Designer Note Page
SPRA404

Proper Usage of TMS320C5x Timing Specifications for High-Speed Memory Interface

Jon Bradley *Digital Signal Processing*

This designer note page describes how to properly interpret the Texas Instruments (TI™) TMS320C5x timing specifications to design an interface to memory and to other external devices. Designing a high-speed interface between the TMS320C5x digital signal processor (DSP) and memory or other parallel devices poses interesting circuit design considerations. The 'C5x device data sheet provides a wide variety of timing specifications for various aspects of device operation. Proper interpretation of these timing specifications is critical to ensure that timing requirements are met over all conditions of read and write operations without excessively constraining the design to a higher speed interface than necessary.

Contents

Figures

Design Problem

How do I properly interpret the TMS320C5x timing specifications to design an interface to memory or other external devices?

Solution

Designing a high-speed interface between the TMS320C5x DSP and memory or other parallel devices poses some interesting circuit design considerations. The ‘C5x device data sheet provides a wide variety of timing specifications for various aspects of device operation. Proper interpretation of these timing specifications is critical in order to ensure that timing requirements are met over all conditions of read and write operations without excessively constraining the design to a higher speed interface than necessary.

The key timings involved in the ‘C5x external parallel interface are those for the address and data busses, the control outputs CLKOUT1, $\overline{\mathrm{STRB}}$, $\overline{\mathrm{RD}}$, $\overline{\mathrm{WE}}$, and $\mathrm{R}/\overline{\mathrm{W}}$, and the ready input. When designing an interface, timing requirements for data and ready inputs must be satisfied based on the address and control outputs. Since some switching characteristics and timing requirements are specified with respect to only one or two key control signals, interface design often requires establishment of timings between any two arbitrary interface signals. This is usually accomplished by first considering the control signal timings.

Although timing between the control outputs does exhibit measurable skew, transitions on these outputs are in fact generated by the same internal clock signals, and therefore track each other extremely closely. Accordingly, for the purposes of interface, these signals are considered to be coincident, and this is accounted for in the guardbanding included in the device timing specifications. Therefore, timings referenced to equivalent edges of one of these signals are considered to be equivalent. For example, in a read operation, the required setup time of data before $\overline{\mathrm{RD}}$ going high can also be applied to the falling edge of CLKOUT1 resulting in an equivalent requirement of data setup prior to CLKOUT1 falling edge.

More specifically, for a read, the falling edges of CLKOUT1 are considered to be coincident with the rising and falling edges of $\overline{\mathrm{STRB}}$, and the rising edge of $\overline{\mathrm{RD}}$. Also during a read, the rising edge of CLKOUT1 is considered to be coincident with the falling edge of $\overline{\mathrm{RD}}$.

For a write, the falling edges of CLKOUT1 are considered to be coincident with the rising and falling edges of $\overline{\mathrm{STRB}}$ and $\overline{\mathrm{WE}}$. Taking these equivalencies into account greatly eases external interface design.

An explicit demonstration of this aspect of interface timing interpretation can be observed directly by considering the READY signal timings as presented in the ‘C5x data sheet (TI document #SPRS030). Note that READY timing requirements are specified as referenced both to CLKOUT1 and to $\overline{\mathrm{RD}}$. READY timings must be specified in this fashion since when beginning a series of wait states, $\overline{\mathrm{RD}}$ remains low, so using this signal with which to reference READY timings in a design is not possible. The READY timings with respect to CLKOUT1 are required as a reference with which to control READY to complete the cycle. The fact that illustrates the above discussion of timing specification equivalencies is that the READY timings on the data sheet with respect to $\overline{\mathrm{RD}}$ are identical to the READY timings with respect to CLKOUT1. (Recall that it was stated that edges of $\overline{\mathrm{RD}}$ are considered to be equivalent to edges of CLKOUT1.) Numerous other examples of this can be found in the ‘C5x device timing specifications.

Note that the timing interpretations discussed here apply only to ‘C5x devices, and not to all TMS320 family members. Interface to other TMS320 family members must be considered independently in their own light.

Data Timings

When designing an interface to ‘C5x devices, one of the main considerations is the proper approach to determining required data timings when performing a read operation. Data valid timings, as well as several other interface timings, may be derived using more than one approach. Not all approaches to deriving an interface timing are valid; some may result in improper timings, while others can result in an interface which is excessively constrained, and causes an interface to be unnecessarily over-designed (i.e., using higher speed components than necessary).

In general, there are potentially three ways to determine interface timing between two signals. If the timing of interest is specified explicitly, this is the most accurate specification of the timing, and deriving the timing through any other method generally produces an excessively constraining specification, and is therefore not recommended.

The next most accurate approach is to derive the required timing by referencing a timing which is specified with respect to an equivalent control signal. Since, as described above, there are several equivalent control signals on the ‘C5x device, using this approach generally produces most additional timing specifications that are required, but that are not explicitly specified.

The third possible approach to deriving timing specifications is to add and subtract specified timings to calculate the required timing as the sum or difference of several of the existing device timings. In this case, however, maximum and minimum values, which are worst-cased in the opposite directions, can potentially cause the interface to be excessively conservative. This method is not as accurate as the other two approaches and should be used only if no other approach is possible.

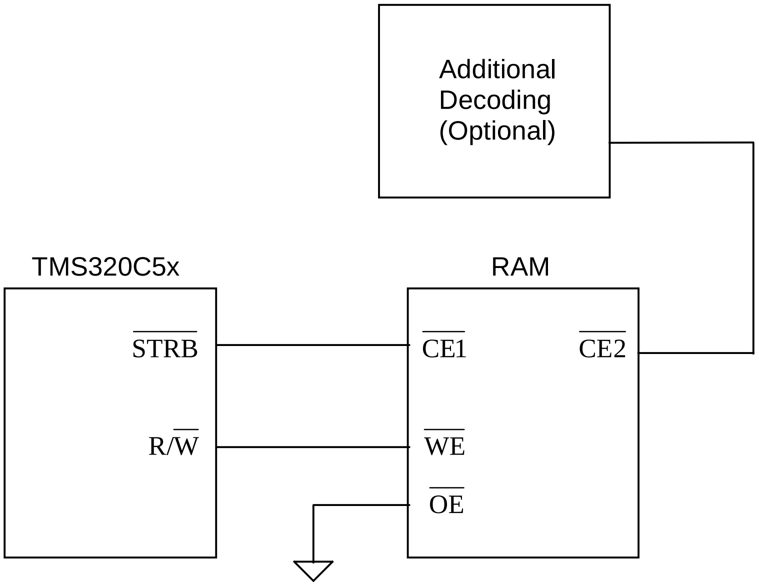
In order to illustrate the potential pitfalls of this approach, consider attempting to determine the read data access time from address by adding tsu(AV-RDL) and ta(RDL-RD). The net result of this sum should yield the data access time from address, but the summation produces 2H-12, rather than 2H-10, which is the actual ta(RDAV) specification. Therefore, if the calculated time (rather than the explicitly specified time) is used to design an interface, memories with a 2 ns faster access time than necessary might be chosen. As mentioned above, making this calculation is unnecessarily constraining since the signals involved track each other and all of the specified device maxima and minima do not occur at the same time on the same device. Accordingly, this approach to deriving interface timings should be used only if no other approach is possible.

Since read data timings on the ‘C5x device are specified directly in three different ways, interface design is simplified. The three read data valid timing requirements are: read data access time after address (ta(RDAV)), data access time after $\overline{\text{RD}}$ falling (ta(RDL-RD)), and data setup time prior to $\overline{\text{RD}}$ going high (tsu(RD-RDH)). As described above, since data timings are specified explicitly, attempting to derive data timings as the sum or difference of other specified timings should not be done. It should also be noted that since each of these timings is tested explicitly and independently on each device during the manufacturing process, only one out of the three timings need be met. Therefore, if any one of these timings is met, although the other timings may appear not to be met, the interface still functions properly.

SRAM Interface

As device clock rates increase, interface at high speed requires increasingly more sophisticated interface design, due to subtleties in timing requirements. If accessing only one bank of SRAM is required at zero wait states, the approach presented in Designer’s Notebook Page (DNP) Number 45 can be used, and the access time required for the memory used is dictated simply by the ‘C5x address access time (ta(RDAV)).

Figure 1. DNP 45 Memory Interface Approach



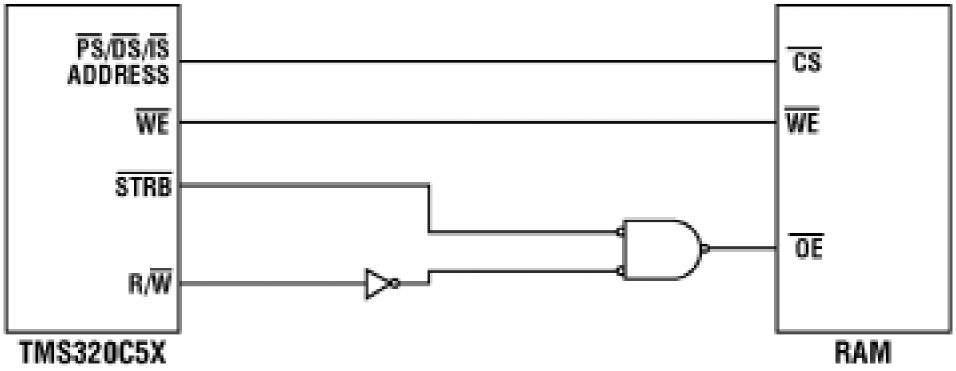
Note that in the circuit shown, if the additional decoding is used on the second chip select input, this signal must be static during RAM accesses or the delay presented by this logic fundamentally impacts the RAM access time as perceived by the ‘C5x, since this delay is directly in series with the access. Accordingly, if full speed with zero wait state qualification of $\overline{\mathrm{CS}}$ is required, then DNP 45 does not provide a glueless interface. Also, if the RAMs used do not implement an additional chip select input, then any device select decoding must be directly in line with the device therefore also additionally constrains the access time (faster memory is required).

In order to design an interface to take fullest advantage of RAM access time, while still maintaining capability to be selected at full speed with zero wait states on the basis of an address bit or an address-related signal such as $\overline{\mathrm{PS}}/\overline{\mathrm{DS}}/\overline{\mathrm{IS}}$ without negatively impacting access time, a different approach to the interface is required.

To do this requires three things: allowing an ungated address bit or address-related signal to drive RAM $\overline{\mathrm{CS}}$, providing an appropriate $\overline{\mathrm{WE}}$ signal, and also ensuring that $\overline{\mathrm{OE}}$ is properly controlled. Since using an address or space select signal for $\overline{\mathrm{CS}}$ does not allow use of the $\overline{\mathrm{CS}}$ controlled write cycle as used in DNP 45, the ‘C5x $\overline{\mathrm{WE}}$ signal must be used for the RAM $\overline{\mathrm{WE}}$, and an appropriate $\overline{\mathrm{OE}}$ control signal must be generated. Proper implementation of the output enable function is critical since otherwise data bus conflicts can arise due to overlap between read and write cycles when the ‘C5x and the memory device are driving the data bus at the same time, which is not allowable.

In the DNP 45 circuit, output enable control is provided because the RAM $\overline{\mathrm{CS}}$ and $\overline{\mathrm{WE}}$ inputs both gate output enable, and therefore, even though output enable is grounded, outputs are only enabled when the RAM is selected and $\mathrm{R}/\overline{\mathrm{W}}$ is high. However the circuit does not allow any full speed qualification of the RAM $\overline{\mathrm{CS}}$ without degrading the access time. Additionally, for some memories whose access time allows their use with the ‘C5x, output enable timing specifications do not allow $\overline{\mathrm{RD}}$ to be used for $\overline{\mathrm{OE}}$ function. For example, the 5 ns output enable time of a 10 ns SRAM is not fast enough to meet the device timing requirement of H-6 or 4 ns with a 50 MHz CLKOUT1 rate.

Figure 2. Improved Circuit Diagram


$\overline{\mathrm{PS}}/\overline{\mathrm{DS}}/\overline{\mathrm{IS}}$
ADDRESS
$\overline{\mathrm{WE}}$
$\overline{\mathrm{STRB}}$
$\mathrm{R}/\overline{\mathrm{W}}$
TMS320C5X
$\overline{\mathrm{CS}}$
$\overline{\mathrm{WE}}$
$\overline{\mathrm{OE}}$
RAM

To properly control output enable, a $\overline{\mathrm{STRB}}$-related signal is generated which occurs only when $\mathrm{R}/\overline{\mathrm{W}}$ is high, therefore implementing the equivalent of the $\overline{\mathrm{WE}}$ signal which occurs for read cycles. This signal is then used to drive the RAM $\overline{\mathrm{OE}}$ input. Although external logic is required, since this logic is not in direct line with the RAM access time, this logic need not be particularly fast, and therefore can be implemented in an inexpensive PAL-type device, or in discrete logic. Therefore, since the $\overline{\mathrm{OE}}$ control is generated significantly earlier than if $\overline{\mathrm{RD}}$ were used for this function, much more timing overhead is provided, therefore allowing this interface to easily meet ‘C5x device timing requirements with RAMs with access times up to the maximum ta(RDAV) specification.

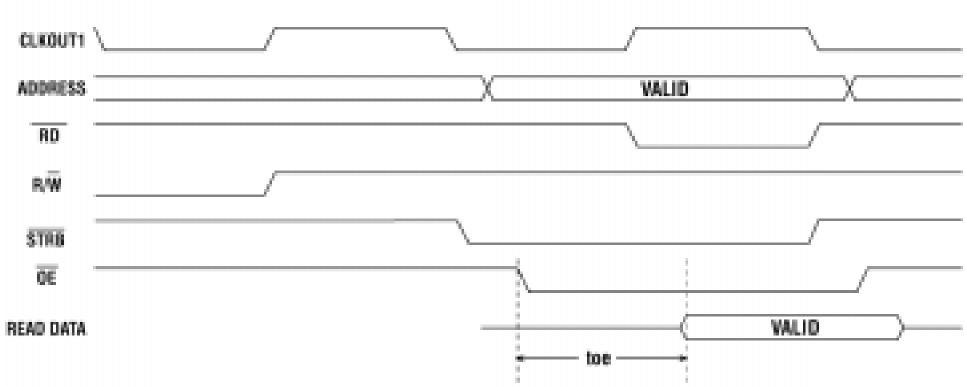
Using this approach, RAM $\overline{\mathrm{CS}}$ input can be driven by an address-related signal and still maintain access time requirements. This allows the RAM to be able to be distinguished from other devices in the system without requiring any decoding logic to be used in the path with the chip select input and therefore impacting access time. However, note that depending on what other devices may be present in the system, if the RAM is accessed contiguously with other devices, data bus conflicts may occur if the other devices are either too fast or too slow in getting on or off of the data bus. Therefore, timing of other devices in the system must be compared with RAM timings in order to ensure that bus conflicts do not occur. If bus conflicts with other devices in the system can occur, this can be avoided either by examining and adjusting the sequence of instructions in the code to ensure that there are no contiguous accesses at bank boundaries, or by executing a short segment of code internally before accessing the next device. Other techniques can be applied in hardware to avoid bus conflicts, but these techniques can be more costly, and are somewhat beyond the scope of this DNP.

Applying the concepts described above to design an actual memory interface, consider the example of connecting a ‘C5x-100 device to an ISSI IS61C256 32Kx8 SRAM, with an address/chip select access time of 10 ns and an output enable time of 5 ns.

For a read operation, before the cycle actually starts, $\mathrm{R}/\overline{\mathrm{W}}$ is already high (and inverted $\mathrm{R}/\overline{\mathrm{W}}$ is low) preparing $\overline{\mathrm{OE}}$ to go low after $\overline{\mathrm{STRB}}$ goes low [see Figure 3]. Therefore, as the cycle begins, RAM $\overline{\mathrm{CS}}$ and $\overline{\mathrm{OE}}$ both go low at approximately the same time (depending on the OR gate propagation delay) and the read access time is determined by the $\overline{\mathrm{CS}}$/address access time, since the output enable time is much faster.

Specifically, since the falling edges of CLKOUT1 are considered to be coincident with the rising and falling edges of $\overline{\mathrm{STRB}}$ and the rising edge of $\overline{\mathrm{RD}}$, the maximum allowable delay for the OR gate can be calculated as 2H-(tsu(RD-RDH)+tOE(RAM)). This yields 9 ns. Therefore, for this interface, with a 20 ns ‘C5x cycle time, the 10 ns RAM address access time and the 5 ns output enable time, any gate faster than 9 ns satisfies the ‘C5x requirements for proper read operation.

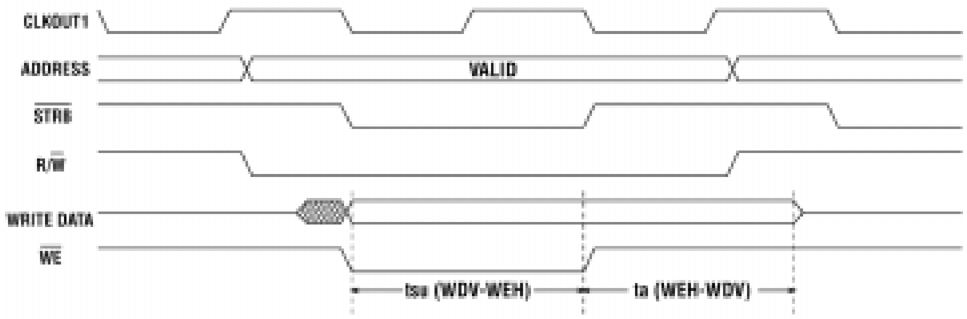
Figure 3. Read Timing Diagram



For a write operation [see Figure 4], all interface timings are also easily met. Since the RAM $\overline{OE}$ is only asserted when $\overline{STRB}$ is low and $R/\overline{W}$ is high, the RAMs outputs are disabled almost a full CLKOUT1 cycle of 20 ns before the write cycle starts. This provides more than adequate assurance that the 'C5x and the RAM will not be driving the data bus at the same time at the beginning of the cycle. The RAM minimum write data setup time of 7 ns is met by the 'C5x tsu(WDV-WEH) of 2H-12 or 8 ns.

The write data hold time with respect to $\overline{WE}$ rising edge (th(WEH-WDV) provided by the 'C5x of H-4 to H+7 or 3 to 17 ns easily meets the data hold time for this and most other RAMs requirements of 0 ns. Since the maximum 'C5x data hold time is also the effective maximum bus drive time, this interface provides reasonable margin before a RAM $\overline{OE}$ can be generated following a write operation. This guarantees that the 'C5x and the RAM will not be driving the data bus at the same time at the end of the write cycle. Other RAM write timing requirements are also easily met by this interface.

Figure 4. Write Timing Diagram



INTERNET

www.ti.com

Register with TI&ME to build custom information pages and receive new product updates automatically via email.

TI Semiconductor Home Page
http://www.ti.com/sc

TI Distributors
http://www.ti.com/sc/docs/distmenu.htm

PRODUCT INFORMATION CENTERS

US TMS320

Hotline	(281) 274-2320
Fax	(281) 274-2324
BBS	(281) 274-2323
email	dsph@ti.com

Americas

Phone	+1(972) 644-5580
Fax	+1(972) 480-7800
Email	sc-infomaster@ti.com

Europe, Middle East, and Africa

Phone	
Deutsch	+49-(0) 8161 80 3311
English	+44-(0) 1604 66 3399
Francais	+33-(0) 1-30 70 11 64
Italiano	+33-(0) 1-30 70 11 67
Fax	+33-(0) 1-30-70 10 32
Email	epic@ti.com

Japan

Phone	
International	+81-3-3457-0972
Domestic	+0120-81-0026
Fax	
International	+81-3-3457-1259
Domestic	+0120-81-0036
Email	pic-japan@ti.com

Asia

Phone	
International	+886-2-3786800
Domestic	
Australia	1-800-881-011

Asia (continued)

TI Number	-800-800-1450
China	10811
TI Number	-800-800-1450
Hong Kong	800-96-1111
TI Number	-800-800-1450
India	000-117
TI Number	-800-800-1450
Indonesia	001-801-10
TI Number	-800-800-1450
Korea	080-551-2804
Malaysia	1-800-800-011
TI Number	-800-800-1450
New Zealand	+000-911
TI Number	-800-800-1450
Philippines	105-11
TI Number	-800-800-1450
Singapore	800-0111-111
TI Number	-800-800-1450
Taiwan	080-006800
Thailand	0019-991-1111
TI Number	-800-800-1450

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.

TI is a trademark of Texas Instruments Incorporated.
Other brands and names are the property of their respective owners.