MOTOROLA
Semiconductor Products Sector Application Note

Order Number: AN1834/D
Rev. 0, 12/1999

Using DSP56300 Interactive Timing Diagrams

By Iantha E. Scheiwe

As systems become faster and more complex, the timing interface between Motorola’s DSP56300 family and other system components becomes increasingly important in system design. To help designers examine DSP interface timings, Motorola provides electronic timing diagrams based on the information from the DSP56300 data sheets.

This application note briefly describes the tools and materials involved in interface timing analysis, including how to acquire them. It then explains library and diagram organization and the basic concepts of constraints and delays. Finally, it demonstrates how to use the Motorola DSP56300 timing diagrams to analyze specific interfaces: DSP-to-SRAM and DSP-to-DRAM.

Contents

MOTOROLA

1 Introduction

To understand why timing libraries are helpful, consider a system design that requires external DRAM connected to a DSP. An engineer wants to find the most efficient and cost-effective way to connect these two devices, and there are more than 60 DRAM timings in this interface. It may be easy to find a speed category (say 60-ns DRAM), but it can be difficult to guarantee that there are no timing violations. Instead of going through each timing number in the data sheet, the engineer can use timing libraries to model the interface visually and accurately, ensuring that there are no timing violations. Entering new device information instantly shows how the new timings affect the interface, allowing the engineer to switch among device timing libraries to find the most cost-effective solution. Motorola provides the DSP56300 timing diagrams in the form of TimingDesigner® files. This application note draws examples from the external memory expansion port (Port A) diagrams, but the set of files from Motorola includes a number of other diagrams, as well (see *Libraries and Diagrams* on page 3).

1.1 Tools and Materials for Timing Analysis

To best understand the information in this application note, you should acquire the appropriate Chronology® Corporation tool and the example timing analysis files from Motorola, as well as the appropriate Motorola data sheet. Then work through the examples in this application note as you read.

1.1.1 Chronology Corporation Tools

Chronology Corporation provides two tools that make it possible to examine timing interfaces visually: TimingDesigner® and TimingViewer™. With either tool, you create timing diagrams and enter data sheet timings in library files. The tool relates the timing diagrams to the libraries. If you change library information, the tool updates the diagrams accordingly.

You can work through the examples in this application note with either TimingDesigner or TimingViewer. Note, however, that TimingDesigner allows you to save your analysis, while TimingViewer does not. To acquire TimingDesigner or download a free copy of TimingViewer, see the Chronology web site:

`http://www.chronology.com`

Note: In this application note, references to red and green indicate an item’s color in the TimingDesigner interface. In monochrome copies of this document, red and green appear as shades of gray.

1.1.2 Motorola Timing Analysis Files and Data Sheet

First, download the Motorola DSP56300 timing analysis files and the DSP56303 data sheet. For convenience, you may want to create a directory (for example, `motorola`) to contain the downloaded materials. Download the files and data sheet from the Motorola DSP web site:

`http://www.motorola.com/SPS/DSP/documentation/DSP56300.html`

After downloading, follow these steps to make the timing analysis files accessible for use in diagrams:

1. Extract the files from the archive.
2. Start TimingDesigner.
3. From the File menu, select **Access Synchrony Data...**

4. Browse to the `dsp56303cdr` directory.
5. Select either **tdproj.res** (on a UNIX® system) or **tdproj.ini** (on a PC).
6. Click **OK**.
7. When TimingDesigner asks whether you want these files to be available in the future, click **yes**.

2 Timing Analysis Files Overview

The Motorola timing analysis files include all of the data sheet timings and diagrams for a specific DSP56300 family device, organized as they are in the associated data sheet. This section describes the diagrams and libraries and explains how TimingDesigner represents them.

2.1 Libraries and Diagrams

Figure 1 shows the directory structure of the diagrams and libraries within the timing analysis file set. The set includes files for clocks, GPIO, ESSI, the host interface (HI08), JTAG, OnCE™, external memory expansion (Port A), reset, SCI and timers.

To compare the timing file structure to the organization of the data sheet, refer to the data sheet section entitled *External Memory Expansion Port (Port A)*, which has subsections for arbitration, DRAM, SRAM, and synchronous timings. Similarly, in the timing analysis file structure, the `porta` directory includes `arbitrat`, `dram`, `sram` and `synchron` subdirectories.

Diagram names in the timing analysis files correspond to those in the data sheet. For example, Figure 2B-12 in the DSP56303 data sheet, revision 2, is the SRAM Read Access timing diagram, so the associated electronic diagram is `fig2B_12.td`. (The bold line in **Figure 1** indicates the path to `fig2B_12.td`.)

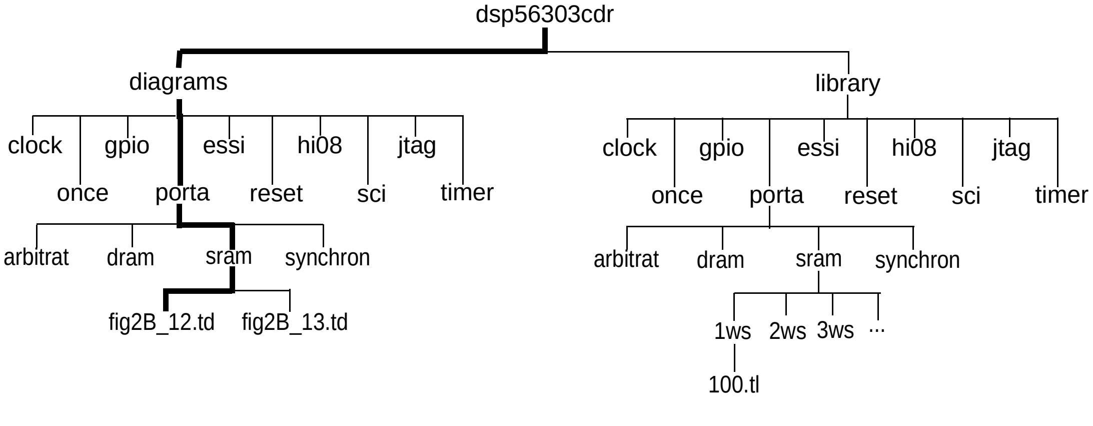


Figure 1. Directory Structure

2.2 Diagrams and Parameters

When you open TimingViewer, it displays a diagram window and a parameter window.[1] Use the File menu to open one of the Motorola diagrams.

To follow the example in this section, open the file named `fig2B_4.td` in the `reset` directory. The diagram window displays the timing relationships associated with synchronous reset timing shown in **Figure 2**.

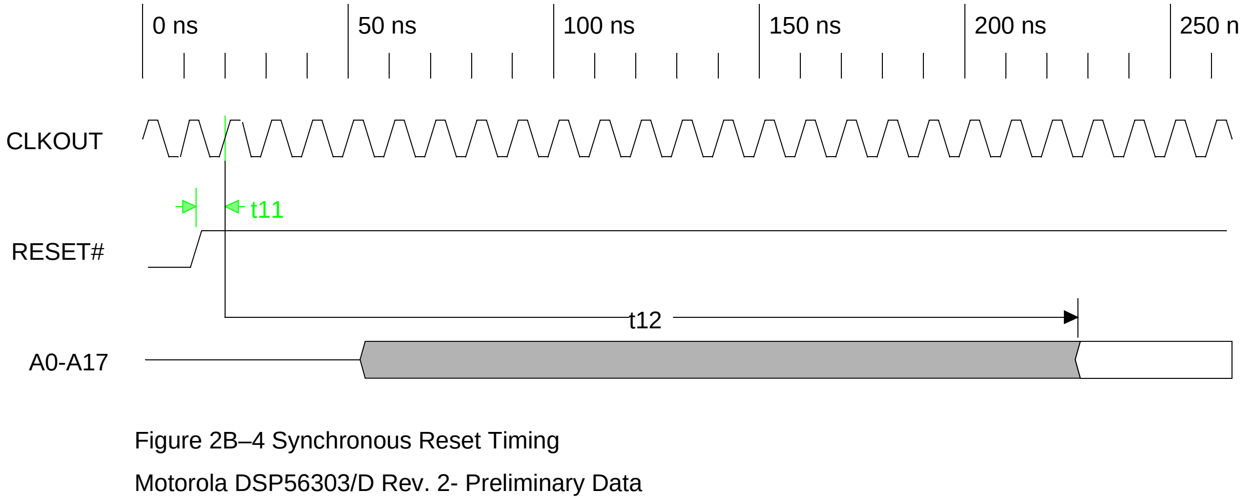


Figure 2B–4 Synchronous Reset Timing

Motorola DSP56303/D Rev. 2- Preliminary Data

Figure 2. Diagram Window for fig2B_4.td

Diagrams visually represent all signals and timing relationships described in the data sheet figure. At the base of each diagram are the figure's title, which describes the timing relationships, and its source. In this example, the diagram is from the DSP56303 data sheet, revision 2.

Note: In the timing diagrams, the # symbol represents signals that are asserted low (for example, RESET#). Otherwise, this application note and the data sheet use overbars for that purpose (for example, $\overline{\text{RESET}}$).

The parameter window shows all timing information and documentation associated with the diagram, including the actual, numerical values for the timings shown in the associated diagrams. This timing information derives from TimingDesigner libraries that Motorola provides and is organized in the same way for all diagrams in the Motorola DSP56303 timing libraries. **Table 1** contains parameter information for the reset diagram.

[1] If either window does not appear, go to the Window menu and select **Parameter** or **Diagram** to open the missing window.

Table 1. Parameter Window for fig2B-4.td

Row		Name	Formula	Min	Max	Margin	Comment
1	V	Part_Number	[,]				Motorola DSP56303 24-Bit General Purpose Digital Signal Processor
2	V	Data_Reference	[,]				Motorola DSP56303/D Rev.2 - Preliminary Data
3	V	Revision_Information	[,]				Revision 2.0 - November 1999 - Motorola
4	V	Mode_Description	[,]				Figure 2B-4 Synchronous Reset Timing
5	V	Aliases	[,]				
6	P	$DSP	dsp56303cdr:library:reset:$freq				This "Part Alias" stores the directory path to the timing parameter libraries referenced in the timing diagram.
7	A	$freq	100				ENTER: <100> in the "Formula" field to select the internal operating frequency (MHz).
8	V	Constants	[,]				
9	V	freq	100	100	100		Enter <x> MHz frequency desired for operation
10	V	TC	((1/freq)*1000)	10	10		ENTER: <clock_cycle_time> in the "Formula" field to set the actual cycle time (ns).
11	V	duty.cycle	50	50	50		
12	V	rising.jitter	0	0	0		
13	V	falling.jitter	0	0	0		
14	V	Input_Constraints	[,]				
15	C	t11	$DSP:t11(TC)	5.9	10	<3.1,1>	Synchronous reset setup time from RESET# deassertion to CLKOUT Transition 1
16	V	Output_Delays	[,]				
17	D	t12	$DSP:t12(TC)	33.5	203.5		Synchronous reset deasserted, delay time from the CLKOUT 1 to the first external address output

The parameter information table contains the following types of information:

- Rows 1 through 4 identify the diagram’s reference information, revision information, and mode description.
- Rows 6 and 7 define aliases for the libraries and the internal operating frequency to be modeled.
- Rows 9 through 13 contain constants that set the clock frequency of the part and duty cycle information. These constants are used in calculating the timings lower in the table. Other constants

can be defined here as well, such as the clock period length (T_c) or wait states constants required by the formulas that represent values in the data sheet.

- Rows 15 and 17 provide the actual timing values, first for constraints and then for delays. The constraint and delay information appears in the table after an associated constraint or delay is drawn in the diagram window.

2.3 A Note on Constraints and Delays

TimingDesigner represents constraints and delays differently. Therefore, before modeling timings, you must determine which ones are constraints and which ones are delays. The TimingDesigner help index defines constraints and delays as follows:

> A constraint establishes a relationship between two edges that must be maintained for the design to work properly. Setup times, hold times, and pulse widths are examples of constraints. A delay represents a cause-and-effect relationship between two edges, such as the propagation delay caused by a signal going through a logic gate. The source of a delay is an input signal transition, and the target is an output signal transition. A delay value expresses the timing delay from the source edge to the target edge.

We separate the two types of timings in the parameter window to make it easier to reference them. In **Figure 2**, timing t11, a constraint, is shown under the Input_Constraints category. Timing t12, a delay, is shown below, under the Output_Delay category. The diagram also represents the two timings differently: t11 is shown in green as a constraint that is met (if not met, this constraint would be red) and t12 is shown in black.

3 DSP-to-SRAM Analysis

With a general idea of the information included in Motorola's timing libraries and diagrams, you should be ready to analyze an interface between the DSP and another device. A commonly used interface is the Port A interface to SRAM. This section illustrates how to analyze all timings in this interface through an example based on the following objective:

> Determine whether a 12-ns asynchronous SRAM device can interface to a DSP56303 running at 100 MHz with 1 wait state, or if a 10-ns memory device is required.

This example uses the Motorola MCM6341 SRAM and refers to the MCM6341 data sheet, which you can download from the Motorola website:

`http://www.mot-sps.com/products/memory/srams/asynchronous`

3.1 Creating a Memory Library

Before starting to define the interface, you must have a library for the memory device. Some memory device libraries are available on the web, but many are not. This section explains how to create a library using the memory data sheet as a guideline.

Note: You can follow the steps in this section with either TimingDesigner or TimingViewer. However, to save the completed library, you must use TimingDesigner (TimingViewer cannot save data and, therefore, cannot create a library). If you do not have TimingDesigner, you may want to download the completed library from the Motorola DSP web site (see *Motorola Timing Analysis Files and Data Sheet* on page 2) and skip to **Section 3.2**, *Relating the Memory and DSP Timings*, on page 7.

From either the diagram or parameter window:

1. Click on the Window menu.
2. Select the **Library** window. This opens a window in which to enter memory timings.
3. Locate the MCM6341 data sheet table for read timings and write timings. (You can separate timings into multiple files, say one library for read timings and one library file for write timings. However, memory timing tables are so short that it may be more convenient to keep all timings in one file.)
4. Enter the timings (in the same way that you enter data into an electronic spreadsheet).

If you follow the format used earlier for the DSP56303 libraries (**Table 1**), the completed memory library resembles **Table 2**.

Table 2. MCM6341 Electronic Library—Read Timings

Row		Name	Formula	Min	Max	Comment
1	V	Part_Number	[,]			Motorola MCM6341 128K x 24 bit Static Random Access Memory
2	V	Data_Reference	[,]			Motorola MCM6341/D Rev.4 - Page 5
3	V	Mode_Description	[,]			Read Cycle Timing
4	V	TAVAV	[10,]	10		Read Cycle Time
5	V	TAVQV	[,10]		10	Address Access Time
6	V	TELQV	[,10]		10	Enable Access Time
7	V	TGLQV	[,4]		4	Output Enable Access Time
8	V	TAXQX	[3,]	3		Output Hold from Address Change
9	V	TELQX	[3,]	3		Enable Low to Output Active
10	V	TGLQX	[0,]	0		Output Enable Low to Output Active
11	V	TEHQZ	[0,5]	0	5	Enable High to Output High-Z
12	V	TGHQZ	[0,5]	0	5	Output Enable High to Output High-Z

5. Save the library to a file. Parameters and diagrams can now refer to this file for all MCM6341 read cycle timings.

3.2 Relating the Memory and DSP Timings

Now that you have both the DSP and memory device library timings, you can begin to create the interface diagrams. The MCM6341 data sheet has four diagrams. Because the DSP56303 provides a read signal separate from the Chip Select, only one MCM6341 read cycle and one write cycle diagram relate to the interface with the DSP56303 (and other DSP56300 devices): Read Cycle 2 and Write Cycle 1.

3.2.1 Adding Constraints

To create the interface diagram for the write cycle, use the DSP56303 SRAM Write Access diagram as a starting point. Open the `dsp56303cdr/diagrams/porta/sram/fig2B_13.td` file in the diagram window.

As the MCM6341 write cycle diagram in the data sheet shows, tDVWH is a setup time from Data valid on the DSP to $\overline{\text{WR}}$ deassertion. Because tDVWH is a setup time, it should be modeled as a constraint.

To add a constraint:

1. To select the constraint tool, click on the **CONST** button to the left of the diagram window.

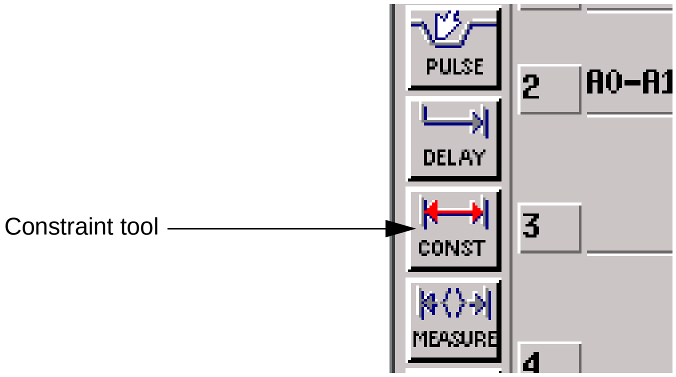


Figure 3. Constraint Tool

2. Click on the constraint source (shown as **Data out edge** in **Figure 4**).
3. Click on the constraint destination (**WR# deassertion edge** in **Figure 4**).

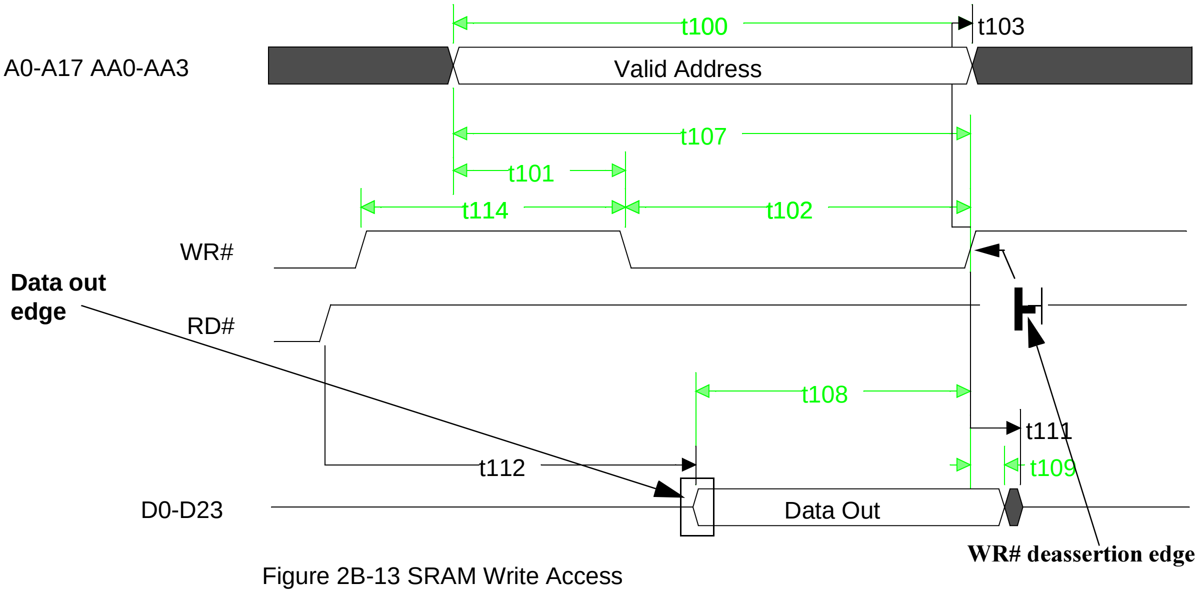
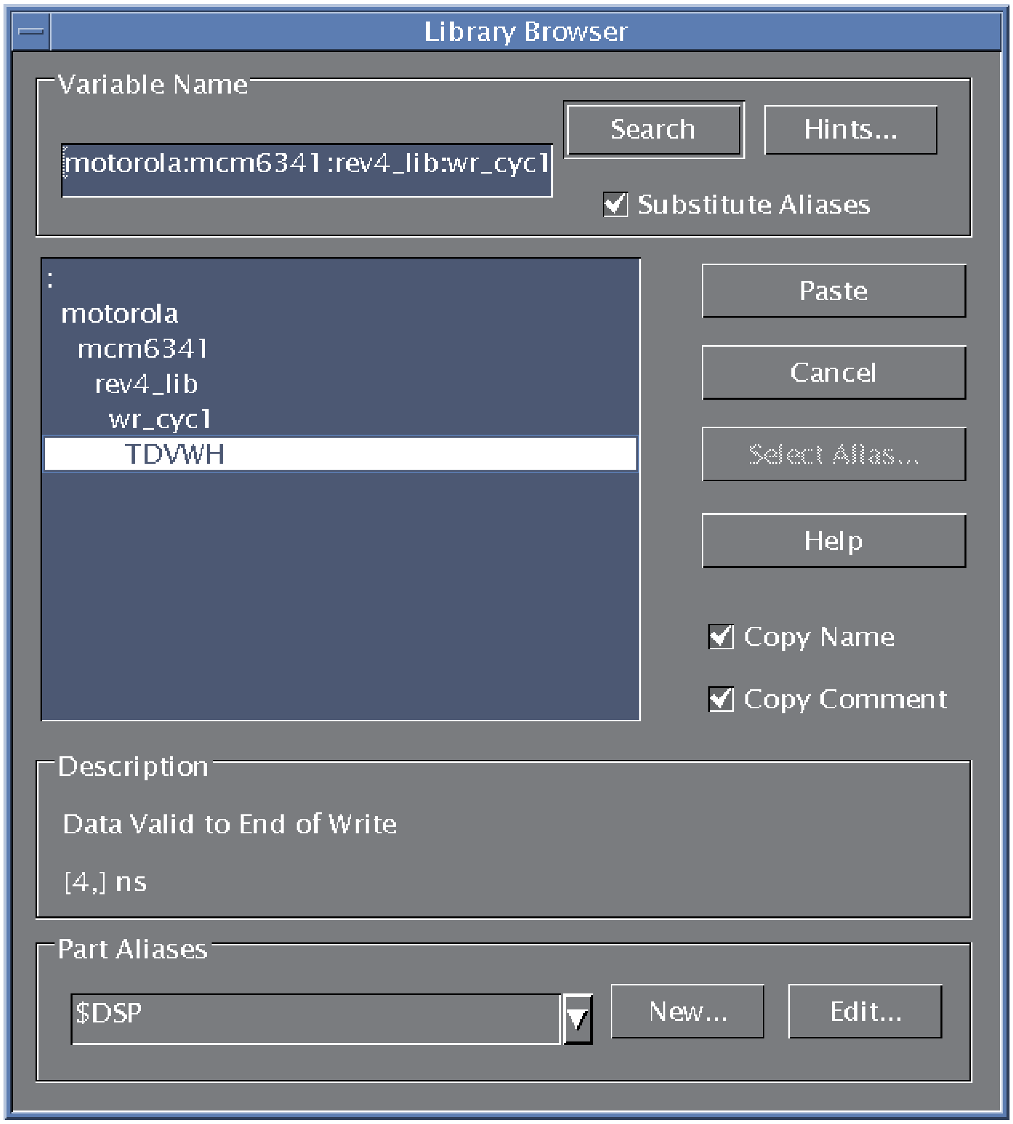


Figure 2B-13 SRAM Write Access
Motorola DSP56303/D Rev. 2- Preliminary Data

Figure 4. Adding Constraints

4. TimingDesigner displays a constraint dialog box that allows you to select the appropriate memory timing from the library file. Click on the **Browse Library . . .** button.

5. In the library browser window, locate the appropriate library timing file, as shown in **Figure 5**. The TDVWH timing is in the `wr_cyc1` library, which is saved in `motorola/mcm6341/rev4_lib` directory. Select **Paste**, which directs TimingDesigner to refer to this library for the value of this timing constraint specification. (If other libraries are available, you can swap among them by changing the path alias value in the parameter window.)



Variable Name

motorola:mcm6341:rev4_lib:wr_cyc1

☑ Substitute Aliases

```
:
 motorola
  mcm6341
   rev4_lib
    wr_cyc1
     TDVWH
```

☑ Copy Name

☑ Copy Comment

Description

Data Valid to End of Write

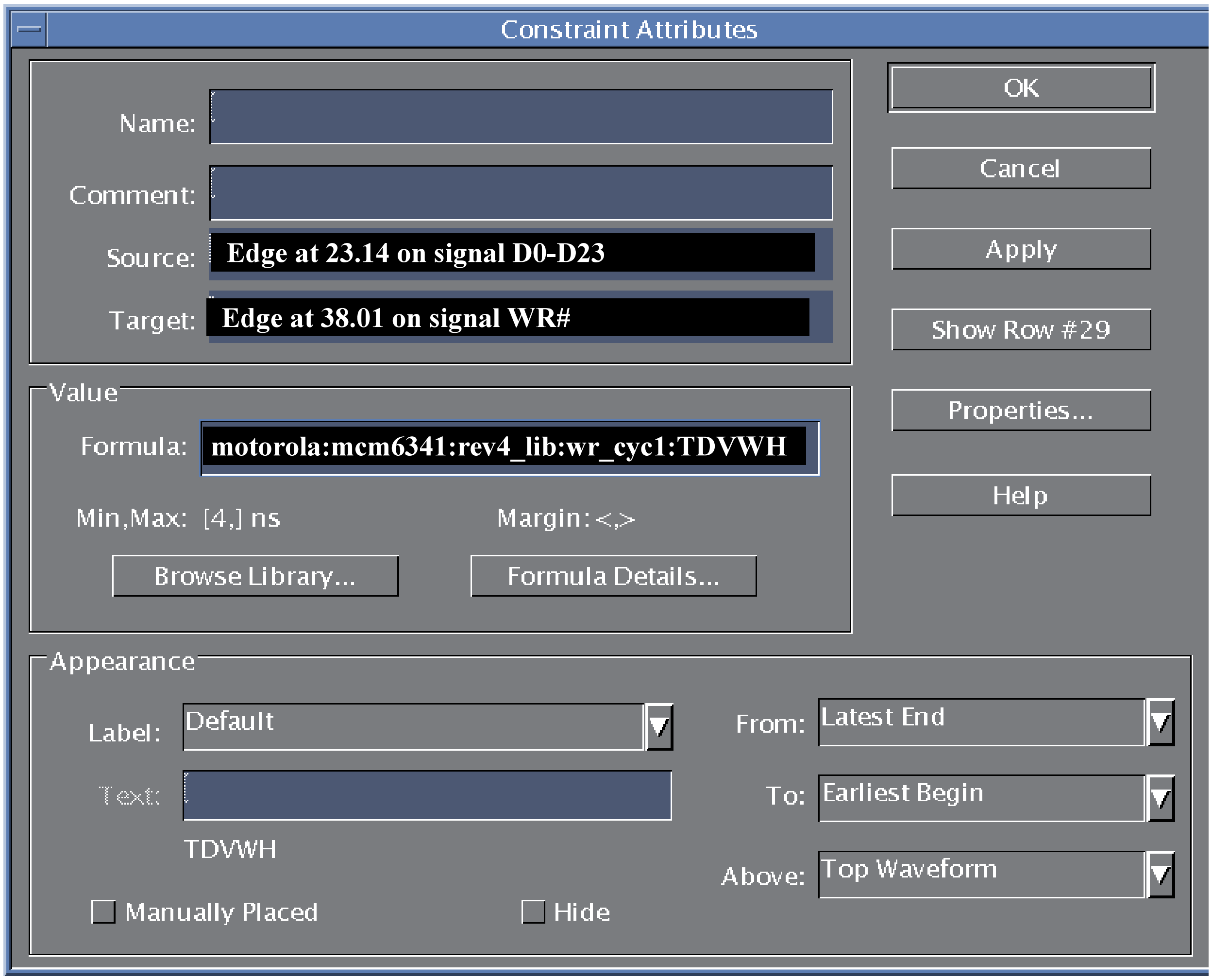
[4,] ns

Part Aliases

$DSP

Figure 5. Browse for Library

Figure 6 shows the constraint dialog box after the library timing file has been selected.



Constraint Attributes

Name:

Comment:

Source: Edge at 23.14 on signal D0-D23

Target: Edge at 38.01 on signal WR#

Value

Formula: motorola:mcm6341:rev4_lib:wr_cyc1:TDVWH

Min,Max: [4,] ns

Margin: <,>

Appearance

Label: Default

Text:

TDVWH

From: Latest End

To: Earliest Begin

Above: Top Waveform

☐ Manually Placed

☐ Hide

Figure 6. Constraint Dialog Box

3.2.2 Adding Signals

The write cycle timing diagram in the MCM6341 data sheet includes an additional signal line, Q, that you must add to the diagram. To add a signal, follow these steps:

1. Click the **SIGNAL** button in the top, left corner of the diagram window (see **Figure 7**). TimingDesigner displays a dialog box (shown completed in **Figure 8**).

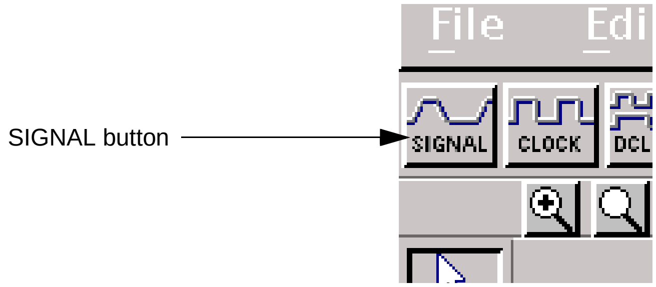
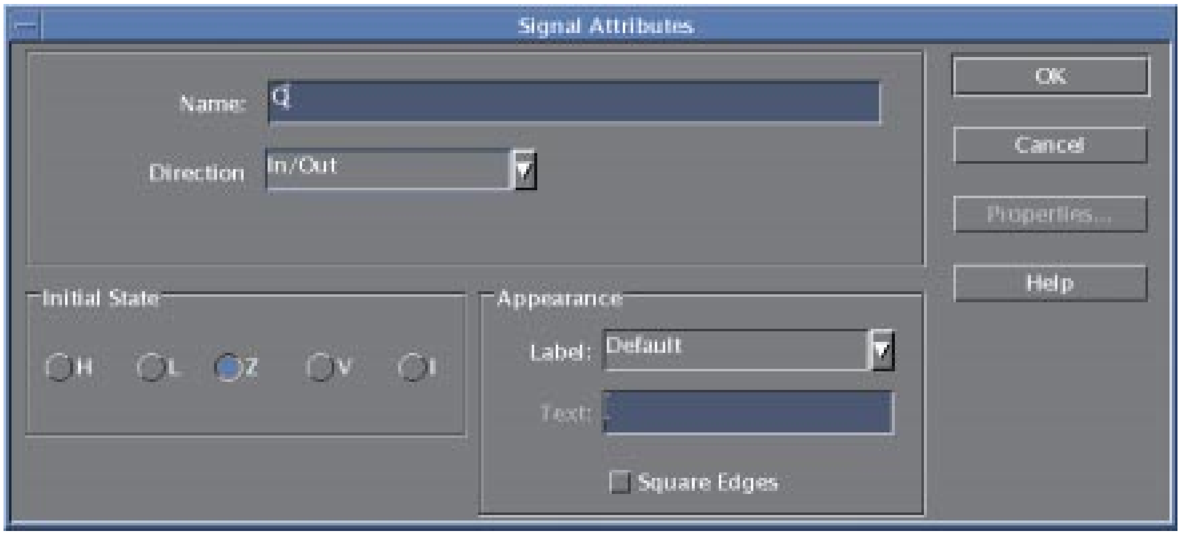


Figure 7. Signal Tool

2. In the dialog box Name field, enter **Q .**
3. From the Direction menu, select **out**.
4. In the initial state box, select **Z**.



Name: Q

Direction: In/Out

○ H ○ L ● Z ○ V ○ I

Label: Default

Text:

□ Square Edges

Figure 8. Signal Attributes Dialog Box

5. Click **OK .** TimingDesigner displays the signal name **Q** at the bottom of your diagram window.
6. Click to the right of the signal name in the diagram area to display the signal.
7. To complete and display the signal, click at the five places indicated in **Figure 9**.

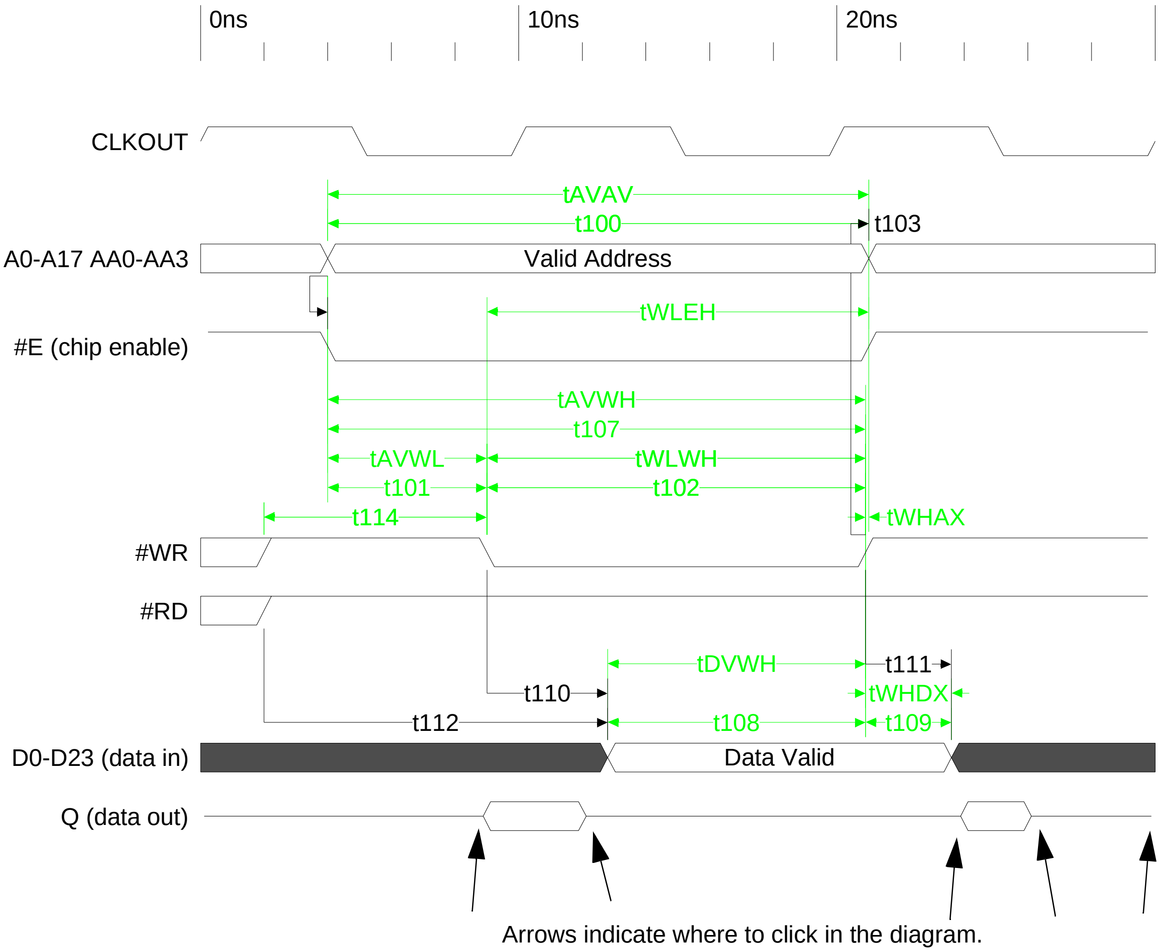


Figure 9. Adding Q Signal

The data sheet shows that the two data areas for signal Q are invalid. Therefore, you must mark them invalid in the diagram, as follows:

1. Click on a data area.
2. Click on the **INVALID** button shown in **Figure 10** and located at the top of the diagram window. **Figure 11** shows the result.

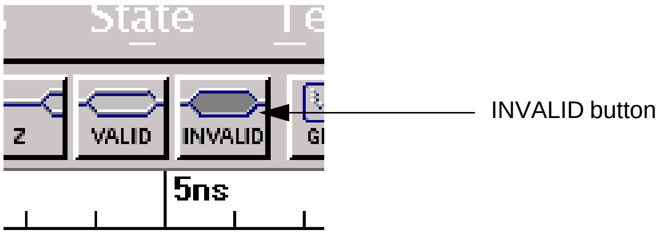


Figure 10. Invalid Tool

3. Repeat steps 1 and 2 to invalidate the other portion of the Q signal line.

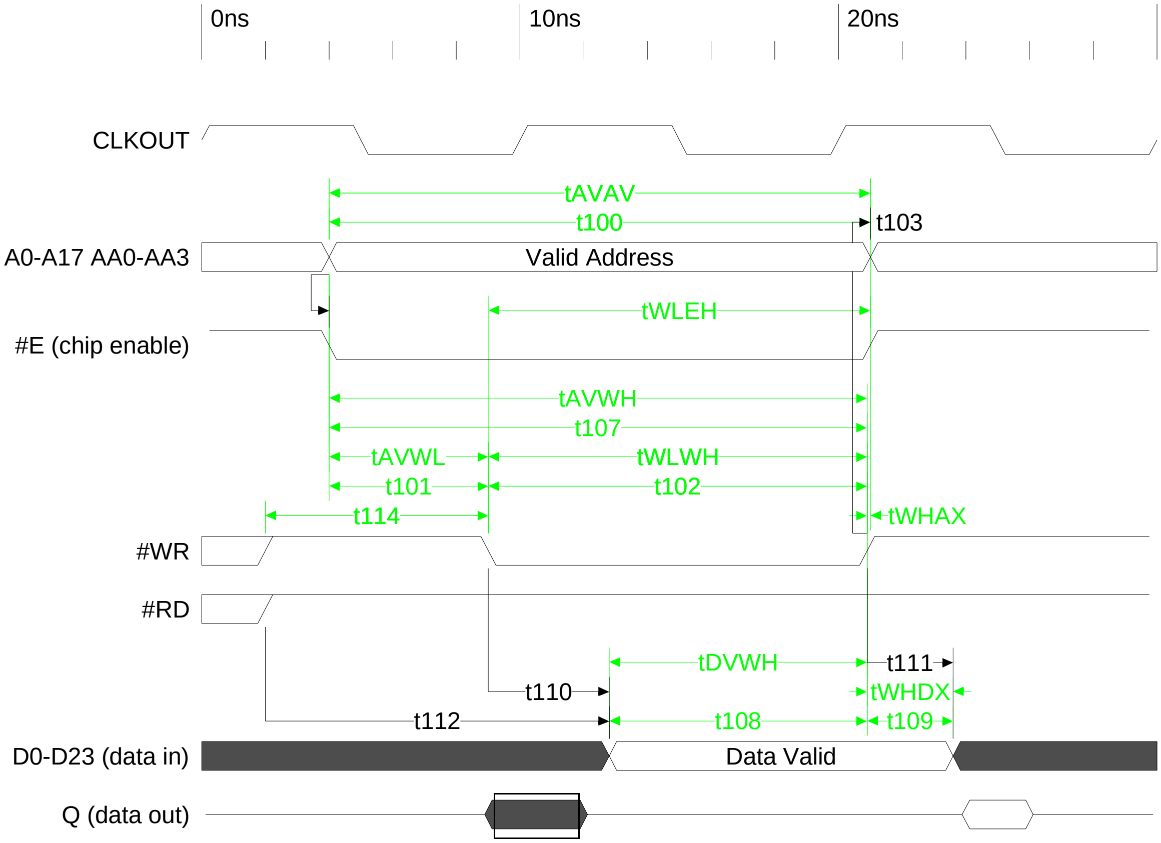


Figure 11. Changing Signal from Valid to Invalid

3.2.3 Adding Delays

Adding delays is similar to adding constraints. As the MCM6341 data sheet shows, TWLQZ is a delay from $\overline{\text{WR}}$ assertion to Data Out High-Z on the Q signal line created earlier. To add this delay to your diagram, follow these steps:

1. Click on the **DELAY** button shown in **Figure 12** and located at the left of the diagram window.

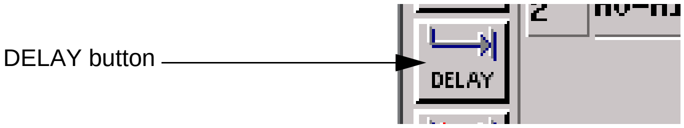
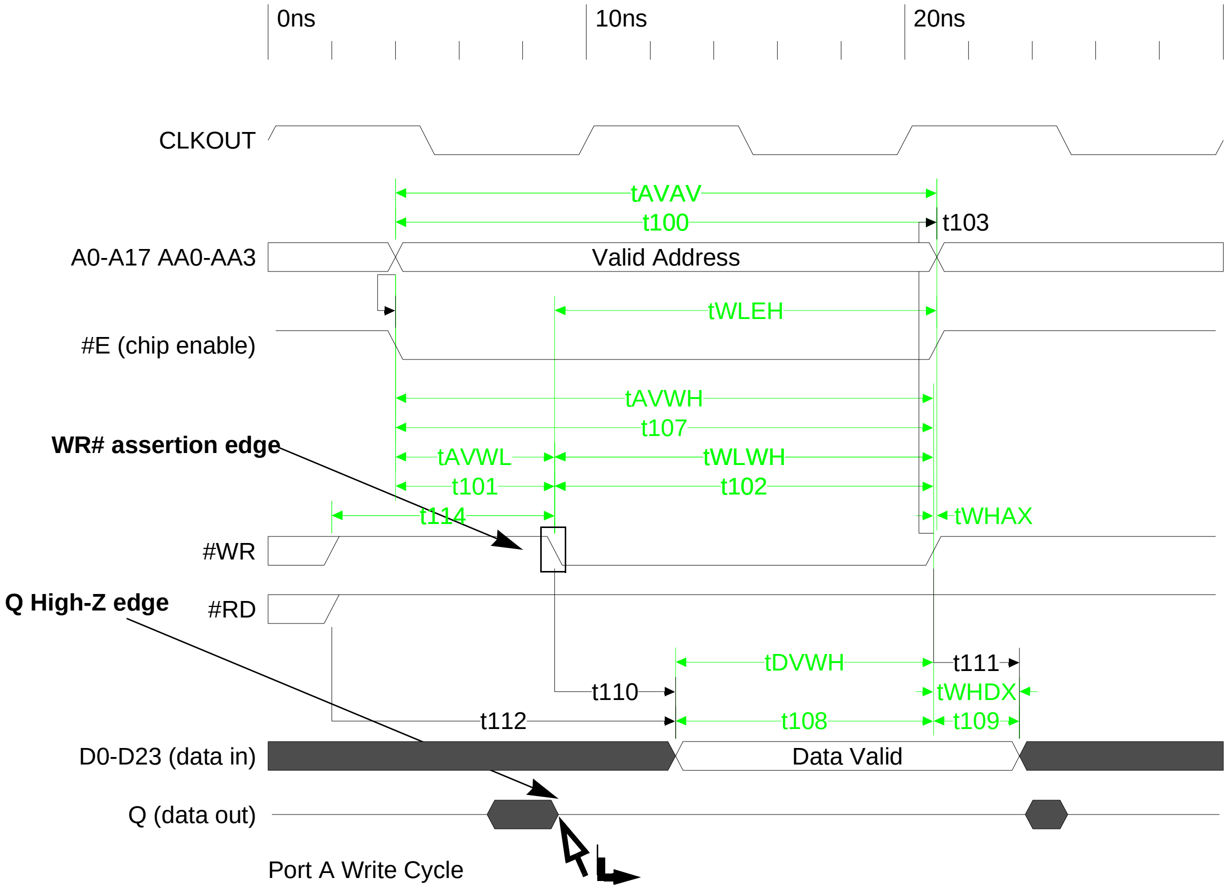


Figure 12. Delay Tool

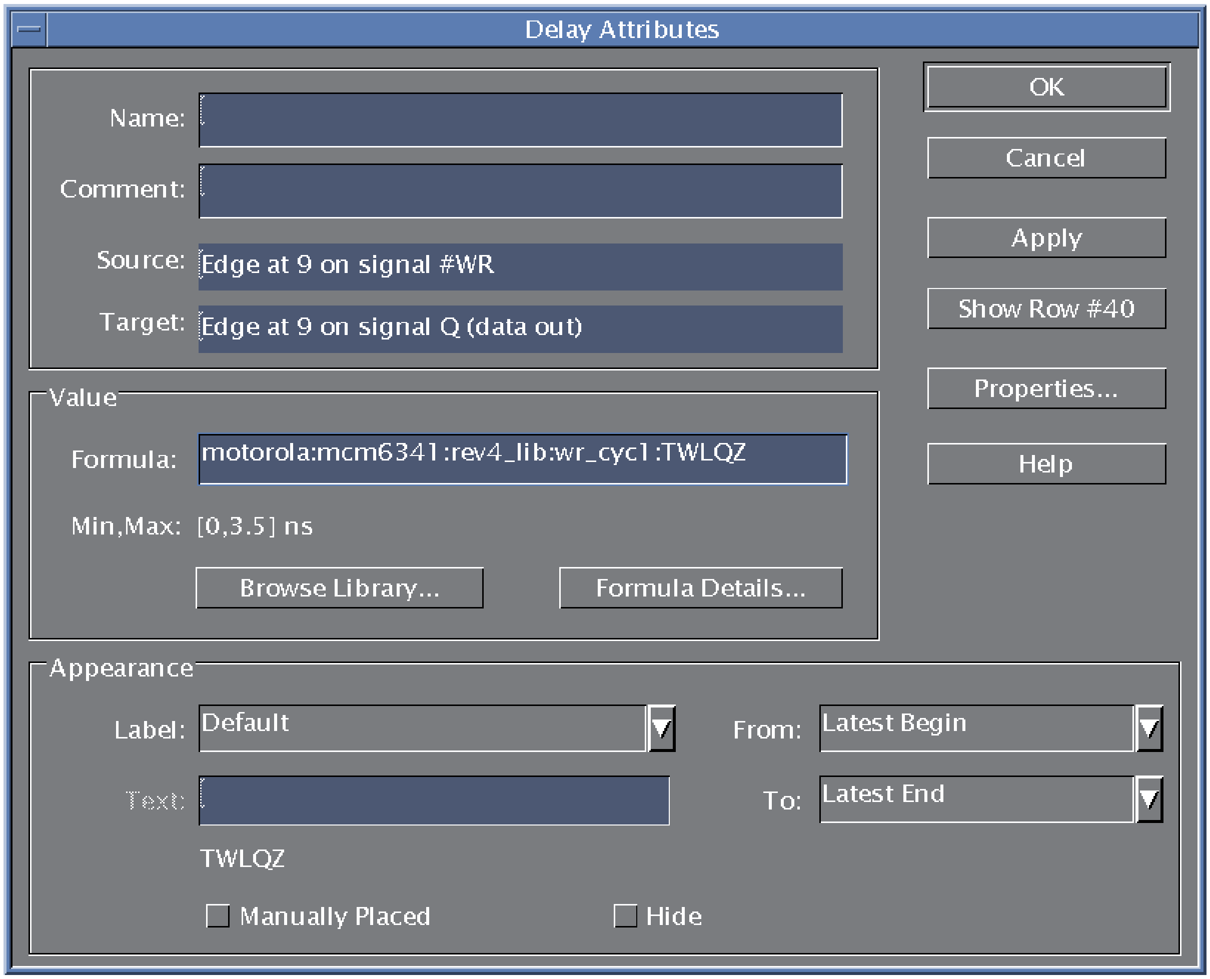
2. Click on the signal edge that is the source of the delay (**#WR assertion edge** in **Figure 13**).
3. Click on the edge that is the destination of the delay (**Q High-Z edge** in **Figure 13**).



Port A Write Cycle
Motorola DSP56303/D Rev. 2 - Figure 2B-13 page 2B-20
Motorola MCM6341/D Rev. 4 - Page 7

Figure 13. Adding Delays

4. When the delay dialog box shown in **Figure 14** appears, select the appropriate delay from the memory library file created previously (see *Creating a Memory Library* on page 6).



Name:

Comment:

Source: Edge at 9 on signal #WR

Target: Edge at 9 on signal Q (data out)

Value

Formula: motorola:mcm6341:rev4_lib:wr_cyc1:TWLQZ

Min,Max: [0,3.5] ns

Appearance

Label: Default

From: Latest Begin

Text:

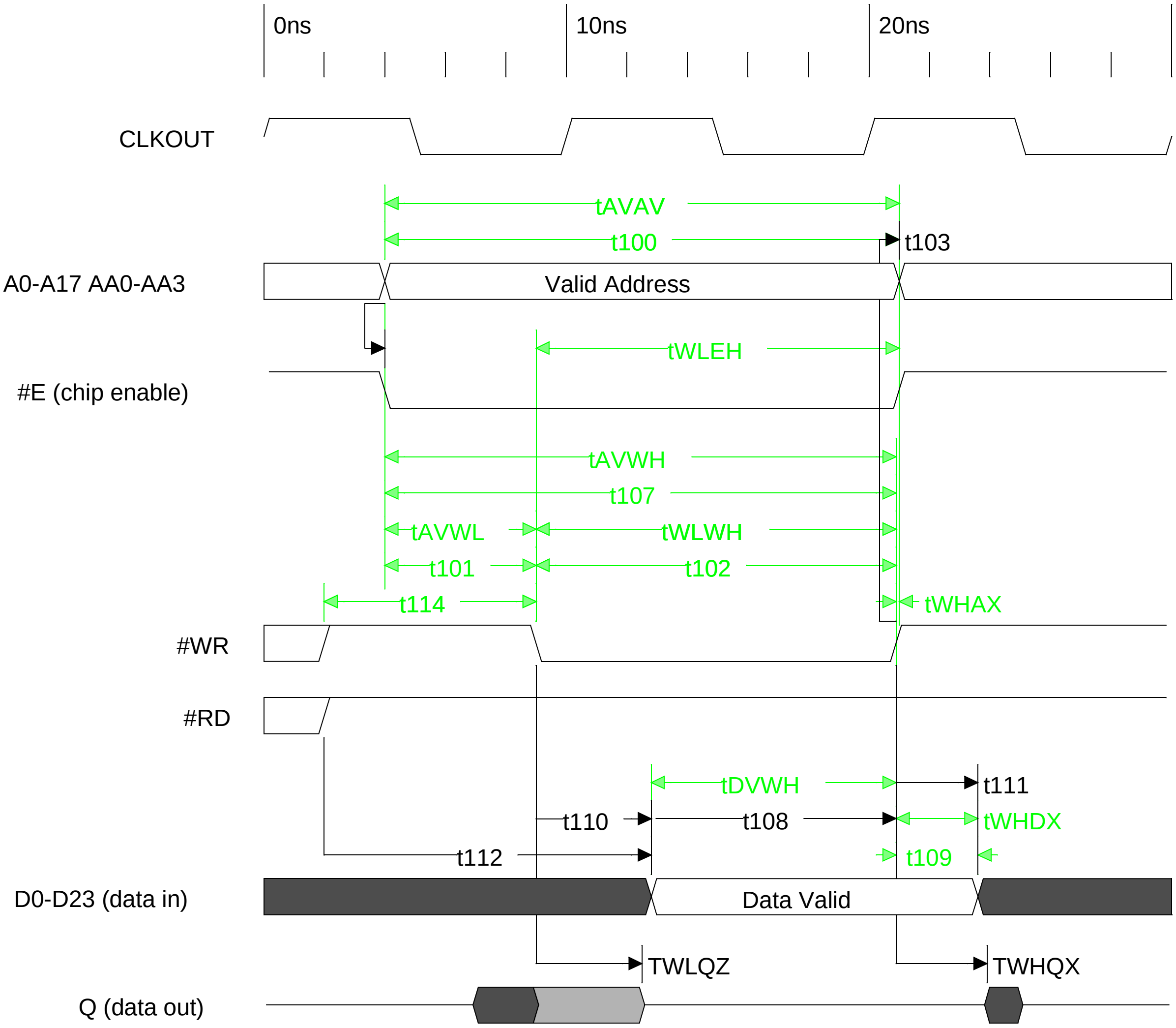
To: Latest End

TWLQZ

☐ Manually Placed ☐ Hide

Figure 14. Delay Dialog Box

5. Add the remaining delays (repeat steps 1–3). Once all memory delay and constraint timings are added, the final diagram should look similar to the one shown in **Figure 15**.



Port A Write Cycle

Motorola DSP56303/D Rev. 2- Figure 2B-13 page 2B-20

Motorola MCM6341/D Rev. 4 - Page 7

Figure 15. Completed DSP56303-MCM6341 Write Interface Diagram

For a more detailed discussion of how to enter constraints and delays, refer to the TimingDesigner or TimingViewer on-line help facility.

3.3 Finding and Correcting Timing Violations

A timing violation occurs when there is no margin between the DSP-specified timing and the memory timing. The on-line TimingDesigner help defines margin as follows:

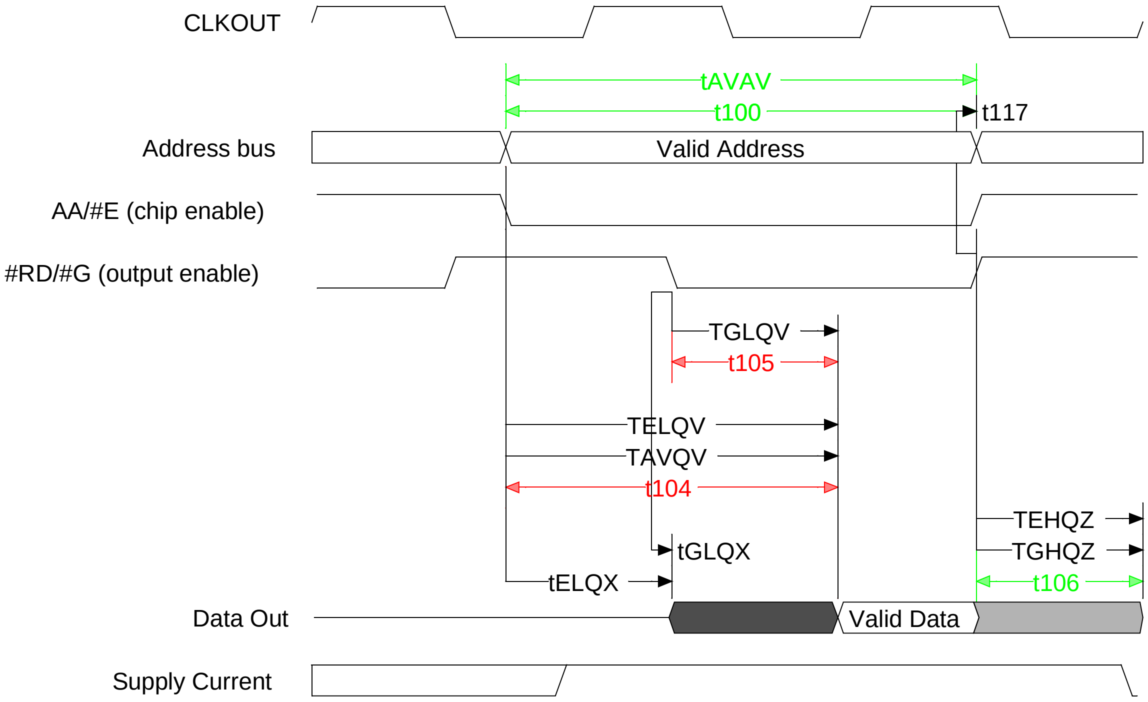
> Margin indicates the amount of time by which a constraint is satisfied or violated and is shown between angle brackets. For satisfied constraints (e.g., <2,>), margin shows the amount of time beyond what is required by the constraint; for violated constraints (e.g., <,-5>), margin shows the amount of time needed in order to satisfy the constraint.

It is often helpful to check constraint boundaries in order to understand the timing interactions, especially when analyzing timing violations. To display timing margins in TimingDesigner, double-click on a timing in the diagram or refer to the appropriate parameter table.

This section explains how to identify, validate and correct timing violations with TimingDesigner.

3.3.1 Identifying Violations

TimingDesigner displays violated timings in red. **Figure 16** shows an interface with violations.



Port A Read Cycle accessing 128Kx24 FSRAM
Motorola DSP56303/D Rev. 2 - Figure 2B-12, page 2B-20
Motorola MCM6341/D Rev. 4 - Page 6

Figure 16. 12-ns MCM6341 to DSP56303 Interface—in Violation

Timings 104 and 105 are access times for the memory device. (Memory interface access timings are the timings most likely to have violations, so check them first during your analysis.) TimingDesigner displays timings 104 and 105 in red because they require valid data before the memory device can provide it. As the parameter table (**Table 3**) shows, t104 must have data within 10.5 ns, and the memory does not guarantee data until 12 ns, resulting in a margin of <,-1.5>.

3.3.2 Validating Timing Violations

Timing diagrams indicate potential violations. Validate all potential violations as follows:

1. Make sure that the library value is correct. Check the timing in the parameter table and compare it to the timing value in the appropriate data sheet. (For example, refer to the DSP56303 data sheet and the parameter table shown in **Table 1** to validate the timing violations shown in **Figure 16**.)

Table 3. Parameter Table for 12-ns Memory Interface Diagram

Row		Name	Formula	Min	Max	Margin	Comment
1	V	DSP_Number	$DSP:Part_Number				Motorola DSP56303 24-Bit General Purpose Digital Signal Processor
2	V	MEMORY_Number	$memory:Part_Number				Motorola MCM6341 128K x 24 bit Static Random Access Memory
3	V	Data_Reference1	$DSP:Data_Reference				Motorola DSP56303/D Rev. 2 - Figure 2B-12 page 2B-19
4	V	Data_Reference2	$memory:Data_Reference				Motorola MCM6341/D Rev.4 - Page 6
5	V	Mode_Description	[,]				Port A Read Cycle accessing 128Kx24 FSRAM
6	V	Aliases	[,]				Aliases required for library path settings
7	A	$DSP	dsp56303cdr:library:porta:sram:$ws:$freq				
8	A	$memory	motorola:mcm6341:rev4_lib:$memspeed				
9	A	$freq	100				Enter <100> MHz
10	A	$ws	1ws				Enter <1ws,2ws,3ws,...,8ws> wait states
11	A	$memspeed	12ns				Enter <10ns, 12ns> Memory Speed
12	V	Constants	[,]				Constants Required for Formulas
13	V	freq	100	100	100		Enter <x> MHz frequency of operation
14	V	TC	((1/freq)*1000)	10	10		1/freq (in ns)
15	V	WS	1	1	1		Required wait states for access to memory device
16	V	duty.cycle	50	50	50		

Table 3. Parameter Table for 12-ns Memory Interface Diagram (Continued)

Row		Name	Formula	Min	Max	Margin	Comment
17	V	rising.jitter	0	0	0		
18	V	falling.jitter	0	0	0		
19	V	Input_Constraints	[,]				
20	C	tAVAV	$memory:TAVAV	12		<5,>	Read Cycle Time
21	C	t100	$DSP:t100(WS,TC)	16		<1,>	Address Valid
22	C	t104	[,10.5]		10.5	**<,-1.5>**	Address and AA valid to input data valid
23	C	t105	[,5.5]		5.5	**<,-0.5>**	#RD assertion to input data valid
24	C	t106	$DSP:t106	0		<0,>	RD deassertion to data not valid (data hold time)
25	V	Output_Delays	[,]				
26	D	TAVQV	$memory:TAVQV		12		Address Access Time
27	D	TELQV	$memory:TELQV		12		Enable Access Time
28	D	TGLQV	$memory:TGLQV		4		Output Enable Access Time
29	D	tELQX	$memory:TELQX	3			Enable Low to Output Active
30	D	tGLQX	$memory:TGLQX	0			Output Enable Low to Output Active
31	D	TEHQZ	$memory:TEHQZ	0	6		Enable High to Output High-Z
32	D	TGHQZ	$memory:TGHQZ	0	6		Output Enable High to Output High-Z
33	D	t117	[0,]	0			RD deassertion to address not valid

2. If the parameter values are correct, check the margin of the violated timing with any other related timings (compare other delays in the parameter table with the data sheet timing values).

If these two checks indicate an actual violation (that is, they reveal no library or diagram creation errors), your next steps are to:

1. Document the violation.

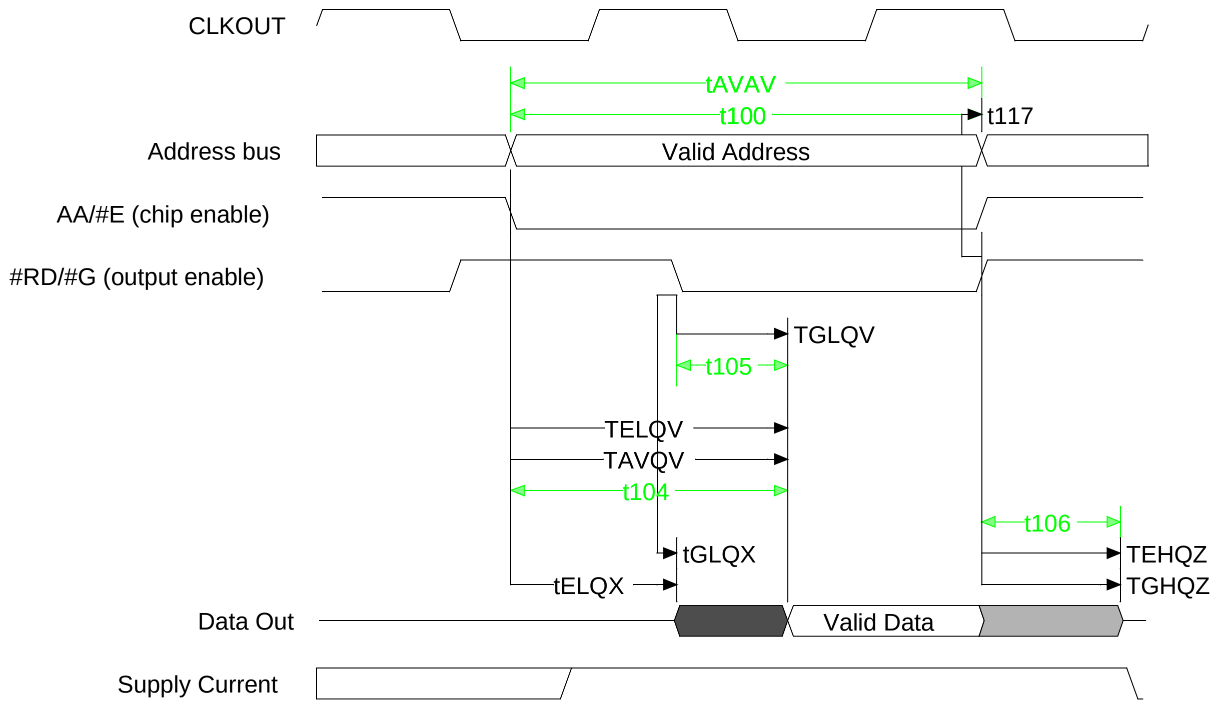
2. Determine whether the violation is an issue in a given system.

If the timing violation causes a problem, you may be able to use a different device.

In the example shown in **Figure 16**, timing t105 is in violation with DSP timing tGLQV. The referenced library values are correct and the timing is clearly in violation when the DSP operates at 100 MHz. This violation has two possible solutions:

- Run the DSP at a slower speed (if acceptable in a specific application).
- Use faster SRAM (if the application requires the DSP to run at 100 MHz).

For example, a 10-ns SRAM might correct the violation shown in **Figure 16**, so you could enter and analyze the 10-ns memory interface.The 10-ns MCM6341 interface shown in **Figure 17** has no violations, so you can use the 10-ns memory in a system with the DSP56303.



Port A Read Cycle accessing 128Kx24 FSRAM
Motorola DSP56303/D Rev. 2- Figure B2-12 on page 2B-20
Motorola MCM6341/D Rev. 4 - Page 6

Figure 17. 10-ns MCM6341 to DSP56303 Interface

3.4 Adjusting an Interface for Various Speeds

With TimingDesigner, you can analyze interface timings at various memory and DSP speeds. Variables in the parameter table select the appropriate DSP and memory speed. **Table 4** shows the variables in the parameter table that can be changed: $mem, $freq, $ws, freq, and WS.

Table 4. Speed Settings in the Parameter Table

Row		Name	Formula	Min	Max	Margin	Comment
6	V	Aliases	[,]				
7	A	$DSP	:dsp56303cdr:library:porta:sram:$ws:$freq				
8	A	$memory	:mcm6341:rev4_lib:$mem				
9	A	$mem	12				
10	A	$freq	100				

Table 4. Speed Settings in the Parameter Table (Continued)

Row		Name	Formula	Min	Max	Margin	Comment
11	A	$ws	1ws				
12	V	Constants	[,]				
13	V	freq	100	100	100		
14	V	TC	((1/freq)*1000)	10	10		
15	V	WS	1	1	1		
16	V	duty.cycle	50	50	50		
17	V	rising.jitter	0	0	0		
18	V	falling.jitter	0	0	0		
19	V	Input_Constraints	[,]				
20	C	tAVAV	$memory:TAVAV	12		<5,>	Read Cycle Time
21	C	t100	$DSP:t100(WS,TC)	16		<1,>	Address Valid

The following examples demonstrate ways to adjust an interface with the parameter table:

- **Change DSP speed**. To model an 80-MHz DSP, change the values of freq and $freq to 80. (DSP timing formulas use the freq variable to determine clock cycle time. The $freq alias selects the 80-MHz library file referenced in row 8.) In this case, you must also change library path, since the 66-MHz and 80-MHz timings are for a different device (one made with UDR, rather than CDR, process technology). To change the library path, change the value of the $DSP alias from dsp56303cdr to dsp56303udr. (It is not always necessary to change the library path, as you can see from the following example based on changing memory device speed.)
- **Change memory device speed**. Change the value of $mem to a different speed (a library file for that speed must exist). For example, in the downloaded file with the MCM6341 libraries, both 10-ns timings and 12-ns timings are available. So, the value of $memspeed can be changed between 10 ns and 12 ns to analyze both interfaces.
- **Change memory access time**. To model a slower memory access time, change the wait state setting. For example, to model for three wait states, change the value of $ws to 3ws (to select the three-wait-state library file from the DSP libraries) and the value of WS to 3 (to provide the correct wait state value for timing calculations.

3.4.1 Substituting Device Libraries in a Diagram

Memory manufacturers often use different naming conventions for a given timing specification. Therefore, after swapping libraries, be sure to check the information in the parameter window to ensure that timing names are still valid. TimingDesigner may show timing specification as valid in the diagram window, but flag them with an **ERROR** in the parameter window. If this occurs, delete the timing in the diagram window and add the specification again—this time browsing for the correct timing name in the appropriate memory library.

4 DSP-to-DRAM Analysis

The steps for DRAM and SRAM analysis are the same. However, DRAM analysis takes longer because DRAM interfaces require many more timing relationships. This section provides additional information to help with DRAM analysis, but does not repeat the procedures detailed in *DSP-to-SRAM Analysis* on page 6.

4.1 Entering DRAM Timings

Enter the DRAM timings in the same way that you entered SRAM timings. Although you could use a different format for the DRAM library, you may find it helpful to use the same format defined for SRAM. Using the same format for each library makes it easier to find information. **Table 5** shows a library created from a DRAM data sheet.

Table 5. DRAM Library for 60-ns EDO DRAM

Row		Name	Formula	Min	Max	Comment
1	V	Part_Number	[,]			Samsung 1Mx16bit CMOS Dynamic RAM with Fast Page Mode
2	V	Data_Reference	[,]			KM416V1200C-6 CMOS DRAM Data Sheet March 1999
3	V	Mode_Description	[,]			All -6 Timings
4	V	tRC	[110,]	110		Random read or write cycle time
5	V	tRWC	[155,]	155		Read-modify-write cycle time
6	V	tRAC	[,60]		60	Access time from /RAS
7	V	tCAC	[,15]		15	Access time from /CAS
8	V	tAA	[,30]		30	Access time from column address
9	V	tCLZ	[0,]	0		/CAS to output in Low-Z
10	V	tOFF	[0,15]	0	15	Output buffer turn-off delay
11	V	tT	[3,50]	3	50	Transition time (rise and fall)
12	V	tRP	[40,]	40		/RAS precharge time
13	V	tRAS	[60,10000]	60	10000	/RAS pulse width
14	V	tRSH	[15,]	15		/RAS hold time
15	V	tCSH	[60,]	60		/CAS hold time
16	V	tCAS	[15,10000]	15	10000	/CAS pulse width
17	V	tRCD	[20,45]	20	45	/RAS to /CAS delay time
18	V	tRAD	[15,30]	15	30	/RAS to column address delay time
19	V	tCRP	[5,]	5		/CAS to /RAS precharge time
20	V	tASR	[0,]	0		Row address set-up time

Table 5. DRAM Library for 60-ns EDO DRAM (Continued)

Row		Name	Formula	Min	Max	Comment
21	V	tRAH	[10,]	10		Row address hold time
22	V	tASC	[0,]	0		Column address set-up time
23	V	tCAH	[10,]	10		Column address hold time
24	V	tRAL	[30,]	30		Column address to /RAS lead time
25	V	tRCS	[0,]	0		Read command set-up time
26	V	tRCH	[0,]	0		Read command hold time referenced to /CAS
27	V	tRRH	[0,]	0		Read command hold time referenced to /RAS
28	V	tWCH	[10,]	10		Write command hold time
29	V	tWP	[10,]	10		Write command pulse width
30	V	tRWL	[15,]	15		Write command to /RAS lead time
31	V	tCWL	[15,]	15		Write command to /CAS lead time
32	V	tDS	[0,]	0		Data set-up time
33	V	tDH	[10,]	10		Data hold time
34	V	tREF1	[,16000000]		16000000	Refresh period (1K, Normal)
35	V	tREF2	[,64000000]		64000000	Refresh period (4K, Normal)
36	V	tREF3	[,128000000]		128000000	Refresh period (L-ver)
37	V	tWCS	[0,]	0		Write command set-up time
38	V	tCWD	[40,]	40		/CAS to /W delay time
39	V	tRWD	[85,]	85		/RAS to /W delay time
40	V	tAWD	[55,]	55		Column address to /W delay time
41	V	tCPWD	[60,]	60		/CAS precharge to /W delay time
42	V	tCSR	[5,]	5		/CAS set-up time (/CAS before /RAS refresh)
43	V	tCHR	[10,]	10		/CAS hold time (/CAS before /RAS refresh)
44	V	tRPC	[5,]	5		/RAS to /CAS precharge time
45	V	tCPA	[,35]		35	Access time from /CAS precharge
46	V	tPC	[40,]	40		Fast Page mode cycle time
47	V	tPRWC	[80,]	80		Fast Page read-modify-write cycle time
48	V	tCP	[10,]	10		/CAS precharge time (Fast Page cycle)4
49	V	tRASP	[60,200000]	60	200000	/RAS pulse width (Fast Page cycle)
50	V	tRHCP	[35,]	35		/RAS hold time from /CAS precharge

Table 5. DRAM Library for 60-ns EDO DRAM (Continued)

Row		Name	Formula	Min	Max	Comment
51	V	tOEA	[,15]		15	/OE access time
52	V	tOED	[15,]	15		/OE to data delay
53	V	tOEZ	[0,15]	0	15	Output buffer turn off delay time from /OE
54	V	tOEH	[15,]	15		/OE command hold time
55	V	tRASS	[100000,]	100000		/RAS pulse width (C-B-R) self refresh
56	V	tRPS	[110,]	110		/RAS precharge time (C-B-R self refresh)
57	V	tCHS	[(-50),]	-50		/CAS hold time (C-B-R self refresh)

4.2 Connecting DSP to DRAM

Figure 18 shows the Page Mode Read interface between the DSP56303 and DRAM. The DRAM interface diagram is considerably more complex than the SRAM diagram, due to the number of timing relationships required in the interface. However, the arrangement is the same: title and reference information are at the base of the diagram and the same conventions relate the signals to each other (for example, address lines are above data lines, which are at the bottom of the diagram).

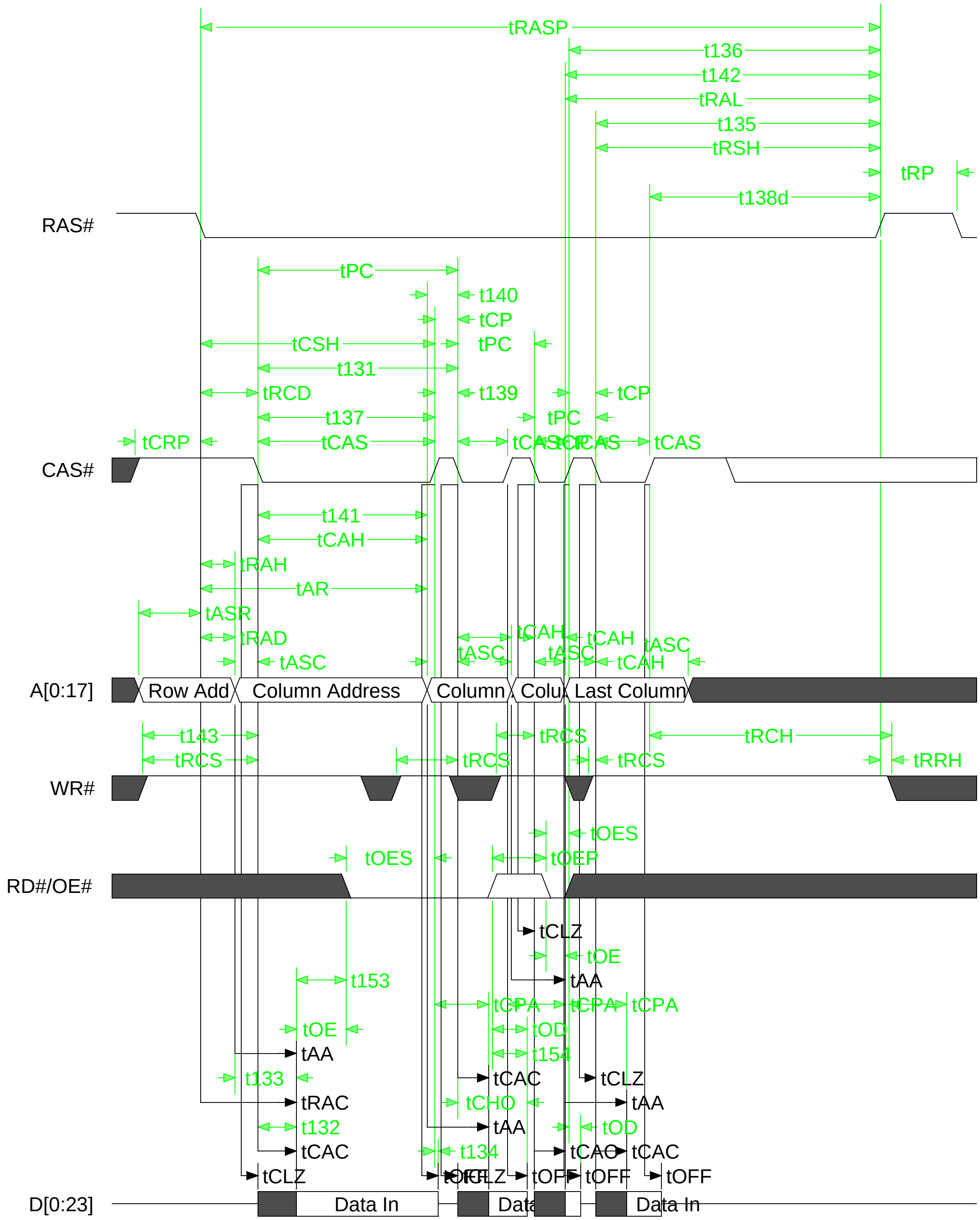


Figure 2-16 DRAM Page Mode Read Access
Motorola DSP56303/D

Figure 18. EDO DRAM to DSP56303 Page Mode Read Interface Diagram

4.3 Analyzing the DSP-to-DRAM Interface

This analysis is similar to the SRAM analysis. While the parameter table for the DRAM interface is larger, due to the number of timings specified in the interface, the order of the information is similar to that in the SRAM parameter table. For this analysis, reuse the SRAM parameter window format:

- Device descriptions (two rows)
- Reference timings source (two rows)
- Interface to be analyzed, in this case, a read cycle (one row)
- Path for the electronic libraries (rows as necessary)
- Definitions of constants the libraries require (rows as necessary)
- Other information that may be helpful (rows as necessary)
- Constraint information (rows as necessary)
- Delay information (rows as necessary)

While SRAM had one read cycle diagram and one write cycle diagram, DRAM has both in-page and out-of-page read and write diagrams, as well as a refresh diagram. The ability to swap libraries without modifying diagrams is an especially efficient way to analyze such complex interfaces.

Having completed the diagram, analyze it for timing violations (for a description of the procedure, refer to *Finding and Correcting Timing Violations* on page 17). Before documenting violations, be sure to check the library values and related timings. If you cannot resolve a violation, substitute a different DRAM library and check it again. When you find an acceptable DRAM, your analysis is complete.

5 Implications for Design

Motorola timing analysis enables quick analysis of data sheet timing interfaces, easy consideration of different devices, and rapid, accurate system design verification. Due to the ease of switching between device choices, a system design may be verified quickly and accurately. However, analyzing the device-to-device interface is only a start. Thorough system design also includes board delays and any other buffer timing issues that an application may require. Adding these timings to the system model increases design accuracy. The following example, together with **Table 6**, demonstrates how to add board delay timing to the DSP-SRAM interface.

Condition: The address bus in a system design has a 1-ns board delay, due to capacitive loading, trace lengths, or other issues.

Design modification:

1. Create a variable called BD_delay (row 15 in **Table 6**) and give it a value of 1 ns.
2. Add this delay to the formula for memory timings that use the address bus as a source.
3. Check the diagram for violations. Notice that timings 104 and 105 are now in violation.
4. Bring the interface back into compliance by shortening trace lengths, lowering capacitive loading, or make other system adjustments.

Table 6. Adding Board Delay Information

Row		Name	Formula	Min	Max	Margin	Comment
1	V	DSP_Number	$DSP:Part_Number				Motorola DSP56303 24-Bit General Purpose Digital Signal Processor
2	V	MEMORY_Number	$memory:Part_Number				Motorola MCM6341 128K x 24 bit Static Random Access Memory
3	V	Data_Reference1	$DSP:Data_Reference				Motorola DSP56303/D Rev. 2 - Figure 2B-12 page 2B-20
4	V	Data_Reference2	$memory:Data_Reference				Motorola MCM6341/D Rev.4 - Page 6
5	V	Mode_Description	[,]				Port A Read Cycle accessing 128Kx24 FSRAM
6	V	Aliases	[,]				
7	A	$DSP	:dsp56303cdr: library:porta:sram:$ws:$freq				
8	A	$memory	:mcm6341:rev4_lib:rd_cyc				
9	A	$freq	100				
10	A	$ws	1ws				
11	V	Constants	[,]				
12	V	freq	100	100	100		
13	V	TC	((1/freq)*1000)	10	10		
14	V	WS	1	1	1		
15	V	**BD_delay**	1	1	1		
16	V	duty.cycle	50	50	50		
17	V	rising.jitter	0	0	0		
18	V	falling.jitter	0	0	0		
19	V	Input_Constraints	[,]				
20	C	tAVAV	$memory:TAVAV	10		<7,>	Read Cycle Time
21	C	t100	$DSP:t100(WS,TC)	16		<1,>	Address Valid
22	C	t104	[,10.5]		10.5	**<,-0.5>**	Address and AA valid to input data valid
23	C	t105	[,5.5]		5.5	**<,-0.5>**	#RD assertion to input data valid
24	C	t106	$DSP:t106	0		<0,>	RD deassertion to data not valid (data hold time)
25	V	Output_Delays	[,]				
26	D	TAVQV	($memory:TAVQV+**BD_delay**)		11		Address Access Time

Table 6. Adding Board Delay Information (Continued)

Row		Name	Formula	Min	Max	Margin	Comment
27	D	TELQV	($memory: TELQV+**BD_delay**)		11		Enable Access Time
28	D	TGLQV	($memory: TGLQV+**BD_delay**)		5		Output Enable Access Time
29	D	tELQX	$memory:TELQX	3			Enable Low to Output Active
30	D	tGLQX	$memory:TGLQX	0			Output Enable Low to Output Active
31	D	TEHQZ	$memory:TEHQZ	0	5		Enable High to Output High-Z
32	D	TGHQZ	$memory:TGHQZ	0	5		Output Enable High to Output High-Z
33	D	t117	[0.5,]	0.5			RD deassertion to address not valid

6 Conclusion

As system speeds and interface complexity increase, in-depth timing analysis becomes critical. With software tools like TimingDesigner, engineers can swap timing libraries to model different parts without wasting time redrawing or rechecking signals. The tool calculates new timings instantly. The engineer can quickly determine whether the design has potential timing violations. Motorola hopes that the timing analysis files and the information in this application note help our customers to prove designs more efficiently and get to market faster.

7 References

This application note refers to the following resources:

- DSP56303 technical data sheet, DSP56303/D
- MCM6341 technical data sheet, MCM6341/D
- KM416V1200C Component Specifications
- TimingDesigner Help Utility

You can download the Motorola documents from the Web at the following URL:

```
http://www.mot.com/SPS/DSP
```

Notes:

Notes:

Order Number: AN1834/D

Rev. 0, 12/1999

OnCE is a registered trademark and Mfax is a trademark of Motorola, Inc.
TimingDesigner is a registered trademark and TimingViewer is a trademark of Chronology Corporation.
UNIX is a registered trademark of The Open Group.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:

USA/Europe/Locations Not Listed:

Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1 (800) 441-2447
1 (303) 675-2140

Motorola Fax Back System (Mfax™):
RMFAX0@email.sps.mot.com
TOUCHTONE (602) 244-6609
USA and Canada ONLY:
1-800-774-184
http://sps.motorola.com/mfax/

Asia/Pacific:
Motorola Semiconductors H.K. Ltd.
8B Tai Ping Industrial Park
51 Ting Kok Road
Tai Po, N.T., Hong Kong
852-26629298

Technical Resource Center:
1 (800) 521-6274

DSP Helpline
dsphelp@dsp.sps.mot.com

Japan:
Nippon Motorola Ltd.
SPD, Strategic Planning Office141
4-32-1, Nishi-Gotanda
3-14-2 Tatsumi Koto-Ku
Shinagawa-ku, Tokyo, Japan
81-3-5487-8488

Customer Focus Center:
1-800-521-6274

Internet:
http://www.mot.com/SPS/DSP/

MOTOROLA