EDRAM® Controller for the 60 MHz TMS320C40 DSP

APPLICATION REPORT: SPRA290

Digital Signal Processing Solutions
June 1998

Texas Instruments

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 and complete.

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 application 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.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Quickcache is a registered trademark of Enhanced Memory Systems.

EDRAM is a registered trademark of Enhanced Memory Systems. Revision 2.0.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

Figures

Tables

EDRAM® Controller for the 60 MHz TMS320C40 DSP

Abstract

The popular Texas Instruments (TI™) TMS320C4X family of 32-bit floating-point digital signal processors (DSPs) has a memory bus that operates at 30 MHz. The 32-bit local external memory bus supports a maximum read bandwidth of 120 Mbytes per second and a maximum write bandwidth of 60 Mbytes per second. The processor has two on-board programmable timers to simplify the generation of a refresh clock. The TMS320C40 is available in 50 and 60 MHz clock speed options.

The enhanced DRAM (EDRAM) memory with Quickcache® from Enhanced Memory Systems provides high performance for the TMS320C40 DSP at 60 MHz processor clock rates. Its fast 12ns access time requires no wait states for any read within a page. Unless preceded by a write, the EDRAM's fast random access performance allows read hit cycles to be completed with no wait states. This performance is achieved with a single non-interleaved memory bank consisting of as few as four 512K x 8 components (2Mbytes). Fast 12ns SRAM memory used for DSP external memory has slightly better performance but is more expensive at this memory size. Standard DRAM memory is too slow for most high performance DSP applications.

This application report describes the design of the 60 MHz EDRAM controller.

Product Support

Related Documentation

The following list specifies product names, part numbers, and literature numbers of corresponding TI documentation.

- *TMS320C40 Digital Signal Processor* Data Sheet, January 1998, Literature number SPRS038
- *TMS320C4x User's Guide*, March 1996, Literature number SPRU063

World Wide Web

Our World Wide Web site at **www.ti.com** contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

Email

For technical issues or clarification on switching products, please send a detailed email to **dsph@ti.com**. Questions receive prompt attention and are usually answered within one business day.

Introduction

The Texas Instruments TMS320C4X family of 32-bit floating-point digital signal processors are popular for DSP applications. These processors have a memory bus that operates at 30 MHz. The 32-bit local external memory bus supports a maximum read bandwidth of 120 Mbytes per second and a maximum write bandwidth of 60 Mbytes per second. The processor has two on-board programmable timers to simplify the generation of a refresh clock. The TMS320C40 is available in 50 and 60 MHz clock speed options.

The enhanced DRAM (EDRAM) memory with Quickcache® from Enhanced Memory Systems provides high performance for the TMS320C40 DSP at 60 MHz processor clock rates. Its fast 12ns access time requires no wait states for any read within a page. Unless it is preceded by a write, the EDRAM's fast random access performance allows read hit cycles to be completed with no wait states. This performance is achieved with a single non-interleaved memory bank consisting of as few as four 512K x 8 components (2Mbytes). Fast 12ns SRAM memory used for DSP external memory has slightly better performance but is more expensive at this memory size. Standard DRAM memory is too slow for most high performance DSP applications.

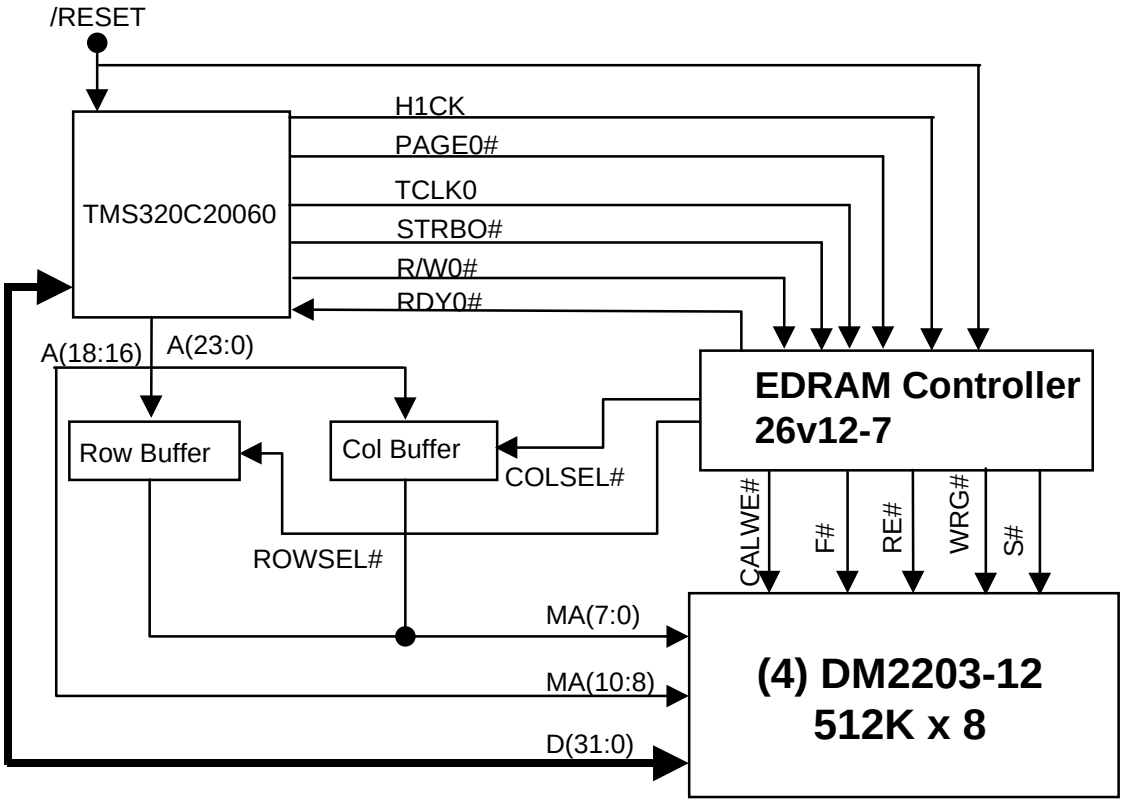
Table 1. Performance Comparison at 30 MHz Memory Bus Clock

Transaction	**EDRAM (12ns)** †	**SRAM (12ns)** †
Page Read Hit	1 Cycle	1 Cycle
Page Read Miss	2 Cycles	1 Cycle
Write	2 Cycles	2 Cycles

† If preceded by a write, add one cycle

A single 26V12-7 PAL device and two FCT buffers (FCT541CT) are the only components necessary to interface a single 2Mbyte bank of EDRAM to the local memory bus. The controller design supports a 60 MHz processor clock rate using 12ns EDRAM. Figure 1 shows the system block diagram.

Figure 1. TMS320C40 System Block Diagram



EDRAM Controller Design

The objective of this controller design is to support TMS320C40 memory transactions with minimum wait states. The controller is designed to support one EDRAM bank of 2 Mbytes. Memory size can be expanded by adding additional control outputs to the logic. The TMS320C40 local bus interface is programmable to control wait states and HOLD operation. The memory interface control register for the local bus should be programmed with the following parameters:

- SWW is programmed to respond only to the external RDY# input for wait states (SWW=00). The internal /RDY_{WTCNT} is ignored.
- WTCNT does not need to be programmed since software programmable wait states are not used.
- STRB0 PAGESIZE is set as shown in the table below:

Table 2. STRB0 PAGESIZE

EDRAM	MSBs Defining a Bank	LSBs Defining a Column	Bank Size	STRB0# PAGE SIZE
512K x 8	A(30:8)	A(7:0)	256	01000

The TMS320C40 supports the following memory transactions:

- 32-bit Page Read Hit
- 32-bit Page Read Miss
- 32-bit Write

To support these bus transactions, the EDRAM and/or EDRAM memory controller must interface with the following processor control and address signals:

A18-A0 Address Bus
STRB0# External Access Strobe
PAGE0# Page Detect Output
R/W0# Read/Write Mode Outputs
RDY0# Ready Input
Reset# Reset Input
H1CLK Processor Clock Output
TCLK0 Timer Zero Output Clock/Pulse

The controller generates the following signals to control the EDRAM:

ROWSEL#	Multiplexed Row Address Enable
COLSEL#	Multiplexed Column Address Enable
RE#	Row Enable
CALWE#	Column Address Strobe/Write Enable
WRG#	Combined W/R and /G
F#	Refresh
S#	Chip Select

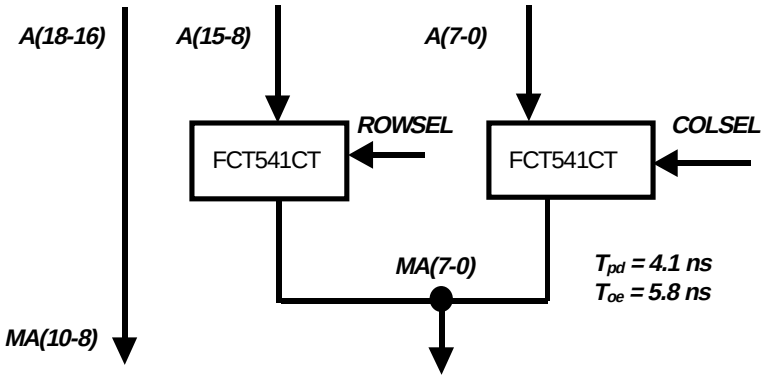
The EDRAM W/R and /G control lines are tied together to form WRG#. The ROWSEL#, COLSEL#, CALWE#, WRG# and F# signals are supplied from the 26V12 PAL. S# can be generated from the PLD or from NAND or NOR gates. When the memory interface is in either the idle or any refresh state, the S# output is brought high to reduce the EDRAM power consumption. Optionally, S# can be tied to ground if the EDRAM power consumption is not an issue. If S# is not used, the EDRAM controller can fit into a 22V10 PLD.

EDRAM Controller Functional Description

This section describes the EDRAM controller block diagram shown in Figure 1. The Refresh Counter is formed using an internal timer in the C40 to generate the periodic refresh signal every 40µsec for the state machine. This refresh signal will trigger an F# refresh on the next available bus cycle. Up to three wait states may be inserted to the processor while a bus transaction is occurring. The Address Multiplexer selects the row or column address to the EDRAM multiplex address lines under control of the state machine. The multiplexer is implemented using two 74FCT541CT 8-bit buffer chips. These chips were selected because of their fast through delay (4.1ns) and fast select delay (5.8ns). Column address is connected to A(7:0) for 512K x 8 EDRAM. The row address comprises the 11 local address bits above the column address.

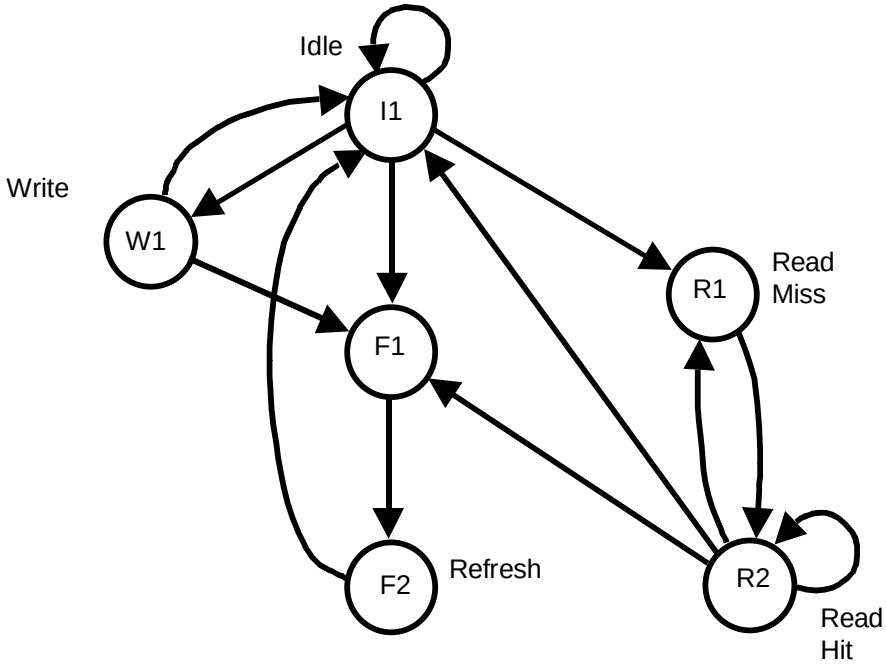
Address lines A(10:0) of the EDRAM are connected directly to A(18:16) of the C40 DSP, respectively. The upper address lines are ignored during column time, so there is no need to feed these address lines through a buffer.

Figure 2. Row/Column Address Multiplexing



The State Machine operates synchronously with the rising edge of the H1CLK.

Figure 3. EDRAM Controller State Machine



One of the following memory control sequences is selected based on the control input status. Read and write sequences are described. All states are determined at the rising edge of H1CLK.

Reset Sequence

When Reset# is low, the processor is in its reset state. The EDRAM controller continuously executes F# refresh cycles until Reset# returns high. This meets the EDRAM initialization requirement of eight F# refresh cycles during startup. It is assumed that a startup control program will be run immediately following reset. This program must configure the TMS320C40 to perform two read miss cycles to different row addresses for each internal EDRAM bank. Row address bits A8 and A9 define the four internal EDRAM banks. This startup procedure must be performed for proper EDRAM operation.

Page Read Miss Sequence

Stating from the idle state, when STRB0# is low and R/W# is high, a page read miss sequence is executed. The row address will be selected, and WRG# signal will be brought low. The RE# signal is then asserted clocked to latch the row address. RDY0# is held high until the middle of cycle R1 to insert a wait state for the processor. Data is available to the processor at the end of cycle two. Because of the TMS320C40's addressing scheme, one wait state is inserted when a read cycle follows a write cycle.

Page Read Hit Sequence

Starting from the R2 state, when STRB# is low, R/W# is high, and PAGE0# is low, a page read hit sequence is executed. The column address is selected, and WRG# is brought low to gate data on to the data bus. The EDRAM can support back-to-back in-page reads from cache in zero wait states. The TMS320C40 processor will hold STRB# and PAGE0# low on back-to-back page hit reads. On a page read miss, STRB0# and PAGE0# will go high to cause a page read miss sequence to be executed.

Write Sequence	If STRB# is low and R/W# is low, a write sequence is executed. The row address is selected, and WRG# is brought high. The RE# signal is asserted to latch the row address. When the column address and write data are stable, the CALWE# signal is asserted to latch the column address and write data. Following the write, RE# and CALWE# are brought high to terminate the write cycle. The write is completed in two cycles or zero wait states. The EDRAM supports single and back-to-back writes in two cycles or zero wait states. The C40 does not support single cycle writes, therefore PAGE0# is not used on write cycles.
Refresh Sequence	If the TCLK0 input is high, an F# refresh sequence is performed after any current memory cycles are completed. The TCLK0 output will be high for 40ns and stay low until the next refresh period. The F# pin is brought low and RE# is clocked to perform an internal refresh using the internal refresh counter. A refresh is executed every 40µsec. During the refresh sequence, RDY# is held high to cause the processor to wait. The processor may need to wait for up to three wait states to complete the current memory transaction.
Idle	If STRB# is high (except during the W2 state), this indicates the cycle is an idle cycle. The WRG# signal is brought high to disable the EDRAM outputs from the data bus until the next active cycle.
Chip Select (S#)	The S# can be generated from the PLD or by using simple NAND or NOR gates. S# is brought high during the idle and refresh states to reduce the power consumption. This circuit is optional. S# can be tied to ground if reduced power consumption is not desired or if the pin count must be reduced.
Programming the TMS320C40 Timer	The Timer Global Control Register must be programmed immediately after power up so that the refresh counter can be initialized properly. The TCLK0 output is used to generate a 40ns minimum high pulse every 40µsec.

The following bit values are programmed:

CLKSRC=1
FUNC-0
I/O=0
HLD=1
Period Register Value=600

$$\text{period register} = \frac{f_{INT}}{f(\text{Timer Clock})} = \frac{15\,\text{MHz}}{25\,\text{KHz}} = 600$$

The remaining bit fields can be left at their default values after power up. The interrupt enable register does not need to be programmed to generate an interrupt when timer zero generates a high pulse every 40µsec. The timer must be started immediately after power up by setting the GO/HLD bits in the Timer Global Control Register to “11”. When the counter value equals the value set in the period register, a 40ns high pulse will be output on TCLK0 pin. This high pulse will cause the EDRAM state machine to record a pending refresh on the next rising edge of H1CLK. It will be cleared after a refresh sequence is performed. The C40 timer does not need to be restarted and no interrupt will be generated. A refresh sequence will not be started until after the second cycle of a read miss or write sequence, if one is in progress.

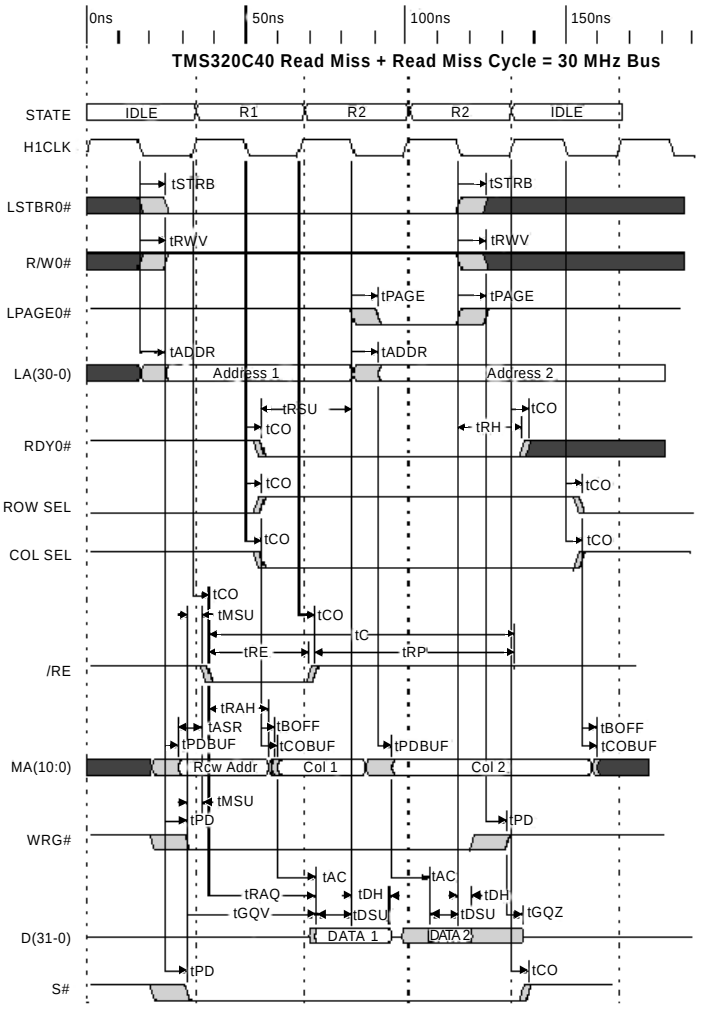
The attached timing diagrams show the timing of the EDRAM controller operating in a 60 MHz TMS320C40 system with the 12ns version of the EDRAM. The EDRAM controller parameters are for 26V12-7 PLD and 74FCT541CT 8 – buffers. The TMS320C40 timing parameters are taken from the TI *TMS320C4x User’s Guide*, 1996.

Summary

A simple EDRAM controller for the 60 MHz Texas Instruments TMS320C40 DSP can be implemented using a simple PLD device and two FCT buffers. The controller supports one EDRAM bank of 2M bytes (using DM2203 components) and can easily be expanded to more memory. The EDRAM system achieves zero wait states on read hit cycles and on all write cycles.

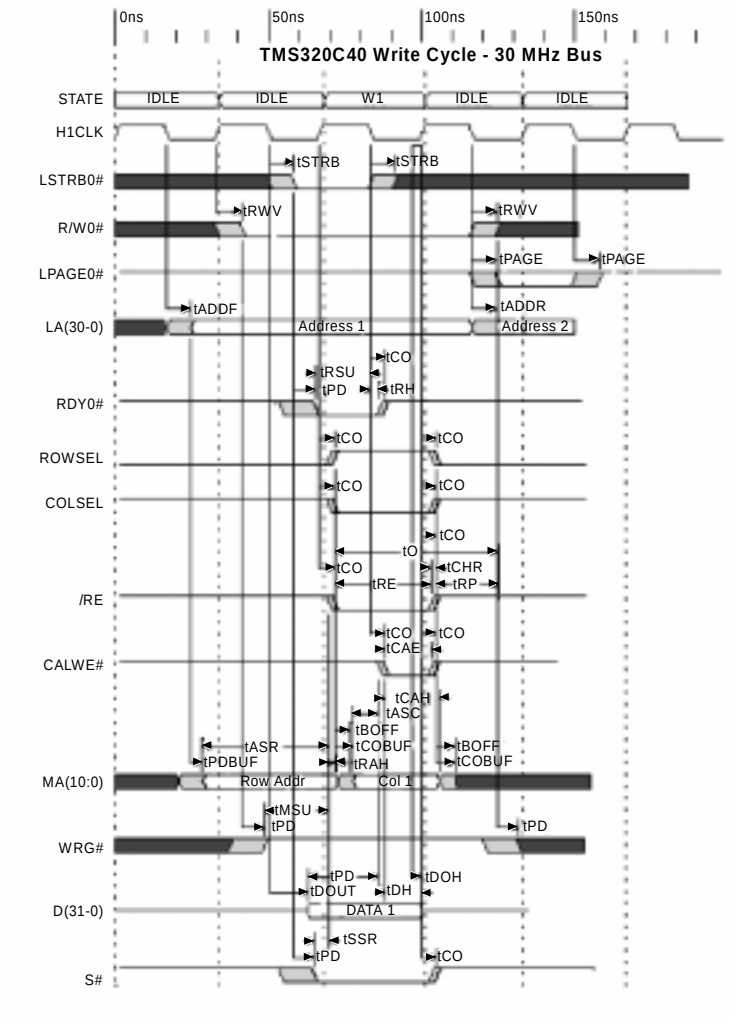
System cost is reduced by utilizing an internal timer in the TMS320C40 to generate the refresh signal. The EDRAM achieves near 12ns SRAM performance while providing much lower memory cost at 4Mb density. The EDRAM provides the best price/performance for DSP applications.

Figure 4. TMS320C40 Read Miss Followed by a Read Hit Cycle – 30 MHz Bus



Note: Based on the 26V12-7 PLD and 12ns EDRAM.

Figure 5. TMS320C40 Write Cycle – 30 MHz Bus



Note: Based on the 26V12-7 PLD and 12ns EDRAM.

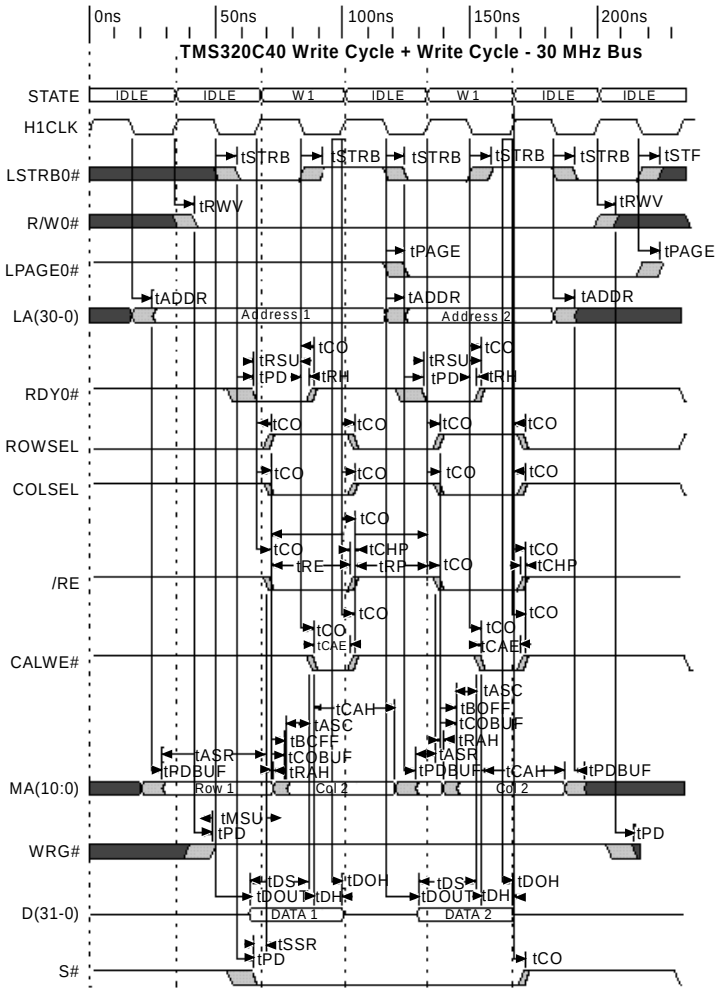
Table 3. Read Miss Followed by a Read Hit Cycle – 30 MHz Bus

Row		Name	Formula	Min	Max	Margin	Comment
1	D	tSTRB	[0,8]	0	8		H1CLK Falling to STRB Valid
2	D	tADDR	[0,8]	0	8		H1CLK Falling to Address Valid
3	D	tRWV	[0,8]	0	8		H1CLK Falling to R/W Valid
4	C	tRE	[30,]	30		<1.33,>	Row Enable Active Time
5	C	tC	[50,]	50		<45.67,>	Row Enable Cycle time
6	D	tAC	12	12	12		Column Address Access Time
7	D	tGQV	5	5	5		Output Enable Access Time
8	C	tRP	[20,]	20		<42.33,>	Row Precharge Time
9	C	tMSU	[5,]	5		<-0.33,><0.33,>	/F and W/R Mode Select Setup Time
10	D	tCO	[3,5]	3	5		Clock to Output Delay
11	D	tBOFF	4	4	4		Buffer Disable Delay
12	D	tCOBUF	5	5	5		Buffer Enable Delay
13	C	tDSU	[9,]	9		<2.33,><0.33,>	Data Setup Time to H1CLK
14	C	tDH	[0,]	0		<11.62,><4,>	Data Hold Time from H1CLK
15	D	tPDBUF	4	4	4		Buffer Propagation Delay
16	C	tASR	[5,]	5		<2.67,>	Row Address Setup Time
17	C	tRAH	[1,]	1		<17.67,>	Row Address Hold time
18	D	tPD	[4,7]	4	7		Propagation Delay
19	D	tPAGE	[0,8]	0	8		H1CLK Falling to PAGE Valid
20	D	tRAC	30	30	30		Row Enable Access Time
21	D	tGQZ	[0,5]	0	5		Output Turn-Off Delay From Output
22	C	tRSU	[18,]	18		<10.33,>	Ready Setup Time to H1CLK Falling
23	C	tRH	[0,]	0		<19.67,>	Ready Hold Time From H1CLK Falling

Table 4. Write Cycle

Row		Name	Formula	Min	Max	Margin	Comment
1	D	tSTRB	[0,8]	0	8		H1CLK Falling to STRB Valid
2	D	tADDR	[0,8]	0	8		H1CLK Falling to Address Valid
3	D	tRWV	[0,8]	0	8		H1CLK Falling to R/W Valid
4	C	tRE	[30,]	30		<1.33,>	Row Enable Active Time
5	C	tC	[50,]	50		<3.33,>	Row Enable Cycle Time
6	C	tRP	[20,]	20		<0,>	Row Precharge Time
7	C	tMSU	[5,]	5		<16.33,>	/F and W/R Mode Select Setup Time
8	D	tCO	[3,5]	3	5		Clock to Output Delay
9	D	tBOFF	[3,5]	3	5		Buffer Disable delay
10	D	tCOBUF	[3.6]	3	6		Buffer Enable Delay
11	D	tPDBUF	4	4	4		Buffer Propagation Delay
12	C	tASR	[5,]	5		<36,>	Row Address Setup Time
13	C	tRAH	[1,]	1		<0,>	Row Address Hold Time
14	D	tPD	[4,7]	4	7		Propagation Delay
15	D	tPAGE	[0,8]	0	8		H1CLK Falling to PAGE Valid
16	D	tDOUT	13	13	13		H1CLK Falling Data Output Valid
17	C	tASC	[5,]	5		<3.67,>	Column Address Setup Time
18	C	tDS	[5,]	5		<18.33,>	Data Input Setup Time
19	C	tDH	[0,]	0		<11.67,>	Data Input Hold Time
20	D	tDOH	0	0	0		Data Output Hold Time
21	C	tCAE	[5,]	5		<9.67,>	Column Address Latch Active Time
22	C	tCAH	[0,]	0		<17.67,>	Column Address Hold Time
23	C	tRSU	[18,]	18		<0.33,>	Ready Setup Time to H1CLK Falling
24	C	tRH	[0,]	0		<3,>	Ready Hold Time from H1CLK Falling
25	G	tCHR	(-[2,])		-2		/CAL Inactive Lead Time to /RE Inactive
26	C	tSSR	[5,]	5		<-0.33,>	Select Setup Time to Row Enable

Figure 6. TMS320C40 Back to Back Write Cycles – 30 MHz Bus

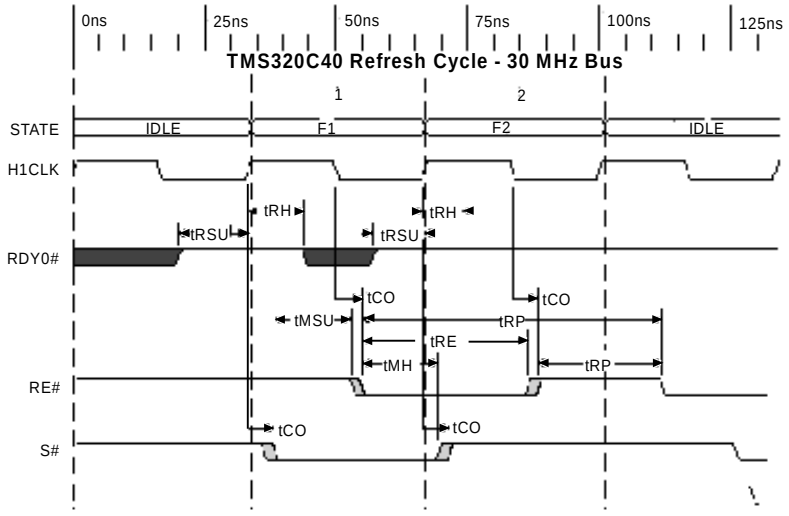


Note: Based on the 26V12-7 PLD and 12ns EDRAM.

Table 5. Back to Back Write Cycles – 30 MHz Bus

Row		Name	Formula	Min	Max	Margin	Comment
1	D	tSTRB	[0,8]	0	8		H1CLK Falling to STRB Valid
2	D	tADDR	[0,8]	0	8		H1CLK Falling to Address Valid
3	D	tRWV	[0,8]	0	8		H1CLK Falling to R/W Valid
4	C	tRE	[30,]	30		<1.33,>	Row Enable Active Time
5	C	tC	[50,]	50		<11.29,>	Row Enable Cycle Time
6	C	tRP	[20,]	20		<7.95,>	Row Precharge time
7	C	tMSU	[5,]	5		<16.33,>	/F and W/R Mode Select Setup Time
8	D	tCO	[3,5]	3	5		Clock to Output Delay
9	D	tBOFF	[3,5]	3	5		Buffer Disable delay
10	D	tCOBUF	[3.6]	3	6		Buffer Enable Delay
11	D	tPDBUF	4	4	4		Buffer Propagation Delay
12	C	tASR	[5,]	5		<36,><2.67,>	Row Address Setup Time
13	C	tRAH	[1,]	1		<0,><0,>	Row Address Hold Time
14	D	tPD	[5,7]	5	7		Propagation Delay
15	D	tPAGE	[0,8]	0	8		H1CLK Falling to PAGE Valid
16	D	tDOUT	13	13	13		Data Output Valid Delay
17	C	tASC	[5,]	5		<3.67,><3.67,>	Column Address Setup Time
18	C	tDS	[5,]	5		<18.33,><18.33,>	Data Setup Time
19	C	tDH	[0,]	0		<11.67,><11.67,>	Data Hold Time
20	D	tDOH	0	0	0		Data Output Hold Time
21	C	tCAE	[5,]	5		<9.67,><9.67,>	Column Address Latch Active Time
22	C	tCAH	[0,]	0		<32.33,><32.33,>	Column Address Hold Time
23	C	tRSU	[18,]	18		<0.33,><0.33,>	Ready Setup Time to H1CLK Falling
24	C	tRH	[0,]	0		<3,><3,>	Ready Hold Time from H1CLk Falling
25	G	tCHR	(-[2,])		-2		/CAL Inactive Lead Time to /RE Inactive
26	C	tSSR	[5,]	5		<-0.33,>	Select Setup Time to Row Enable

Figure 7. TMS320C40 Refresh Cycle – 30 MHz Bus



Notes: 1) All other signals are “don’t care.”
2) Based on the 26V12-7 and FCT541 buffers.

Table 6. Refresh Cycle

Row		Name	Formula	Min	Max	Margin	Comment
1	C	tRSU	[6]	6		<7.33,><3.67,>	Setup, RDY# Before H1CLK Rising
2	C	tRH	[0,]	0		<10.67,><7.33,>	Hold, RDY# After H1CLK Rising
3	C	tRE	[30,]	30		<1.33,>	Row Enable Active Time
4	C	tC	[55,]	55		<2,>	Row Enable Cycle Time
5	C	tMSU	[5,]	5		<9.67,>	/F and W/R Mode Select Setup Time
6	C	tMH	[5,]	5		<9.67,>	/F and W/R Mode Select Hold Time
7	C	tRP	[20,]	20		<3.67,>	Row Precharge time
8	D	tCO	[3,5]	3	5		Clock to Output Delay

```
" TMS320C40 PLD Equations - 60 MHz Core, 30 MHz Bus
" 26V12-7
" Revision 2.0
" Enhanced Memory Systems

INPUT H1CLK;
LOW_TRUE INPUT RESET;
LOW_TRUE INPUT RW0;
LOW_TRUE INPUT STRB0;
LOW_TRUE INPUT PAGE0;
INPUT TCLK0;

LOW_TRUE OUTPUT RDY0;
LOW_TRUE OUTPUT RE ;
LOW_TRUE OUTPUT CALWE;
LOW_TRUE OUTPUT WRG ;
LOW_TRUE OUTPUT F ;
LOW_TRUE OUTPUT S ;
LOW_TRUE OUTPUT ROWSEL ;
LOW_TRUE OUTPUT COLSEL ;

NODE q2..q0 CLOCKED_BY H1CLK;
D_FLOP NODE REF_PENDING CLOCKED_BY H1CLK;

MACRO ON 1;
MACRO OFF 0;

" Equations
WRG = /RW0;

" We need to store the current cycle state to generate EDRAM timings
STATE_MACHINE TMS320C40
    STATE_BITS [q2..q0]
    CLOCKED_BY H1CLK;

STATE IDLE :
    REF_PENDING = OFF;
    RDY0 = OFF;
    RE = OFF;
    CALWE = OFF;
    F = OFF;
    S = STRB0;
    ROWSEL = ON;
    COLSEL = OFF;

IF (RESET) THEN
    GOTO READ1;
    ELSE
    IF (TCLK0) THEN
        GOTO REFRESH1;
    ELSE
```

```
       IF (STRB0) THEN
          IF (RW0) THEN
             GOTO WRITE1;
          ELSE
             GOTO READ1;
          END IF;
       END IF;
    END IF;
END IF;

STATE REFRESH1 :
    REF_PENDING = OFF;
    RDY0 = OFF;
    RE = /H1CLK;
    CALWE = OFF;
    F = ON;
    S = OFF;
    ROWSEL = OFF;
    COLSEL = OFF;

    GOTO REFRESH2;

STATE REFRESH2 :

    REF_PENDING = OFF;
    RDY0 = OFF;
    RE = H1CLK;
    CALWE = OFF;
    F = OFF;
    S = OFF;
    ROWSEL = OFF;
    COLSEL = OFF;

    GOTO IDLE;

STATE READ1 :

    REF_PENDING = (TCLK0 + REF_PENDING);
    RDY0 = /H1CLK;
    RE = ON;
    CALWE = OFF;
    F = OFF;
    S = ON;
    ROWSEL = /H1CLK;
    COLSEL = H1CLK;

    GOTO READ2;

STATE READ2 :

    RDY0 = ON;
    RE = OFF;
```

```
CALWE = OFF;
F = OFF;
S = ON;
ROWSEL = OFF;
COLSEL = ON;

IF (RESET) THEN
   GOTO IDLE;
ELSE
   IF (TCLK0 + REF_PENDING ) THEN GOTO REFRESH1;
ELSE
   IF (/RW0 * /PAGE0) THEN
   GOTO READ1;
ELSE
   IF (/RW0 * PAGE0) THEN
      GOTO READ2;
   ELSE
      IF (RW0) THEN
         GOTO WRITE1;
      ELSE
         GOTO IDLE;
END IF;
END IF;
END IF;
END IF;
END IF;

STATE WRITE1:

   REF_PENDING = (TCLK0 + REF_PENDING);
   RDY0 = /H1CLK;
   RE = ON;
   CALWE = /H1CLK;
   F = OFF;
   S = ON;
   ROWSEL = ON;
   COLSEL = OFF;

   GOTO IDLE;

END TMS320C40;
```