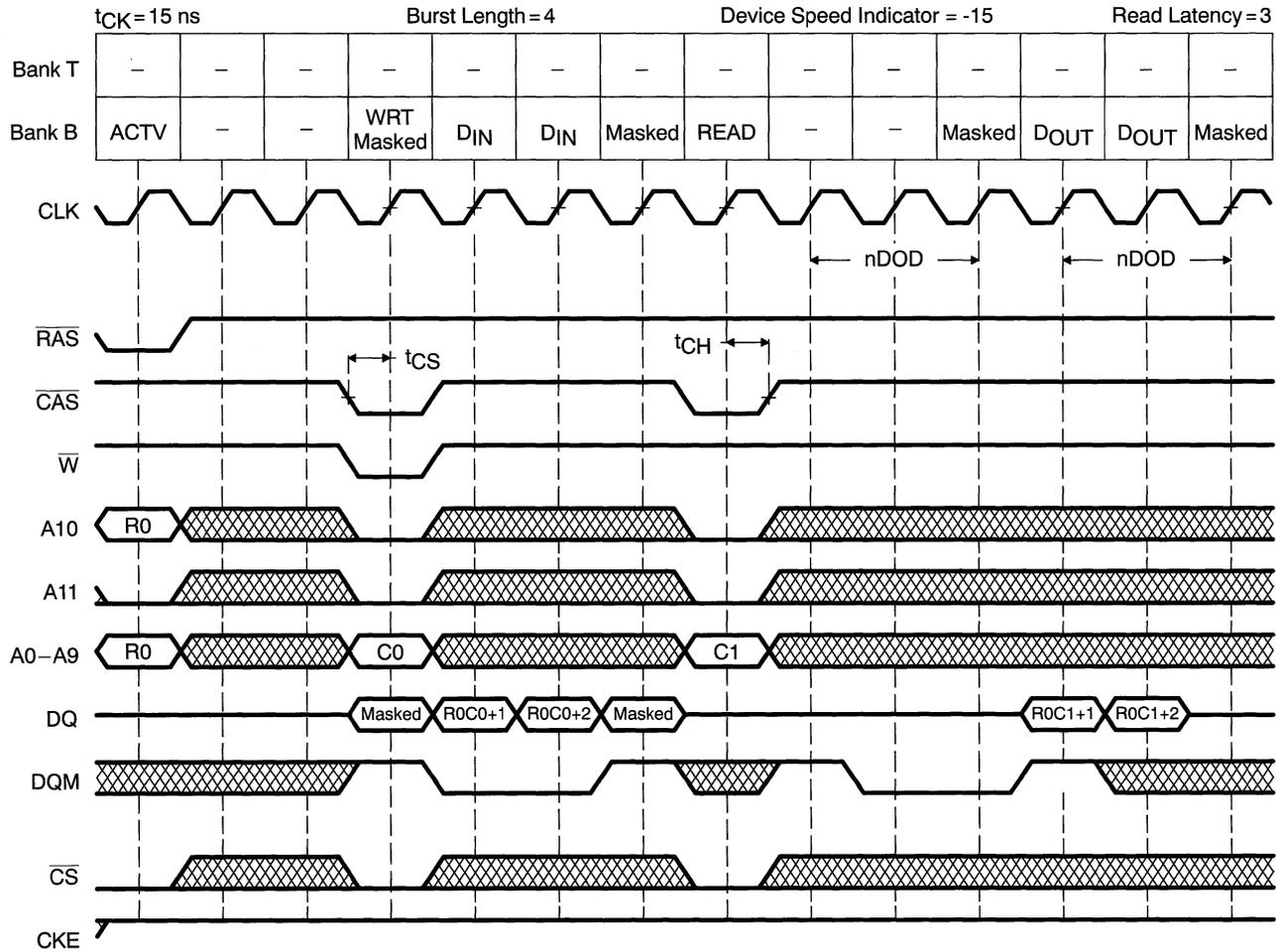
### 3.18 Data-Masking Operation

The data-masking feature of the SDRAM provides another way to perform read or write operations that are shorter than the programmed burst length. This feature is useful in situations where the majority of data accesses are four bits in duration but a small percentage of the accesses are only two bits long. A data-masking operation can be performed on both read and write cycles, and is effective on a cycle-by-cycle basis. This allows the user to mask any single data cycle in a burst sequence. The DQM input is used to signal whether the data in a given cycle is to be masked or enabled. The data to be read or written is enabled by keeping DQM at a logic low level during the appropriate cycle. The data is masked when DQM is held high, and the data is enabled when DQM is held low. These operations are performed by the MASK and ENBL commands, respectively. For write operations, the state of the input, either masked or enabled, is determined by the logic level of DQM in that same cycle. In other words, there is no latency associated with DQM input for write operations.

There is, however, a latency associated with the DQM for read operations. This delay is specified by the timing parameter nDOD and is always two cycles. Therefore, the state of the output during a read operation is determined by the logic level of the DQM input two cycles before the expected data output cycle. The reason for the two-cycle delay is related to the way the SDRAM outputs data. Since data is output by the rising clock edge of the cycle before the data is to be valid, the MASK command must be recognized one cycle before that in order to stop the data from being output. The result is the two-cycle latency for the MASK and ENBL commands.

The example in Figure 3–20 shows the timing relationships for both the mask and enable operations during write and read cycles. The first and last bits of both the read and write bursts are masked by bringing DQM high. As discussed previously, the data input on the write cycles when DQM is high is ignored by the SDRAM. For the read cycles, the outputs of the SDRAM are placed in the high-impedance state in the second cycle after DQM is brought high.

Figure 3–20. Data-Masking Operation





## Glossary

---

---

---

### A

**ACTV command:** See bank activation.

**autodeactivate read (READ-P command):** Banks are automatically deactivated at the end of the read access; a separate command does not have to be issued.

**autodeactivate write (WRT-P command):** Banks are automatically deactivated at the end of the write access; a separate command does not have to be issued.

### B

**bank activation (ACTV command):** Command sequence that causes one or both memory banks to be in the active state.

**bank deactivation (DCAB, DEAC commands):** Banks can be active only for a certain length of time. Bank deactivation is a command sequence that causes one or both banks to be in the inactive or precharge state.

**burst:** The capability of a memory device to fetch multiple addresses given only the starting address.

**burst interrupt, burst interruption:** A command issued after the beginning of a burst sequence but before the completion of the burst, which begins a new transaction without causing adverse effects. This interrupting command sequence should be entered only on even numbers of cycles from the initial burst command.

**burst length:** The number of addresses that can be fetched internally by the SDRAM before the next column address is sent.

**burst sequence:** Defines the amount and type of data the SDRAM can handle without requiring a secondary instruction. Burst sequences can be 8, 4, 2, or 1 unit in length and specify either serial or interleaved addressing.

## C

**CLK suspend:** The state the SDRAM enters when CKE goes low during a read or write operation. Clocks are ignored and the operation in progress is put in hold. The SDRAM resumes operation from the point at which it was placed in suspension once CKE is returned high. If CLK suspend is entered while both banks are deactivated, the SDRAM enters the power-down mode.

## D

**data masking input (DQM):** Inhibits read/write for the cycle. Data is not output/input during cycle, rather in the high-impedance state.

**DCAB command:** Deactivate both memory banks. See bank deactivation.

**DEAC command:** Deactivate one memory bank. See bank deactivation.

**DESL command:** Deselect command. The device remains in its current state or continues the active process for each clock cycle without need for a new command. If no process is active, the SDRAM is idle and banks are precharging. DESL is equivalent to the NOOP command.

## H

**HOLD command:** Command that initiates CLK-suspend/power-down mode.

## I

**interrupted bursts:** See burst interruption.

**interleaving:** Process by which transfer operations can occur by switching back and forth between the two banks internal to the SDRAM.

## J

**JEDEC:** Solid State Products Engineering Council (formerly Joint Electron Device Engineering Council) of the Electronic Industries Association (EIA). This council operates under EIA administrative and legal procedures and publishes JEDEC standards and publications. This council also continuously develops and maintains these standards as required by the industry.

## L

**latency:** Number of clock cycles until command takes effect. The state of the device changes after the latency period.

## M

**MASK command:** Command that initiates data or output masking; see masking.

**masking:** Process by which individual data cycles within a burst sequence can be ignored or disabled (placed in the high-impedance state).

**mode-register set:** The command process by which the programmable features of the SDRAM are defined. Such features include serial or interleave burst type, defining system read latency and defining burst length.

**MRS command:** Mode-register set command. See mode-register set.

## N

**nBSD:** Number of cycles from entry of STOP command to entry of next command.

**nCCD:** Column-to-column address delay, or initial-command-burst-to-interrupting-command delay (in cycles).

**nCLE:** Number of cycles from either the HOLD command entry to the suspended CLK edge or from CLK-suspend exit to entry of any valid command.

**nEP:** Number of cycles from final data out to bank deactivation.

**nDOD:** Number of cycles from an ENBL or MASK command to output data valid.

**nHZP:** Number of cycles from bank-deactivation (DEAC or DCAB) interrupt of a read burst to the data outputs going into the high-impedance state.

**NOOP command:** Similar to DESL command. See DESL command.

**nRSA:** The minimum number of cycles that must elapse between MRS command entry and entry of any other valid command.

**nWCD:** The number of cycles from a valid WRT command to the first valid data at the inputs.

**P**

**PDE command:** See power-down enable command.

**precharge:** The state of the memory bank when deactivated.

**ping-pong operation:** Process of interleaving accesses in both banks by going back and forth between bank T and bank B.

**power-down enable command (PDE):** Puts part into low power consumption mode, thereby saving stand-by power.

**power-down mode:** The power-saving mode the SDRAM enters if CKE goes low while both banks are inactive. To ensure validity of data during power-down mode, execute the SLFR command concurrently with the PDE command to activate self refresh.

**R**

**READ command:** Initiates a read access to the activated bank of the SDRAM.

**read latency:** The delay between the read command (READ) and the first output burst (also referred to as  $\overline{\text{CAS}}$  latency).

**READ-P command:** See autodeactivate read.

**REFR command:** Initiates a  $\overline{\text{CAS}}$ -before- $\overline{\text{RAS}}$  refresh of the SDRAM after both banks have been deactivated.  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  must be low and  $\overline{\text{W}}$  must be high on the rising edge of CLK.

**refresh:** DRAM or SDRAM operation by which data is retained in the memory. In the SDRAM, refresh can be accomplished by performing an activate/deactivate command sequence to every row in both banks, by performing 4096 REFR commands, or by placing the device in self refresh.

**S**

**SDRAM:** Synchronous dynamic random-access memory. A memory device in which DRAM operations are synchronized to the system clock.

**seamless (gapless) operation:** Process by which interleaved operations of the SDRAM tightly coincide such that data is transferred each clock cycle without a gap or interruption. This is the fast mode of operation the SDRAM can perform with potential rates of 100 MHz depending on the device speed indicator of the product selected.

**SLFR command:** See self refresh.

**STOP command:** Discontinues acceptance of data at the inputs during a write access, or completes only the read access in process, whichever is applicable. The bank remains active, but a new command must be entered to start any operation.

**self refresh:** Type of memory refresh that requires CKE to be held low in addition to  $\overline{\text{CAS}}$  and  $\overline{\text{RAS}}$  being low and  $\overline{\text{W}}$  being high for one cycle. Sustaining the self-refresh action for subsequent cycles requires only that CKE remain low. Data is retained indefinitely while power is maintained. Power consumption is reduced to a minimum in self refresh.

**W**

**write latency:** The delay between the write command (WRT) and the time data is accepted at the memory inputs. There is no write latency for the SDRAM.

**WRT command:** Initiates a write access to the activated bank of the SDRAM.

**WRT-P command:** See autodeactivate write.



# Index

## A

ACTV command 2-5, 2-9, 3-4, 3-9, 3-17  
autodeactivate read (READ-P) 2-6, 3-9 to 3-11  
autodeactivate write (WRT-P) 2-6, 3-17 to 3-19,  
3-44

## B

bank activate command (ACTV) 2-5  
bank activation 1-3, 2-5  
bank deactivation (precharge) 1-2, 2-6  
block diagram 2-2  
burst interruptions 2-10  
burst length 1-3  
burst sequence 2-3

## C

$\overline{\text{CAS}}$ -before- $\overline{\text{RAS}}$  (CBR) refresh 2-9  
chip select ( $\overline{\text{CS}}$ ) 2-6  
CLK-suspend mode 2-7, 3-38 to 3-40  
clock suspend 3-38 to 3-40

## D

data throughput 1-3  
data-masking operation 3-48  
data/output mask 2-7  
DCAB command 2-5, 2-6, 3-2, 3-28, 3-36  
DEAC command 2-5, 2-6, 2-9, 3-2, 3-4 to 3-8,  
3-17 to 3-20, 3-28, 3-36  
design comparison: JEDEC-standard versus TI's  
SDRAM 2-10 to 2-12

DESL command 2-6, 3-44  
detailed operations on the 16M-bit  
SDRAM 3-1 to 3-50  
differences between the SDRAM and the  
DRAM 1-2  
DQM input 3-23

## E

ENBL command 2-7, 3-48

## F

functional comparison: SDRAM versus  
DRAM 1-2 to 1-4

## G

gapless (seamless) accesses 2-5, 3-5, 3-21  
general description of 16M-bit SDRAM 2-1 to 2-12

## H

HOLD command 2-7, 3-38

## I

interleaving operation 1-2, 1-3, 2-3, 2-5, 3-4, 3-9,  
3-13, 3-17  
interrupted bursts 2-10  
interrupting a read burst 3-21 to 3-30  
*See also* read burst  
interrupting a write burst 3-30 to 3-37  
*See also* write burst  
introduction 1-1 to 1-4

**J**

JEDEC-standard SDRAM features 2-10  
 prefetch and pipeline comparison 2-11  
 versus TI SDRAM 2-10

**L**

latency 2-4, 2-7, 3-4, 3-13, 3-17, 3-26, 3-38, 3-48

**M**

MASK command 2-7, 3-26, 3-48  
 masking 2-7, 3-23, 3-48  
 mode-register set 2-8  
 MRS command 2-8, 3-2

**N**

nCCD specification 2-10, 3-21, 3-23, 3-26, 3-28,  
 3-30, 3-32, 3-34, 3-36  
 nCLE specification 3-38, 3-39  
 nDOD specification 3-23, 3-48  
 nEP specification 3-5 to 3-8, 3-9 to 3-12, 3-28  
 nHZP specification 3-28  
 NOOP command 2-6, 3-2, 3-44  
 nRSA specification 3-2  
 nWCD specification 3-13

**O**

overview of the SDRAM 2-2 to 2-3

**P**

PDE command 2-7, 3-42, 3-44  
 ping-pong operation 2-5  
 power-down mode (PDE command) 2-7, 3-38  
 power-down-mode entry — both banks active 3-42  
 power-down-mode entry — both banks  
 deactivated 3-44 to 3-46  
 power-up sequence 3-2 to 3-4

precharge (bank deactivation), 1-2, 2-6  
 program features  
 burst sequence 2-3  
 burst type 2-3

**R**

read  
 with bank interleaving — random row  
 address autodeactivate 3-9 to 3-10  
 with bank interleaving — seamless 3-4 to 3-11  
 read burst  
 interrupted by a DEAC/DCAB command 3-28  
 interrupted by a READ command 3-21  
 interrupted by a STOP command 3-26  
 interrupted by a WRT command 3-23 to 3-25  
 READ command 3-4 to 3-8, 3-21, 3-23,  
 3-26 to 3-28, 3-28 to 3-31  
 read latency 1-3, 3-4  
 READ-P command 2-6, 3-9 to 3-12  
 interruption not allowed 3-21  
 reduced-power operating mode 3-42  
*See also* power-down mode  
 REFR command 2-9, 3-2, 3-46  
 refresh 2-9  
 $\overline{\text{CAS}}$ -before- $\overline{\text{RAS}}$  (CBR) refresh 2-9  
 self refresh 2-9 to 2-12  
 row-address entry. *See* bank activation

**S**

SDRAM  
 block diagram 2-2  
 overview 2-2  
 versus DRAM 1-2  
 seamless (gapless) operation 2-5, 3-4, 3-5, 3-21  
 seamless read with bank interleaving 3-4 to 3-6  
 self-refresh operation 2-9, 3-46  
 SLFR command 2-9, 3-46  
 STOP command 3-26, 3-34  
 penalty associated with 3-26  
 synchronous DRAM definition 1-2  
 system clock input (CLK) 2-2

**T**

throughput 1-3  
two-bank column-access operation 2-5  
two-bank operation 2-5  
two-bank row-access operation 2-5

**W**

write  
with bank interleaving 3-13 to 3-14  
with bank interleaving — random row  
address autodeactivate 3-17 to 3-18

write burst  
interrupted by a DEAC/DCAB  
command 3-36 to 3-37  
interrupted by a READ command 3-30 to 3-31  
interrupted by a STOP command 3-34 to 3-35  
interrupted by a WRT command 3-32  
WRT command 2-5, 3-13, 3-23, 3-26, 3-32, 3-36  
WRT-P command 2-6, 3-17, 3-23, 3-30, 3-32  
interruption not allowed 3-30



# TI Worldwide Sales and Representative Offices

**AUSTRALIA / NEW ZEALAND:** Texas Instruments Australia Ltd.: Sydney [61] 2-910-3100, Fax 2-805-1186; Melbourne 3-696-1211, Fax 3-696-4446.  
**BELGIUM:** Texas Instruments Belgium S.A./N.V.: Brussels [32] (02) 242 75 80, Fax (02) 726 72 76.  
**BRAZIL:** Texas Instrumentos Electronicos do Brasil Ltda.: Sao Paulo [55] 11-535-5133.  
**CANADA:** Texas Instruments Canada Ltd.: Montreal (514) 335-8392; Ottawa (613) 726-3201; Toronto (416) 884-9181.  
**DENMARK:** Texas Instruments A/S: Ballerup [45] (44) 68 74 00.  
**FINLAND:** Texas Instruments/OY: Espoo [358] (0) 43 54 20 33, Fax (0) 46 73 23.  
**FRANCE:** Texas Instruments France: Vellyz-Villacoublay Cedex [33] (1) 30 70 10 01, Fax (1) 30 70 10 54.  
**GERMANY:** Texas Instruments Deutschland GmbH.: Freising [49] (08161) 80-0, Fax (08161) 80 45 16; Hannover (0511) 90 49 60, Fax (0511) 64 90 331; Ostfildern (0711) 34 03 0, Fax (0711) 34 032 57.  
**HONG KONG:** Texas Instruments Hong Kong Ltd.: Kowloon [852] 956-7288, Fax 956-2200.  
**HUNGARY:** Texas Instruments Representation: Budapest [36] (1) 269 8310, Fax (1) 267 1357.  
**INDIA:** Texas Instruments India Private Ltd.: Bangalore [91] 80 226-9007.  
**IRELAND:** Texas Instruments Ireland Ltd.: Dublin [353] (01) 475 52 33, Fax (01) 478 14 63.  
**ITALY:** Texas Instruments Italia S.p.A.: Agrate Brianza [39] (039) 68 42.1, Fax (039) 68 42.912; Rome (06) 657 26 51.  
**JAPAN:** Texas Instruments Japan Ltd.: Tokyo [81] 03-769-8700, Fax 03-3457-6777; Osaka 06-204-1881, Fax 06-204-1895; Nagoya 052-583-8691, Fax 052-583-8696; Ishikawa 0762-23-5471, Fax 0762-23-1583; Nagano 0263-33-1060, Fax 0263-35-1025; Kanagawa 045-338-1220, Fax 045-338-1255; Kyoto 075-341-7713, Fax 075-341-7724; Saitama 0485-22-2440, Fax 0425-23-5787; Ohta 0977-73-1557, Fax 0977-73-1583.  
**KOREA:** Texas Instruments Korea Ltd.: Seoul [82] 2-551-2800, Fax 2-551-2828.  
**MALAYSIA:** Texas Instruments Malaysia: Kuala Lumpur [60] 3-230-6001, Fax 3-230-6805.  
**MEXICO:** Texas Instruments de Mexico S.A. de C.V.: Colina del Valle [52] 5-639-9740.  
**NORWAY:** Texas Instruments Norge A/S: Oslo [47] (02) 264 75 70.  
**PEOPLE'S REPUBLIC OF CHINA:** Texas Instruments China Inc.: Beijing [86] 1-500-2255, Ext. 3750, Fax 1-500-2705.  
**PHILIPPINES:** Texas Instruments Asia Ltd.: Metro Manila [63] 2-817-6031, Fax 2-817-6096.  
**PORTUGAL:** Texas Instruments Equipamento Electronico (Portugal) LDA.: Maia [351] (2) 948 10 03, Fax (2) 948 19 29.  
**SINGAPORE / INDONESIA / THAILAND:** Texas Instruments Singapore (PTE) Ltd.: Singapore [65] 390-7100, Fax 390-7062.  
**SPAIN:** Texas Instruments España S.A.: Madrid [34] (1) 372 80 51, Fax (1) 372 82 66; Barcelona (3) 31 791 80.  
**SWEDEN:** Texas Instruments International Trade Corporation (Sverigefilialen): Kista [46] (08) 752 58 00, Fax (08) 751 97 15.  
**SWITZERLAND:** Texas Instruments Switzerland AG: Dietikon [41] 886-2-3771450.  
**TAIWAN:** Texas Instruments Taiwan Limited: Taipei [886] (2) 378-6800, Fax 2-377-2718.  
**UNITED KINGDOM:** Texas Instruments Ltd.: Bedford [44] (0234) 270 111, Fax (0234) 223 459.

**UNITED STATES:** Texas Instruments Incorporated: **ALABAMA:** Huntsville (205) 430-0114; **ARIZONA:** Phoenix (602) 244-7800; **CALIFORNIA:** Irvine (714) 660-1200; San Diego (619) 278-9600; San Jose (408) 894-9000; Woodland Hills (818) 704-8100; **COLORADO:** Aurora (303) 368-8000; **CONNECTICUT:** Wallingford (203) 265-3807; **FLORIDA:** Orlando (407) 260-2116; Fort Lauderdale (305) 425-7820; Tampa (813) 882-0017; **GEORGIA:** Atlanta (404) 662-7967; **ILLINOIS:** Arlington Heights (708) 640-2925; **INDIANA:** Indianapolis (317) 573-6400; **KANSAS:** Kansas City (913) 451-4511; **MARYLAND:** Columbia (410) 312-7900; **MASSACHUSETTS:** Boston (617) 895-9100; **MICHIGAN:** Detroit (303) 553-1500; **MINNESOTA:** Minneapolis (612) 828-9300; **NEW JERSEY:** Edison (908) 906-0033; **NEW MEXICO:** Albuquerque (505) 345-2555; **NEW YORK:** Poughkeepsie (914) 897-2900; Long Island (516) 454-6601; Rochester (716) 385-6770; **NORTH CAROLINA:** Charlotte (704) 522-5487; Raleigh (919) 876-2725; **OHIO:** Cleveland (216) 765-7258; Dayton (513) 427-6200; **OREGON:** Portland (503) 643-6758; **PENNSYLVANIA:** Philadelphia (215) 825-9500; **PUERTO RICO:** Hato Rey (809) 753-8700; **TEXAS:** Austin (512) 250-6769; Dallas (214) 917-1264; Houston (713) 778-6592; **WISCONSIN:** Milwaukee (414) 798-1001.

## North American Authorized Distributors

### COMMERCIAL

Almac / Arrow  
Anthem Electronics  
Arrow / Schweber  
Future Electronics (Canada)  
Hamilton Hallmark  
Marshall Industries  
Wyle

### OBSOLETE PRODUCTS

Rochester Electronics 508/462-9332

### MILITARY

Alliance Electronics Inc  
Future Electronics (Canada)  
Hamilton Hallmark  
Zeus - An Arrow Company  
**CATALOG**  
Allied Electronics  
Arrow Advantage  
Newark Electronics

*For Distributors outside North America, contact your local Sales Office.*

A1194

**Important Notice:** Texas Instruments (TI) reserves the right to make changes to or to discontinue any product or service identified in this publication without notice. TI advises its customers to obtain the latest version of the relevant information to verify, before placing orders, that the information being relied upon is current. Please be advised that TI warrants its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. TI assumes no liability for applications assistance, software performance, or third-party product information, or for infringement of patents or services described in this publication. TI assumes no responsibility for customers' applications or product designs.



© 1994 Texas Instruments Incorporated

Printed in the U.S.A.

