Hardware Interfacing to the TMS320C2x

APPLICATION REPORT: SPRA014B

George Troullinos
John Bradley
Digital Signal Processor Products
Semiconductor Group
Texas Instruments

Digital Signal Processing Solutions

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.

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.

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

Hardware Interfacing to the TMS320C2x

Abstract

This chapter suggests hardware design techniques for interfacing memories and peripherals to the TMS320C2x and notes appropriate differences between the TMS320C2x and TMS32020.

The first section discusses ready generation techniques and describes interfaces to the TMS320C2x for

- PROMs
- EPROMs
- SRAM (Static RAMs)

The second section describes hardware interfaces to the TMS320C2x to the following converters:

- Combo-codec (PCM coder-decoder)
- Analog-to-digital converter
- Digital-to-analog converter

The chapter includes appropriate timing diagrams and interface line drawings and considers both direct interfaces and those that utilize address decoding.

Product Support

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

Each member of the TMS320 Second-Generation Digital Signal Processors family has the power and flexibility to satisfy a wide range of system requirements. The second-generation TMS320 line includes the TMS32020, TMS320C25, TMS320C25-50, TMS320E25, and TMS320C26. Please refer to the Second-Generation TMS320 User's Guide[1] for details on device-to-device variation.

All TMS320 second-generation DSPs are pin-compatible and thus have the same set of external interface signals. For convenience, the following notation will be used throughout this report: Second-generation TMS320 devices refer to all members of this family, TMS320C2x refers to all members of the second-generation family except the TMS32020 (i.e., TMS320C25, TMS320C25-50, TMS320E25, and TMS320C26). In other TI literature, TMS320C2x normally refers to the entire second-generation family. This report will focus on TMS320C2x hardware interfacing.

All second-generation TMS320 devices can address 64K 16-bit words in data space, 64K words in program space, and 16 16-bit wide I/O ports. The 128K-word address space for program and data memory can be utilized in applications that require large amounts of memory by interfacing external memories using the control signals of second-generation TMS320 devices. In other applications, the internal program and data resources of second-generation TMS320 devices can be used to implement single-chip solutions. Peripheral devices can be interfaced to second-generation TMS320 devices to perform analog signal acquisition at different levels of signal quality.

This report suggests hardware design techniques for interfacing memories and peripherals to the TMS320C2x. Differences between the TMS320C2x and the TMS32020 are pointed out when appropriate. The first section presents the design interfaces of PROMs, EPROMs, and static RAMs (SRAM) to the TMS320C2x. Timing requirements of the processor and external memories are considered. The second section discusses the interface of a combo-codec (PCM coder-decoder), an analog-to-digital converter, and a digital-to-analog converter to the TMS320C2x. All interfaces in this report have been built and tested to verify their operation.

Ready Generation Techniques

This section describes techniques for generating the READY input signal for the TMS320C2x. READY can be used to extend external bus cycles by an integer number of machine cycles. The READY input thereby provides a means of interfacing the TMS320C2x to external devices that cannot be accessed at full speed, such as memory devices having access times longer than those required by the TMS320C2x.

The access time (t_a) of a given device determines the number of dormant cycles (wait-states) required for each access of that device. In general, N wait-states are required for a particular access if

$$[\, t_{c(C)} * (N-1) + t_{a(A)} \,] < t_a < [\, t_{c(C)} * N + t_{a(A)} \,] \,, \; N > 0$$

where $t_{c(C)}$ is the period of CLKOUT1/2 (the reciprocal of the machine rate) and $t_{a(A)}$ is the access time from address specified in the appropriate second-generation TMS320 device electrical specification, Table 1 gives appropriate values of N for several ranges of t_a for a TMS320C25 operating

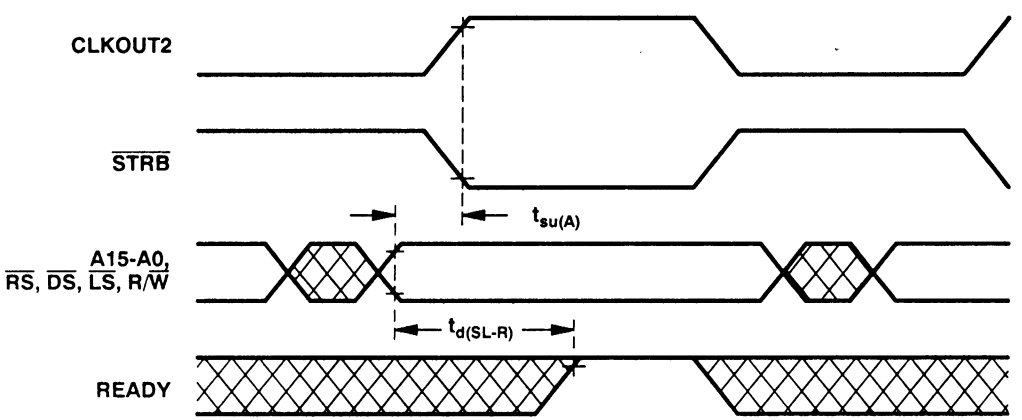
with a 100 ns instruction cycle time and a TMS320C25-50 operating with a 80 ns instruction cycle time.

Table 1. Number of Wait-States Required for a Memory or Peripheral Access

TMS320C25		TMS320C25-50	
Access Time	Number of Wait States Required	Access Time	Number of Wait States Required
t_a < 40 ns	0	ta < 29 ns	0
40 ns < t_a < 140 ns	1	29 ns < t_a < 109 ns	1
140 ns < t_a < 240 ns	2	109 ns < t_a < 189 ns	2
240 ns < t_a < 340 ns	3	189 ns < t_a < 269 ns	3
340 ns < t_a < 440 ns	4	269 ns < t_a < 349 ns	4

The timing requirements for generation of the READY signal are specified in the TMS320C25 electrical specifications by $t_{su(A)}$ and $t_{d(SL-R)}$ or $t_{d(C2H-R)}$.

Figure 1. Ready Timing Requirement



READY (see Figure 1) must be valid no later than $t_{su(A)} + t_{d(SL-R)}$ after the address bus and interface control signals (except $\overline{STRB}$) become valid. This evaluates to

$$t_{su(A)} + t_{d(SL-R)} = (Q-11) + (Q-20) = 9 \text{ ns}$$

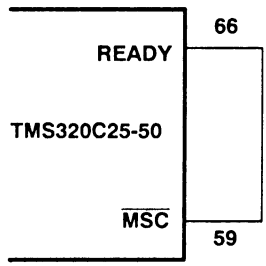
for a TMS320C25-50 operating with an input clock frequency of 50.0 MHz, and

$$t_{su(A)} + t_{d(SL-R)} = (Q-12) + (Q-20) = 18 \text{ ns}$$

for a TMS320C25 operating with an input clock frequency of 40.0 MHz. Note that for bus cycles with wait-states, CLKOUT2 serves as the timing reference, whereas for no-wait cycles either $\overline{STRB}$ or CLKOUT2 can be used as the timing reference. Any skew between these two signals may be disregarded as $t_{d(SL-R)}$ and $t_{d(C2H-R)}$ are guaranteed independently.

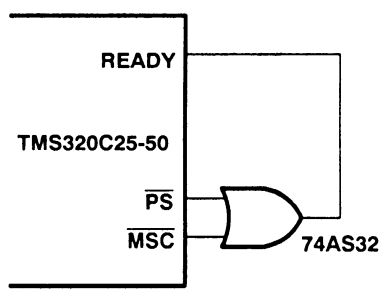
If all external bus cycles are to occur with no wait-states, READY can simply be tied high with a pull-up resistor. Extending all external bus cycles with one wait-state can easily be accomplished by connecting the $\overline{MSC}$ output to READY as shown in Figure 2.

Figure 2. Connection for One Wait-State External Accesses



Similarly, $\overline{MSC}$ and the $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ signals can be used to generate wait-state mixes such as that resulting from the circuit in Figure 3. With this circuit, all program space accesses are one wait-state accesses while all data space and I/O accesses occur at full speed.

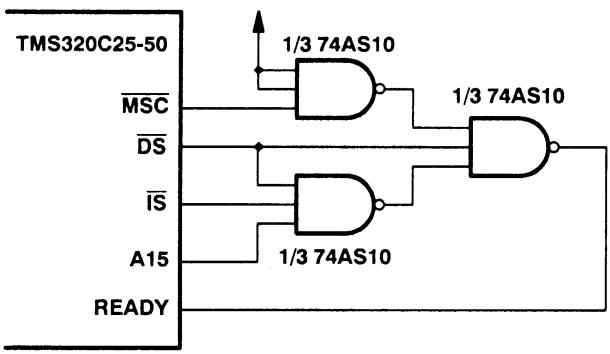
Figure 3. Ready Generation for One Wait-State Program Space Accesses



Applications having sufficiently simple address partitioning can make use of one or more levels of standard logic gates to generate READY. The circuit shown in Figure 4 has the following wait-state map:

External Space	Address Range	Number of Wait-States
Program	0000h–7FFFh	1
Program	8000h–FFFFh	0
Data	0000h–FFFFh	0
I/O	0000h–000Fh	1

Figure 4. Ready Generator with Simple Address Partitioning



Note that this circuit just meets the READY specification of the TMS320C25-50 with READY guaranteed valid no later than 9 ns from address valid.

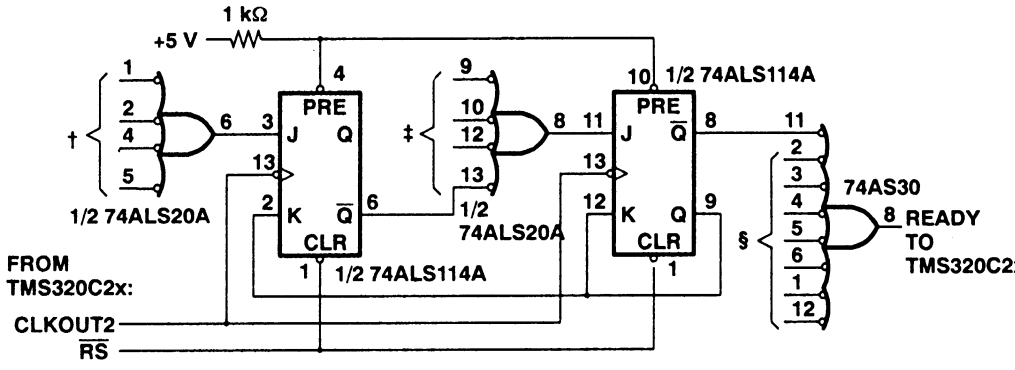
TMS320C25-50 applications requiring more extensive address decoding will in most cases require the use of a high-speed programmable logic device to generate READY sufficiently fast. Two such devices are listed in Table 2.

Table 2. High-Speed Programmable Logic Devices

Manufacturer	Part Number	t_{pd} (ns)
TI	TIBPAL16L8-7	7.5
AMD	PAL16L8-7	7.5

The wait-state generator shown in Figure 5 can be used to generate the READY signal for a TMS320C25 interfaced to external devices requiring up to 2 wait-states. A timing diagram for this circuit is shown in Figure 6.

Figure 5. Two Wait-State Generator Design

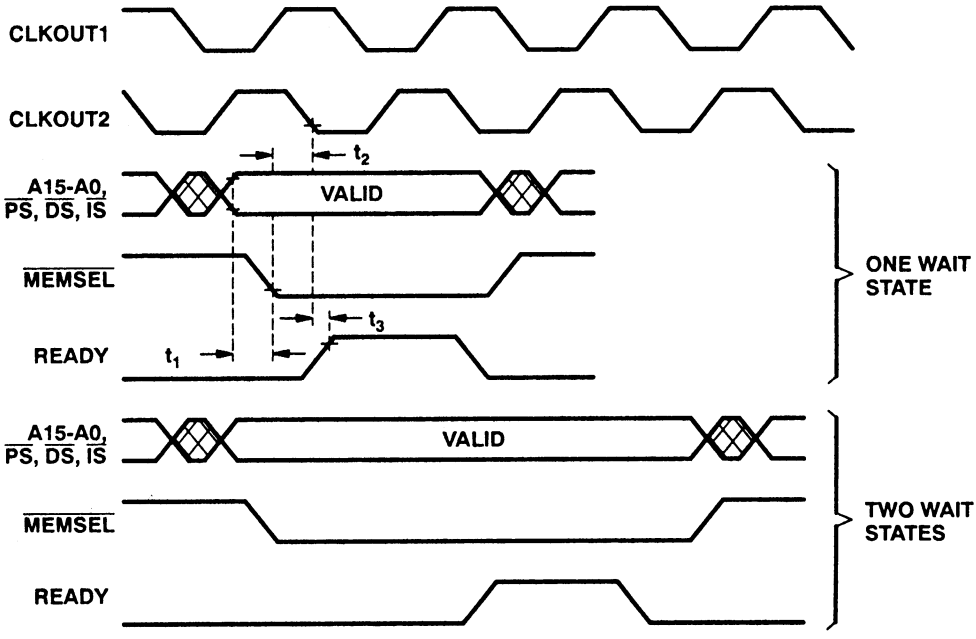


† Connections to other devices in the system that require two wait states. (Inputs not used by other devices should be pulled up.)

‡ Connections to other devices in the system that require one wait state. (Inputs not used by other devices should be pulled up.)

§ Connections to other devices in the system that require zero wait states. (Inputs not used by other devices should be pulled up.)

Figure 6. Timing Diagram for Two Wait-State Generator Design



With this arrangement, READY is driven by a multiple-input NAND gate. This can be a standard gate such as a 74AS30 or can be part of the logic implemented by a high-speed programmable logic device. The output of this gate is low unless at least one of the inputs is low. The propagation delay of READY decode logic selecting zero wait-state devices in addition to the NAND delay

must be short enough to satisfy the READY specification discussed above. For zero wait-state accesses, the flip-flop J inputs are low, the $\overline{Q}$ outputs are high and neither flip-flop switches state.

Now consider the circuit operation when a one or two wait-state device is selected. The $\overline{Q}$ output of each JK flip-flop is high at the start of the access, which can be considered to begin with the falling edge of CLKOUT2. All the inputs to the NAND gate generating READY are high and thus READY is low during the first cycle and the TMS320C25 inserts one wait-state. If a one wait-state device is decoded, the J input of the first flip-flop goes high. The Q output goes low on the next falling edge of CLKOUT2 and READY goes high.

If a two wait-state device is decoded, the J input of the second flip-flop goes high. Two cycles are required for this signal to propagate to the READY line. For each cycle, one wait-state is inserted.

Referring to Figure 6, the following two inequalities must be satisfied in order for the setup time specification of the flip-flops to be met:

1) $t_{(decode)} + t_{(NAND)} + t_{su(74ALS114A)} < t_{su(A)} + 2Q$
2) $t_{p(74ALS114A)} + t_{(NAND)} + t_{su(74ALS114A)} < 4Q$

where $t_{(decode)}$ is the propagation delay of the decode logic for the selected device, $t_{(NAND)}$ is the delay associated with the NAND gate at the flip-flop input, $t_{su(74ALS114A)}$ and $t_{p(74ALS114A)}$ are the data setup time and prop delay of the 74ALS114A, respectively, and $Q = 1/4t_{c(C)}$. In Figure 6,

$t_1 = t_{(decode)}$
$t_2 = t_{(NAND)} + t_{su(74ALS114A)}$ and
$t_3 = t_{p(74ALS114A)} + t_{(NAND)}$.

A third inequality must be satisfied for the READY specification to be met:

3) $t_{p(74ALS114A)} + t_{(NAND)} < t_{d(C2H-R)} + 2Q$

For a TMS320C25-50 operating at 50 MHz, inequality (1) evaluates to

1) $t_{(decode)} + 5\text{ ns} + 22\text{ ns} < 9\text{ ns} + 40\text{ ns}$

or

$t_{(decode)} < 22\text{ ns}$

This inequality specifies the maximum decode time in order for the setup time specification of the pertinent flip-flop to be met.

The remaining two inequalities are satisfied:

2) $19\text{ns} + 5\text{ns} + 22\text{ns} < 80\text{ns}$
3) $19\text{ns} + 5\text{ns} < 0\text{ns} + 40\text{ns}$

All three of these inequalities should be considered if different flip-flops and/or gates are used to implement the wait-state generator.

Note that special considerations should be made with respect to READY timing if the TI Extended Development Support (XDS) in-circuit emulator is used. Please refer to *TMS320 Second-Generation User's Guide*[1] and/or *Extended Development Support Products User's Guide* (literature number SPYF001) for further details on READY timing requirements.

Interfacing Memories to the TMS320C25

This section describes interfaces of external memory devices to the 40 MHz speed version of the TMS320C25. Interfaces to PROMS, EPROMs, and SRAMS are included. A separate section is included in this document to describe memory interfaces to the TMS320C25-50.

The TMS320C2x offers 544 words of RAM and 4K words of masked ROM. For prototyping and/or system expansion, however, external memories may be required. The speed, cost, and power limitations imposed by a particular application determine the selection of a specific memory device. If speed and maximum throughput are desired, the TMS320C2x can run with no wait-states. In this case, memory accesses are performed in a single machine cycle. Alternatively, slower memories can be accessed by introducing an appropriate number of wait-states or by slowing down the system clock. The latter approach is more appropriate when interfacing to memories with access times slightly longer than those required by the TMS320C2x at full speed.

When wait-states are required, the number of wait-states depends on the memory access time (see Table 1 on page 2). With no wait-states, the READY input to the TMS320C2x can be pulled high. If one or more wait-states are required, the READY input must be driven low during the cycles in which the TMS320C2x enters a wait-state.

The TMS320C2x implements two separate and distinct memory spaces: program space (64K words) and data space (64K words). Distinction between the two spaces is made through the use of the PS (program space) and DS (data space) pins. A third space, the I/O space, is also available for interfacing with peripherals. This space is selected by the IS (I/O space) pin, and is discussed in the Interfacing Peripherals section of this report.

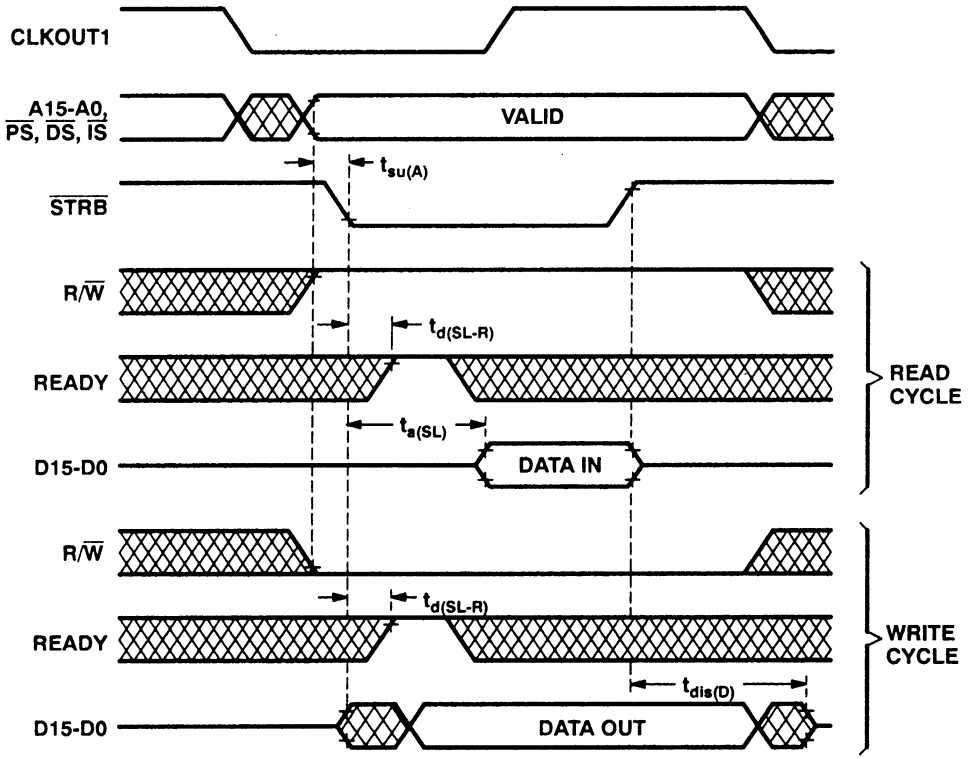
The following brief discussion describes the TMS320C2x read and write cycles. A more complete discussion is contained in the *Second-Generation TMS320 User's Guide.*[1] Throughout this report, Q is used to indicate the duration of a quarter-phase of the output clock (CLKOUT1 or CLKOUT2). Memory interfaces discussed in this report assume that the TMS320C2x is running at 40 MHz; i.e., Q = 25 ns. The memory read and write timings are shown in Figure 7. In a read cycle, the following sequence occurs:

1) Near the beginning of the machine cycle (CLKOUT1 goes low), the address bus and one of the memory select signals ($\overline{PS}$, $\overline{DS}$, or $\overline{IS}$) becomes valid. R/W goes high to indicate a read cycle.
2) $\overline{STRB}$ goes low in not less than $t_{su(A)} = (Q - 12)$ ns after the address bus becomes valid.
3) Early in the second half of the cycle, the READY input is sampled. READY must be stable (low or high) at the TMS320C2x no later than $t_{d(SL-R)} = (Q - 20)$ ns after $\overline{STRB}$ goes low.
4) With no wait-states (READY is high), data must be available no later than $t_{a(SL)} = (2Q - 23)$ ns after $\overline{STRB}$ goes low.

The sequence of events that occurs during an external write cycle is the same as the above, with the following differences:

1) R/$\overline{W}$ goes low to indicate a write cycle.
2) The data bus begins to be driven approximately concurrently with $\overline{STRB}$ going low.
3) The data bus enters a high-impedance state no later than $t_{dis(D)} = (Q + 15)$ ns after $\overline{STRB}$ goes high.

Figure 7. Read and Write Timings



Interfacing with a PROM

A convenient means of implementing program memory in a TMS320C2x system is provided through the use of PROMs. Two separate approaches for interfacing PROMs to the TMS320C2x are considered. The first approach does not require address decoding since the system contains only a small amount of one type of memory. The second approach illustrates an interface that utilizes address decoding to distinguish between two or more memory types with different access times.

Direct PROM Interface

An example of a no wait-state memory system is the direct PROM interface design shown in Figure 8. In this design, the TMS320C2x is interfaced with the Texas Instruments TBP38L165-35, a low-power, 2K × 8-bit PROM. The interface timing for the design of Figure 8 is shown in Figure 9.

Figure 8. Direct Interface of the TBP38L165-35 to the TMS320C2x

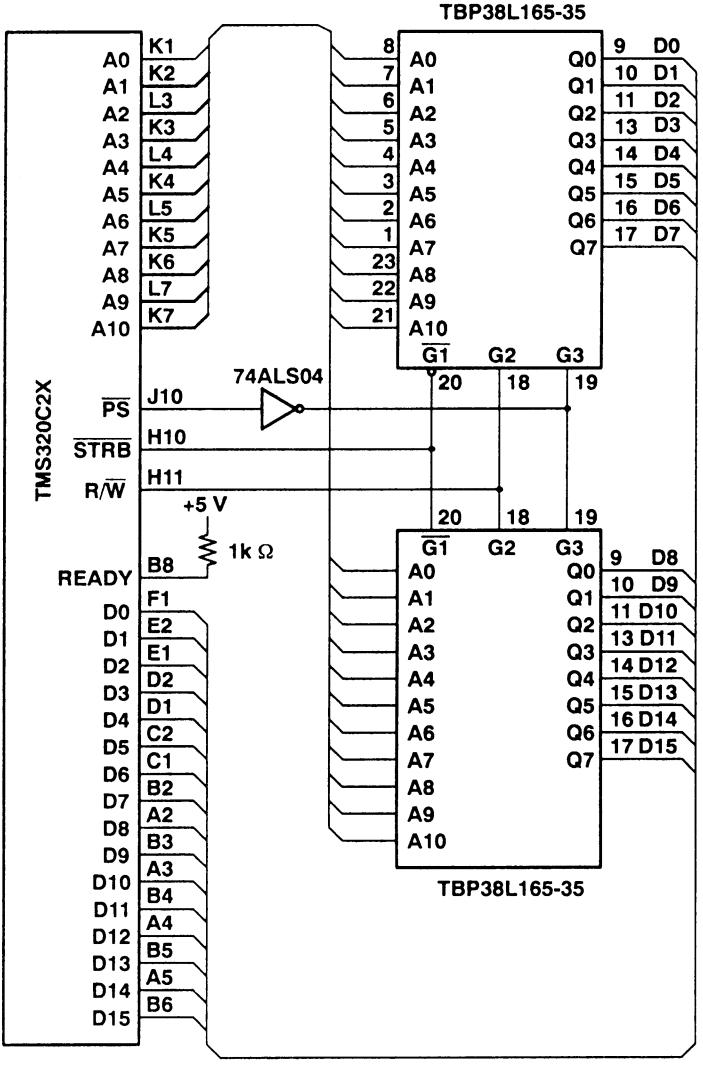
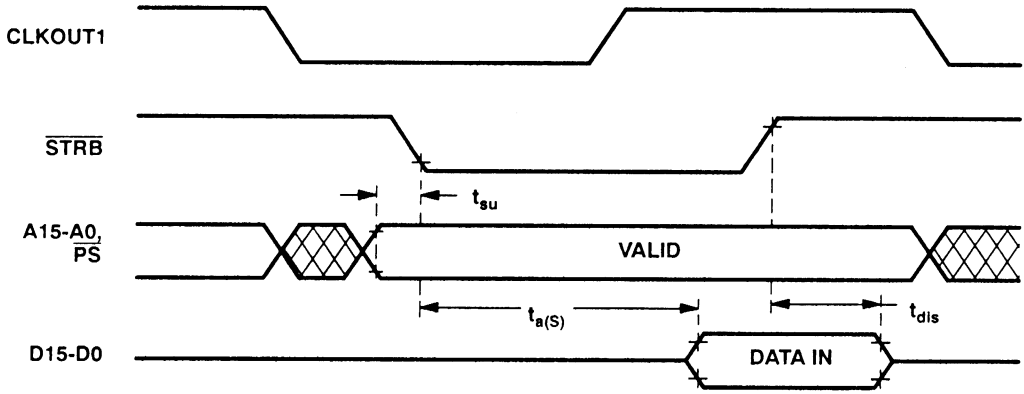


Figure 9. Interface Timing of the TBP38L165-35 to the TMS320C2x



As discussed earlier, the TMS320C2x expects data to be valid no later than (2Q – 23) ns after STRB goes low; this is 27 ns for a TMS320C2x operating at 40 MHz. The access times of the TBP38L165-35 are 35 ns maximum from address ($t_{a(A)}$), and 20 ns maximum from chip enable ($t_{a(S)}$). On the TMS320C2x, address becomes valid a minimum of t_{su} = (Q – 12) ns = 13 ns before STRB goes low (see Figure 1). The memory is not enabled, however until STRB goes low. Therefore, the data appears on the data bus within 27 ns after STRB goes low, as required by the TMS320C2x.

Bus conflict may occur when a TMS320C2x write cycle is followed by a memory read cycle. In this case, the TMS320C2x data lines must enter a high-impedance state before the memory starts driving the data bus. In a write cycle, the TMS320C2x enters a high-impedance state no later than 15 ns after the beginning of the next cycle. Since the design of Figure 8 utilizes STRB to enable the TBP38L165s, these memories cannot drive the data bus before STRB goes low, i.e., Q ns after the beginning of the cycle. Therefore, bus conflict is avoided since 25 ns > 15 ns.

Note that the TMS320C2x R/$\overline{W}$ line is connected to the G_2 enable line on both TBP38L165s. Therefore, the PROMs are disabled whenever R/$\overline{W}$ goes low, even if STRB is active. This prevents the bus conflict that occurs if the PROMs are written to when using the TBLW instruction, which transfers data from the data memory space to the program memory space.[1] Such transfers, however, were intended to be made only when RAMs are used in the program space.

The most critical timing parameters of the TBP38L165-35 direct interface to the TMS320C2x are summarized in Table 3.

Table 3. Timing Parameters of the TBP38L165-35 Direct Interface to the TMS320C2x

Description	Symbol Used in Figure 9	Value
Address setup time	t_{su}	13 ns (min)
TBP38L165-35 access time from chip enable	$t_{a(S)}$	20 ns (max)
TBP38L165-35 disable time	t_{dis}	15 ns (max)

PROM Interface with Address Decoding

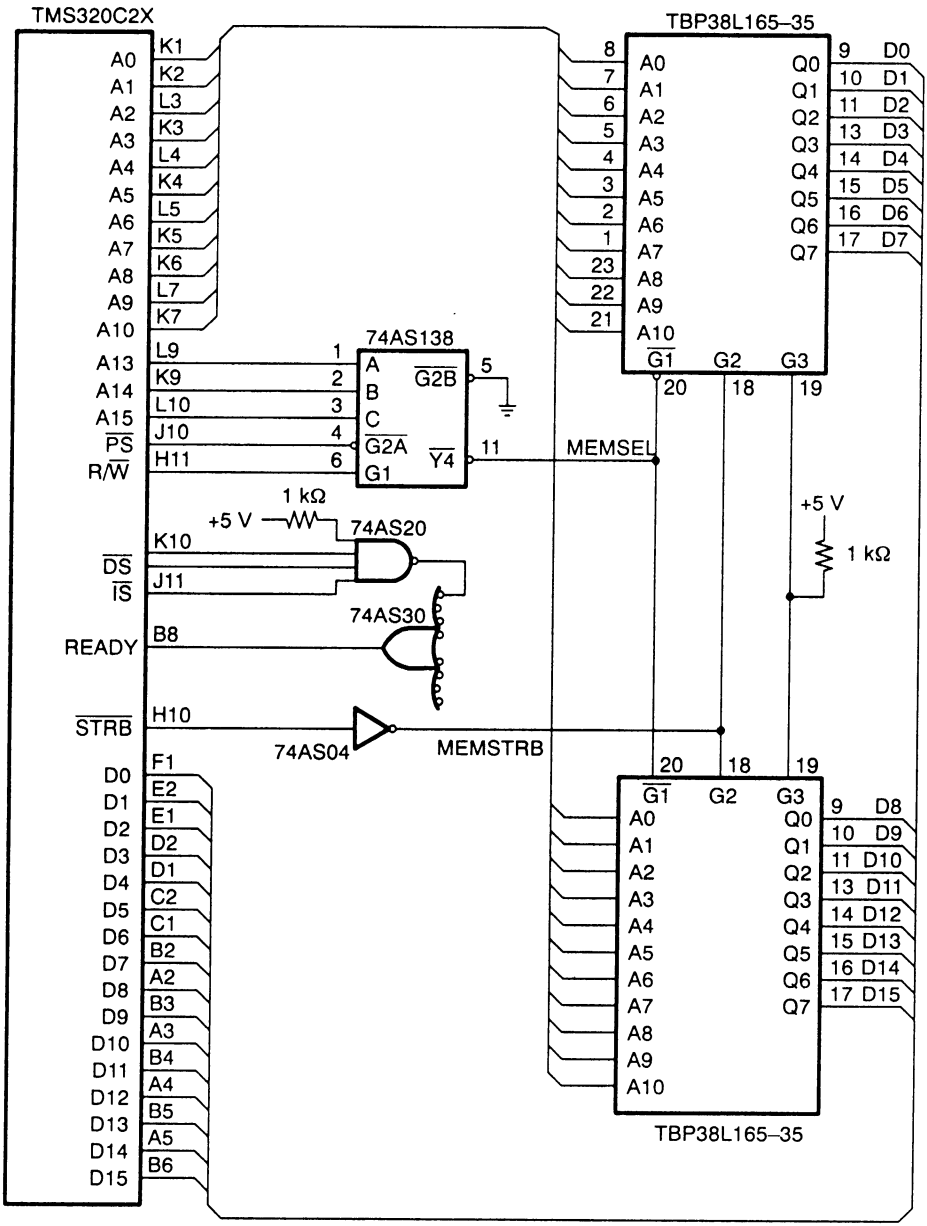
The second design example considers the interface of PROMs to the TMS320C2x using address decoding. A major issue when designing an interface with address decoding is that the TMS320C2x requires the READY signal to be stable no later than (Q – 20) ns after $\overline{\text{STRB}}$ goes low. Since the setup time for the address is (Q – 12) ns, the TMS320C2x requires (worst case) a stable READY at least (2Q – 32) ns after the address has been stabilized. This is 18 ns at 40 MHz. Proper address decoding may require two levels of gating. A third level of gating is required when more than one type of memories or peripherals with different numbers of wait-states is used. Using 'AS interface logic (the fastest currently available), these three levels of gating have a total propagation delay of 15 ns (worst case). Using a 74AS138 three-to-eight-line decoder to implement the first two levels of gating does will not result in any significant improvement in the propagation delay. (The 74AS138 has a maximum propagation delay of 9.5 ns for a high-to-low transition.)

An approach that can be used to meet the READY timing requirements is shown in Figure 10. This design utilizes one address decoding scheme to generate READY, and a second address decoding scheme to enable the different memory banks.

In this design, the memories with no wait-states are mapped at the upper half (upper 32K) of the program space. The lower half is used for memories with one or more wait-states. This decoding is implemented with the 74AS20 four-input NAND gate. The output of this gate is low when the following are true:

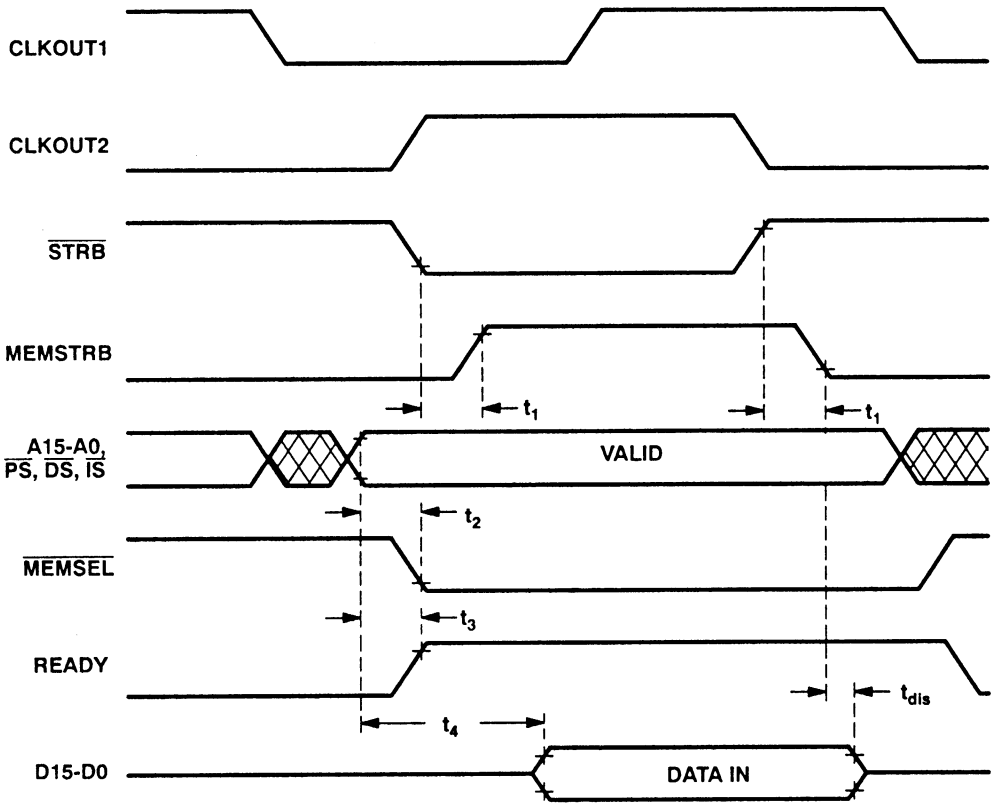
1) Address line A15 is high; i.e., the upper 32K words are selected.
2) DS and IS are high; i.e., an external program memory cycle is in progress.

Figure 10. Interface of the TBP38L165-35 to the TMS320C2x



The timing of READY is shown in Figure 11. READY goes high 10 ns (worst case) after the address has become valid.

Figure 11. Interface Timing of the TBP38L165-35 to the TMS320C2x (with Address Decoding)



Address decoding is implemented by the 74AS138. This decoding separates the program space into eight segments of 8K words each. The first four of these segments (lower 32K of address space) are enabled by the Y0, Y1, Y2, and Y3 outputs of the 74AS138. These segments are used for memories with one or more wait-states. The other four segments select memories with no wait-states (the TBP38L165s are mapped in segment #5 starting at address 8000h). Note that in Figure 10, R/$\overline{W}$ is used to enable the 74AS138. This prevents a bus conflict from occurring if an attempt is made to write to the PROMs.

In Figure 10, $\overline{\text{MEMSEL}}$ goes low no later than 10 ns (time t_2 in Figure 11) after address is valid. The PROMs are not enabled, however, until MEMSTRB goes high, i.e., a minimum of 5 ns after $\overline{\text{STRB}}$ goes low (time t_1 in Figure 11). Valid data appears on the data bus within 25 ns later. This meets the 27 ns or (2Q – 23) ns access time required from $\overline{\text{STRB}}$ low by the TMS320C2x. Note that in the design of Figure 10, $\overline{\text{STRB}}$ is used to enable the PROMs so that no bus conflict occurs

if the memory read cycle is followed by a write cycle. As seen in Figure 11, the memory enters a high-impedance state within ($t_1 + t_{dis}$) = 20 ns after $\overline{\text{STRB}}$ goes high. Therefore, if a memory read cycle is followed by a write cycle, no bus conflict occurs since the TMS320C2x starts driving the data bus no earlier than Q ns after the beginning of the write cycle.

The most critical timing parameters of the TBP38L165-35 interface with address decoding to the TMS320C2x are summarized in Table 4.

Table 4. Timing Parameters of the TBP38L165-35 Interface with Address Decoding to the TMS320C2x

Description	Symbol Used in Figure 11	Value
Propagation delay through the 74AS04	t_1	5 ns (max)
Propagation delay through the 74AS138	t_2	10 ns (max)
Address valid to READY	t_3	10 ns (max)
TBP38L165-35 disable time	t_{dis}	15 ns (max)

In summary, when interfacing to PROM memories with the TMS320C2x, two different approaches can be taken depending on whether or not any of the memories in the system require wait-states. When no wait-states are required for any of the memories, READY can be tied high, and the interface to the PROMs becomes a direct connection. When some of the system memories require wait-states, address decoding must be performed, and a valid READY signal that meets the TMS320C2x timing requirements must be provided. An efficient method of accomplishing this is to use one section of circuitry to generate the address decode, and a second, independent section to generate the READY signal.

EPROM Interfacing

EPROMs may be used to debug TMS320C2x algorithms. Three different EPROM interfaces to the TMS320C2x are presented in this subsection. First, the direct interface of an EPROM that requires no wait-states is discussed. This is followed by descriptions of EPROM interfaces that require one and two wait-states.

Direct EPROM Interface with No Wait-States

A Texas Instruments TMS27C292-35 EPROM can interface directly to the TMS320C2x with no wait-states, as shown in Figure 12. The TMS27C292-35 is a CMOS EPROM with access times of 35 ns from valid address and 25 ns from chip select. The timing of the interface is shown in Figure 13.

Figure 12. Direct Interface of the TMS27C292-35 to the TMS320C2x

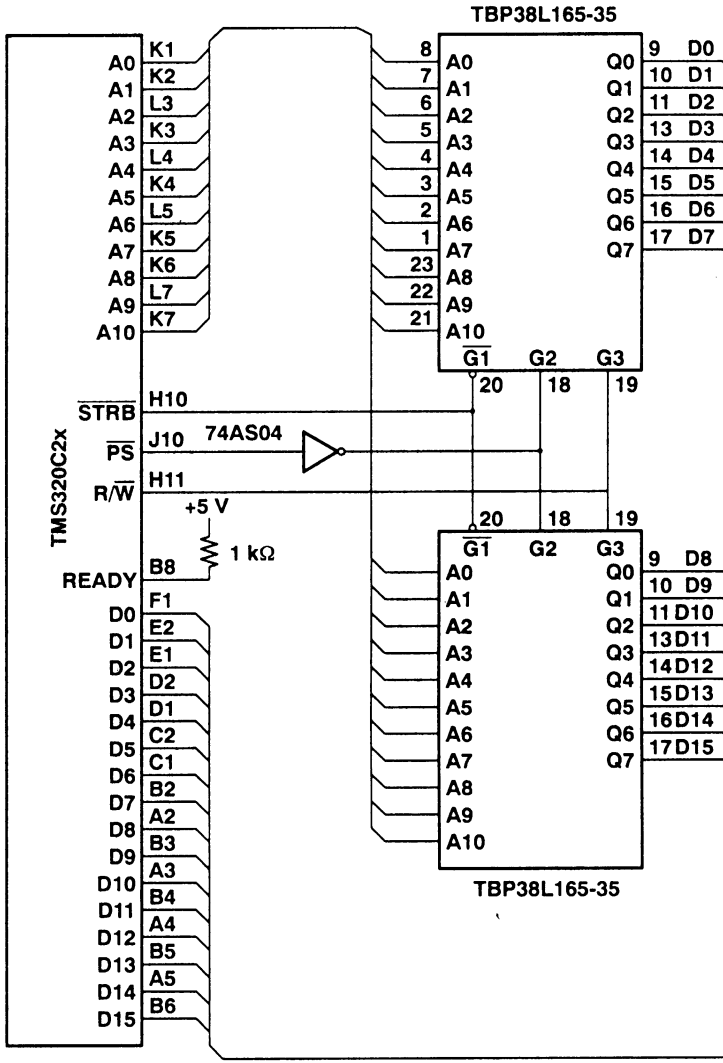
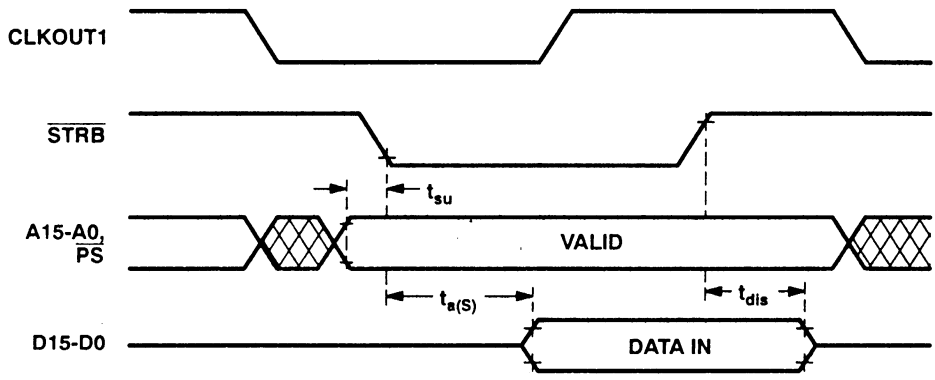


Figure 13. Interface Timing of the TMS27C292-35 to the TMS320C2x



As shown in Figure 13, the EPROMs are not enabled until $\overline{\text{STRB}}$ goes low. Since the address has been valid for at least t_{su} = 13 ns before $\overline{\text{STRB}}$ goes low, valid data appear on the data bus $t_{a(S)}$ = 25 ns (max) later. The EPROMs are disabled with $\overline{\text{STRB}}$ going high, and their output buffers enter a high-impedance state, t_{dis} = 25 ns (max) later. Therefore, no bus conflict occurs even if the memory read cycle is followed by a write cycle.

The most critical timing parameters of the TMS27C292-35 direct interface to the TMS320C2x are summarized in Table 5.

Table 5. Timing Parameters of the TMS27C292-35 Direct Interface to the TMS320C2x

Description	Symbol Used in Figure 11	Value
Address setup time	t_{su}	13 ns (min)
TMS27C292-35 access time from chip enable	$t_{a(S)}$	25 ns (max)
TMS27C292-35 disable time	t_{dis}	25 ns (max)

EPROM Interface with One Wait-State

The hardware interface of the Wafer Scale WS57C64F-12 (8K × 8-bit EPROMs) to the TMS320C2x is shown in Figure 14. The WS57C64F-12s are mapped at address 2000h. The interface timing diagram is provided in Figure 15.

Figure 14. Interface of the WS57C64F-12 to the TMS320C2x

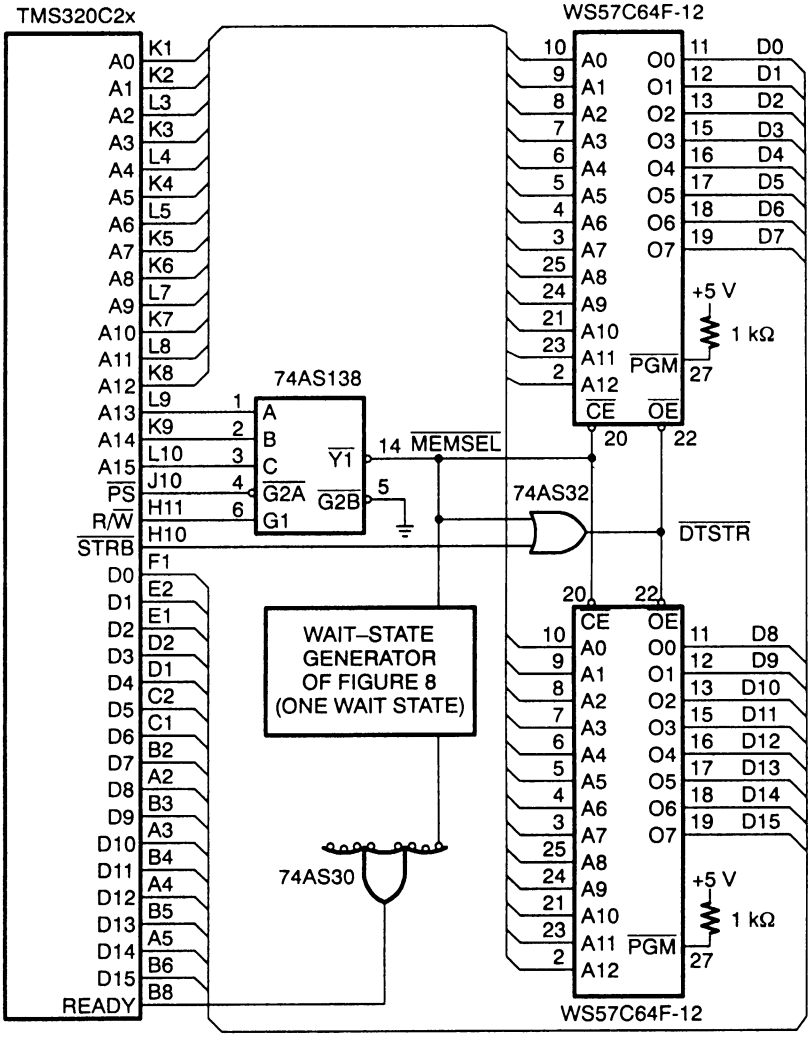
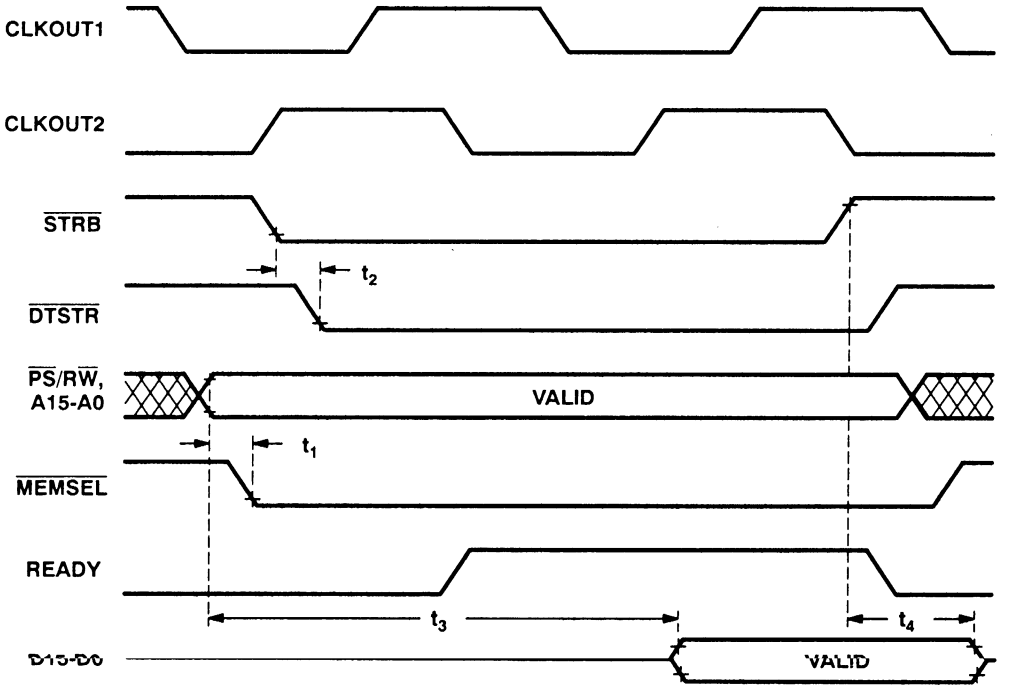


Figure 15. Interface Timing of the WS57C64F-12 to the TMS320C2x



The WS57C64-12 access times from valid address, chip select, and output enable are $t_{a(A)}$ = 120 ns (max), $t_{a(CE)}$ = 120 ns (max), and $t_{a(OE)}$ = 35 ns (max), respectively. As shown in Figure 14, the 74AS138 is used for address decoding. $\overline{PS}$ and R/$\overline{W}$ are used to drive the $\overline{G2A}$ and G1 enable inputs of the 74AS138, respectively. The latter prevents any bus conflict resulting from an accidental write (using the TBLW instruction) to the program space. $\overline{MEMSEL}$ going low t_1 = 10 ns (max) after address valid (see Figure 15) is used for two purposes:

1) to drive the wait-state generator, as discussed earlier; and
2) to generate a strobe signal, $\overline{DTSTR}$, that activates the output buffers of the WS57C64-12s.

Time t_3 in Figure 15, is the time from valid address to valid data on the data bus, i.e., $t_3 = t_1 + t_{a(CE)}$ = 130 ns (max). Since 40 ns < t_3 < 140 ns, one wait-state is required. The wait-state generator of Figure 14 may be used to implement this wait-state. Also, note that the WS57CF64-12 is the slowest member of the WS57C64F EPROM series, and still meets the specifications for one wait-state.

With $\overline{STRB}$ going high, the read has been completed. $\overline{DTSTR}$ is then used to turn off the memory output buffers. The output disable time of the WS57C64F-12 is t_{dis} = 35 ns (max). Time t_4 in Figure 15 is used to indicate the time from $\overline{STRB}$ high to output entering a high-impedance state. With a propagation delay of t_p = 5.8 ns (max) through the 74AS32, $t_4 = t_p + t_{dis}$ = 40.8 ns (max). Since this time is less than 50 ns (the earliest the TMS320C2x can start driving the data bus when the next instruction is a write), there is no bus conflict.

Table 6 summarizes the most critical timing parameters of the WS57C64F-12 interface to the TMS320C2x.

Table 6. Timing Parameters of the WS57C64F-12 Interface to the TMS320C2x

Description	Symbol Used in Figure 11	Value
Address valid to $\overline{\text{MEMSEL}}$ low	t_1	10.5 ns (max)
$\overline{\text{STRB}}$ to $\overline{\text{DTSTR}}$ low	t_2	5.8 ns (max)
TMS320C2x address valid to WS57C64F-12 data valid	t_3	130.0 ns (max)
$\overline{\text{STRB}}$ high to WS57C64F-12 output disable	t_4	40.8 ns (max)

EPROM Interface with Two Wait-States

The interface of the TMS27C64-20 to the TMS320C2x is shown in Figure 16. The TMS27C64-20 is a CMOS 8K × 8-bit EPROM with an access time of 200 ns. The timing diagram is shown in Figure 17.

Figure 16. Interface of the TMS27C64-20 to the TMS320C2x

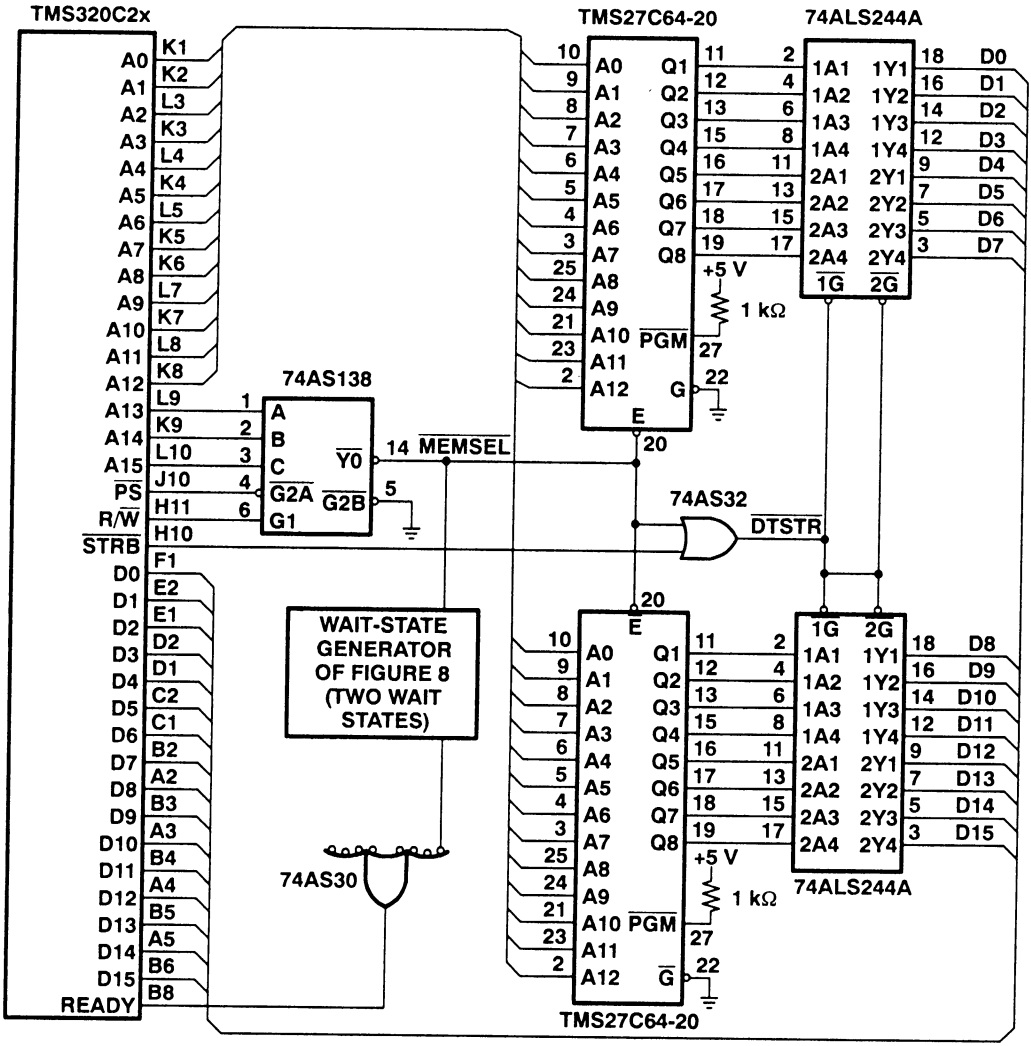
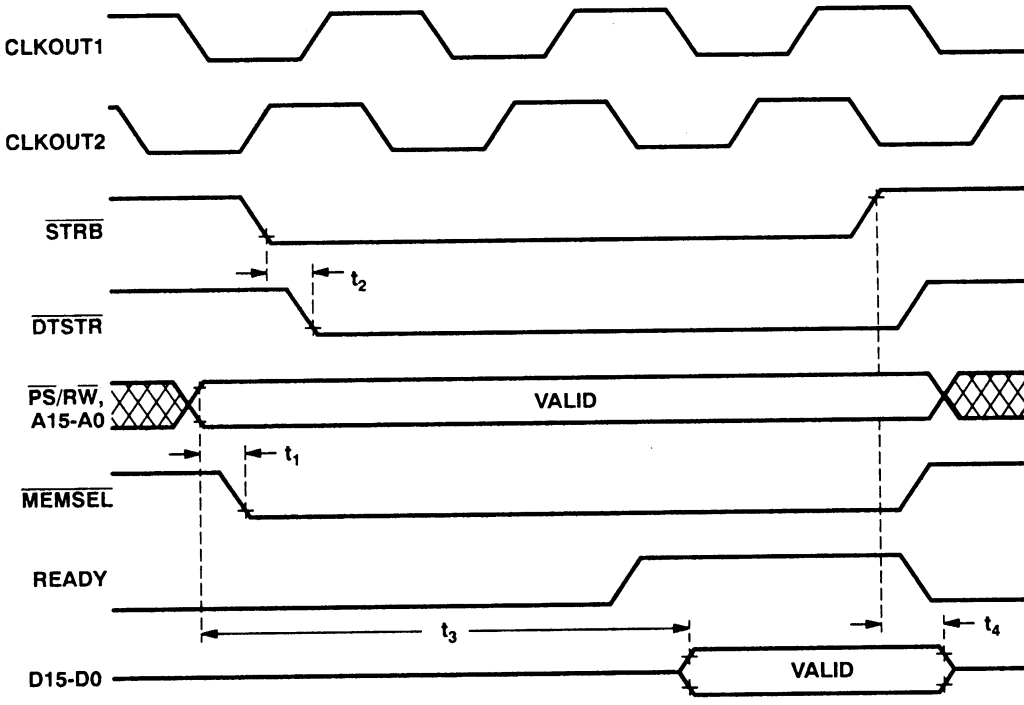


Figure 17. Interface Timing of the TMS27C64-20 to the TMS320C2x



With a 200-ns access time, two wait-states are needed. These can be implemented using the wait-state generator of Figure 14(a). Address decoding is similar to that used for the WS57C64F-12, and the TMS27C64 is mapped at address 0000h. The memory cycle starts with address valid. $\overline{\text{MEMSEL}}$ becomes low t_1 = 10 ns (max) later, due to propagation delay through the 74AS138. With $\overline{\text{MEMSEL}}$ active, valid data appear on the TMS27C64 data lines, t_a = 200 ns (max) later. As shown in Figure 16, the 74ALS244A octal buffers are used to buffer the memories from the TMS320C2x. These buffers are enabled with $\overline{\text{DTSTR}}$, which is a logical-OR signal of both $\overline{\text{MEMSEL}}$ and $\overline{\text{STRB}}$. The maximum propagation delay through these buffers is t_p = 10 ns. Therefore, valid data appear on the TMS320C2x data bus no later than $t_3 = t_1 + t_a + t_p$ = 220 ns from valid address. This is the overall access time, and 140 ns < t_3 < 240 ns, i.e., two wait-states are sufficient.

With $\overline{\text{STRB}}$ going high, the TMS320C2x has completed the memory read. $\overline{\text{DTSTR}}$ follows $\overline{\text{STRB}}$, and t_2 = 5.8 ns (maximum propagation delay through the 74AS32) after STRB goes high; $\overline{\text{DTSTR}}$ also goes high. This forces the 74ALS244As to enter a high-impedance state 13 ns (max) later. Therefore, no later than t_4 = (13 + 5.8) ns = 18.8 ns after $\overline{\text{STRB}}$ goes high, the outputs of the 74ALS244As are in a high-impedance state (see Figure 12). Buffers were used because the disable time of the TMS27C64-20 is 60 ns, which will generate a conflict on the data bus.

Table 7 summarizes the most critical timing parameters of the TMS27C64-20 interface to the TMS320C2x.

Table 7. Timing Parameters of the TMS27C64-20 Interface to the TMS320C2x

Description	Symbol Used in Figure 11	Value
Address Valid to $\overline{\text{MEMSEL}}$ low	t_1	10.5 ns (max)
$\overline{\text{STRB}}$ low to $\overline{\text{DTSR}}$ low	t_2	5.8 ns (max)
TMS320C2x address valid to TMS27C64-20 data valid	t_3	220.0 ns (max)
$\overline{\text{STRB}}$ high to TMS27C64-20 output disable	t_4	18.8 ns (max)

In summary, EPROMs can be a valuable tool during the prototyping stages of a design, and may even be desirable for production. When EPROMs that are fast enough are used with the TMS320C2x, a direct interface similar to that used for PROMs may be used. When slower, less costly EPROMs are used, a simple flip-flop circuit can be used to generate one or more wait-states. With slower EPROMs, however, data output turnoff can be slow, and must be taken into consideration in the design. The same advantages are offered by the TMS320E25, which has an on-chip 4K-word EPROM in place of the 4K-word on-chip ROM of the TMS320C25.

Interfacing SRAMS

The TMS320C2x can utilize SRAM as either program or data memory. When used as program memory, object code can be downloaded into the RAM and executed. SRAM can also be used as data memory to extend the TMS320C2x's 544 words of internal RAM. In the first case, the SRAM is mapped into the TMS320C2x program space, while the second case maps the SRAM into the data space.

The SRAM chosen for this interface is the Cypress Semiconductor CY7C169-25 4K × 4-bit SRAM. This SRAM has a 25-ns access time from address ($t_{a(A)}$) and a 15-ns access time from chip enable ($t_{a(CE)}$). Note that these access times are fast enough that a wait-state generator is not required for this interface. If, however, RAMs that require wait-states are used in the system,the wait-state generator described in the Interfacing EPROMs subsection can be used.

RAMs with a 4K × 4-bit organization are used in this application to minimize the package count for the desired number of words of memory being implemented. In this case, only four packages are required. In contrast, if 16K × 1-bit memories had been used, 16 packages would have been required, and much of the memory might have gone unused. In general, the choice of memory organization for a particular system should be based on the amount of memory required and the organization of the memories currently available in the industry.

The hardware interface to this RAM is shown in Figure 18, and a timing diagram of the interface is presented in Figure 19.

Figure 18. Interface of the CY7C169-25 to the TMS320C2x

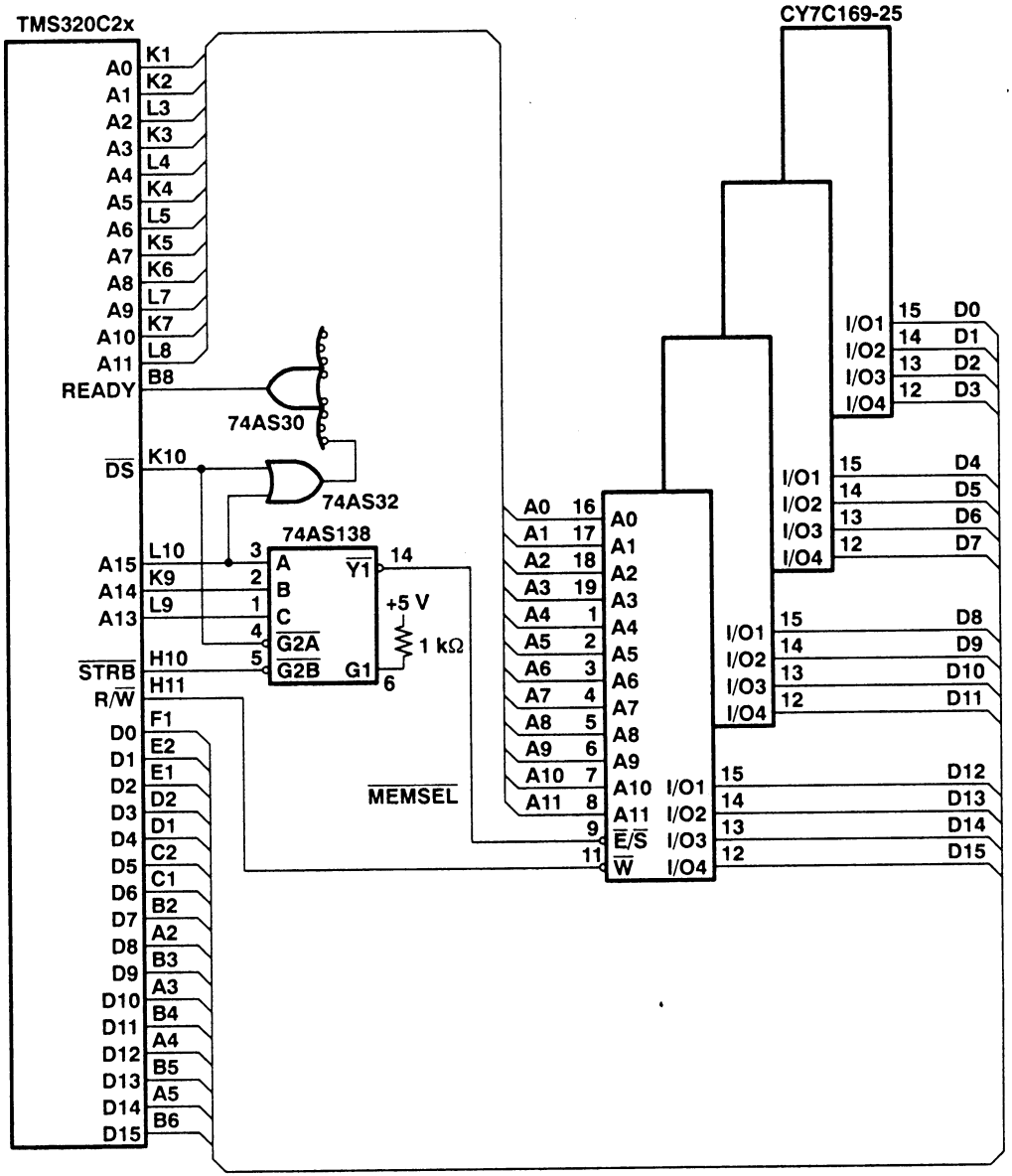
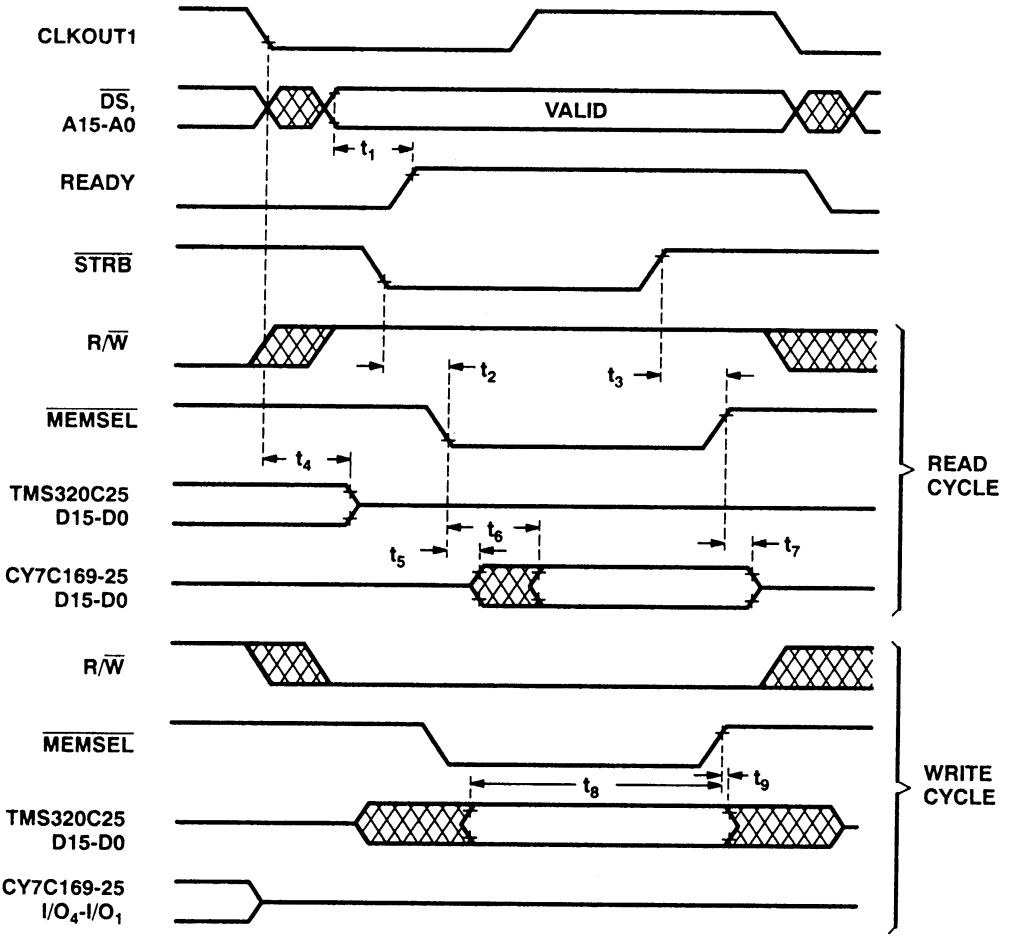


Figure 19. Interface Timing of the CY7C169-25 to the TMS320C2x



The design of Figure 18 utilizes a similar approach to the one described in the Interfacing PROMs and Interfacing EPROMs subsections; i.e., one address decoding scheme is used to generate READY, and a second address decoding scheme is used to enable the SRAM. In this design, RAMs with no wait-states are mapped at the lower half (lower 32K words) of the TMS320C2x data space. The upper half is used for memories with one or more wait-states. This decoding is implemented with the 74AS32 two-input OR gate. The output of this gate is low (active) when $\overline{DS}$ is low (i.e., access to external data space requested), and A15 is low (i.e., lower 32K words selected). Time t_1 in Figure 19 indicates the time from valid address to READY going high. The maximum value for t_1 is

$$t_1 = t_{p(74AS32)} + t_{p(74AS30)} = (5.8 + 5)\ ns = 10.8\ ns$$

where $t_{p(X)}$ denotes the maximum propagation delay through device X.

As shown in Figure 18, address decoding that enables the RAM is implemented with the 74AS138. This decoding separates the data space into eight segments with 8K words per segment. The first four segments are enabled by the $\overline{Y0}$, $\overline{Y1}$, $\overline{Y2}$, and Y3 outputs of the 74AS138. These segments are used for memories with no wait-states. Note, in Figure 18, that the CY7C169s are enabled by $\overline{Y1}$; i.e., the memories are mapped at address 2000h. The rest of those segments, enabled by the other outputs of the 74AS138 decoder, are used for memories with one or more wait-states.

Memory Read Cycle

Figure 19 shows the timing for memory read and write cycles. In a read cycle, R/W goes high concurrently with valid address, indicating that a read rather than a write cycle has been initiated. With $\overline{STRB}$ used to enable the 74AS138, $\overline{MEMSEL}$ goes low no later than t_2 = 8.5 ns after $\overline{STRB}$ goes low. This is the maximum propagation delay of the 74AS138 before outputting a high-to-low transition from the $\overline{G}$ enable pin. The CY7C169s begin driving the data bus no earlier than t_5 = 5 ns after $\overline{MEMSEL}$ goes low. By then, all of the devices having access to the data bus must have entered a high-impedance state. Figure 19 shows the TMS320C2x data lines entering a high-impedance state no later than t_4 = 15 ns after the beginning of the read cycle. This is the case when the present read cycle is preceded by a write cycle.

The RAMs provide valid data no later than t_6 = 15 ns after $\overline{MEMSEL}$ goes low. Therefore, the worst-case access time from $\overline{STRB}$ going low is $t_2 + t_6$ = 23.5 ns. This meets the 27-ns access time required by the TMS320C2x operating at 40 MHz.

The TMS320C2x read cycle is concluded with $\overline{STRB}$ going high. $\overline{MEMSEL}$ follows $\overline{STRB}$ and goes high within t_3 = 7.5 ns. This time is the maximum propagation delay through the 74AS138 for a low-to-high transition. The CY7C169 data lines enter a high-impedance state no later than t_7 = 15 ns after $\overline{MEMSEL}$ goes high. Therefore, no bus conflict occurs if the present read cycle is followed by a write cycle.

Memory Write Cycle

As shown in Figure 19, the memory write cycle is similar to the read cycle with the exception that R/$\overline{W}$ is low. The TMS320C2x begins driving the data bus as soon as $\overline{STRB}$ goes low, while $\overline{MEMSEL}$ follows $\overline{STRB}$ within t_2 = 8.5 ns. Since R/$\overline{W}$ is low when $\overline{MEMSEL}$ goes low, the CY7C169s do not drive the data bus.

Data is clocked into the CY7C169s on the rising edge of $\overline{MEMSEL}$. Time t_8 in Figure 19 is the time that data is valid before $\overline{MEMSEL}$ goes high. This time is no less than the TMS320C2x minimum data setup time before $\overline{STRB}$ goes high (t_8 = (2Q – 20) ns = 30 ns when operating at 40 MHz) plus the 2-ns minimum propagation delay through the 74AS138. Therefore, t_8 is equal to or greater than 32 ns. Note that this time meets the 10-ns minimum data setup time required by the CY7C169.

Table 8 summarizes the most critical timing parameters that must be considered when interfacing the CY7C169s with the TMS320C2x.

Table 8. Timing Parameters of the CY7C169-25 Interface to the TMS320C2x

Description	Symbol Used in Figure 11	Value
Address valid to READY valid	t_1	10.8 ns (max)
STRB low to MEMSEL low	t_2	8.5 ns (max)
STRB high to MEMSEL high	t_3	7.5 ns (max)
CLKOUT1 low to TMS320C2x data bus entering the high-impedance state	t_4	15.0 ns (max)
MEMSEL low to CY7C169-25 driving	t_5	5.0 ns (min)
MEMSEL low to CY7C169-25 data valid	t_6	15.0 ns (max)
MEMSEL high to CY7C169-25 entering the high-impedance state	t_7	15.0 ns (max)
Data setup time for a write	t_8	32.0 ns (min)
Data hold time	t_9	7.5 ns (min)

In summary, interfacing external RAM to the TMS320C2x is quite useful for expanding the internal data memory or implementing additional RAM program memory. In cases where RAMs of different execution times are used, separate schemes for address decoding and READY generation can be used to meet READY timing requirements in a similar manner to that used for the PROM interface as described in this report. RAMs with similar access times may then be grouped together in one segment of memory.

Interfacing Memories to the TMS320C25-50

TMS320C25-50 memory interfaces are similar or identical in form to those of the 40-MHZ version of the TMS320C25. In many cases, the interfacing techniques given in the preceding section can be used, with higher-speed versions of the memory devices substituted.

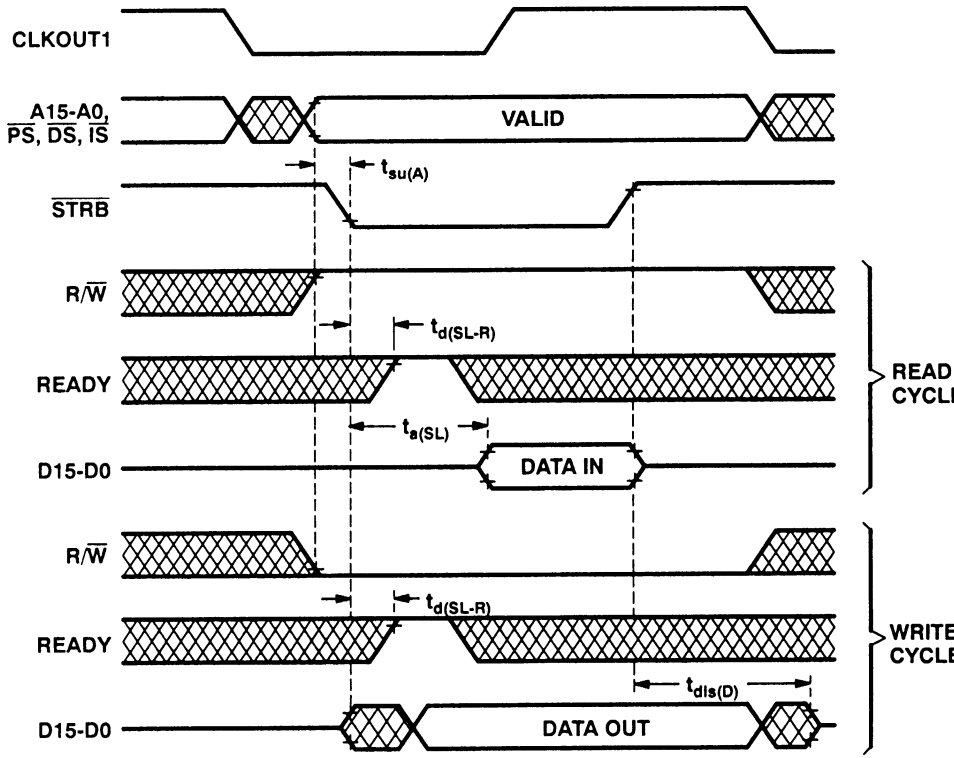
This section describes the memory interface timing requirements of the TMS320C25-50. Determining appropriate memory device speeds requires an understanding of TMS320C25-50 external bus cycles and the timing specification of the device.

The following excerpt from the TMS320C25-50 Electrical Specification and Figure 20 show the information necessary to determine the minimum memory device speed for a given application.

	Min	Max	Units
$t_{a(A)}$		3Q–31	ns
$t_{su(A)}$	Q–11		ns
$t_{su(D)R}$	17		ns

Figure 20 shows a TMS320C25-50 memory read and write cycle. Either of two timing requirements must be satisfied to guarantee a successful read operation. These two requirements are specified by $t_{a(A)}$ and $t_{su(D)R}$. Note that it is not necessary to satisfy both requirements, as each parameter is guaranteed independently.

Figure 20. TMS320C25-50 Memory Read and Write Cycle



A timing requirement of special interest is the memory access time measured from the falling edge of $\overline{STRB}$. The specification of this requirement is jointly implied by the device $t_{a(A)}$ and $t_{su(D)R}$ specifications as shown in the following.

$t_{a(A)}$ is defined as follows:

$$t_{a(A)} = t_{su(A)}\text{min} + t_{w(SL)} + t_{r(C)} - t_{su(D)R}\text{min}$$

For convenience, define $t_{w(S)}$ as follows:

$$t_{w(S)} = t_{w(SL)} + t_{r(C)}$$

Then $t_{a(A)}$ is given by

$$t_{a(A)} = t_{su(A)}\text{min} + t_{w(S)} - t_{su(D)R}\text{min}$$

The $t_{a(A)}$ specification guarantees that

$$t_{a(A)} > t_{a(A)\text{max}}$$

or

$$t_{su(A)\text{min}} + t_{w(S)} - t_{su(D)R\text{min}} > t_{a(A)\text{max}}$$

The above inequality is potentially confusing in that it guarantees a minimum on a parameter with a *max* subscript. As with any parameter specified as a *maximum*, the measured $t_{a(A)}$ value of a given device must be *greater* than the specified maximum in order for the device to pass the $t_{a(A)}$ test performed on the device. In this way, all values of $t_{a(A)}$ *less* than $t_{a(A)\text{max}}$ are guaranteed to meet the device $t_{a(A)}$ requirement.

$t_{a(A)\text{max}}$ is specified as

$t_{a(A)\text{max}}$ = 3Q–35 ns (40 MHz TMS320C25)
$t_{a(A)\text{max}}$ = 3Q–31 ns (TMS320C25-50)

Thus, the following inequalities are guaranteed:

Q–12 + $t_{w(S)}$ – 23 > 3Q–35 (40 MHz TMS320C25)
Q–11 + $t_{w(S)}$ – 17 > 3Q–31 (TMS320C25-50)

which evaluate to

$t_{w(S)}$ > 2Q (40 MHz TMS320C25)
$t_{w(S)}$ > 2Q–3 (TMS320C25-50)

The ta(A) specification thus implies a minimum value for $t_{w(S)}$.

On a memory read cycle, data must be valid no later than $t_{su(D)R\text{min}}$ prior to $\overline{STRB}$ going high. The maximum access time from $\overline{STRB}$ low (define this as $t_{a(SL)}$) is thus

$$t_{a(SL)\text{max}} = t_{w(S)\text{min}} - t_{su(D)R\text{min}}$$
$$= 2Q - 23 \qquad \text{(40 MHz TMS320C25)}$$

or

$$= (2Q–3) - 17 = 2Q–20 \qquad \text{(TMS320C25-50)}$$

The specification of $t_{a(SL)}$ typically determines the maximum access time from chip select and/or output enable for a memory device, as discussed in the following sections. Note that the

specification of the minimum value of $t_{w(SL)}$ ($\overline{STRB}$ – low pulse width) is in no way involved in assessing access time from address or from $\overline{STRB}$ going low.

Full-Speed Interfaces

The TMS320C25-50 can be interfaced to fast SRAM with no wait-states. Two key memory device specifications for such an interface are access time from address valid and access time from chip select and/or output enable. The key TMS320C25-50 timing requirements are specified by $t_{a(SL)}$ and $t_{a(A)}$.

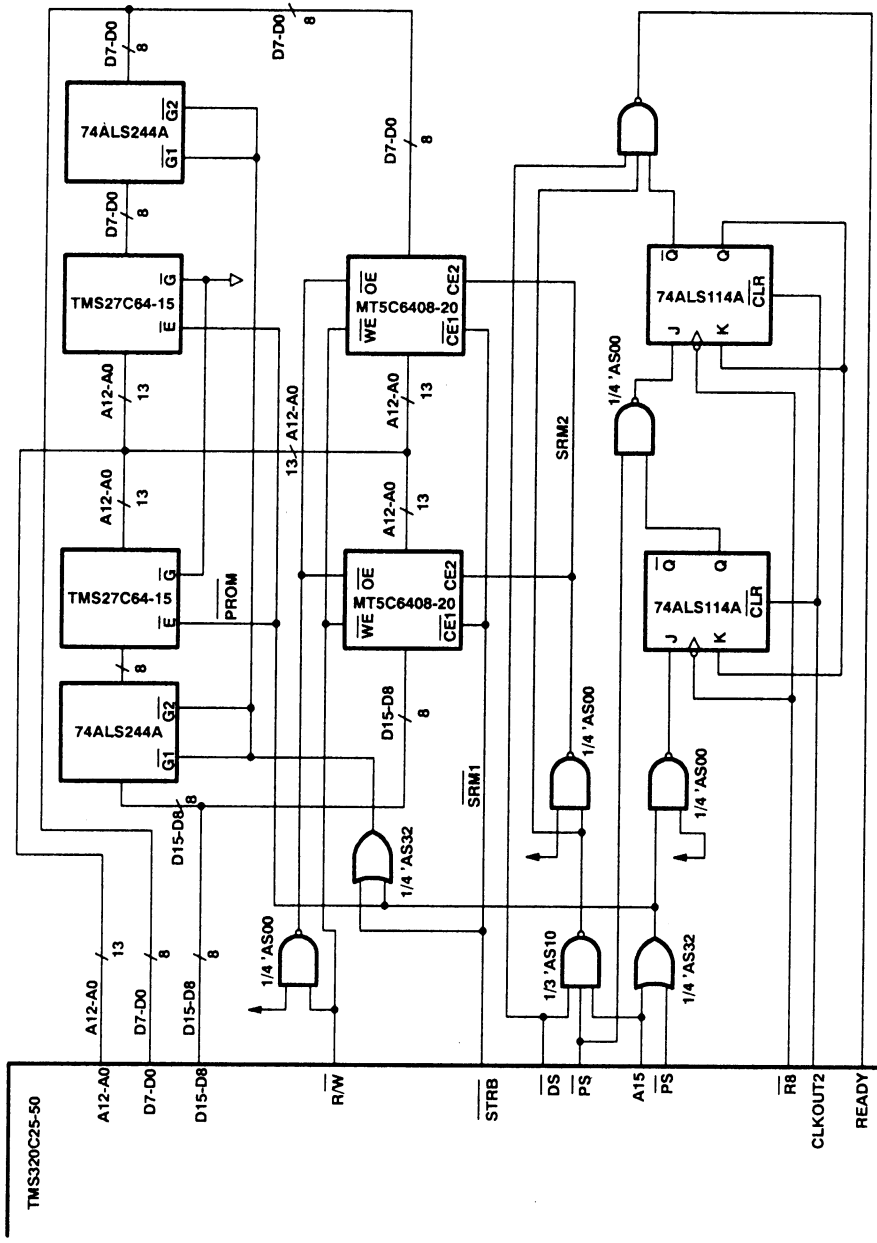
If $\overline{STRB}$ is an input to logic that generates the chip select and/or output enable signal for a memory device, data must be guaranteed valid no later than $t_{a(SL)} - t_d$ from $\overline{STRB}$ falling, where t_d is the delay imposed by the logic used to generate the chip select or output enable signal.

Typically, devices with both chip select and output enable signals can more easily accommodate the $t_{a(SL)}$ requirement, as $\overline{STRB}$ can directly serve as the output enable signal (active low), resulting in the condition $t_d = 0$. Logic internal to the memory device enables the device's input or output buffers (depending on the state of R/$\overline{W}$) only if the chip is selected via its chip select input.

Interfaces to memory devices having a chip select input but no output enable input will include chip select logic having $\overline{STRB}$ as one of its inputs. In these cases t_d is nonzero and thus the requirement on access time from chip select is tightened.

Figure 21 shows a TMS320C25-50 interfaced to 8K-words of full-speed SRAM and 8K-words of two wait-state EPROM. The operation of this circuit is discussed in the following section.

Figure 21. TMS320C25-50 Interfaced to Full-Speed SRAM and Two Wait-State EPROM



Full-Speed SRAM in Program Space

The cost and/or availability of non-volatile memory devices able to support TMS320C25-50 full-speed program execution may be prohibitive for some applications. (One such device is the Cypress Semiconductor 2K × 8 EPROM, part number CY7C291A-25.) The program code for Figure 21 can be stored in EPROM and self-booted into the SRAM devices at powerup for subsequent full-speed execution.

Table 9 shows the wait-state map for this circuit. Note that the READY generation logic for this arrangement is simple enough that inexpensive gates can be used for its implementation. Refer to the Ready Generation Techniques section earlier in this report for details of operation of the READY generation logic.

Table 9. Wait-State Map for Circuit of Figure 21.

External Space	Address Range	Number of Wait-States
Program	0000h–7FFFh	2
Program	8000h–FFFFh	0
Data	0000h–FFFFh	0
I/O	0000h–000Fh	1

The TI TMS27C64 EPROM devices reside in the two wait-state portion of program space at locations 0000h–1FFFh; the Micron MT5C6408-20 SRAM devices reside in the zero-wait portion of program space at locations 8000h–9FFFh.

Timing Analysis

Figure 22 shows the interface timing for accesses of the TMS27C64 EPROMs. Key timings are listed in Table 10. The output disable time of the TMS27C64 is too long to guarantee that no bus conflict will occur if an external write cycle follows a TMS27C64 read cycle; this is solved by buffering the data lines with TMS74ALS244A octal buffer ICs.

Figure 22. Interface Timing for Accesses of TMS27C64-15 to the TMS320C25-50

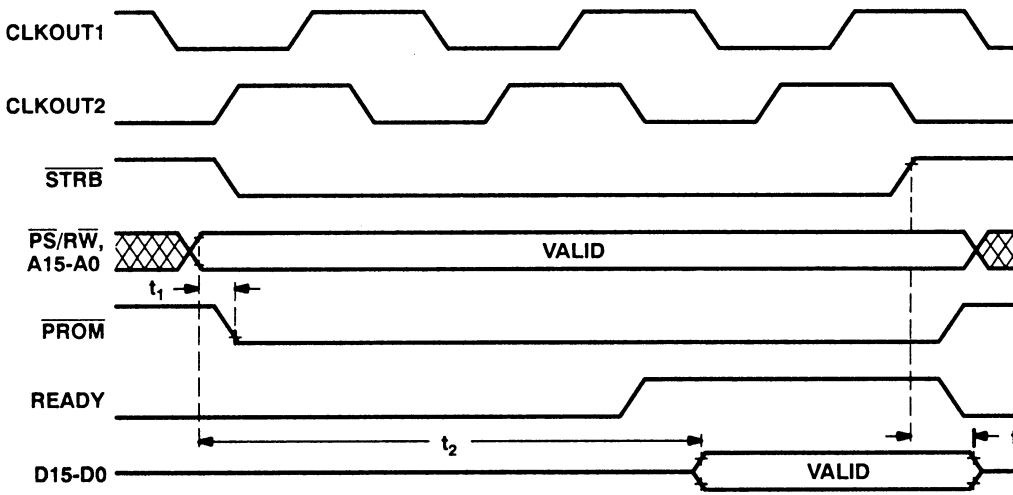


Table 10. TMS27C64 Interface Timing Parameters

Parameter Name	Designation in Figure 22	Time Duration
Address valid to $\overline{\text{PROM}}$ valid	t_1	5.8 ns (max)
$\overline{\text{PROM}}$ valid to TMS27C64 data valid		150 ns (max)
Address valid to TMS320C25 data valid	t_2	165.8 ns (max)
$\overline{\text{STRB}}$ high to TMS74ALS244A outputs high-Z	t_3	18.8 ns (max)

As shown in Figure 11, data is valid on the TMS27C64 data lines 5.8 ns + 150 ns (max) after address becomes valid. The delay through the TMS74ALS244A buffers is 10 ns (max). Data is valid on the TMS320C25-50 data bus $t_1 + t_2 + 10 = 165.8$ ns (max) after address valid. Thus the inequality $t_1 + t_2 + 10$ (max) $< t_{a(A)} + N_{tc(C)}$ is satisfied; 165.8 ns < 29 ns + 2 * 80 ns. Note that $t_{c(C)}$ is assumed to equal 80 ns. The buffer outputs are set in the high-impedance state $t_3 = 5.8$ ns + 13 ns = 18.8 ns (max) after $\overline{\text{STRB}}$ goes high.

Figure 23 shows the interface timing for accesses of the MT5C6408 SRAMs. Key interface timing parameters are given in Table 11.

Figure 23. Interface Timing for Accesses of the MT5C6408-20 to the TMS320C25-50

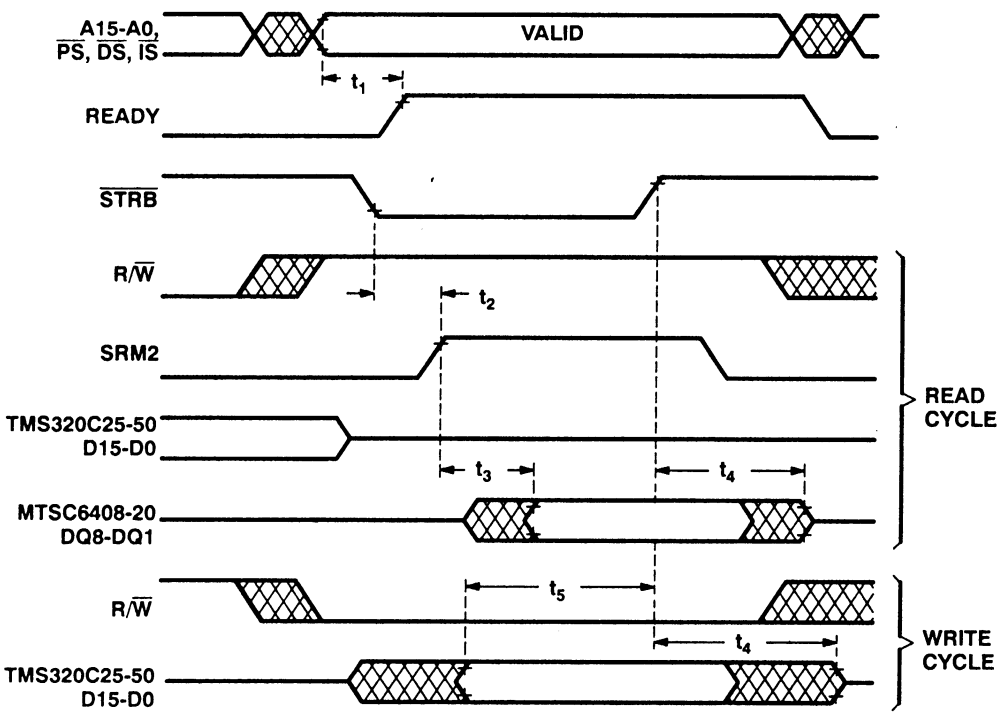


Table 11. MT5C6408-20 Interface Timing Parameters

Read Cycle		
Parameter Name	**Designation in 23**	**Time Duration**
Address valid to READY valid	t_1	9 ns (max)
Address valid to $\overline{SRM2}$ valid	t_2	9 ns (max)
Address valid to SRM1 valid		9 ns (max)
$\overline{SRM1}$/SRM2 valid to data valid	t_3	20 ns (max)
$\overline{STRB}$ high to data bus high-Z	t_4	15 ns (max)
Write Cycle		
Parameter Name	**Designation in Figure 23**	**Time Duration**
Address valid to READY valid	t_1	9 ns (max)
Address valid to $\overline{SRM2}$ valid	t_2	9 ns (max)
Address valid to $\overline{SRM1}$ valid		9 ns (max)
Data valid before $\overline{STRB}$ high	t_5	23 ns (max)
$\overline{STRB}$ high to data bus high-Z	t_4	15 ns (max)

The SRAMs are enabled if $\overline{CE1}$ is low and CE2 is high. CE2 is high when $\overline{IS}$, $\overline{DS}$, and A15 are high. (Making use of the fact that the 3 external spaces are mutually exclusive and exhaustive, 1 gate delay is saved by using $\overline{IS}$ and $\overline{DS}$ rather than $\overline{PS}$. This is crucial for satisfying the READY timing requirement.) $\overline{CE1}$ is driven directly by $\overline{STRB}$.

The function of the $\overline{OE}$ input of the MT5C6408s is the inverse of that of the $\overline{WE}$ input.

Read Cycle

As shown in Table 11, both chip enable inputs are valid no later than 9 ns from address valid. Data is valid no later than 20 ns after $\overline{CE1}$ and CE2 are valid, thus satisfying the condition $t_{a(SL)} \leq t_{a(SL)max}$. The outputs are tristated no later than 15 ns from $\overline{STRB}$ high.

Write Cycle

As shown in Table 11, both chip enable inputs are valid no later than 9 ns from address valid. Data is valid 23 ns (min) prior to $\overline{STRB}$ going high, satisfying the MT5C6408 data setup time requirement of 12 ns (min). The outputs are tristated no later than 35 ns from $\overline{STRB}$ high.

The complete electrical specifications and additional information pertaining to the TMS320C25-50 may be found in the *Second-Generation TMS320 User's Guide*.[1]

System Control Circuitry

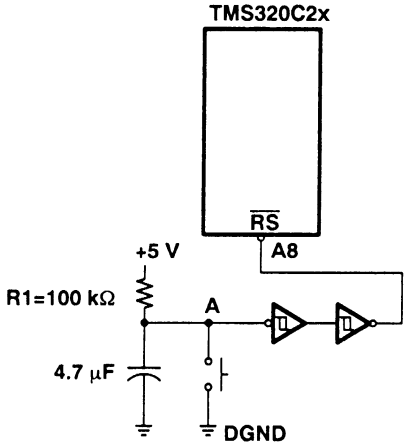
A system control circuitry performs important functions in system initialization and operation. A powerup reset circuit design and a crystal oscillator circuit design are presented in this section.

Reset Circuit

The reset circuit shown in Figure 24 performs a power-up restart operation; i.e., the TMS320C2x is reset when power is applied. Note that the switch circuit must contain debounce

circuitry. Driving the RS signal low initializes the processor. Reset affects several registers and status bits. For a detailed description of the effect of reset on the processor status, refer to the *Second-Generation TMS320 User's Guide*.[1]

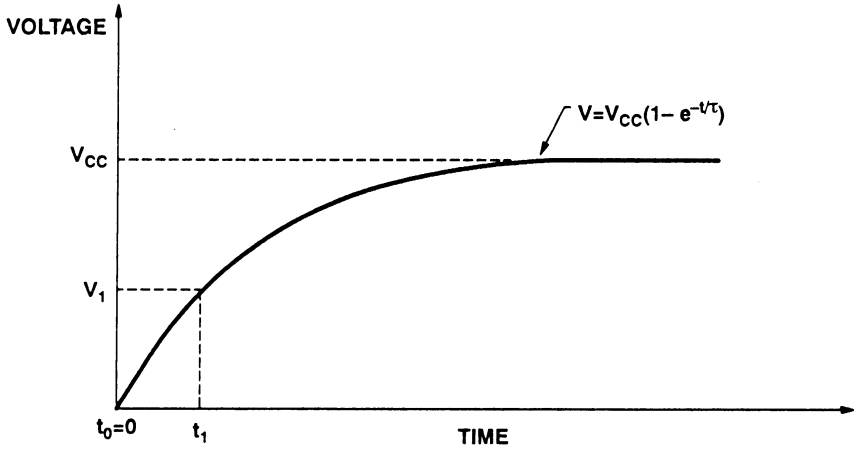
Figure 24. Powerup Reset Circuit



For proper system initialization, the reset signal must be applied for at least three CLKOUT cycles; i.e., 300 ns for a TMS320C2x operating at 40 MHz. Upon powerup, however, it can take up to hundreds of milliseconds before the system oscillator reaches a stable operating state. Therefore, the powerup reset circuit should generate a low pulse on the reset line until the oscillator is stable (between 100 and 200 ms). Once a proper reset pulse has been applied, processor operation begins at program memory location 0 which normally contains a branch (B) statement to direct program execution to the system initialization routine.

The voltage on node A is controlled by the R_1C_1 network (see Figure 24). After a reset, the voltage rises exponentially to the time constant R_1C_1, as shown in Figure 25.

Figure 25. Voltage on the TMS320C2x Reset Pin



The duration of the low pulse on the reset pin is approximately t_1, which is the time it takes for the capacitor C_1 to fully charge; i.e., 1.5 V. This is approximately the voltage at which the reset input switches from a logic level 0 to a logic level 1. The capacitor's voltage is given by

$$V = V_{CC}\left[1 - e^{-\frac{t}{\tau}}\right]; \quad (1)$$

where $\tau = R_1C_1$ is the reset circuit time constant.

Solving (1) for t gives:

$$t = -R_1R_1 \ln\left[1 - \frac{V}{V_{CC}}\right]. \quad (2)$$

Setting the following:

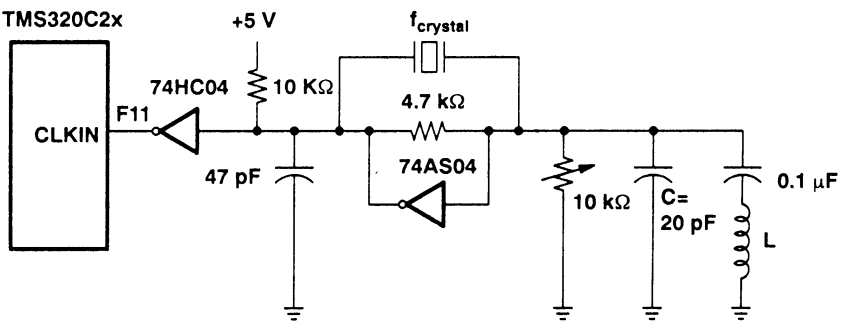
$R1 = 1\ M\Omega$
$C1 = 0.47\ \mu F$
$V = V_1 = 1.5\ V$
$V_{CC} = 5\ V$

gives $t = t_1 = 167$ ms. The Schmitt triggers shown in Figure 25 appropiately reshape the signal on node A.Therefore, the reset circuit of Figure 24 can generate a low pulse of an appropriate duration (167 ms) to ensure the stabilization of the system oscillator when most systems are powered.

Crystal Oscillator Circuit

The crystal oscillator circuit shown in Figure 26 is suitable for providing the input clock signal to any TMS320C2x device except the TMS32020. Since crystals with fundamental oscillation frequencies of 30 MHz and above are not readily available, a parallel-resonant third-overtone oscillator is used. If a packed clock oscillator is used, oscillator design is of no concern.

Figure 26. Crystal Oscillator Circuit



The 74AS04 inverter in Figure 26 provides the 180-degree phase shift that a parallel oscillator requires. The 4.7-kΩ resistor provides the negative feedback that keeps the oscillator in a stable

state; i.e., the poles of the system are constrained in a narrow region about the j axis of the s-plane (analog domain). The 10-kΩ potentiometer is used to bias the 74AS04 in the linear region. This potentiometer is adjusted as follows: Before the crystal is placed on the system board, adjust the potentiometer so that the voltage at the input of the inverter is in the transition region between a logic level 0 and a logic level 1 (i.e., approximately 1.5 V). Then install the crystal.

In a third-overtone oscillator, the crystal fundamental frequency must be attenuated so that oscillation is at the third harmonic. This is achieved with an LC circuit that filters out the fundamental, thus allowing oscillation at the third harmonic. The impedance of the LC network must be inductive at the crystal fundamental frequency and capacitive at the third harmonic. The impedance of the LC circuit is given by:

$$Z(\omega) = \frac{\frac{L}{C}}{j\left[L - \frac{1}{\omega C}\right]} \tag{3}$$

Therefore, the LC circuit has a pole at:

$$\omega_p = \frac{1}{\sqrt{LC}} \tag{4}$$

At frequencies significantly lower than ω_p, the $1/(\omega C)$ term in (3) becomes the dominating term while ωL can be neglected. This gives:

$$z(\omega) = j\omega L, \quad \text{for} \quad \omega << \omega_p \tag{5}$$

In (5), the LC circuit appears inductive at frequencies lower than ωp. On the other hand, at frequencies much higher than ωp, the ωL term is the dominant term in (3), and $1/(\omega C)$ can be neglected. This gives:

$$z(\omega) = \frac{1}{j\omega C} \qquad \text{for } \omega >> \omega_p \tag{6}$$

The LC circuit in (6) appears increasingly capacitive as frequency increases above ω_p. This is shown in Figure 27, which is a plot of the magnitude of the impedance of the LC circuit of Figure 26 versus frequency.

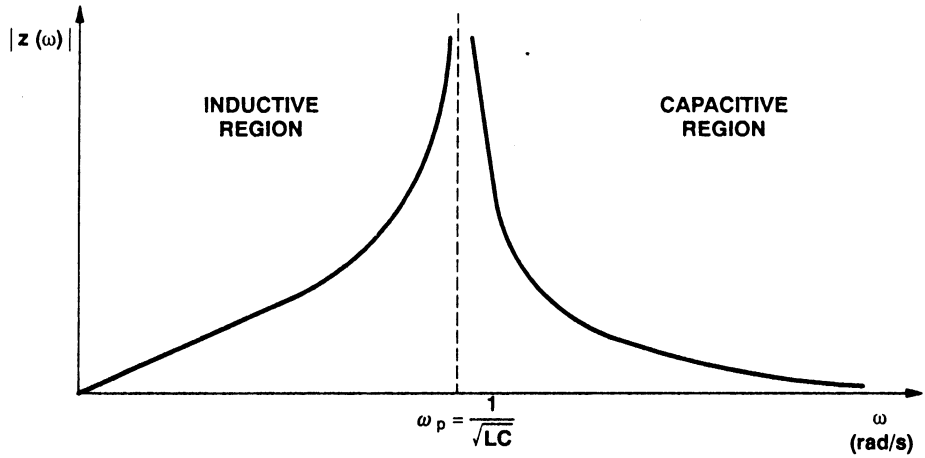
Based on the discussion above, the design of the LC circuit proceeds as follows: Choose the pole frequency ω_p approximately halfway between the crystal fundamental and the third harmonic. The circuit now appears inductive at the fundamental frequency and capacitive at the third harmonic.

In the oscillator of Figure 26, choose ω_p = 166.5 rads/s for the 40.96 MHz design or ω_p = 223.6 for the 51.2 MHz design. These angular frequencies lie approximately halfway between the respective fundamentals and third harmonics. Choose C = 20 pF. The appropriate value of L may then be computed using (4). Values of L for three different TMS320C2x devices operating at different frequencies are tabulated in Table 12.

Table 12. Values of $f_{crystal}$ and L for TMS320C2x Devices

	$f_{crystal}$ (MHz)	L (μH)
TMS320C25	40.96	1.8
TMS320C25-50	51.20	1.0
TMS320E25	40.96	1.8

Figure 27. Magnitude of the Impedance of the Oscillator LC Network



The 0.1 μF capacitor in series with the 1.8 μH inductor is a coupling capacitor, requiring no DC path to ground. The 74AS04 inverter is included to shorten the rise and fall times of the waveform generated by the oscillator.

Consider the case where the TTL inverter goes low. In this case, the current flowing through the 10-kΩ resistor is less than 5 V/10-kΩ = 0.5 mA. This is an acceptable current level since the 74AS04 inverter can sink up to 20 mA.

The output of the oscillator drives the CLKIN input of the TMS320C2x, thus providing the four phases required for each machine cycle. With a 40.96 MHz input clock frequency, the TMS320C2x machine cycle is 97.6 ns.

In summary, the system control circuitry performs functions that, while often overlooked, are critical for proper system initialization and operation. The powerup reset circuit assures that a reset of the part occurs only after the oscillator is running and stabilized. The oscillator circuit described allows the use of third-overtone crystals that are more readily available at frequencies above 20 MHz.

Interfacing Peripherals

Most DSP systems implement some amount of I/O using peripherals in addition to any memory included in the system. Quite commonly this includes analog input and output, which can

be performed through the parallel and serial I/O ports on the TMS320C2x. In this section, hardware interfaces of the TMS320C2x to a codec, an analog-to-digital converter (A/D), and a digital-to-analog converter (D/A) are described. Interfacing TMS320 devices to the Texas Instruments TLC32040 Analog Interface Chip is described in the applications report *Interfacing the TMS320 Family to the TLC32040 Family* found in this book.

Combo-Codec Interface

In speech, telecommunications, and many other applications that require low-cost analog-to-digital and digital-to-analog converters, a combo-codec may be used. Combo-codecs are single-chip pulse-code-modulated encoders and decoders (PCM codecs). They are designed to perform the encoding (A/D conversion) and decoding (D/A conversion), as well as the antialiasing and smoothing filtering functions. Since combo-codecs perform these functions in a single 300-mil DIP package at low cost, they are extremely economical for providing system data conversion functions. The design presented here uses a Texas Instruments TCM29C16 codec, interfaced using the serial port of the TMS320C2x.

TMS320C2x Serial Port

The TMS320C2x serial port provides direct synchronous communication with serial devices. The interface signals are compatible with codecs and other serial components so that minimum external hardware is required. Externally, the serial port interface is implemented using the following pins on the TMS320C2x:

- DX (transmitted serial data)
- CLKX (transmit clock)
- FSX (transmit framing synchronization signal)
- DR (received serial data)
- CLKR (receive clock)
- FSR (receive framing synchronization signal)

Data on DX and DR are clocked by CLKX and CLKR, respectively. These clocks are only required during serial transfers. Note that this is different from the TMS32020 serial port in which the clocks must be present at all times if the serial port is being used. Also, the TMS320C2x serial port is double-buffered while that of the TMS32020 is not.

Serial port transfers are initiated by framing pulses on the FSX and FSR pins for transmit and receive operations respectively. For transmit operations, the FSX pin can be configured as an input or output. This option is selected by the transmit mode (TXM) bit of status register ST1.[1] In this design, FSX is assumed to be configured as an input; therefore, transmit operations are initiated by a framing pulse on the FSX pin. Upon completion of receive and transmit operations, an RINT (serial port receive interrupt) and an XINT (serial port transmit interrupt) are generated, respectively.

The format (FO) bit of status register ST1 is used to select the format (8-bit byte or 16-bit word) of the data to be received or transmitted. For interfacing the TMS320C2x to a codec, the format bit should be set to one, formatting the data in 8-bit bytes.[1]

After the information from the codec is received by the TMS320C2x, the μ- or A-law companded data must be converted back to a linear representation for use in the TMS320C2x. Software companding routines appropriate for use on the TMS320C2x are provided in the book, *Digital Signal Processing with the TMS320 Family Volume 1.*[2]

The software required to initialize the TMS320C2x-codec interface is shown next. The initialization routine should include the following:

```
INIT        DINT            ; Disable interrupts
            FORT   1        ; Set 8-bit data format
            LACK   10h
            LDPK   0
            SACL   DMA4     ; Enable RINT (through IMR)
            *
            *
            *
            EINT            ; Enable interrupts
```

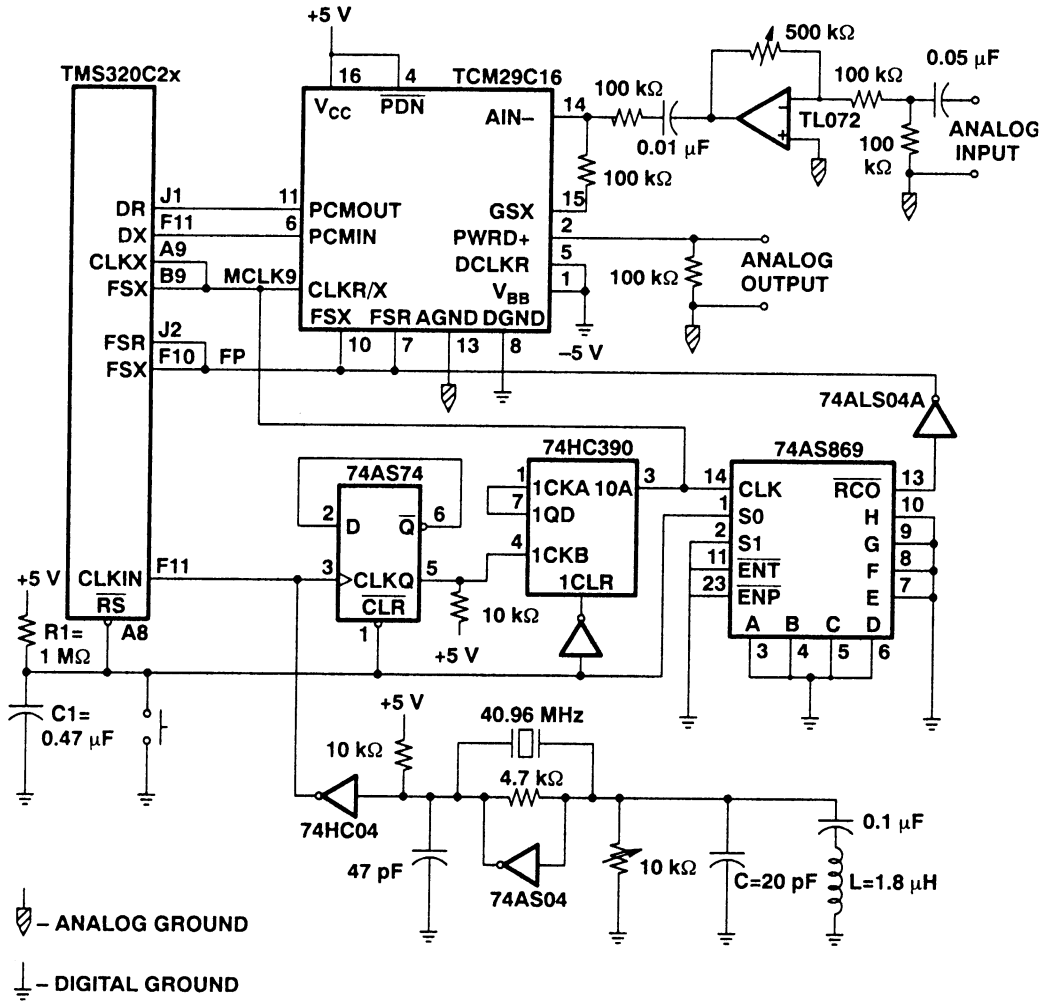
Note that since reset initializes the TXM (transmit mode) and FSM (frame synchronization mode) bits to the values required by this interface, it was not necessary to explicitly initialize these values in the routine shown above. However, in digital communications with peripherals/devices/ports (T_1 trunks) that do not require a framing pulse for every byte/word transmitted, the FSM bit must be set to 0 using the RFSM instruction.[1]

The interrupt mask register (IMR) located at data memory location 4h of the TMS320C2x data memory is used to enable the serial port receive interrupts (RINT). To access that memory location, the data page pointer must be set to zero. Also, the data page pointer must be initialized after reset since its contents are random at powerup. A value of 10h in the IMR enables only the RINT; all other interrupt sources are disabled.

Interrupts are disabled upon reset. Before exiting the initialization routine, interrupts are reenabled with the EINT instruction.

The hardware interface between the TMS320C2x and the TCM29C16 combo-codec is shown in Figure 28.

Figure 28. Interface of the TMS320C2x to the TCM29C16 Codec



Clock Divider Circuit

A combo-codec configured in the fixed-data-rate mode requires the following external clock signals:

A 2.048-MHz clock to be used as the masterclock, and

8-kHz framing pulses required to initialize the data transfers.

Both of these signals can be derived from the 40.96 MHz system clock with appropriate divider circuitry. This is the primary justification for selecting 40.96 MHz as the system clock fre-

quency. The clock divider circuit consists of a 74AS74 D-type flip-flop, a 74HC390 decade counter, and a 74AS869 8-bit up/down counter. The hardware connections between these devices are shown in Figure 28.

To generate the 2.048-MHz master clock for the combo-codec, a division by 20 of the 40.96-MHz system clock is required. The 74HC390 contains on-chip two divide-by-2 and two divide-by-5 counters. Since the 74HC390 cannot be clocked with frequencies above approximately 27 MHz, a 74AS74 configured as a T-type flip-flop is used. This implements a divide-by-2 of the 40.96-MHz clock, thus making the output of the 74AS74 slow enough (20.48 MHz) to properly clock the 74HC390. The 10-kΩ pullup resistor shown in Figure 28 is used to ensure the compatibility between the logic levels of the TTL (74AS74) and HCMOS (74HC390) devices.

The 74HC390 is first used to implement a divide-by-5, which appears at the output pin $1Q_D$ (pin #7) of the 74HC390 (see Figure 28). This in turn drives the divide-by-2 counter, at the output of which (pin $1Q_A$) the 2.048 MHz clock appears. Note that the divide-by-5 precedes the divide-by-2 because the codec requires a clock with a minimum duty cycle of 40 percent, while the output of the divide-by-5 has a duty cycle of only 20 percent. By following the divide-by-5 counter with the divide-by-2, the duty cycle at the output of the 74HC390 is 50 percent.

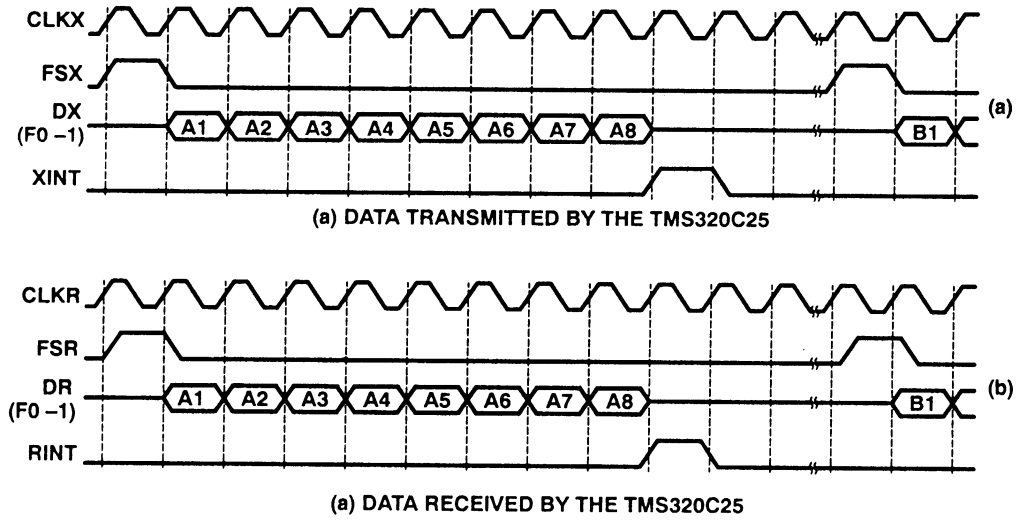
The 74AS869 is configured to count down (S0 = 1 and S1 = 0 in Figure 28); therefore, the counting sequence is 255, 254, ..., 1, 0, 255, ..., and so on. The ripple carry output generates a low-level pulse while the count is zero. The duration of this pulse is one input clock cycle, i.e., 488 ns. The frequency of the ripple carry output is 2.048 MHz/256 = 8 kHz. By inverting this signal, positive pulses at 8 kHz are generated. These pulses are used by the TMS320C2x and codec as framing pulses to initiate data transfers.

TMS320C2x-Codec Interface

The TMS320C2x interfaces directly to the codec, as shown in Figure 28, with no additional logic required. The PCM μ-law data generated by the codec at the PCMOUT pin is read by the TMS320C2x from the data receive (DR) pin, which is internally connected to the receive serial register (RSR).[1] The data transmitted from the data transmit (DX) pin of the TMS320C2x is received by the PCMIN input of the codec. During the digital-to-analog conversion, this data is converted from μ-law PCM to linear. The resulting analog waveform is lowpass-filtered by the codec's internal smoothing filter. Therefore, no additional filtering is required at the codec output (PWRO+).

The timing diagram of the TMS320C2x-codec interface is shown in Figure 29.

Figure 29. Interface Timing of the TMS320C2x to the TCM29C16 Codec



As indicated in Figure 29, both the transmit and receive operations are initiated by a framing pulse on the FSX and FSR pins of the TMS320C2x and the codec. The receive and transmit interrupts shown in Figure 29 occur only if they are enabled. Note that Figure 29 corresponds to the burst-mode serial port operation of the TMS320C2x.[1] Continuous-mode operation using framing pulses or without framing pulses is also available.

Analog Input

The level of the analog input signal is controlled using the TL072 opamp connected in the inverting configuration (see Figure 28). Using the 500-kΩ potentiometer, the gain of this circuit can be varied from 0 to 5. The output of the 0.01-μF coupling capacitor drives the TCM29C16's internal opamp. This opamp is connected in the inverting configuration with unity gain (feedback and input impedances having the same value of 100 kΩ).

In summary, codecs, combo-codecs in particular, are most effective in serving DSP system data-conversion requirements. These inexpensive devices interface directly to the TMS320C2x, occupy minimal board space, and perform both filtering and data conversion functions. Codecs interface to the TMS320C2x by means of the serial port and provide a companded, PCM-coded digital representation of analog input samples. This PCM code is easily translated into a linear form by the TMS320C2x for use in processing. Interface to the codec on the serial port is initialized by a simple software routine in the TMS320C2x.

Interfacing an Analog-to-Digital (A/D) Converter

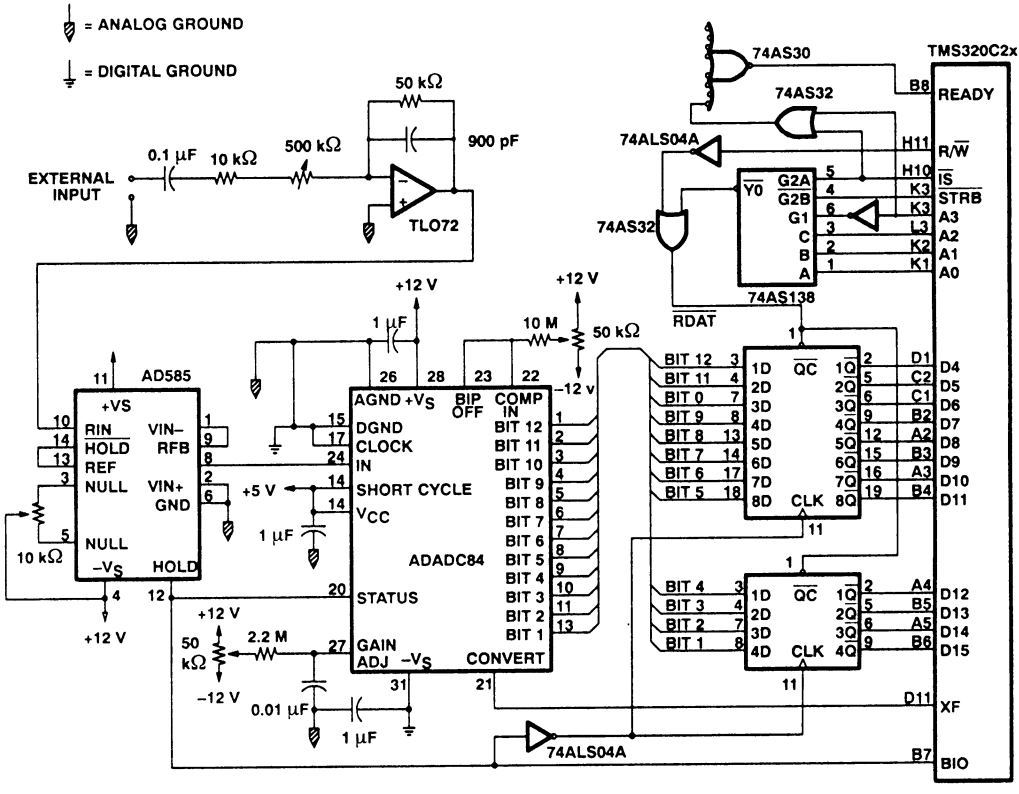
Many digital signal processing applications require a higher level of signal quality than that offered by the eight companded bits of a combo-codec. For these applications, linear analog-to-dig-

ital converters with 10, 12, or 14 bits are commonly used. The improved signal quality obtained with these converters, however, is accompanied by increased system complexity and higher cost.

The hardware interface of a 12-bit linear analog-to-digital (A/D) converter to the TMS320C2x is discussed in this subsection. In this design, the A/D is mapped into the input/output (I/O) space of the TMS320C2x. The distinction between the I/O space and the program and data spaces is made by using the $\overline{IS}$ pin. This pin goes active (low) when the I/O space is accessed. The TMS320C2x space contains 16 ports that can be read from or written to. These ports are accessed with the IN and OUT instructions.[1]

The hardware design of this interface is shown in Figure 30. This design utilizes an antialiasing (lowpass) filter, the Analog Devices' AD585 sample-and-hold and ADADC84 analog-to-digital converter, two 74AS534 octal D-type flip-flops, plus additional logic to generate the READY signal.

Figure 30. Interface of the ADADC84 to the TMS320C2x



The design of Figure 30 consists of two sections: the analog-to-digital conversion and the interface to the TMS320C2x. Each of these sections is considered separately.

Analog-to-Digital Conversion

The analog-to-digital conversion section of this interface performs the function of sampling and coding the input waveform. This circuit consists of the antialiasing filter, the sample-and-hold, and the analog-to-digital converter.

To avoid distortion during an analog-to-digital conversion, the sampling theorem states that the analog signal must contain no frequency components greater than half the sampling frequency. If this condition is not met, distortion occurs in the form of aliasing; i.e., high-frequency components are superimposed on the low frequencies of the signal spectrum. To avoid this phenomenon, an antialiasing (lowpass) filter is used.

In the design of Figure 30, the antialiasing filter is implemented using a TL072 opamp connected in the inverting configuration. The gain of the opamp is determined by the values of two fixed resistors (10 kΩ and 50 kΩ) and a 500-kΩ potentiometer. The resistance of the potentiometer inversely varies the gain of the opamp. The minimum gain of 0.098 (50 kΩ/510 kΩ) is reached when the potentiometer is 500 kΩ. The maximum gain of 5 (50 kΩ/10kΩ) is achieved when the potentiometer is decreased to zero resistance.

To satisfy the sampling theorem, the cutoff frequency of the antialiasing filter must be less than half the sampling rate. In the design of Figure 30, the 900 pF capacitor in the feedback path introduces a pole at the frequency f defined by:

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi(50k\Omega)(0.9)nF)} = 3.5kHz$$

After 3.5 kHz, the frequency response of the filter drops by 6 dB per decade. This rejection, however, may not be adequate for some applications. In such cases, a lowpass filter of higher order is required. Such a filter is presented in the next subsection.

The output of the antialiasing filter is connected to the input of the AD585 sample-and-hold, which is configured for a gain of –1. The operation of this device is controlled by the HOLD input. When HOLD is low, the output of the sample-and-hold (V_{OUT}) follows the input (lowpass version of the external input). When HOLD is high, the output stays constant. The time from HOLD high to output stable is referred to as the aperture time, specified as 35 ns for the AD585.

A/D conversions are implemented by the ADADC84, a 12-bit linear A/D converter in which data is represented in complementary two's-complement form. A conversion begins when the CONVERT input goes high. The XF (external flag) output of the TMS320C2x is used to drive the CONVERT input. Since the XF pin is software controlled, the TMS320C2x internal timer may be used to generate programmable sampling rates. This is discussed in more detail later.

When CONVERT goes high, the ADADC84 begins the conversion and STATUS goes high. This puts the AD585 in the hold mode. The A/D conversion lasts for 10 μs, with the MSB decision made approximately 820 ns after STATUS goes high. Note that the aperture time of the AD585 is only 35 ns, and as a result, the input to the A/D converter is stable well before the time the MSB decision is made. The LSB decision is made at least 40 ns before STATUS goes low. When STATUS goes low, the AD585 enters the sample mode with a gain of –1; i.e., the output follows the inverted

input waveform. As shown in Figure 30, the $\overline{BIO}$ pin of the TMS320C2x is connected to STATUS. By polling $\overline{BIO}$, the TMS320C2x can detect when an A/D conversion is completed.

The falling edge of STATUS generates a rising edge at the clock inputs of the 74AS534s. This rising edge clocks the ADADC84 data into the 74AS534s. Since the LSB decision is made 40 ns before STATUS goes low, the 3 ns setup time for the 74AS534s is met. Since the 74AS534s are inverting-type flip-flops, the ADADC84 outputs are complemented to give data in two's-complement form. This data, however, does not appear on the TMS320C2x data bus until the output buffers of the 74AS534s are enabled.

Interface to the TMS320C2x

The interface logic in Figure 30 is used to perform the following functions:

- Generate READY, and
- Enable the output buffers of the 74AS534s so that the TMS320C2x can read the data from the A/D conversion

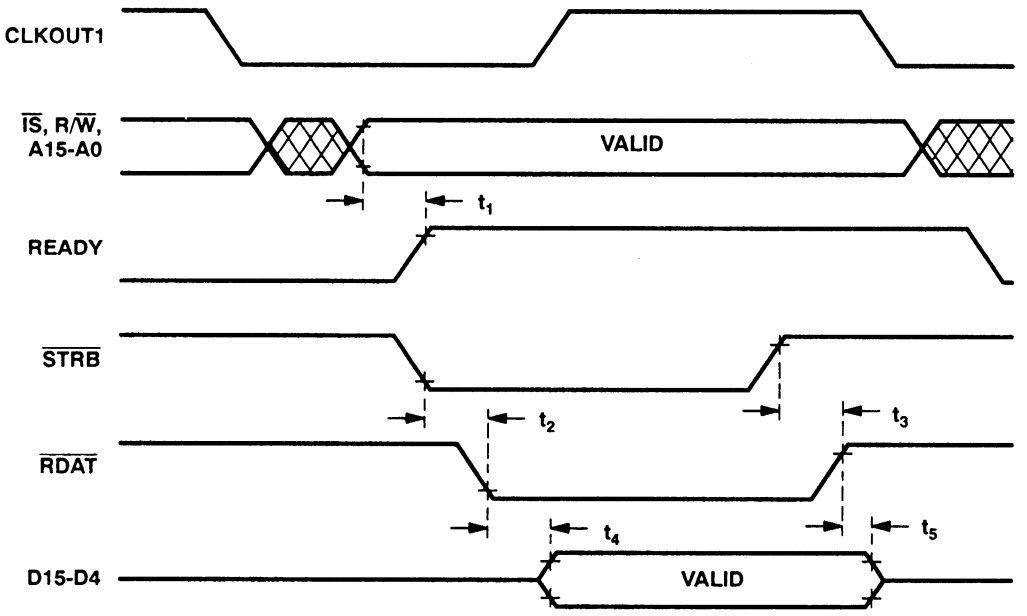
To meet the TMS320C2x READY timing requirements, two separate address decoding schemes are used to implement these two functions. One decoding scheme is used for READY, and a second is used to enable the I/O-mapped devices.

The address decoding for READY is implmented with the 74AS32 positive-OR gate. The output of the 74AS32 goes low when both $\overline{IS}$ and A3 go low; i.e., access to ports 0 through 7 is requested. This scheme generates READY for devices that do not require wait-states. I/O devices that generate one or more wait-states can utilize ports 8 through 15.

To enable the I/O devices, a 74AS138 is used. Outputs $\overline{Y0}$ through $\overline{Y7}$ of the 74AS138 can be used to enable the devices 0 through 7, respectively. In Figure 30, $\overline{Y0}$ is used to enable a read from the A/D converter. Note that $\overline{Y0}$ is ORed with the inverted R/$\overline{W}$. This prevents the bus conflict that occurs if the TMS320C2x writes to port 0.

The timing diagram of a TMS320C2x read from port 0 is shown in Figure 31.

Figure 31. Interface Timing of the ADADC84 to the TMS320C2x



Time t_1 in Figure 31 indicates the time from valid address to READY high. This is less than 10.8 ns, the maximum propagation delay through the READY generation logic. Therefore, the 18-ns READY timing requirement (at 40 MHz) is met.

RDAT in Figure 31 is used to enable the output buffers of the 74AS534s. $\overline{\text{RDAT}}$ goes active (low) no later than $t_2 = t_{p(74AS138)} + t_{p(74AS32)} = 14.3$ ns after $\overline{\text{STRB}}$ goes low ($\overline{\text{STRB}}$ is used to enable the 74AS138). With a low level on the output control ($\overline{\text{OC}}$) of the 74AS534s, valid data appears on the TMS320C2x data bus within $t_4 = 10$ ns. The worst-case access time is $t_2 + t_4 = 24.3$ ns from $\overline{\text{STRB}}$ going low, which is less than the 27 ns required by the TMS320C2x.

When $\overline{\text{STRB}}$ goes high, $\overline{\text{RDAT}}$ follows within $t_3 = 13.3$ ns. With a high logic level on the output control ($\overline{\text{OC}}$), the output buffers of the 74AS534s enter a high-impedance state within $t_5 = 6$ ns. Since $t_3 + t_5 = 19.3$ ns after $\overline{\text{STRB}}$ goes high, the 74AS534s have entered a high-impedance state, and no bus conflict will occur if a write cycle follows the present read cycle.

Table 13 summarizes the most critical timing parameters of the ADADC84 interface to the TMS320C2x.

Table 13. Timing Parameters of the ADADC84 Interface to the TMS320C2x

Description	Symbol Used in Figure 3	Value
Address valid to READY valid	t_1	10.8 ns (max)
$\overline{\text{STRB}}$ low to $\overline{\text{RDAT}}$ low	t_2	14.3 ns (max)
$\overline{\text{STRB}}$ high to $\overline{\text{RDAT}}$ high	t_3	13.3 ns (max)
Progagation delay through the 74AS534 ($\overline{\text{QC}}$ to $\overline{\text{Q}}$)	t_4	10.0 ns (max)
74AS534 disable time	t_5	6.0 ns (max)

Controlling A/D Conversions with the TMS320C2x Timer

The TMS320C2x timer can generate periodic interrupts that may be used to set the A/D sampling frequency. The TMS320C2x timer logic consists of a 16-bit timer register and a 16-bit period register. At every CLKOUT1 cycle, the timer register is decremented by one. When the count reaches zero, a timer interrupt (TINT) is generated. In the next cycle, the contents of the period (PRD) register are loaded into the timer register. Therefore, a timer interrupt is generated every PRD + 1 cycle of CLKOUT1, and the frequency of these interrupts is CLKOUT1/(PRD + 1).

As an example, consider a TMS320C2x operating at 40 MHz. The design of Figure 30 is utilized to interface the A/D converter to the TMS320C2x. A sampling rate of 10 kHz is desired.

To generate timer interrupts at the 10 KHz sampling rate, the value of the period register is calculated as follows: Since

$$f_s = \frac{\text{CLKOUT1}}{\text{PRD} + 1}$$

the period register is

$$\text{PRD} = \frac{\text{CLKOUT}}{f_s} - 1$$

With CLKOUT1 = 10 MHz and f_S = 10 kHz, the value of the period register is PRD = 999. By loading the period register (data memory location 3) with 999, timer interrupts (if enabled) occur at a 10 kHz frequency. This can be implemented with the following TMS320C2x source code:

```
LDPK   0        ; Point to Data Page #0
LALK   999      ; ACC 999
SACL   DMA3     ; Period Register ACC
LACL   8        ; Enable TINT
OR     DMA4     ; through
SACL   DMA4     ; the IMR
```

To start the A/D conversion, the interrupt service routine must generate a positive pulse on the XF output. This can be implemented with the following code:

```
ISE    SXF      ; Set external flag (XF)
       RXF      ; Clear external flag (XF)
       EINT     ; Enable interrupts
       RET
```

Note that upon entering the interrupt service routine, the interrupts are disabled. Interrupts are reenabled by the EINT instruction just before exiting the interrupt service routine. Also, the conversion pulse that this routine generates is 100 ns long, easily meeting the 50-ns minimum conversion pulse width required by the ADADC84.

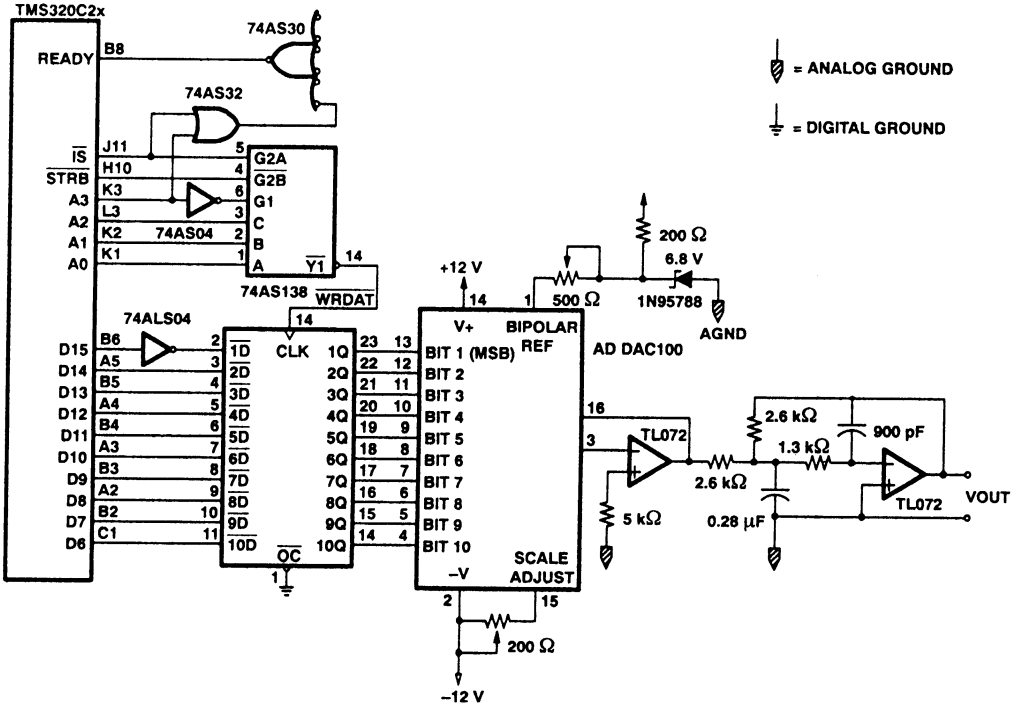
To summarize, 10-bit to more than 14-bit linear A/D converters are often used to perform data conversions in DSP systems that require more resolution than is provided by codecs. The circuit shown in Figure 30 describes the interface of an A/D conversion subsystem to the

TMS320C2x. This subsystem contains antialiasing filters, a sample-and-hold circuit, and a 12-bit A/D converter. Communication with the TMS320C2x is provided via the I/O space. The A/D converter is isolated from the processor's data bus by high-impedance buffers when data transfers are not being performed. The TMS320C2x's internal timer is used to establish the A/D sample rates, thus reducing system logic requirements.

Interfacing a Digital-to-Analog (D/A) Converter

This subsection discusses the hardware interface of a 10-bit digital-to-analog converter to the TMS320C2x. The design, shown in Figure 32, utilizes the Analog Device's ADDAC100 digital-to-analog converter, a 74AS822 10-bit flip-flop, a smoothing filter, plus additional logic to generate READY.

Figure 32. Interface of the ADDAC100 to the TMS320C2x



This design consists of three sections: the interface to the TMS320C2x, the D/A converter, and the smoothing filter. Each of these sections is considered separately.

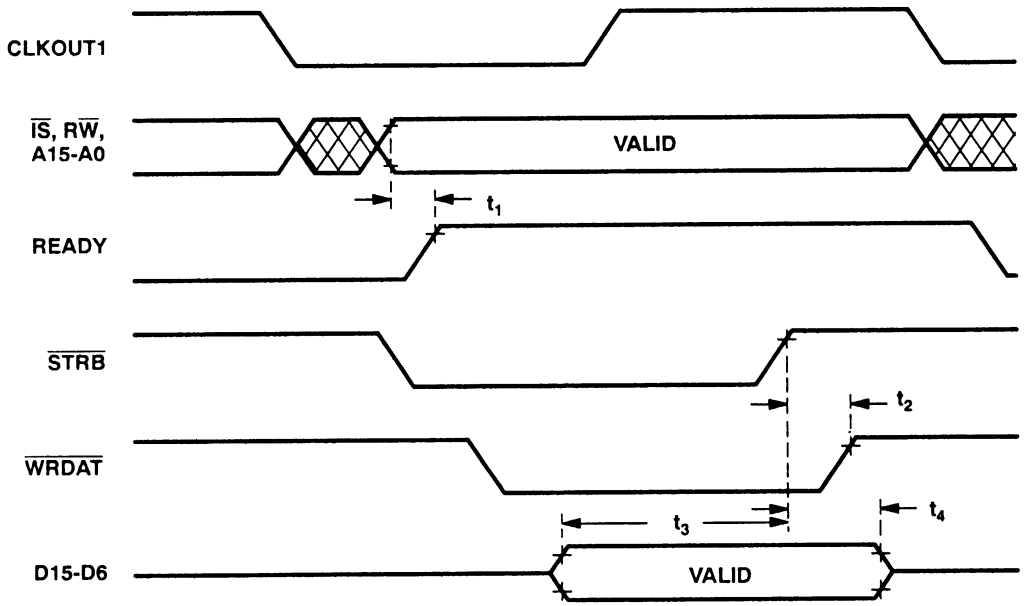
Interface to the TMS320C2x

The 74AS822 is used to latch the data from the TMS320C2x. Since the output control ($\overline{OC}$) of the 74AS822 is always active (grounded), the latched data is available at the inputs of the D/A converter immediately following a write from the TMS320C2x. In bipolar mode, the ADDAC100 accepts data in complementary offset binary form. By inverting the MSB of the two's-complement

data from the TMS320C2x, the data input to the 74AS822 is converted to offset binary form. This data is inverted by the 74AS822 so that the input to the ADDAC100 becomes complementary offset binary form.

The circuit shown in Figure 32 utilizes the same address decoding technique used for the analog-to-digital converter interface. This technique maps devices that require no wait-states into ports 0 through 7. Ports 8 through 15 are used for devices that require one or more wait-states. In this design, the D/A converter is mapped into port 1 of the TMS320C2x I/O space. The timing diagram for a write to the D/A is shown in Figure 33.

Figure 33. Interface Timing of the ADDAC100 to the TMS320C2x



When port 1 is addressed, $\overline{\text{WRDAT}}$ goes low. No later than t_2 = 7.5 ns after $\overline{\text{STRB}}$ goes high, $\overline{\text{WRDAT}}$ follows. This rising edge of $\overline{\text{WRDAT}}$ clocks the data into the 74AS822. The minimum setup time for the data before $\overline{\text{WRDAT}}$ goes high is t_3 min + t_2 min (see Figure 33). Time t_3 min is the minimum setup time for the TMS320C2x data before $\overline{\text{STRB}}$ goes high (30 ns), minus the maximum propagation delay through the 74ALS04 (11 ns). Time t_2 min is the minimum propagation delay through the 74AS138 (2 ns). Therefore, the minimum setup time for the data before $\overline{\text{WRDAT}}$ goes high is 21 ns, which is greater than the 6-ns minimum setup time required by the 74AS822.

Table 14 summarizes the most critical timing parameters of the ADDAC100 interface to the TMS320C2x.

Table 14. Timing Parameters of the ADDAC100 Interface to the TMS320C2x

Description	Symbol Used in Figure 3	Value
Address valid to READY valid	t_1	10.8 ns (max)
STRB high to WRDAT high	t_2	7.5 ns (max)
Data setup time before STRB high	t_3	19.0 ns (min)
Data setup time before WRDAT high	$t_3 + t_2$	21.0 ns (min)
Data hold time from STRB high	t_4	15.0 ns (min)
Data hold time from WRDAT high	$t_4 - t_2$	7.5 ns (min)

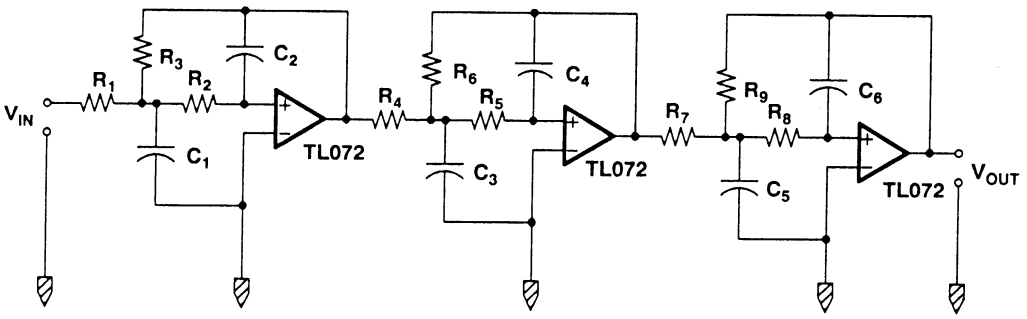
D/A Converter

The ADDAC100 10-bit digital-to-analog converter converts a digital input to an output current. The standard current-to-voltage conversion is implemented using the TL072 opamp. This is the opamp closest to the ADDAC100 in Figure 32. The offset and gain ajustments are implemented with the 500-Ω and 200-Ω potentiometers, respectively.

Smoothing Filter

The output of the ADDAC100 contains high-frequency components to be removed by the smoothing filter. In the design of Figure 32, this filter is implemented with the TL072 opamp configured to implement a second-order lowpass filter with a cutoff frequency around 1.7 KHz. For some applications, however, a rejection of 12 dB per decade is not adequate. A design that implements a sixth-order lowpass filter is shown in Figure 34. This design is a cascade of three opamps, each implementing a second-order section.

Figure 34. Sixth-Order Lowpass Filter Used for Antialiasing and Smoothing Filter Operations



= Analog Ground

The design of Figure 34 is used to implement the antialiasing and smoothing filtering operations in the TMS32010 Analog Interface Board. The cutoff frequency of this filter depends on the values of the passive components. The values of these components for several cutoff frequencies are shown in Table 15.[3]

Table 15. Lowpass Filter Component Values for Various Frequencies

f	1.7 kHz	4.7 kHz	7.7 kHz	10 kHz	12 kHz	16 kHz	20 kHz
R1	2.588	2.588	2.588	2.588	2.588	2.588	2.588
C1	0.280	0.101	0.0617	0.0475	0.0396	0.0297	0.0238
R2	1.294	1.294	1.294	1.294	1.294	1.294	1.294
R3	2.588	2.588	2.588	2.588	2.588	2.588	2.588
C2	0.00936	0.00339	0.00207	0.00160	0.00133	0.000995	0.000796
R4	7.071	7.071	7.071	7.071	7.071	7.071	7.071
C3	0.0375	0.0136	0.00827	0.00637	0.00531	0.00398	0.00318
R5	3.536	3.536	3.536	3.536	3.536	3.536	2.536
R6	7.071	7.071	7.071	7.071	7.071	7.071	7.071
C4	0.00936	0.00339	0.00207	0.00160	0.00133	0.000995	0.000796
R7	9.659	9.659	9.659	9.659	9.659	9.659	9.659
C5	0.0201	0.00726	0.00443	0.00341	0.00284	0.00213	0.00171
R8	4.830	4.830	4.830	4.830	4.830	4.830	4.830
R9	9.659	9.659	9.659	9.659	9.659	9.659	9.659
C6	0.00936	0.00339	0.00207	0.00160	0.00133	0.000995	0.000796

Note: The unit for resistance is kΩ
The unit for capacitance is μF
The above values are not industry-standard values

In summary, the 10-bit linear D/A converter provides analog output for the TMS320C2x. The D/A converter is interfaced to the processor through the I/O space and is driven by latches that store the digital data for the current sample until the next sample period. A smoothing filter provides final analog signal reconstruction by eliminating extraneous high-frequency components in the output waveform.

Summary

The interface of memories and peripherals to the TMS320C2x has been described in this application report. Both direct interfaces and interfaces that utilize address decoding have been considered, with special attention given to READY timing requirements. The design techniques used in these interfaces can be extended to encompass interface of other devices to the TMS320C2x.

References

1) *Second-Generation TMS320 User's Guide* (literature number SPRU014A), Texas Instruments (1989).
2) *Digital Signal Processing Applications with the TMS320 Family, Volume 1* (literature number SPRA012A), Texas Instruments (1986).
3) *TMS32010 Analog Interface Board User's Guide* (literature number SPRU006), Texas Instruments (1983).
4) The *TTL Data Book Volume 2* (literature number SDLD001), Texas Instruments (1985).
5) The *TTL Data Book Volume 3* (literature number SDAD001A), Texas Instruments (1984).
6) *MOS Data Book*, Micron Technology, Inc. (1990).
7) *CMOS/BiCMOS Data Book*, Cypress Semiconductor (1989).