TEXAS INSTRUMENTS

EDRAM™ Memory Controller for the TMS320C31 DSP

Application Report

1998 Digital Signal Processor Solutions

Texas Instruments

Printed in U.S.A., January 1998

SPRA172

EDRAM™ Memory Controller for the TMS320C31 DSP

Enhanced Memory Systems
1850 Ramtron Drive, Colorado Springs, CO 80921
web: http://www.edram.com

SPRA172
January 1998

Printed on Recycled Paper

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.

Contents

List of Figures

List of Tables

EDRAM™ Memory Controller for the TMS320C31 DSP

ABSTRACT

This report provides an overview of a controller chip set to improve processor memory management and transactions while using the Texas Instruments (TI™) TMS320C31 digital signal processor (DSP). This application uses a minimum number of wait states which increases processor throughput.

1 EDRAM Controller Design

This Enhanced DRAM controller design supports TMS320C31 DSP memory transactions with a minimum of wait states. This EDRAM controller is designed to support one EDRAM bank of 2M bytes; however, memory size can expanded by adding additional control outputs to the logic. Figure 1 shows a functional block diagram of the TMS320C31 DSP System. The TMS320C31 DSP primary-bus interface can be programmed to control wait states and hold operations. The memory-control register for the local bus is programmed with the following parameters:

- SWW is programmed to respond only to the external $\overline{\text{RDY}}$ input for wait states (SWW=00). The internal $\overline{\text{RDY}}_{\text{wtcnt}}$ is ignored.
- WTCNT does not need to be programmed since software programmable wait states are not used.
- BNKCMP is set as shown below:

EDRAM	MSBs Defining a Bank	Bank Size	BNKCMP
512K x 8	23–8	256	10000

The TMS320C31 supports the following memory transactions:

- 32-bit page-read hit
- 32-bit page-read miss
- 32-bit write (hit or miss)

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

- A18–A0 – address bus
- $\overline{\text{STRB}}$ – external-access strobe
- R/$\overline{\text{W}}$ – read-write-mode outputs
- $\overline{\text{RDY}}$ – ready input
- $\overline{\text{RESET}}$ – reset input
- H1CLK – processor-clock output
- TCLK0 – timer zero output clock/pulse

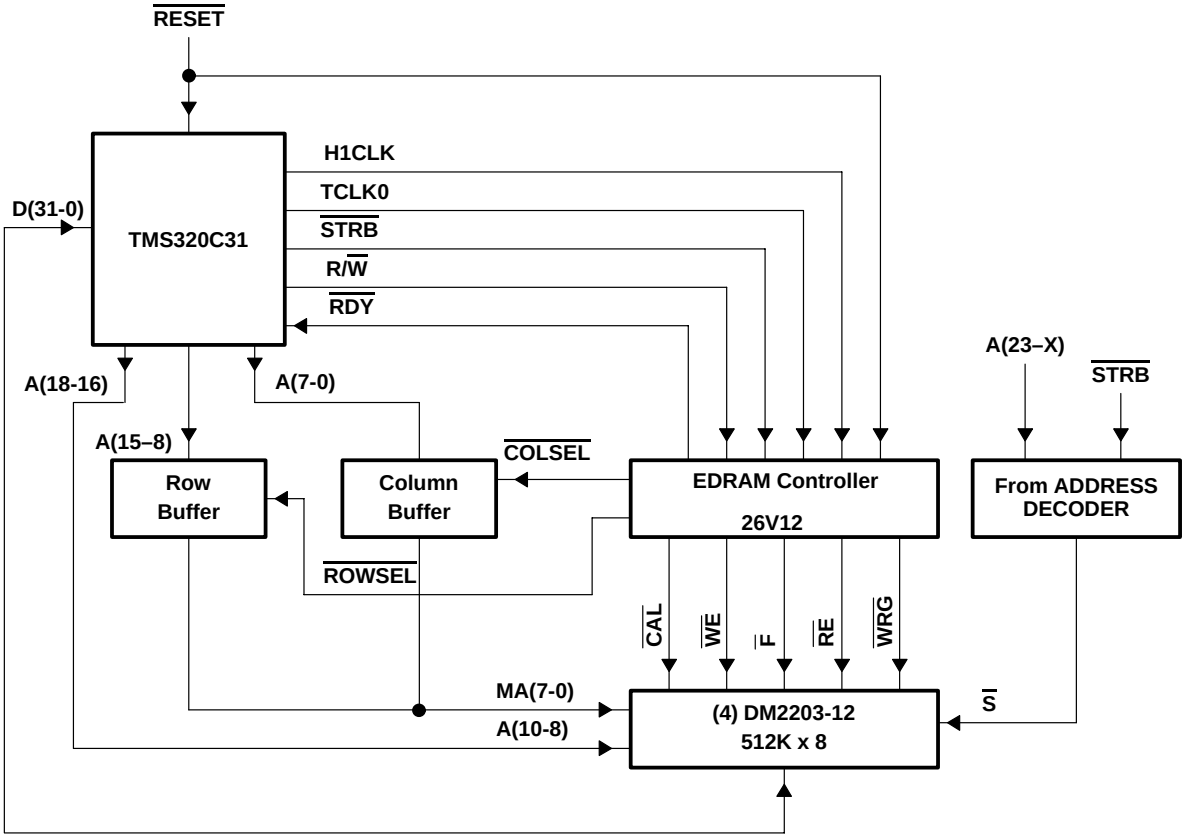

Figure 1. TMS320C31 System Block Diagram

The controller generates the following signals to control the EDRAM:

- $\overline{\text{ROWSEL}}$ – multiplexed row-address enable
- $\overline{\text{COLSEL}}$ – multiplexed column-address enable
- $\overline{\text{RE}}$ – row enable
- $\overline{\text{CAL}}$ – column-address strobe
- $\overline{\text{WE}}$ – write enable
- $\overline{\text{WRG}}$ – combined W/$\overline{\text{R}}$ and $\overline{\text{G}}$
- $\overline{\text{F}}$ – refresh
- $\overline{\text{S}}$ – chip select

The EDRAM W/$\overline{\text{R}}$ and $\overline{\text{G}}$ control lines are tied together to form $\overline{\text{WRG}}$**.** The $\overline{\text{ROWSEL}}$, $\overline{\text{COLSEL}}$, $\overline{\text{CAL}}$, $\overline{\text{WE}}$, $\overline{\text{WRG}}$, and $\overline{\text{F}}$ logic signals are supplied from the 26V12 PLD.

$\overline{\text{S}}$ is an enable pin for the EDRAM and should be active for the desired page(s) of DSP memory. Because of the limited number of outputs of the PAL26V12, $\overline{\text{S}}$ is not included within the PAL. A simple decode might be to connect $\overline{\text{STRB}}$ to $\overline{\text{S}}$ where all external memory accesses enable the EDRAM, or in combination with an upper address line. Note that disabling the EDRAM will also lower the EDRAM power consumption. Optionally, the $\overline{\text{S}}$ input signal can be tied to ground if EDRAM power consumption is not an issue.

2 EDRAM Controller Functional Description

This section describes the EDRAM controller block diagram shown in Figure 2. The refresh counter is formed by using an internal timer in the TMS320C31 DSP to generate a refresh signal every 64 μs for the state machine. This refresh signal triggers an $\overline{F}$ refresh on the next available bus cycle. Up to three wait states can be inserted for the processor while a bus transaction is occurring.

Figure 3, the address multiplexer, selects a row and a column address for the EDRAM multiplex address lines under control of the state machine. The multiplexer is implemented using the two 74FCT541CT 8-bit buffer chips. These chips were selected for their short throughput delay time (4.1 ns) and short select-delay time (5.8 ns), respectively. A column address is connected to pins A7–A0 for 512K x 8 EDRAM. The row address comprises the 11 local address bits above the column address. Address lines A10–A8 of the EDRAM are connected directly to A18–A16 of the TMS320C31 DSP. During column select times, the upper address lines are ignored so there is no need to feed these address lines through a buffer.

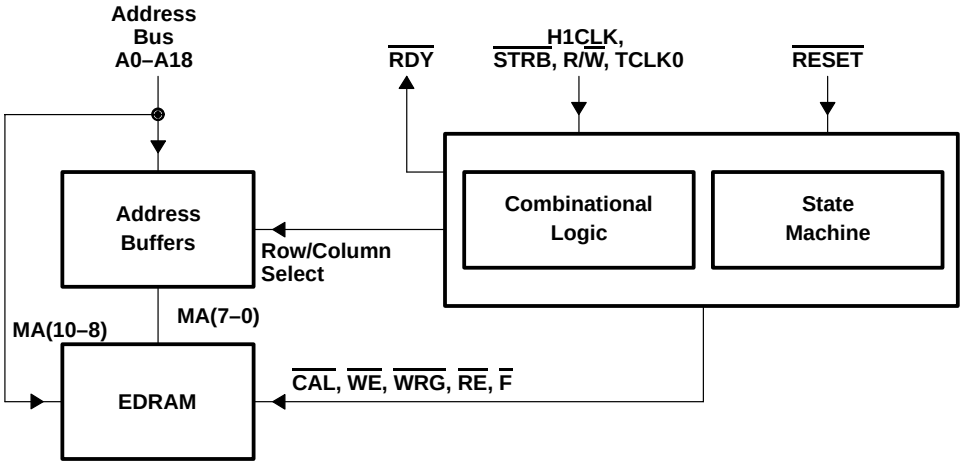

Figure 2. EDRAM Controller Block Diagram

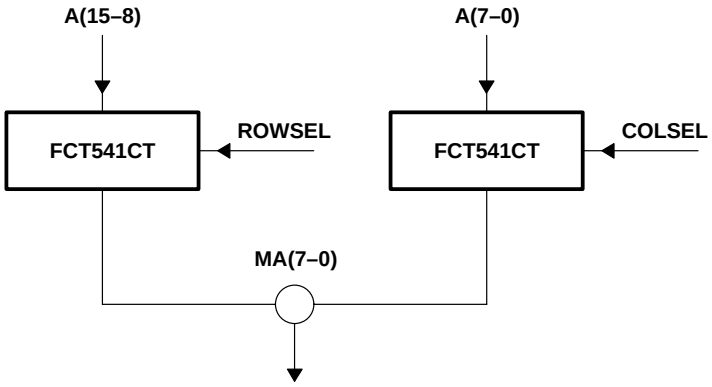

Figure 3. Address Multiplexer Using Two 74FCT541CT Buffers

The state machine operates synchronously with the rising edge of H1CLK.

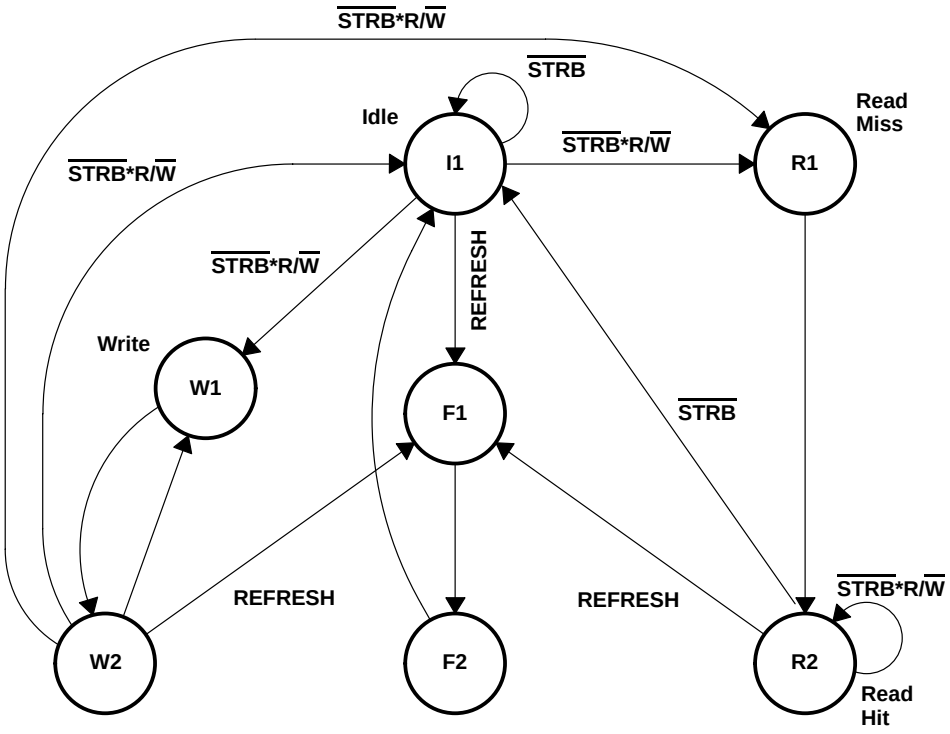

Figure 4. EDRAM Controller State Machine

The following memory-control sequences are selected, based on control-input-line status. Read and write sequences are described. All logic states are determined at the rising edge of H1CLK.

- Reset Sequence – When $\overline{\text{RESET}}$ is at logic 0, the processor is in the reset state. The EDRAM controller continuously executes $\overline{\text{F}}$ refresh cycles until $\overline{\text{RESET}}$ returns to logic 1 level. This meets the EDRAM initialization requirements of eight $\overline{\text{F}}$ refresh cycles during start up. It is assumed that a startup control program is run immediately following a reset. This program must configure the TMS320C31 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 – Starting from the idle state, when $\overline{\text{STRB}}$ is at logic 0 and R/$\overline{\text{W}}$ is at logic 1, a page-read-miss sequence is executed. The row address is selected, and $\overline{\text{WRG}}$ signal is set to logic 0. The $\overline{\text{RE}}$ signal is then clocked to load a new page of memory into the internal DRAM cache column, which takes 30-ns. $\overline{\text{RDY}}$ is brought to logic 1 at the start 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 TMS320C31 addressing system, one wait state is inserted when a read cycle follows a write cycle.
- Page-read-hit sequence – Starting from the R1 state, when $\overline{\text{STRB}}$ is at logic 0 and R/$\overline{\text{W}}$ is logic 1, a page-read-hit sequence is executed. The column address is selected, and $\overline{\text{WRG}}$ is brought to logic 0 to gate data onto the data bus. The EDRAM can support back-to-back page reads from cache in zero wait states. The TMS320C31 processor holds $\overline{\text{STRB}}$ at logic 0 during back-to-back page hit reads since the programmable bank switching feature is being used. On a page-read miss, $\overline{\text{STRB}}$ goes to logic 1, which also causes an idle state to be inserted.
- Write sequence – If $\overline{\text{STRB}}$ is at logic 0 and R/$\overline{\text{W}}$ is also at logic 0, a write sequence is executed. The row address is selected and write-data operation is stable and the $\overline{\text{WE}}$ signal is clocked to post the write data operation into an internal latch. When $\overline{\text{CAL}}$ falls, write data is written to the EDRAM array. Following the write-data operation, $\overline{\text{RE}}$ is brought to logic 1 to terminate the write-data cycle. The write operation is completed in two cycles or zero wait states. The EDRAM supports back-to-back writes in two cycles or zero wait states.

- Refresh sequence – If TCKL0 input is at logic 1, an $\overline{F}$ refresh sequence is performed after any current memory cycles are completed. The TCKL0 output is at logic 1 for 40 ns and it then goes to logic 0 and remains at logic 0 until the next refresh period. The $\overline{F}$ pin is brought to logic 0 and $\overline{RE}$ is clocked to perform an internal refresh using the internal-refresh counter. A refresh sequence is executed every 62-µs. During a refresh sequence, $\overline{RDY}$ is held at logic 1 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 $\overline{STRB}$ is at logic 1 (except at the start of the W2 state), this indicates the cycle is an idle cycle. The $\overline{WRG}$ is brought to logic 1 to disable the EDRAM outputs from the data bus until the next active cycle.
- Chip select ($\overline{S}$) – The $\overline{S}$ is generated using a NAND**,** NOR or equivalent gate to decode the Q2–Q0 state machine outputs. $\overline{S}$ is brought to logic 1 during the idle and refresh states to reduce power consumption. This circuit is optional and $\overline{S}$ can be tied to ground if reduced power consumption is not desired.
- Programming the TMS320C31 DSP 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 40-ns minimum high pulse every 62-µs. The bit values to be programmed are as follows:

CLKSRC = 1
FUNC = 1
C/P = 0
I/O = 1
HLD = 1
INV = 0

$$\text{period register} = \frac{f(\text{Timer clock})}{f\text{INT}} = \frac{12.5\text{MHz}}{16\text{KHz}} = 780$$

Period Register Value = 780d

The remaining bit fields can remain 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 62-µs. 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 40-ns logic 1 pulse is output on TCLK0 pin. This logic 1 pulse causes the REF_PENDING register output-logic level to remain at logic 1 starting with the next rising edge of H1CLK**.** This level is cleared only after a refresh sequence is performed. The TMS320C31 DSP timer does not need to be restarted and no interrupt is generated. A refresh sequence is not started until after the second cycle of a read-miss or write sequence, if one is in progress.

Timing of the EDRAM controller is shown in Figure 5, Figure 6, and Figure 7. The EDRAM controller operates with a 50-MHz TMS320C31 DSP system using the 12-ns version of the EDRAM as shown. The EDRAM–controller parameters are for a 26V12-7 programmable logic device (PLD) and Pericom 74FCT541CT 8-bit buffers. The TMS320C31DSP timing parameters are from the TMS320C31 DSP Data Manual (literature number SPRU031D).

3 Summary

An EDRAM controller for the 50 MHz and 60 MHz versions of the TI TMS320C31 DSP can be implemented using a simple PLD and two Pericom 74FCT541CT data buffers. The controller supports one EDRAM bank of 2M bytes (using DM2203 components) and can be expanded easily to include more memory. The EDRAM system as shown in this application report achieves zero wait states on read-hit cycles and on all write cycles.

System cost is reduced significantly by using an internal timer in the TMS320C31 to generate the refresh signals. The EDRAM achieves near 12 ns static RAM (SRAM) performance, while providing much lower memory cost at 4M-bit density. The EDRAM/TMS320C31 DSP provides an excellent ratio of cost/performance for DSP applications.

Example 1. PLD Equations

```
* TMS320C31 PLD Equations – 25 MHz Bus
* 26V12–7 PLD
* Revision 2.0
* Vantis MACH–XL 5.0 Compiler

INPUT H1CLK;
INPUT TCLK0;
LOW_TRUE INPUT RESET;
LOW_TRUE INPUT RW;
LOW_TRUE INPUT STRB;

LOW_TRUE OUTPUT RDY;
LOW_TRUE OUTPUT RE;
LOW_TRUE OUTPUT CAL;
LOW_TRUE OUTPUT WE;
LOW_TRUE OUTPUT WRG;
LOW_TRUE OUTPUT F;
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

* We need to store the current cycle state to generate EDRAM
* timings
STATE_MACHINE TMS320C31
       STATE_BITS [q2..q0]
       CLOCKED_BY H1CLK;
```

```
STATE IDLE:
       REF_PENDING = OFF;
       RDY = STRB * RW;
       RE = OFF;
       CAL = OFF;
       WE = OFF;
       WRG = OFF;
       F = OFF;
       ROWSEL = ON;
       COLSEL = OFF;

       IF (RESET) THEN
          GOTO REFRESH1;
          ELSE
           IF (TCLK0) THEN
              GOTO REFRESH1;
          ELSE
               IF (STRB) THEN
                  IF (RW) THEN
                     GOTO WRITE1;
                     ELSE
                     GOTO READ1;
                  END IF;
               END IF;
          END IF;
       END IF;


STATE REFRESH1 :
       REF_PENDING = OFF;
       RDY = OFF;
       RE = /H1CLK;
       CAL = OFF;
       WE = OFF;
       WRG = OFF;
       F = ON;
       ROWSEL = OFF;
       COLSEL = OFF;
       GOTO REFRESH2;


STATE REFRESH2 :
       REF_PENDING = OFF;
       RDY = OFF;
       RE = H1CLK;
       CAL = OFF;
       WE = OFF;
       WRG = OFF;
       F = OFF;
       ROWSEL = OFF;
       COLSEL = OFF;
       GOTO IDLE;
```

```
STATE READ1 :
       RDY = /H1CLK;
       RE = /H1CLK;
       CAL = OFF;
       WE = OFF;
       WRG = ON;
       F = OFF;
       ROWSEL = H1CLK;
       COLSEL = /H1CLK;
       REF_PENDING = TCLK0;
       GOTO READ2;

STATE READ2 :
       RDY = /H1CLK;
       RE = /H1CLK;
       CAL = OFF;
       WE = OFF;
       WRG = ON;
       F = OFF;
       ROWSEL = OFF;
       COLSEL = ON;
       REF_PENDING = TCLK0

       IF (RESET) THEN
            GOTO IDLE
           ELSE IF (TCLK0 + REF_PENDING) THEN
            GOTO REFRESH1;
           ELSE IF (STRB * /RW) THEN
            GOTO READ2;
           ELSE
            GOTO IDLE;
           END IF;

STATE WRITE1 :
       RDY = OFF;
       RE = /H1CLK;
       CAL = OFF;
       WE = /H1CLK;
       WRG = OFF;
       F = OFF;
       ROWSEL = H1CLK;
       COLSEL = /H1CLK;
       REF_PENDING = TCLK0;
       GOTO WRITE2;

STATE WRITE2 :
       RDY = STRB * RW;
       RE = H1CLK;
       CAL = H1CLK;
       WE = H1CLK;
       WRG = OFF;
       F = OFF;
       ROWSEL = OFF;
       COLSEL = ON;
```

```
       REF_PENDING = TCLK0;

IF (RESET) THEN
          GOTO IDLE;
       ELSE IF (TCLK0 + REF_PENDING)
          THEN GOTO REFRESH1;

       ELSE IF (/STRB * RW)
          THEN GOTO WRITE1;

       ELSE IF (STRB * /RW)
          THEN GOTO READ1;

       ELSE IF (/STRB * /RW)
          THEN GOTO IDLE;

       ELSE GOTO IDLE;

END IF;

END TMS320C31;
```

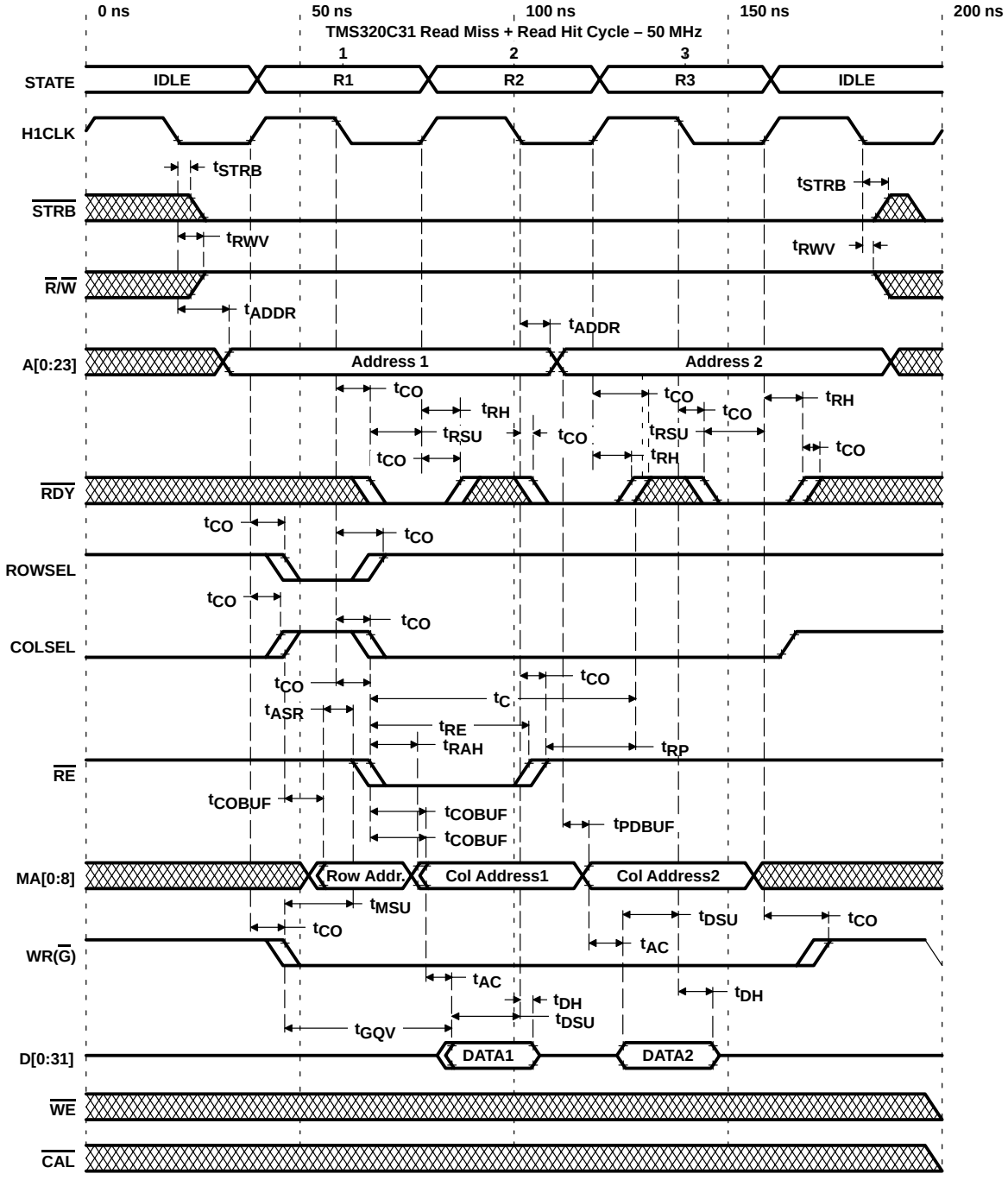

Figure 5. TMS320C31 Read-Miss and Read-Hit Cycles @ 50 MHz

Table 1. TMS320C31 Read-Miss and Read-Hit Cycles @ 50 MHz

NO.		NAME	FORMULA	MIN	MAX	MARGIN	COMMENTS
1	V	t_{ASR}	5	5			Setup time, row address
3	C	t_{RSU}	6	5		<8.><8.><8.>	Setup time, $\overline{RDY}$ before H1 high
2	V	t_{RAH}	1,5	1,5			Hold time, row address
4	C	t_{RH}	0	6		<4.><4.><4.>	Hold time, $\overline{RDY}$ after H1 high
5	C	t_{ASR}	5	5		<7.>	Setup time, row address
6	D	t_{STRB}	4	4			Valid time, H1CLK to $\overline{STRB}$
7	D	t_{ADDR}	9	9			Valid time, H1CLK to address
8	D	t_{RWV}	4	4			Valid time, H1CLK to R/W
9	C	t_{RE}	30	30		<8.>	Active time, row enable
10	C	t_{RAH}	1	1		<3.>	Hold time, row enable
11	C	t_{C}	55	55		<7.>	Cycle time, row enable
12	D	t_{AC}	12	12	12		Access time, column address
13	D	t_{GQV}	5		5		Access time, output enable
14	C	t_{RP}	20	20		<2.>	Precharge time, row
15	C	t_{MSU}	5	5		<13.>	Setup time, $\overline{F}$ and W/R mode select
16	D	t_{COBUF}	6	6	6		Delay time, clock to output
17	C	t_{DSU}	10	10		<6.><5.>	Setup time, ’C31 data
18	C	t_{DH}	0	0		<3.>	Hold time, ’C31 data
19	D	t_{PDBUF}	4	4	4	<3.><3.>	Delay time, propagation
20	D	t_{CO}	4,6	4	6		Delay time, clock to output

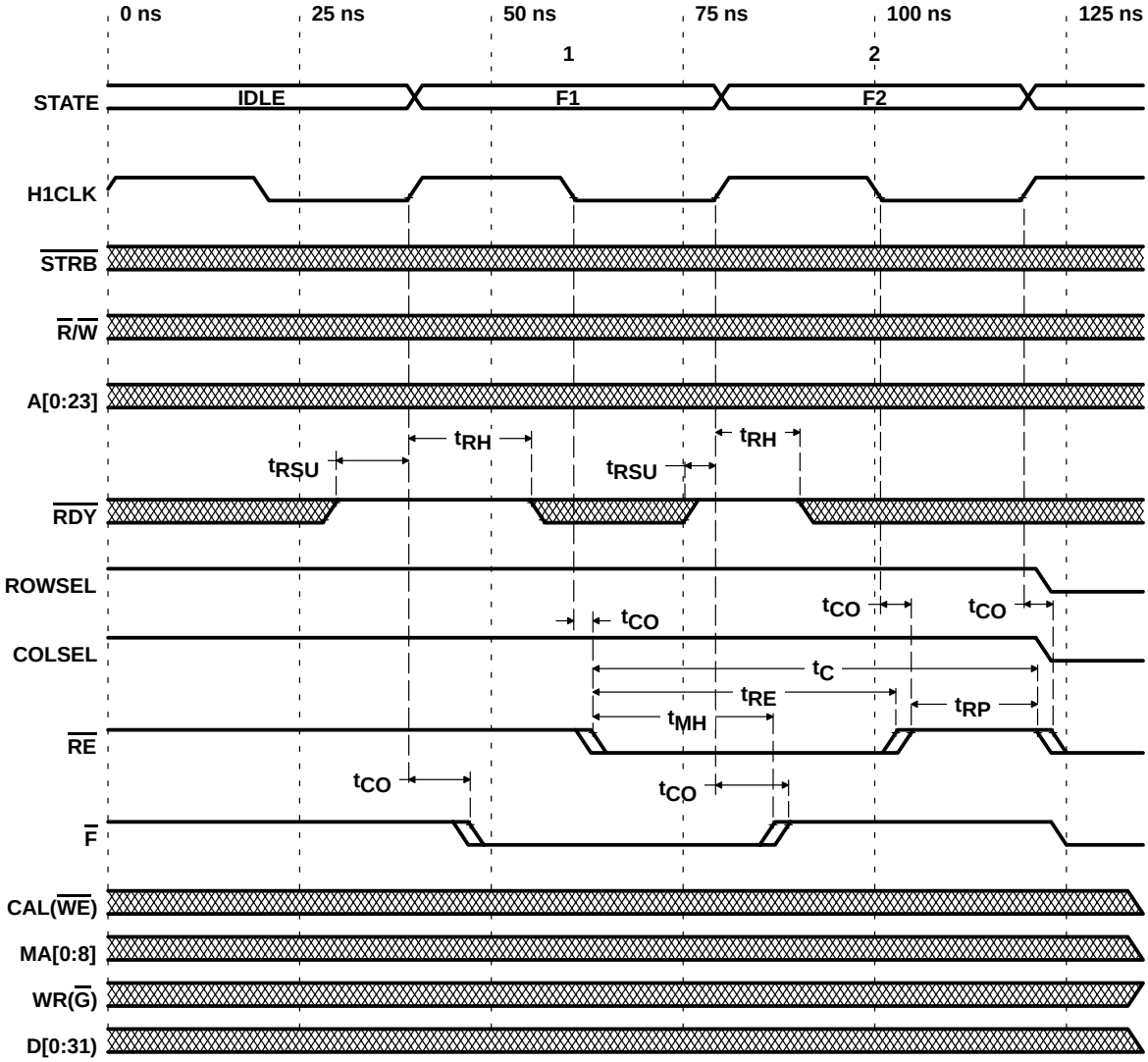

Figure 6. TMS320C31 Refresh After Idle Cycle @ 50 MHz

Table 2. TMS320C31 Refresh After Idle Cycle @ 50 MHz

NO.		NAME	FORMULA	MIN	MAX	MARGIN	COMMENTS
1	V	t_{ASR}	5	5			Setup time, row address
2	V	t_{RAH}	1	1			Hold time, row address
3	C	t_{RSU}	6	6		<3.><2.>	Setup time, $\overline{RDY}$ before H1 high
4	C	t_{RH}	0	0		<14><8.>	Hold time, $\overline{RDY}$ after H1 high
5	C	t_{RE}	30	30		<8.>	Active time, row enable
6	C	t_C	55	55		<3.>	Cycle time, row enable
7	C	t_{MSU}	5	5		<13.>	Setup time, $\overline{F}$ and W/R mode select
8	C	t_{MH}	5	5		<13.>	Hold time, $\overline{F}$ and W/R mode select
9	C	t_{RP}	20	20		<–2.>	Precharge time, row
10	D	t_{CO}	4,6	4	6		Delay time, clock to output

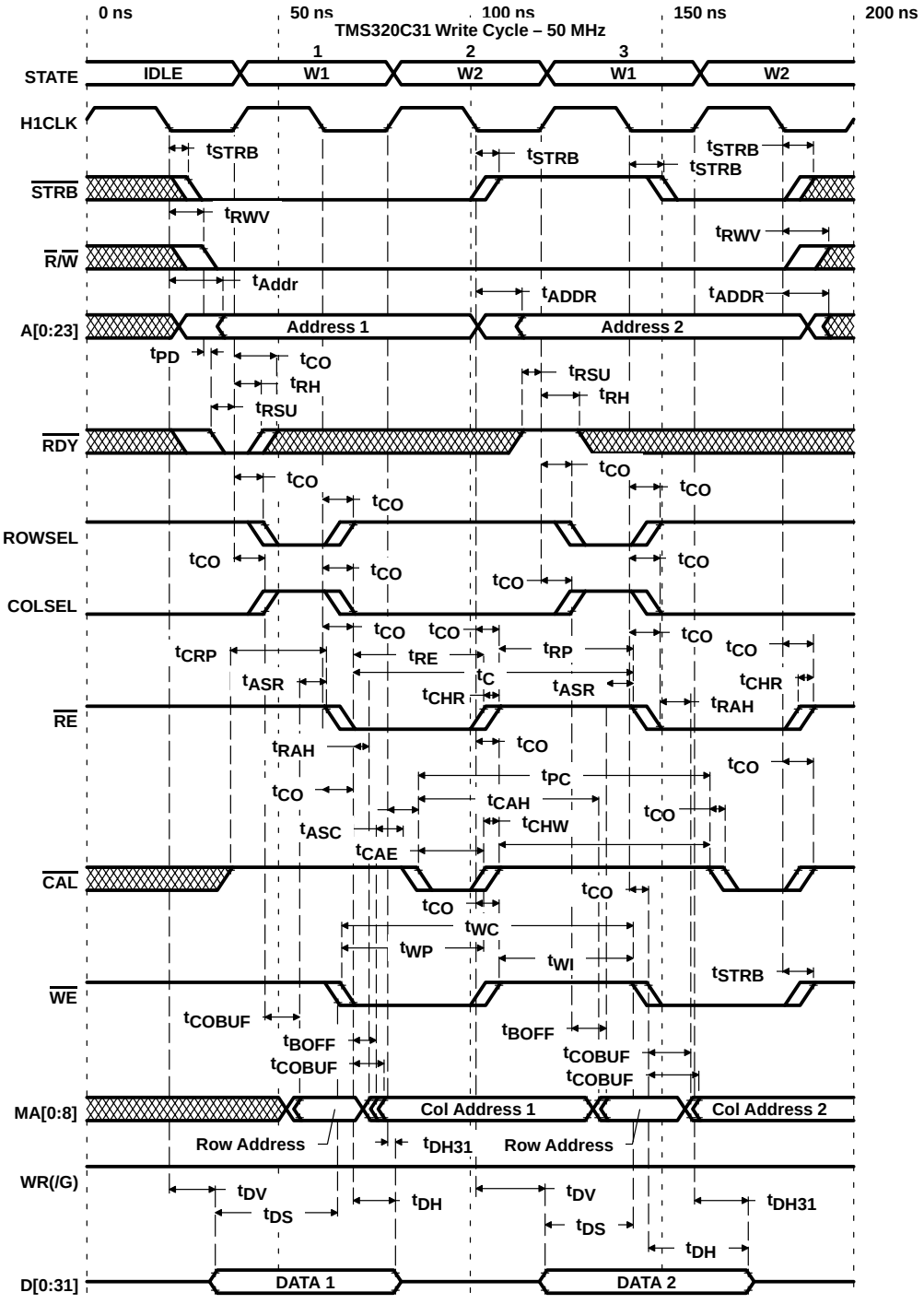

Figure 7. TMS320C31 Write Cycle – 50 MHz

Table 3. TMS320C31 Write Cycle – 50 MHz

		NAME	FORMULA	MIN	MAX	MARGIN	COMMENTS
1	C	t_{RSU}	6	6		<0.><1.>	Setup time, $\overline{RDY}$ before H1 high
2	C	t_{RAH}	0	0		<4.><8.>	Hold time, $\overline{RDY}$ after H1 high
3	C	t_{ASR}	5	5		<7.><7.>	Setup time, row address
4	D	t_{STRB}	2,4	2	4		Valid time, H1CLK to $\overline{STRB}$
5	C	tR_E	30	30		<8.>	Hold time, row address
6	C	t_{RAH}	1	1		<1.><3.>	Cycle time, row enable
7	C	t_C	55	55		<23.>	Cycle time, row enable
8	C	t_{ASC}	5	5		(7.>	Setup time, column address
9	C	t_{DH31}	0	0		<12.><2.>	Hold time, 'C31 data
10	C	t_{CRP}	5	5		<22.>	Setup time, column address latch to row enable
11	D	t_{RWV}	2,7	2	7		Valid time, H1CLK to R/W
12	D	t_{ADDR}	2,9	2	9		Valid time, H1CLK to address
13	D	t_{BOFF}	4	4	4		Buffer output invalid
14	D	t_{COBUF}	6	6	6		Delay time, clock to output
15	D	t_{DV}	12	14	14		Valid time, 'C31 data
16	C	t_{BP}	20	20		<18.>	Precharge time, row
17	C	t_{CHB}	–2	–2		<0.><0.>	Setup time, $\overline{CAL}$ high to $\overline{RE}$ high
18	D	t_{CO}	4,6	4			Delay time, clock to output
19	D	t_{PD}	3,7	3	7		Delay time, propagation
20	C	t_{DS}	5	5	6	<25.><25.>	Setup time, 'C31 data
21	C	t_{DH}	0	0		<16.><26.>	Hold time, 'C31 data
22	C	t_{CAH}	0	0		<44.>	Hold time, column address
23	C	t_{CAE}	5	5		<13.>	Active time, column address latch
24	C	t_{PC}	12	12		<66.>	Cyce time, column address latch
25	C	t_{WP}	5	5		<33.>	Active time, write enable
26	C	t_{WC}	12	12		<66.>	Cycle time, write enable
27	C	t_{CH}	5	5		<53.>	Column address latch high time (latch transparent)
28	C	t_{WI}	5	5		<33.>	Inactive time, write enable
29	C	t_{CHW}	0	0		<–2.>	Column address latch high to write enable low

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.