SPRA014A

Hardware Interfacing to the TMS320C25

Digital Signal Processing Product Application

TEXAS INSTRUMENTS

Hardware Interfacing to the TMS320C25

Product Application

George Troullinos   Jon Bradley

Digital Signal Processing
Applications Engineering

TEXAS
INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes in the devices or the device specifications identified in this publication without notice. TI advises its customers to obtain the latest version of device specifications to verify, before placing orders, that the information being relied upon by the customer is current.

In the absence of written agreement to the contrary, TI assumes no liability for TI applications assistance, customer's product design, or infringement of patents or copyrights of third parties by or arising from use of semiconductor devices described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor device might be or are used.

Contents

List of Illustrations

List of Tables

Introduction

The TMS320C25 Digital Signal Processor, a CMOS pin-compatible version of the NMOS TMS32020, has the power and flexibility to satisfy a wide range of system requirements. The 128K 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 the TMS320C25. In other applications, the internal program and data resources of the TMS320C25 can be used to implement single-chip solutions. Peripheral devices can be interfaced to the TMS320C25 to perform analog signal acquisition at different levels of signal quality.

This report suggests hardware design techniques for interfacing memories and peripherals to the TMS320C25. Differences between the TMS320C25 and the TMS32020 are pointed out when appropriate. The first section presents the design interfaces of PROMs, EPROMs, and static RAMs to the TMS320C25. 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 TMS320C25. All the interfaces in this report have been built and tested to verify their operation.

Interfacing Memories

This section describes the interface of the TMS320C25 with PROMs, EPROMs, and static RAMs. The TMS320C25 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 TMS320C25 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 TMS320C25.

When wait states are required, the number of wait states depends on the memory access time (see the Interfacing EPROMs subsection). With no wait states, the READY input to the TMS320C25 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 TMS320C25 enters a wait state.

The TMS320C25 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 $\overline{\text{PS}}$ (program space) and $\overline{\text{DS}}$ (data space) pins. A third space, the I/O space, is also available for interfacing with peripherals. This space is selected by the $\overline{\text{IS}}$ (I/O space) pin, and is discussed in the Interfacing Peripherals section.

The following brief discussion describes the TMS320C25 read and write cycles. A more complete discussion is contained in the *TMS320C25 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 TMS320C25 is running at 40 MHz; i.e., $Q = 25$ ns. The memory read and write timings are shown in Figure 1. 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 ($\overline{\text{PS}}$, $\overline{\text{DS}}$, or $\overline{\text{IS}}$) becomes valid. R/$\overline{\text{W}}$ goes high to indicate a read cycle.

2. $\overline{\text{STRB}}$ goes low no less than $t_{su(A)} = Q - 12$ ns after the address bus is valid.

3. Early in the second half of the cycle, the READY input is sampled. READY must be stable (low or high) at the TMS320C25 no later than $t_{d(SL-R)} = Q - 20$ ns after $\overline{\text{STRB}}$ goes low.

4. With no wait states (READY is high), data must be available no later than $t_{a(SL)} = 2Q - 23$ ns after $\overline{\text{STRB}}$ 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/$\overline{\text{W}}$ goes low to indicate a write cycle.

2. The data bus begins to be driven approximately concurrently with $\overline{\text{STRB}}$ going low.

3. The data bus enters a high-impedance state no later than $t_{dis(D)} = Q + 15$ ns after $\overline{\text{STRB}}$ goes high.

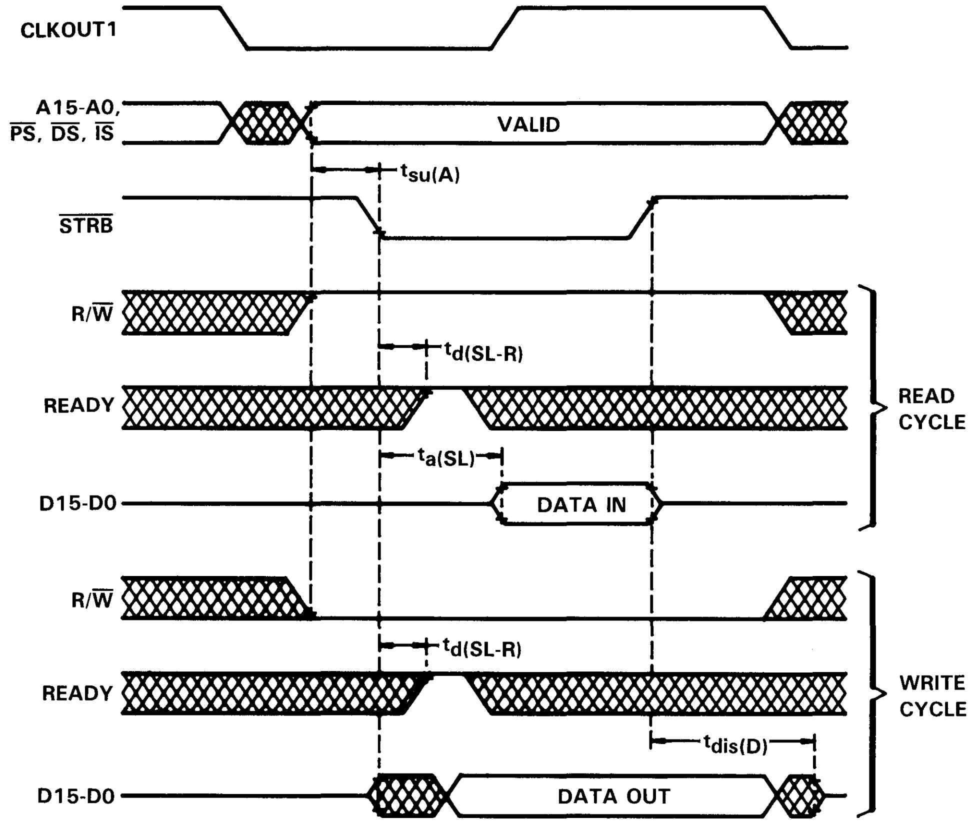


Figure 1. Read and Write Timings

Interfacing PROMs

A convenient means of implementing program memory in a TMS320C25 system is provided through the use of PROMs. Two separate approaches for interfacing PROMs to the TMS320C25 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

A design using the first approach is shown in Figure 2. In this design, the TMS320C25 is interfaced with the Texas Instruments TBP38L165-35, a low-power 2K × 8-bit PROM. The interface timing for the design of Figure 2 is shown in Figure 3.

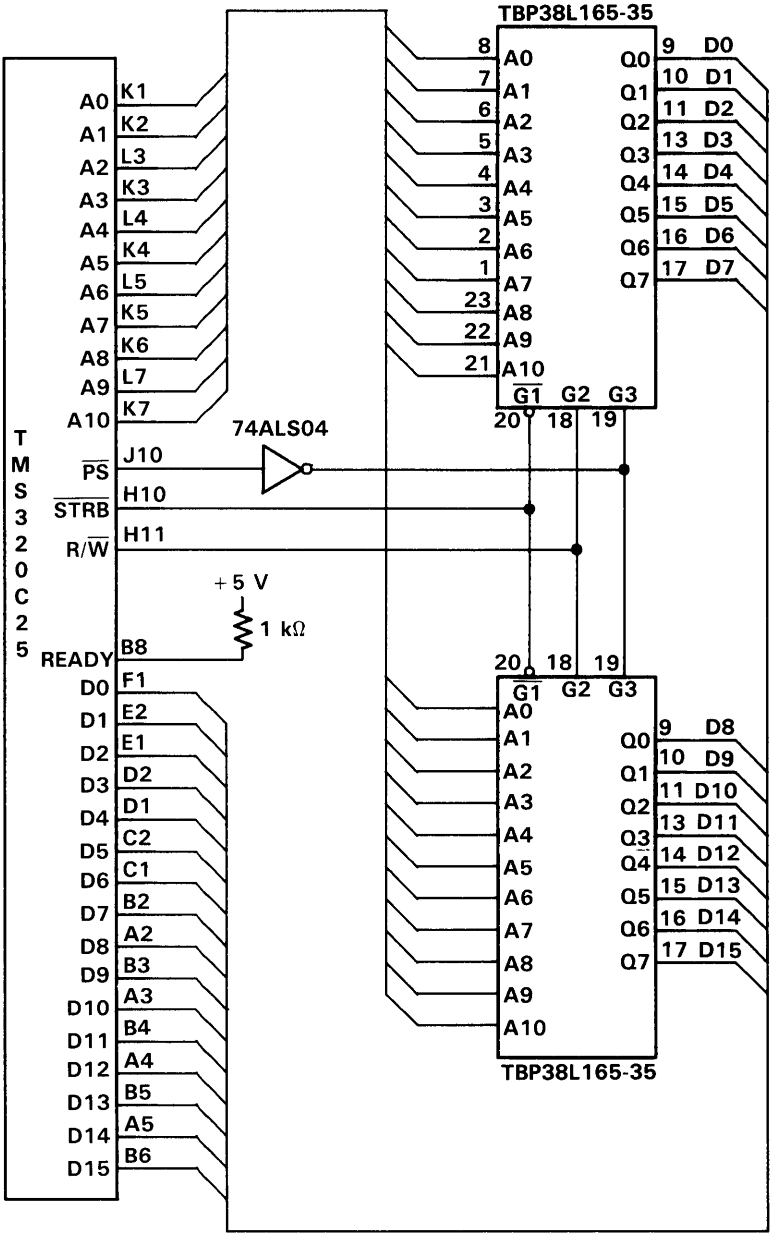


Figure 2. Direct Interface of the TBP38L165-35 to the TMS320C25

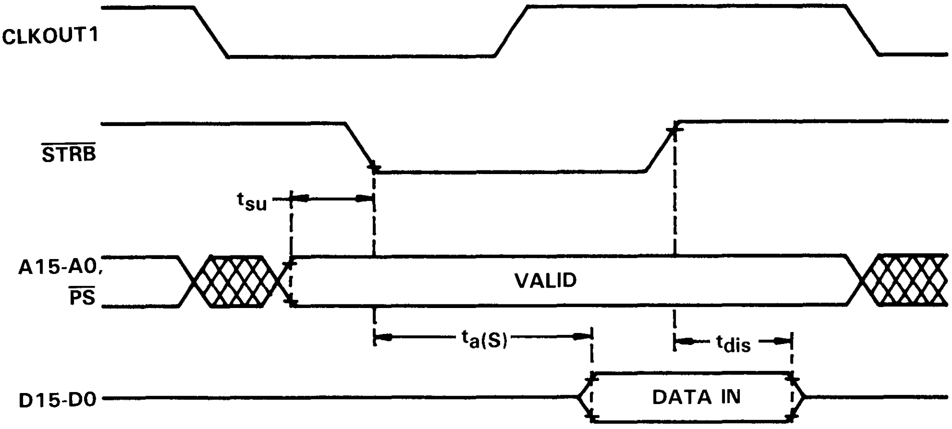


Figure 3. Interface Timing of the TBP38L165-35 to the TMS320C25

As discussed earlier, the TMS320C25 expects data to be valid no later than 2Q – 23 ns after $\overline{\text{STRB}}$ goes low. (This is 27 ns for a TMS320C25 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 TMS320C25, address becomes valid a minimum of t_{su} = Q – 12 ns = 13 ns before $\overline{\text{STRB}}$ goes low (see Figure 1). The memory is not enabled, however, until $\overline{\text{STRB}}$ goes low. Therefore, the data appears on the data bus within 27 ns after $\overline{\text{STRB}}$ goes low, as required by the TMS320C25.

When a read cycle is followed by a write cycle, care must be taken to avoid bus conflicts. In this case, the TMS320C25 begins driving the data bus as soon as $\overline{\text{STRB}}$ goes low, i.e., Q ns after the beginning of the write cycle. At that time, the system designer must guarantee that the outputs of the external memories have entered a high-impedance state. Consider now the design of Figure 2. The memory is disabled when $\overline{\text{STRB}}$ goes high. The disable time for the TBP38L165-35 is t_{dis} = 15 ns max. Therefore, the memory outputs have entered a high-impedance state no later than 15 ns after $\overline{\text{STRB}}$ goes high, and bus conflict is avoided.

Another case with a potential bus conflict is when a TMS320C25 write cycle is followed by a memory read cycle. In this case the TMS320C25 data lines must enter a high-impedance state before the memory starts driving the data bus. In a write cycle, the TMS320C25 enters a high-impedance state no later than 15 ns after the beginning of the next cycle. Since the design of Figure 2 utilizes $\overline{\text{STRB}}$ to enable the TBP38L165s, these memories cannot drive the data bus before $\overline{\text{STRB}}$ goes low, i.e., Q ns after the beginning of the cycle. Therefore, bus conflict is avoided (25 ns > 15 ns).

Note that the TMS320C25 $R/\overline{W}$ line is connected to the G_2 enable of the TBP38L165s. Therefore, the PROMs are disabled when $R/\overline{W}$ goes low, even if $\overline{STRB}$ 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 TMS320C25 are summarized in Table 1.

Table 1. Timing Parameters of the TBP38L165-35 Direct Interface to the TMS320C25

Description	Symbol Used in Figure 3	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 TMS320C25 using address decoding. A major issue when designing an interface with address decoding is that the TMS320C25 requires the READY signal to be stable no later than Q − 20 ns after $\overline{STRB}$ goes low. Since the setup time for the address is Q − 12 ns, the TMS320C25 requires (worst case) a stable READY 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 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 4. 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 are selected.
2. $\overline{DS}$ and $\overline{IS}$ are high; i.e, an external program memory cycle is in progress.

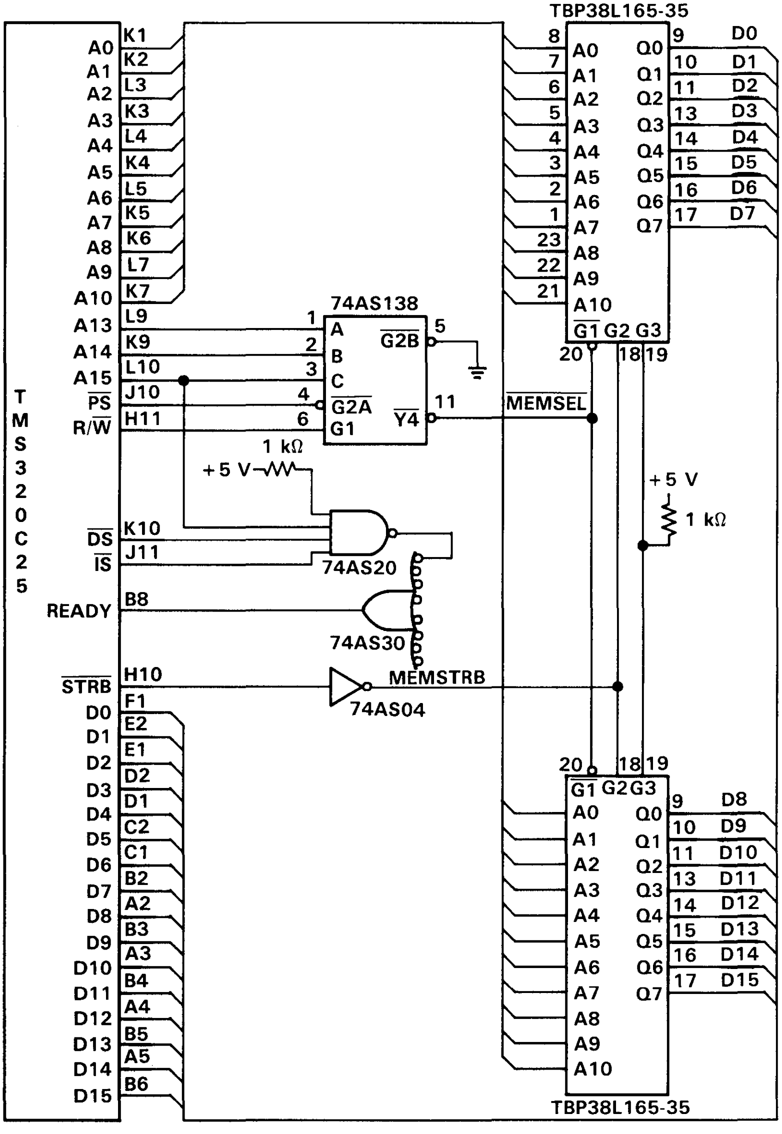


Figure 4. Interface of the TBP38L165-35 to the TMS320C25

The timing of READY is shown in Figure 5. READY goes high 10 ns (worst case) after the address has become valid.

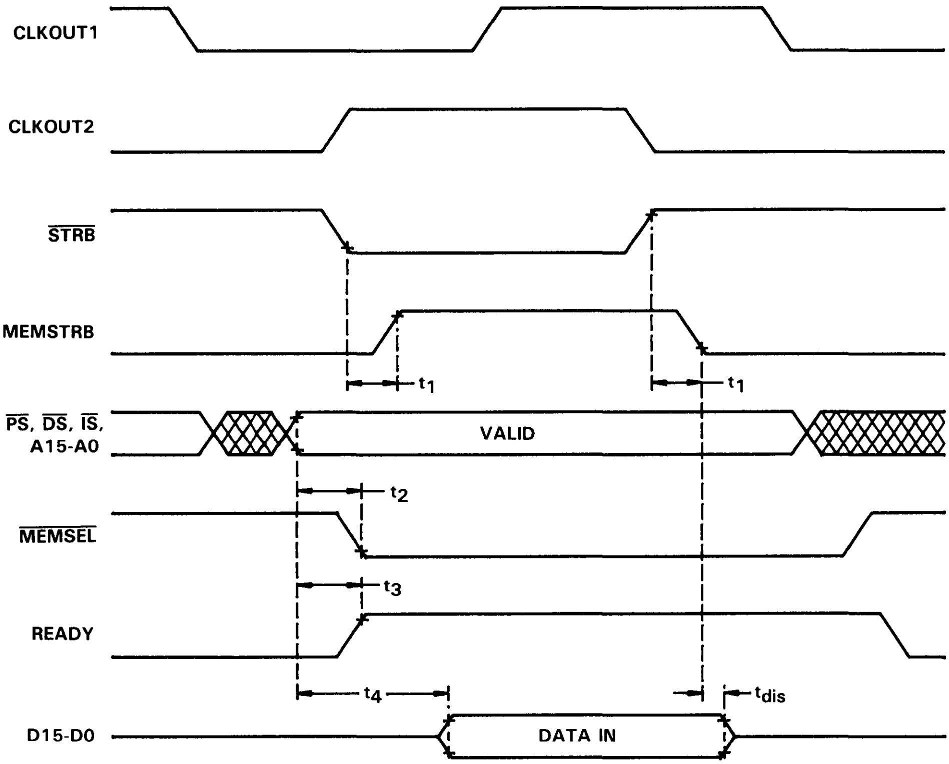

CLKOUT1
CLKOUT2
$\overline{\text{STRB}}$
MEMSTRB
t_1
t_1
$\overline{\text{PS}}$, $\overline{\text{DS}}$, $\overline{\text{IS}}$, A15-A0
VALID
t_2
$\overline{\text{MEMSEL}}$
t_3
READY
t_4
t_{dis}
D15-D0
DATA IN

Figure 5. Interface Timing of the TBP38L165-35 to the TMS320C25 (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 $\overline{\text{Y0}}$, $\overline{\text{Y1}}$, $\overline{\text{Y2}}$, and $\overline{\text{Y3}}$ 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 >8000). Note that in Figure 4, R/$\overline{\text{W}}$ 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 4, $\overline{\text{MEMSEL}}$ goes low no later than 10 ns (time t_2 in Figure 5) after address is valid. The PROMs are not enabled, however, until MEMSTRB goes high, i.e., a maximum of 5 ns after $\overline{\text{STRB}}$ goes low (time t_1 in Figure 5). Valid data appears on the data bus within 25 ns later. This meets the 27 ns (2Q − 23) access time required from $\overline{\text{STRB}}$ low by the TMS320C25. Note that in the design of Figure 4, $\overline{\text{STRB}}$ 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 5, 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 TMS320C25 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 TMS320C25 are summarized in Table 2.

Table 2. Timing Parameters of the TBP38L165-35 Interface with Address Decoding to the TMS320C25

Description	Symbol Used in Figure 5	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 TMS320C25, 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 TMS320C25 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.

Interfacing EPROMs

EPROMs may be used to debug TMS320C25 algorithms. Three different EPROM interfaces to the TMS320C25 are presented in this subsection. First, the direct interface of an EPROM that requires no wait states is discussed. Next, wait-state generator design is described. Finally, EPROM interfaces that require one and two wait states are considered.

Direct EPROM Interface with No Wait States

A Texas Instruments TMS27C292-35 EPROM can interface directly to the TMS320C25 with no wait states, as shown in Figure 6. 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 of Figure 6 is shown in Figure 7.

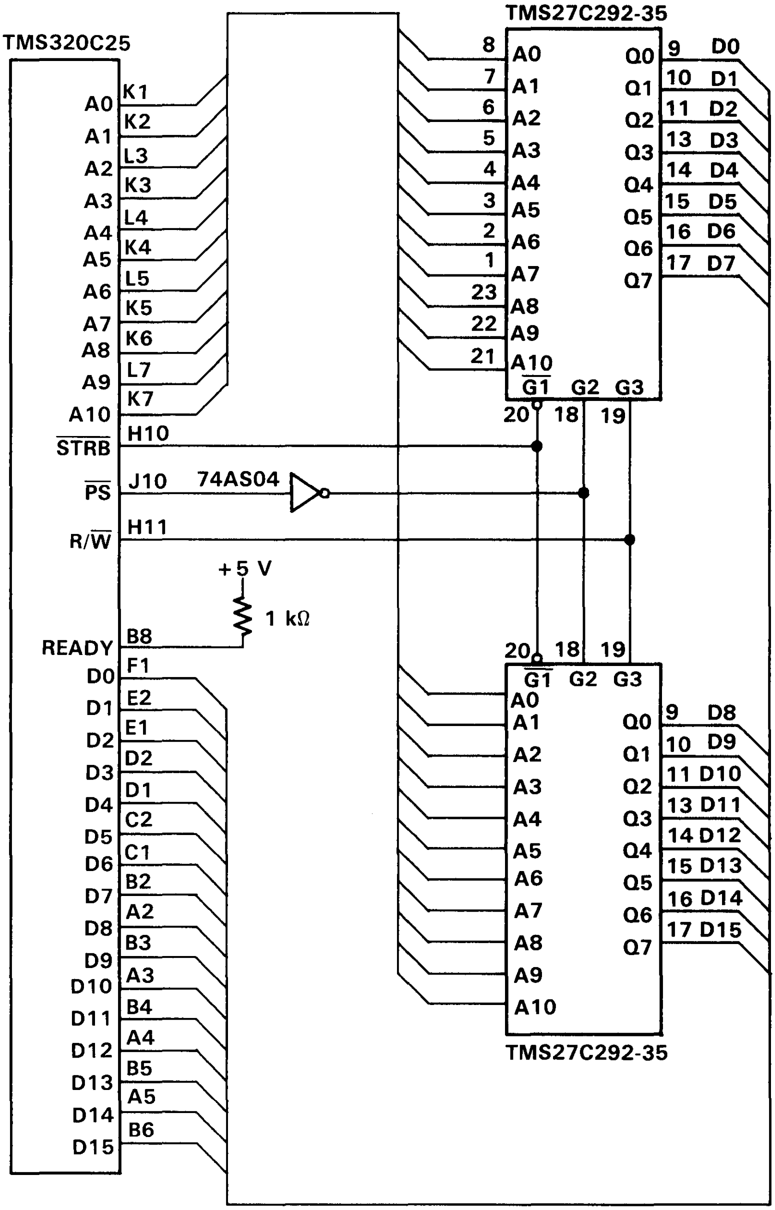


Figure 6. Direct Interface of the TMS27C292-35 to the TMS320C25

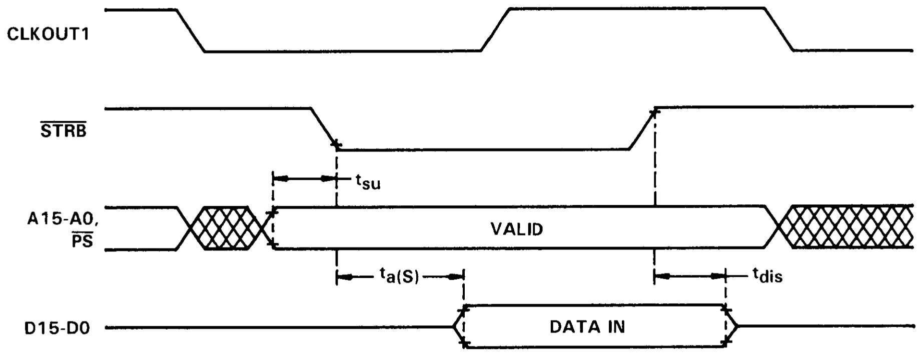


Figure 7. Interface Timing of the TMS27C292-35 to the TMS320C25

As shown in Figure 7, 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 TMS320C25 are summarized in Table 3.

Table 3. Timing Parameters of the TMS27C292-35 Direct Interface to the TMS320C25

Description	Symbol Used in Figure 7	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)

Wait-State Generator

The READY input allows the capability to interface with memory and peripherals that cannot be accessed in a single cycle. READY must be valid (low or high) no later than $Q-20$ ns $= 5$ ns after $\overline{\text{STRB}}$ goes low. If READY is high, then the memory/peripheral access is completed with the present machine cycle. If READY is low, the access is extended to the next machine cycle; i.e., a wait state is introduced. The number of wait states required depends on the access time t_a of the particular memory device or peripheral. If $t_a < 40$ ns, no wait states are required. If 40 ns $< t_a <$ 140 ns, one wait state must be inserted. In general, N wait states are required for a particular access if

$$[100(N-1) + 40]\text{ns} < t_a < [100N + 40]\text{ns}$$

The information on the number of wait states required for a memory or peripheral access is summarized in Table 4.

Table 4. Number of Wait States Required for a Memory or Peripheral Access

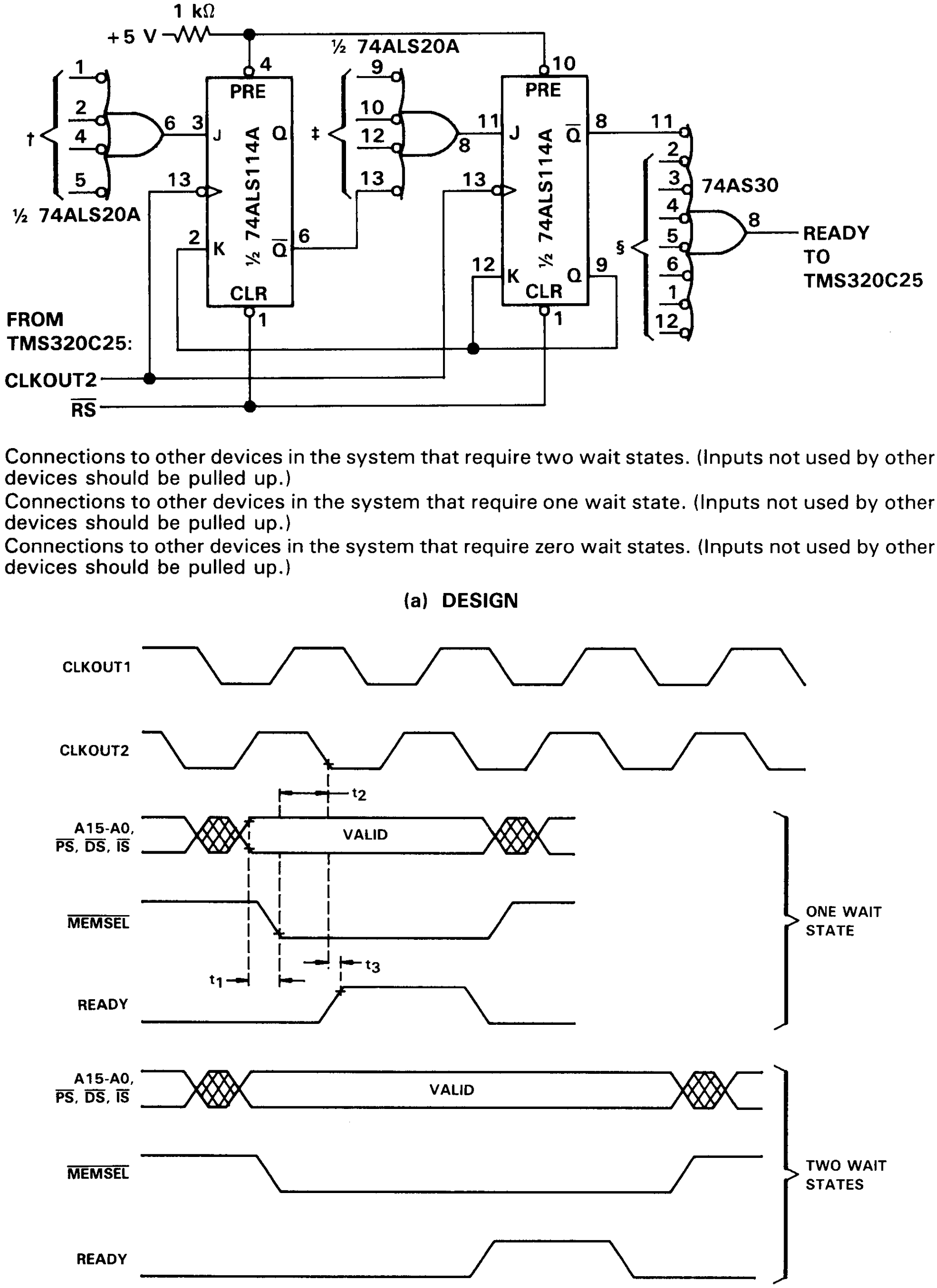
Access Time	Number of Wait States Required
$t_a < 40$ ns	0
40 ns $< t_a <$ 140 ns	1
140 ns $< t_a <$ 240 ns	2
240 ns $< t_a <$ 340 ns	3
340 ns $< t_a <$ 440 ns	4

In the design approach presented in this report, the READY input is driven by a 74AS30 eight-input NAND gate. The output of this gate is normally low unless one of the inputs goes low. In that case, READY goes high. If no wait states are required, the READY generation logic must drive one of the inputs of the 74AS30 low by the time that $\overline{\text{STRB}}$ goes low. Then, with a 5-ns maximum propagation delay through the NAND gate, READY will be valid (high) on time. (Such a circuit was discussed in the Interfacing PROMs subsection.) On the other hand, if one or more wait states are introduced, the READY generation logic has 100N (N being the number of wait states) additional ns to respond.

A wait-state generator design and timing are shown in Figures 8(a) and 8(b), respectively.

This design utilizes a 74ALS114A dual J-K type of flip-flop. Both flip-flops are clocked by the CLKOUT2 of the TMS320C25. First, consider the case of one wait state. Time t_1 in Figure 8(b) is the time from address valid to memory select of the particular device that requires the wait state. This corresponds to the propagation delay through the address decode logic. For a 74AS138 decoder, t_1 = 10 ns (max). The memory select of the device is connected to one of the inputs of the second NAND gate, as shown in Figure 8(a). When memory select goes low, the output of the NAND gate goes high. At the second flip-flop, $J=1$ and $K=0$; i.e., the flip-flop will be set when CLKOUT2 goes low. However, before this happens, the TMS320C25 has sampled READY (shortly after CLKOUT1 goes high). As shown in Figure 8(b), READY is low, and the TMS320C25 enters a wait state.

Consider time t_2 in Figure 8(b). This is the time from memory select going low to CLKOUT2 going low. Time t_2 must be greater than the maximum propagation delay t_p through the 74ALS20A and the setup time t_{su} of the J-K flip-flop, i.e., $t_p + t_{su}$ = 11 ns + 20 ns = 31 ns. The time from valid address to CLKOUT2 going low is 63 ns minimum (13 ns minimum setup time for the address plus 50 ns for CLKOUT2 high). If a 74AS138 (10 ns maximum propagation delay) is used for address decoding,



† 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.)

(a) DESIGN

(b) TIMING

Figure 8. Wait-State Generator

t_2 = 63 ns − 10 ns = 53 ns; therefore, $t_2 > 31$ ns. This analysis also implies that the maximum propagation delay of the address decoding logic is limited to 63 ns − 31 ns = 32 ns.

When CLKOUT2 goes low, the second flip-flop is set, i.e., Q = 1 and $\overline{\text{Q}}$ = 0. Since $\overline{\text{Q}}$ drives one of the inputs of the 74AS30, READY (output of 74AS30) goes high. Time t_3 in Figure 8(b) is the time from CLKOUT2 going low to READY going high. The READY input to the TMS320C25 must be valid 55 ns after CLKOUT2 goes low (50 ns for CLKOUT2 low plus 5 ns between CLKOUT2 going high and READY valid). Therefore, t_3 must satisfy the requirement: $t_3 < 55$ ns. The maximum propagation delays through the 74ALS114A and the 74AS30 are 19 ns and 5 ns, respectively. Therefore, t_3 = 24 ns (max), satisfying the 55-ns requirement.

READY must remain high until it is sampled again, shortly after CLKOUT1 goes high. In the design of Figure 8, READY remains high well after CLKOUT1 goes high. At the falling edge of CLKOUT2, the inputs to the J-K flip-flop are J = 1 and K = Q = 1, and the flip-flop is in the toggle mode. When CLKOUT2 goes low, $\overline{\text{Q}}$ goes back to logic 1. READY goes low and stays low until one of the inputs of the 74AS30 is pulled low.

To implement two wait states, a second J-K flip-flop is utilized as shown in Figure 8(a). This delays READY going high by an additional machine cycle. The timing diagram of the two wait-state generator is shown in Figure 8(b). If more wait states are required, additional J-K flip-flops in the wait-state generator of Figure 8(a) must be included.

EPROM Interface with One Wait State

The hardware interface of the Wafer Scale WS57C64F-12 (8K × 8-bit EPROMs) to the TMS320C25 is shown in Figure 9. The WS57C64F-12s are mapped at address >2000. The interface timing diagram is provided in Figure 10.

The WS57C64F-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 9, the 74AS138 is used for the address decoding. $\overline{\text{PS}}$ and R/$\overline{\text{W}}$ are used to drive the $\overline{\text{G2A}}$ 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. $\overline{\text{MEMSEL}}$ going low t_1 = 10 ns (max) after address valid (see Figure 10) is used for two purposes: (1) to drive the wait-state generator, as discussed earlier; and (2) to generate a strobe signal $\overline{\text{DTSTR}}$ that activates the output buffers of the WS57C64-12s. Time t_3 in Figure 10 is the time from address valid 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 8(a) may be used to implement this wait state. Also, note that the WS57C64F-12 is the slowest member of the WS57C64F EPROM series, and still meets the specifications for one wait state.

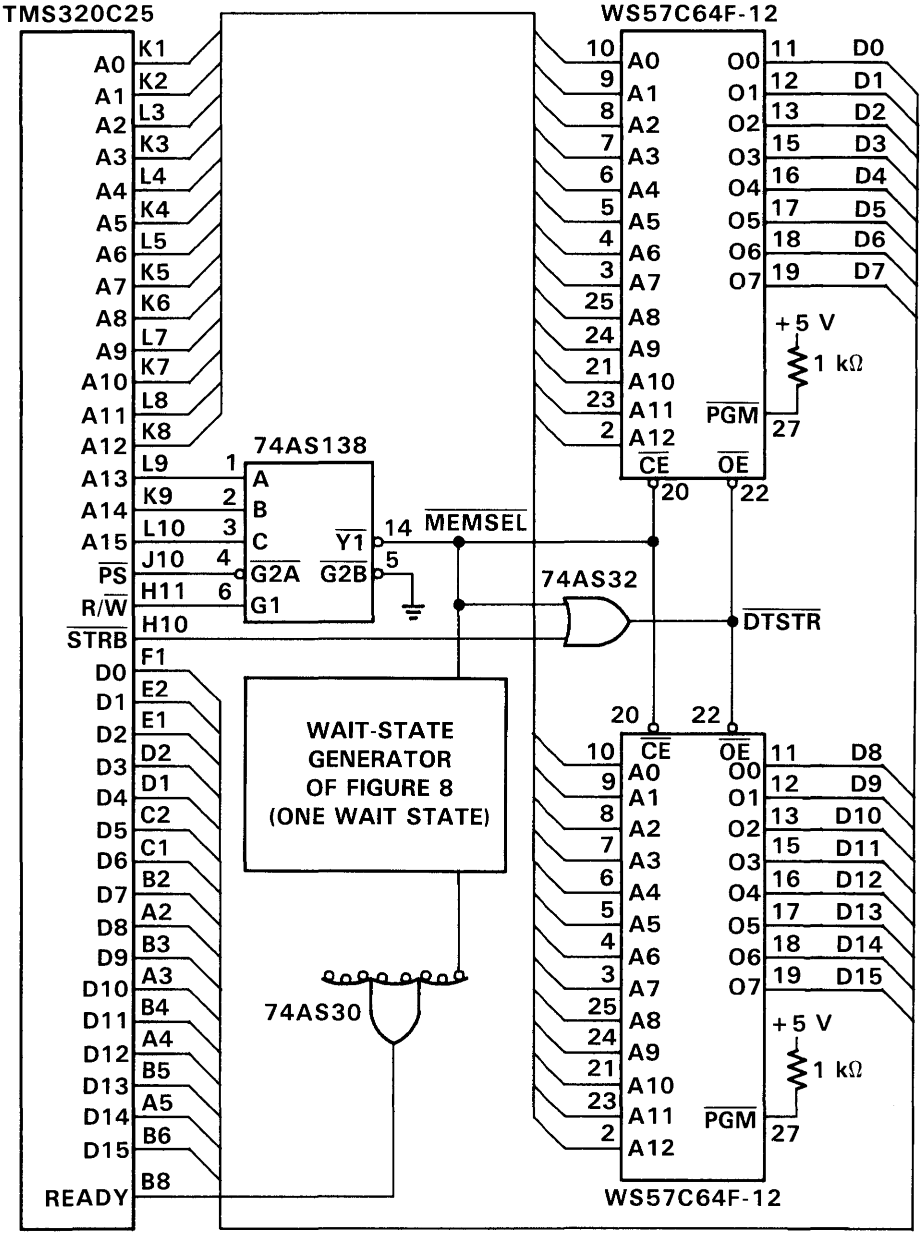


Figure 9. Interface of the WS57C65F-12 to the TMS320C25

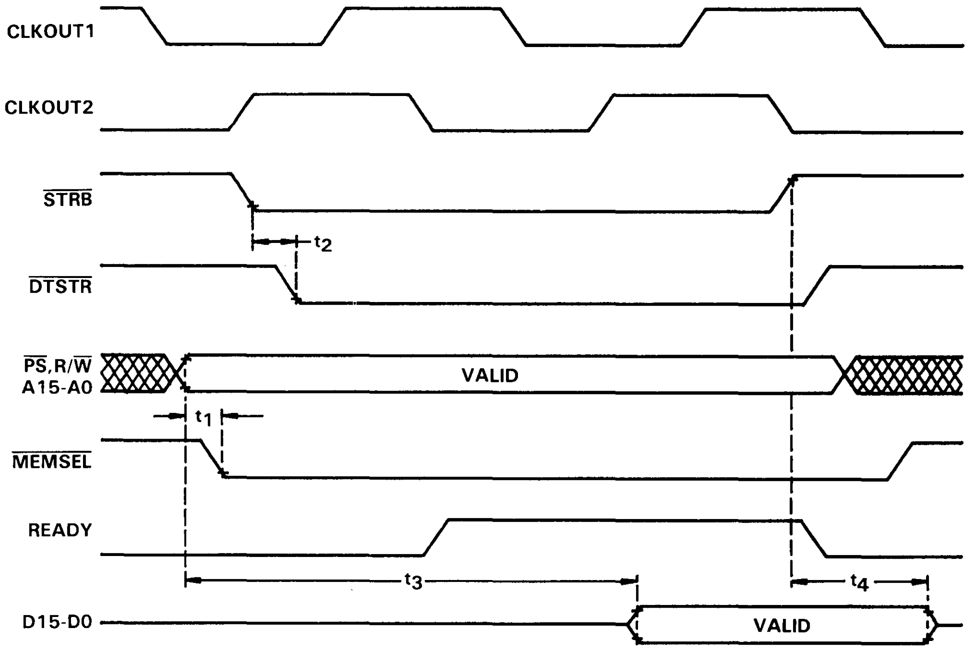


Figure 10. Interface Timing of the WS57C64F-12 to the TMS320C25

With STRB going high, the read has been completed. 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 10 is used to indicate the time from 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 TMS320C25 can start driving the data bus when the next instruction is a write), there is no bus conflict.

Table 5 summarizes the most critical timing parameters of the WS57C64F-12 interface to the TMS320C25.

Table 5. Timing Parameters of the WS57C64F-12 Interface to the TMS320C25

Description	Symbol Used in Figure 10	Value
Address valid to MEMSEL low	t_1	10 ns (max)
STRB low to DTSTR low	t_2	5.8 ns (max)
TMS320C25 address valid to WS57C64F-12 data valid	t_3	130 ns (max)
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 TMS320C25 is shown in Figure 11. The TMS27C64-20 is a CMOS 8K × 8-bit EPROM with an access time of 200 ns. The timing diagram is shown in Figure 12.

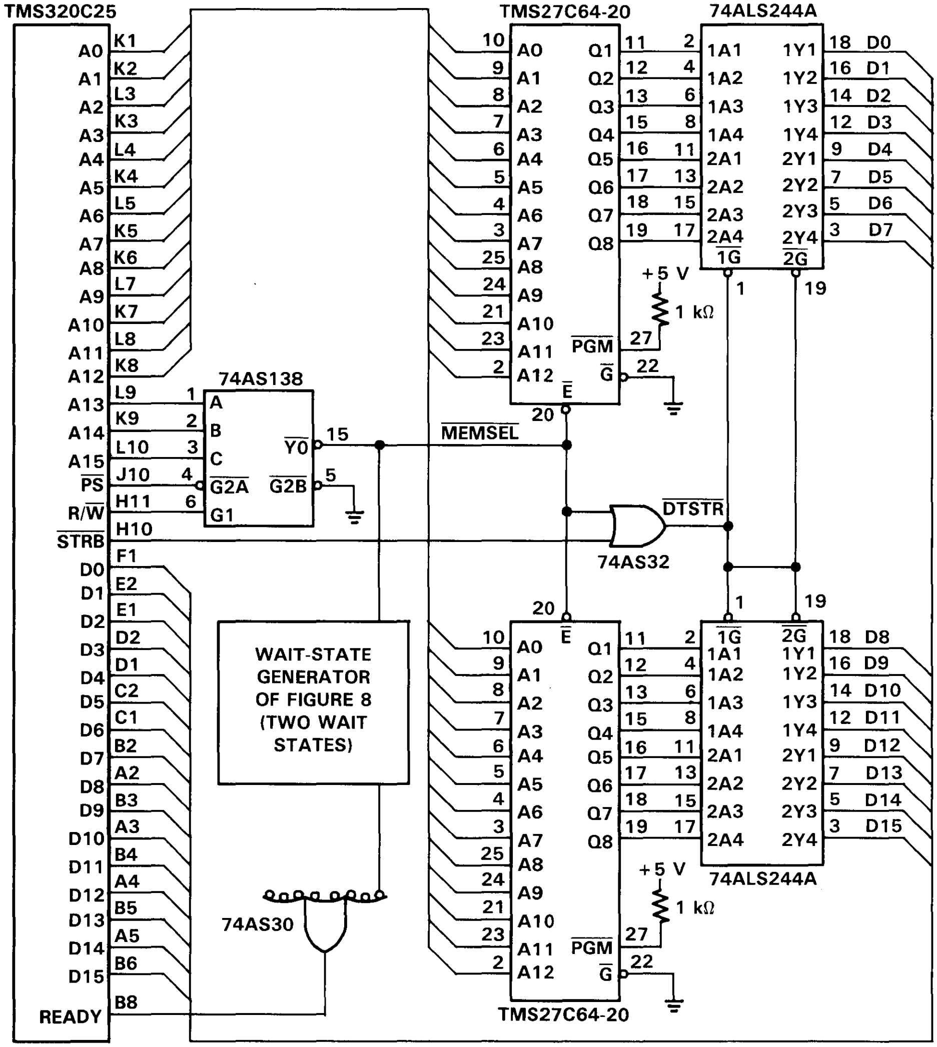


Figure 11. Interface of the TMS27C64-20 to the TMS320C25

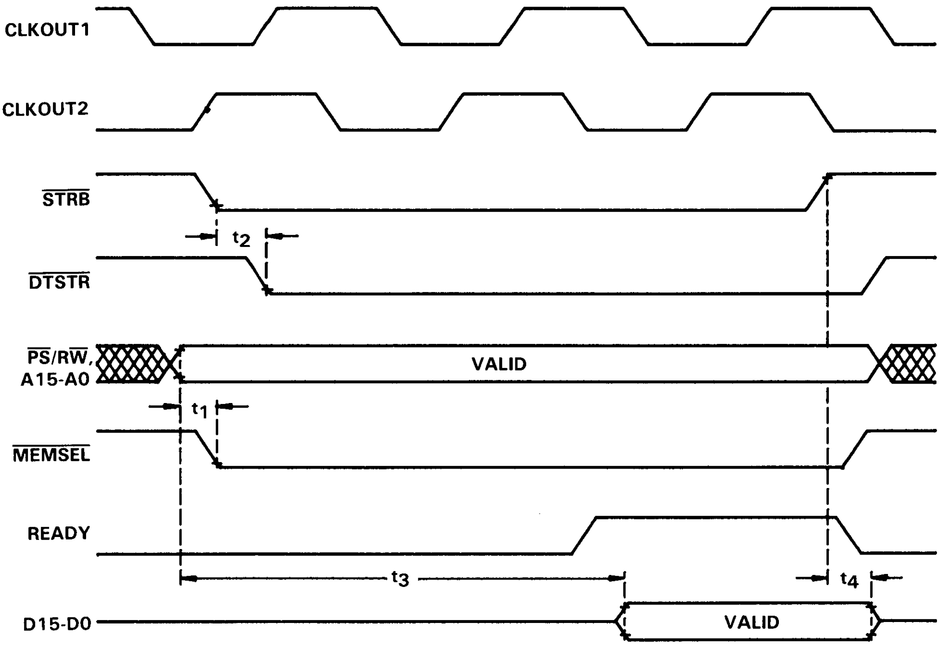


Figure 12. Interface Timing of the TMS27C64-20 to the TMS320C25

With a 200-ns access time, two wait states are needed. These can be implemented using the wait-state generator of Figure 8(a). Address decoding is similar to that used for the WS57C64F-12, and the TMS27C64 is mapped at address >0000. The memory cycle starts with address valid. $\overline{\text{MEMSEL}}$ becomes low $t_1 = 10$ ns (max) later (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 11, the 74ALS244A octal buffers are used to buffer the memories from the TMS320C25. These buffers are enabled with $\overline{\text{DTSTR}}$, which is the logical-OR of $\overline{\text{MEMSEL}}$ and $\overline{\text{STRB}}$. The maximum propagation delay through these buffers is $t_p = 10$ ns. Therefore, valid data appear on the TMS320C25 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 TMS320C25 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 $\overline{\text{STRB}}$ 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 $t_4 = 13 \text{ ns} + 5.8 \text{ ns} = 18.8$ 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 would generate a conflict on the data bus.

Table 6 summarizes the most critical timing parameters of the TMS27C64-20 interface to the TMS320C25.

Table 6. Timing Parameters of the TMS27C64-20 Interface to the TMS320C25

Description	Symbol Used in Figure 12	Value
Address valid to $\overline{\text{MEMSEL}}$ low	t_1	10 ns (max)
$\overline{\text{STRB}}$ low to $\overline{\text{DTSTR}}$ low	t_2	5.8 ns (max)
TMS320C25 address valid to TMS27C64-20 data valid	t_3	220 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 fast enough EPROMs are used with the TMS320C25, 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.

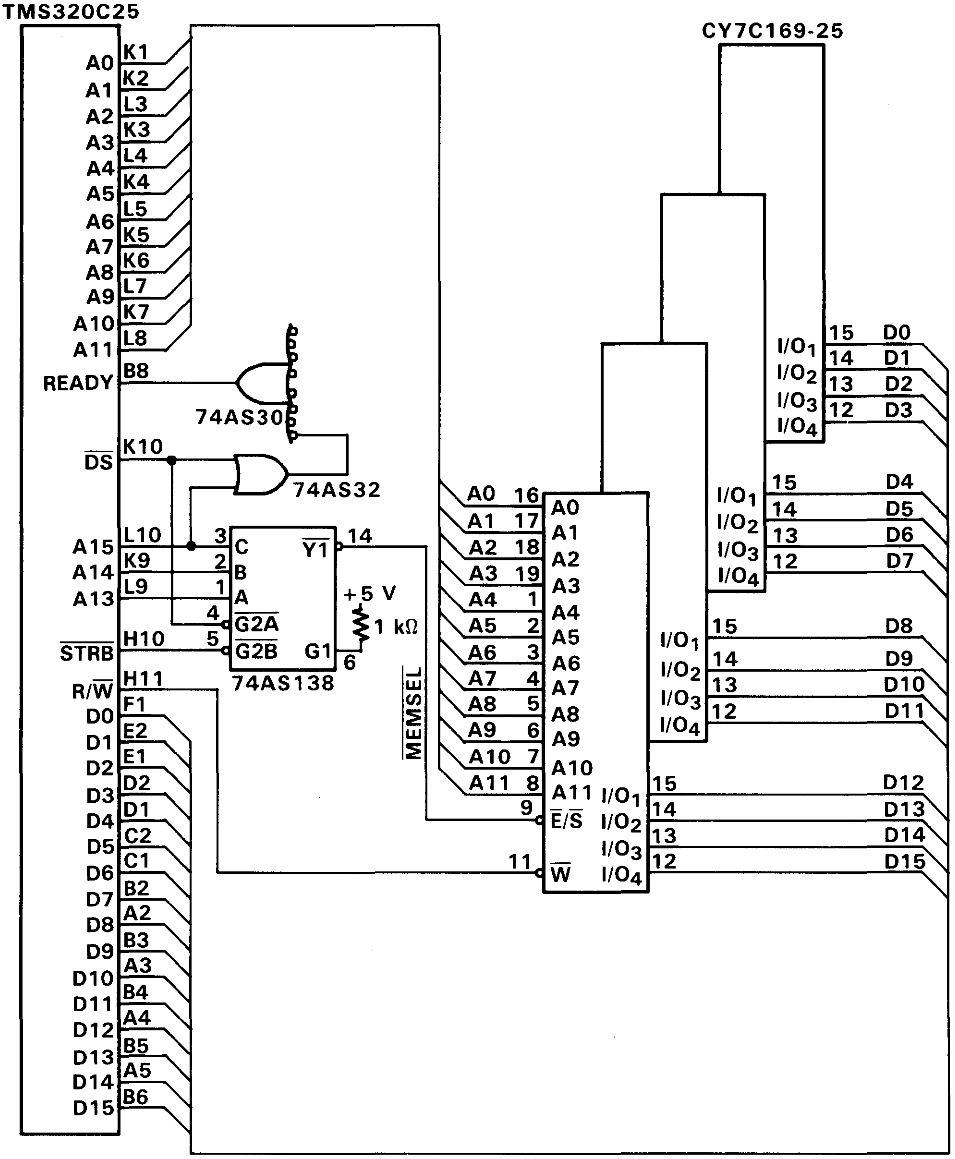
Interfacing Static RAMs

The TMS320C25 can utilize static RAM as either program or data memory. When used as program memory, object code can be downloaded into the RAM and executed. Static RAM can also be used as data memory to extend the TMS320C25's 544 words of internal RAM. In the first case, the static RAM is mapped into the TMS320C25 program space, while in the second case it is mapped into the data space.

The static RAM chosen for this interface is the Cypress Semiconductor CY7C169-25 4K × 4-bit static RAM. This RAM 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 13, and a timing diagram of the interface is presented in Figure 14.



TMS320C25
CY7C169-25
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11
K1 K2 L3 K3 L4 K4 L5 K5 K6 L7 K7 L8
READY
B8
74AS30
DS
K10
74AS32
A15 A14 A13
L10 K9 L9
C B A
Y1
14
G2A G2B G1
+5 V
1 kΩ
STRB
H10
R/W
H11
74AS138
MEMSEL
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
F1 E2 E1 D2 D1 C2 C1 B2 A2 B3 A3 B4 A4 B5 A5 B6
E/S
W
I/O1 I/O2 I/O3 I/O4

Figure 13. Interface of the CY7C169-25 to the TMS320C25

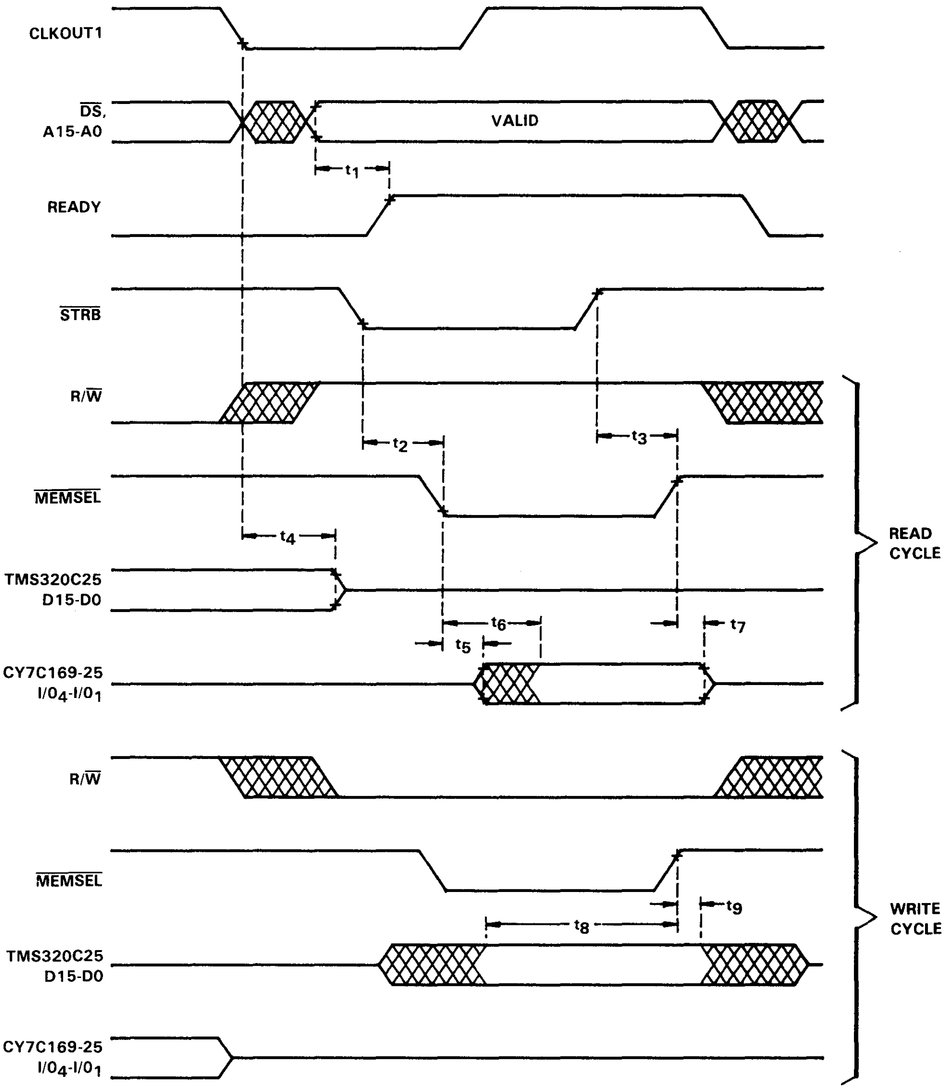


Figure 14. Interface Timing of the CY7C169-25 to the TMS320C25

The design of Figure 13 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 static RAM. In this design, RAMs with no wait states are mapped at the lower half (lower 32K words) of the TMS320C25 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 $\overline{\text{DS}}$ 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 14 indicates the time from valid address to READY going high. The maximum value for t_1 is

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

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

As shown in Figure 13, address decoding to enable the RAM is implemented with the 74AS138. This decoding separates the data space into eight segments of 8K words each. The first four segments are enabled by the $\overline{\text{Y0}}$, $\overline{\text{Y1}}$, $\overline{\text{Y2}}$, and $\overline{\text{Y3}}$ outputs of the 74AS138. These segments are used for memories with no wait states. (Note that in Figure 13 the CY7C169s are enabled by $\overline{\text{Y1}}$; i.e., the memories are mapped at address >2000.) The other four segments, enabled by the other outputs of the decoder, are used for memories with one or more wait states.

Memory Read Cycle

Figure 14 shows the timing for memory read and write cycles. In a read cycle, R/$\overline{\text{W}}$ goes high concurrently with valid address, indicating that a read rather than a write cycle has been initiated. With $\overline{\text{STRB}}$ used to enable the 74AS138, $\overline{\text{MEMSEL}}$ goes low no later than $t_2 = 8.5$ ns after $\overline{\text{STRB}}$ goes low. This is the maximum propagation delay of the 74AS138 from the $\overline{\text{G}}$ enable to output for a high-to-low transition. The CY7C169s begin driving the data bus no earlier than $t_5 = 5$ ns after $\overline{\text{MEMSEL}}$ goes low. By then, all of the devices having access to the data bus must have entered a high-impedance state. Figure 14 shows the TMS320C25 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 $\overline{\text{MEMSEL}}$ goes low. Therefore, the worst-case access time from $\overline{\text{STRB}}$ going low is $t_2 + t_6 = 23.5$ ns. This meets the 27-ns access time required by the TMS320C25 operating at 40 MHz.

The TMS320C25 read cycle is concluded with $\overline{\text{STRB}}$ going high. $\overline{\text{MEMSEL}}$ follows $\overline{\text{STRB}}$ 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 $\overline{\text{MEMSEL}}$ 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 14, the memory write cycle is similar to the read cycle with the exception that R/$\overline{\text{W}}$ is low. The TMS320C25 begins driving the data bus as soon as

$\overline{\text{STRB}}$ goes low, while $\overline{\text{MEMSEL}}$ follows $\overline{\text{STRB}}$ within t_2 = 8.5 ns. Since R/$\overline{\text{W}}$ is low when $\overline{\text{MEMSEL}}$ goes low, the CY7C169s do not drive the data bus.

Data is clocked into the CY7C169 by the rising edge of $\overline{\text{MEMSEL}}$. Time t_8 in Figure 14 is the time that data is valid before $\overline{\text{MEMSEL}}$ goes high. This time is no less than the TMS320C25 minimum data setup time before $\overline{\text{STRB}}$ goes high (2Q-20 = 30 ns when operating at 40 MHz), plus the minimum propagation delay through the 74AS138 (2 ns). Therefore, t_8 is greater than or equal to 32 ns. Note that this time meets the 10-ns minimum data setup time required by the CY7C169.

Table 7 summarizes the most critical timing parameters of the CY7C169-25 interface to the TMS320C25.

Table 7. Timing Parameters of the CY7C169-25 Interface to the TMS320C25

Description	Symbol Used in Figure 14	Value
Address valid to READY valid	t_1	10.8 ns (max)
$\overline{\text{STRB}}$ low to $\overline{\text{MEMSEL}}$ low	t_2	8.5 ns (max)
$\overline{\text{STRB}}$ high to $\overline{\text{MEMSEL}}$ high	t_3	7.5 ns (max)
CLKOUT1 low to TMS320C25 data bus entering the high-impedance state	t_4	15 ns (max)
$\overline{\text{MEMSEL}}$ low to CY7C169-25 driving the data bus	t_5	5 ns (min)
$\overline{\text{MEMSEL}}$ low to CY7C169-25 data valid	t_6	15 ns (max)
$\overline{\text{MEMSEL}}$ high to CY7C169-25 entering the high-impedance state	t_7	15 ns (max)
Data setup time for a write	t_8	32 ns (min)
Data hold time	t_9	7.5 ns (min)

In summary, interfacing external RAM to the TMS320C25 can be very useful for expanding internal data memory or implementing additional RAM program memory. In cases where RAMs of different speeds 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 described in this report. RAMs with similar access times may then be grouped together in one segment of memory.

System Control Circuitry

The 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 15 performs a powerup reset; i.e., the TMS320C25 is reset when power is applied. Driving the $\overline{\text{RS}}$ 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 *TMS320C25 User's Guide.*[1]

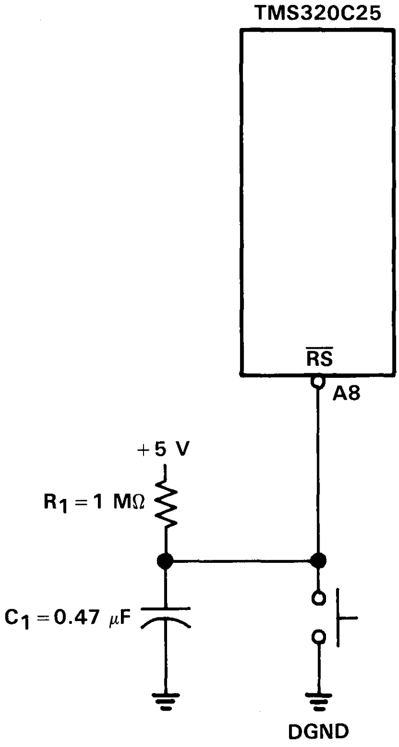


Figure 15. Reset Circuit

For proper system initialization, the reset signal must be applied for at least three CLKOUT cycles, i.e., 300 ns for a TMS320C25 operating at 40 MHz. Upon powerup, however, it takes several milliseconds before the system oscillator reaches a stable operating state. Therefore, the powerup reset circuit should generate a low pulse on the reset line for 100 to 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 the reset pin ($\overline{RS}$) is controlled by the R_1C_1 network (see Figure 15). After a reset, this voltage rises exponentially according to the time constant R_1C_1, as shown in Figure 16.

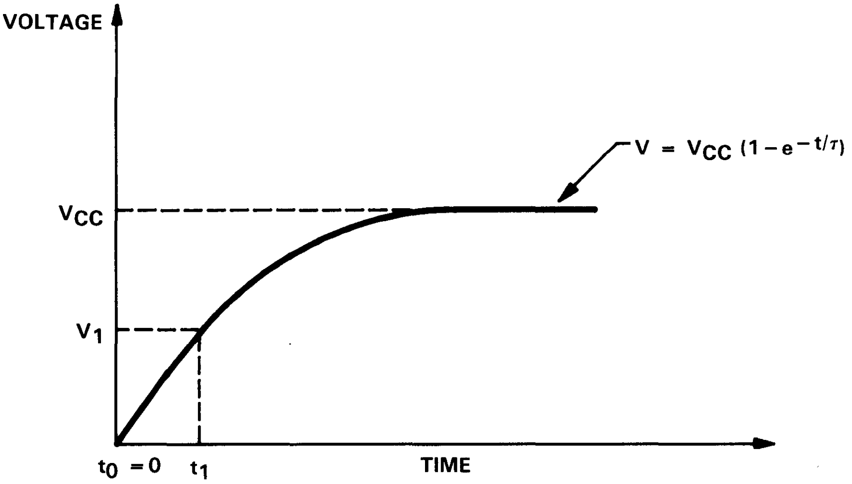


Figure 16. Voltage on the TMS320C25 Reset Pin

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 be charged to 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 voltage is given by

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

where $\tau = R_1C_1$ is the reset circuit time constant. Solving (1) for t gives

$$t = -R_1C_1 \ln \left[1 - \frac{V}{V_{CC}} \right] \qquad (2)$$

Setting the following:

$$R_1 = 1\ \text{M}\Omega$$

$$C_1 = 0.47\ \mu\text{F}$$

$$V_{CC} = 5\ \text{V}$$

$$V = V_1 = 1.5\ \text{V}$$

gives $t = t_1 = 167$ ms. Therefore, the reset circuit of Figure 15 provides a low pulse of long enough duration to ensure the stabilization of the system oscillator upon powerup.

Crystal Oscillator Circuit

The crystal oscillator circuit shown in Figure 17 is designed to operate at 40.96 MHz. Justification for this choice of frequency is discussed later in connection with the clock divider circuit. 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.

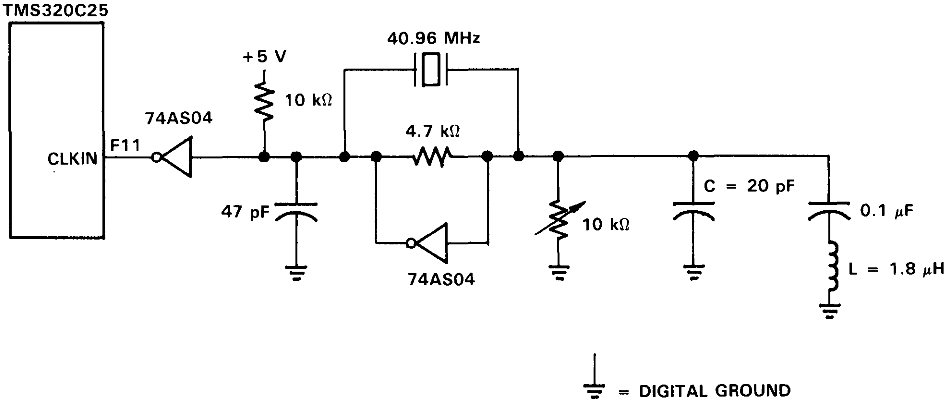


Figure 17. Crystal Oscillator Circuit

The 74AS04 inverter in Figure 17 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

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

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

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

In (5), the LC circuit appears inductive at frequencies lower than ω_p. 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 18, which is a plot of the magnitude of the impedance of the LC circuit of Figure 17 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 17, choose $\omega_p = 26.5$ MHz, which is approximately halfway between the fundamental and the third harmonic. Choose C = 20 pF. Then, using (4), L = 1.8 μH.

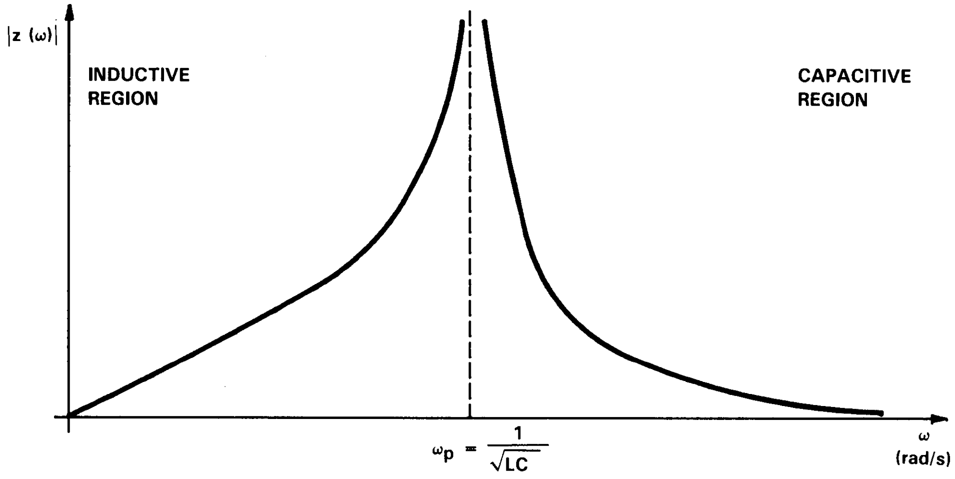


Figure 18. Magnitude of the Impedance of the Oscillator LC Network

The 0.1-μF capacitor in series with the 1.8-μH inductor is a coupling capacitor, requiring no DC path to the ground. The 74AS04 inverter is included to shorten the rise and fall times of the waveform generated by the oscillator. This is desirable because although the 'AS' parts have smaller propagation delays, they switch slower than the 'HC' parts. For a logic level 1, compared with TTL devices, HC devices have a higher minimum voltage for both the input and the output. Therefore, HC devices can directly drive TTL devices. When TTL devices drive HC devices, however, and the TTL output is at a logic level 1, an output voltage can be produced as low as 3 V. On the other hand, the HC device expects a minimum of approximately 3.5 V to recognize it as a logic level 1. Therefore, a 10-kΩ pullup resistor is included as shown in Figure 17. The maximum current entering the HC device is 1 μA. When the 74AS04 inverter drives high, the voltage on the line is greater than 5 V − (1 μA × 10 kΩ) = 4.99 V. This is correctly recognized as a logic 1 by the HC device.

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 TMS320C25, thus providing the four phases required for each machine cycle. With a 40.96-MHz input clock frequency, the TMS320C25 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 guarantees 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 TMS320C25. In this section, hardware interfaces of the TMS320C25 to a codec, an analog-to-digital converter (A/D), and a digital-to-analog converter (D/A) are described.

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 TMS320C25.

TMS320C25 Serial Port

The TMS320C25 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 TMS320C25:

- 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 TMS320C25 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 an 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 TMS320C25 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 TMS320C25, the μ- or A-law companded data must be converted back to a linear representation for use in the TMS320C25. Software companding routines appropriate for use on the TMS320C25 are provided in the book, *Digital Signal Processing Applications with the TMS320 Family*.[2]

The software required to initialize the TMS320C25-codec interface is shown next. The initialization routine should include the following:

```
INIT      DINT             ; Disable interrupts
          FORT    1        ; Set 8-bit data format
          LACK    >10
          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 (T1 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 >4 of the TMS320C25 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 >10 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 TMS320C25 and the TCM29C16 combo-codec is shown in Figure 19.

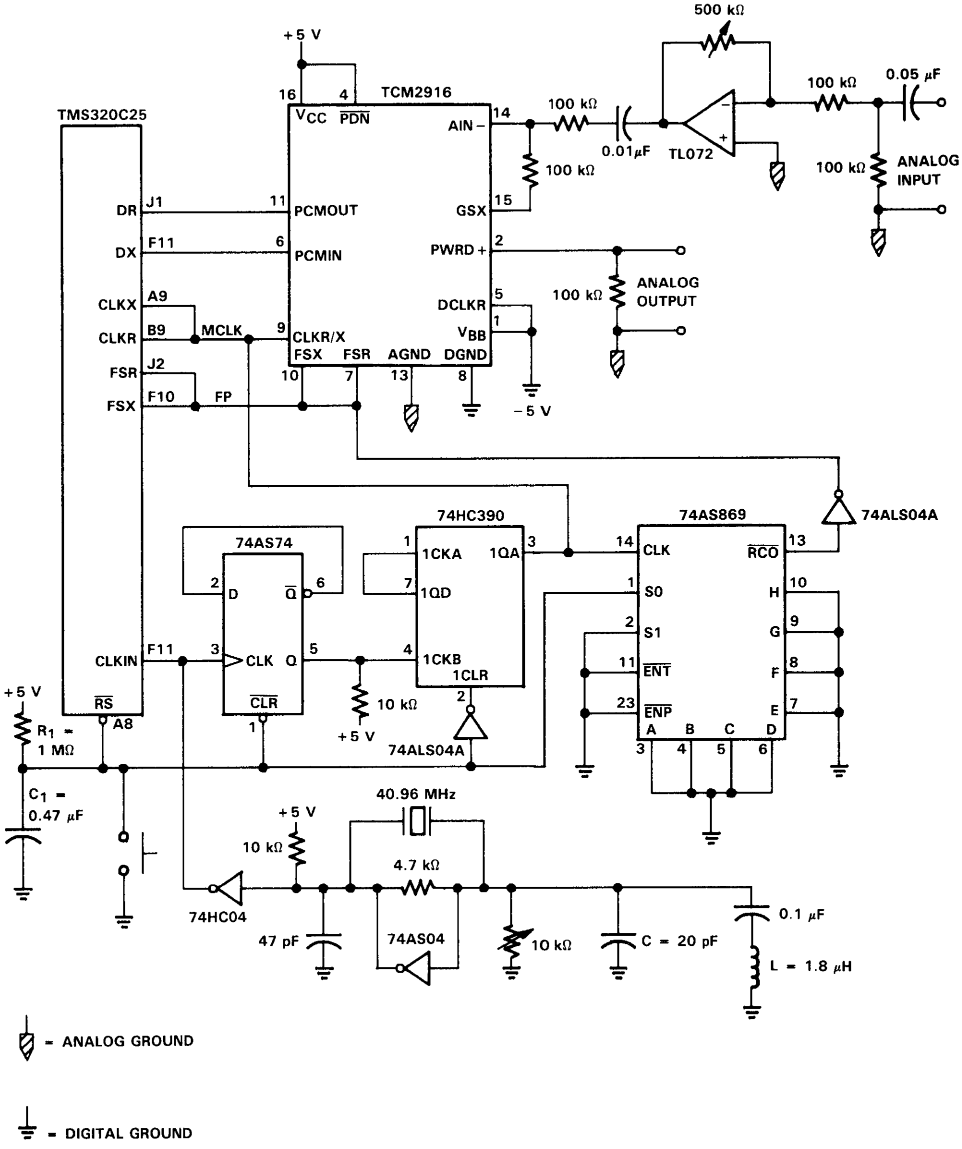


Figure 19. Interface of the TMS320C25 to the TCM29C16 Codec

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 frequency. The clock divider circuit consists of a 74AS74 D-type flip-flop, 74HC390 decade counter, and 74AS869 8-bit up/down counter. The hardware connections between these devices are shown in Figure 19.

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 19 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 19). 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 19); 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 TMS320C25 and codec as framing pulses to initiate data transfers.

TMS320C25-Codec Interface

The TMS320C25 interfaces directly to the codec, as shown in Figure 19, with no additional logic required. The PCM μ-law data generated by the codec at the PCMOUT pin is read by the TMS320C25 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 TMS320C25 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 TMS320C25-codec interface is shown in Figure 20.

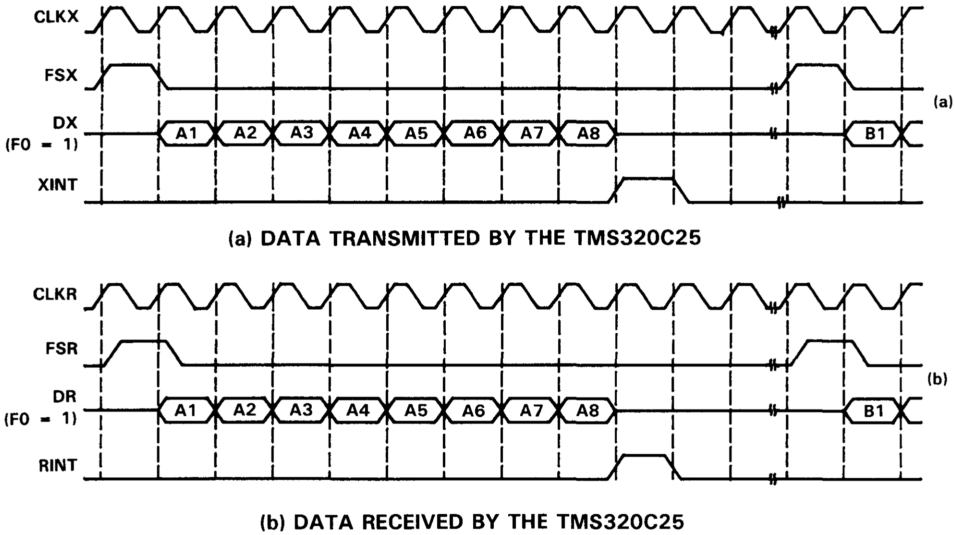


Figure 20. Interface Timing of the TMS320C25 to the TCM29C16 Codec

As indicated in Figure 20, both the transmit and receive operations are initiated by a framing pulse on the FSX and FSR pins of the TMS320C25 and the codec. The receive and transmit interrupts shown in Figure 20 occur only if they are enabled. Note that Figure 20 corresponds to the burst-mode serial port operation of the TMS320C25.[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 19). 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 and combo-codecs in particular are most effective in serving DSP system data-conversion requirements. These inexpensive devices interface directly to the TMS320C25, occupy minimal board space, and perform both filtering and data conversion functions. Codecs interface to the TMS320C25 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 bv the TMS320C25 for use in processing. Interface to the codec on the serial port is initialized by a simple software routine in the TMS320C25.

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-digital 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 TMS320C25 is discussed in this subsection. In this design, the A/D is mapped into the input/output (I/O) space of the TMS320C25. The distinction between the I/O space and the program and data spaces is made by using the $\overline{\text{IS}}$ pin. This pin goes active (low) when the I/O space is accessed. The TMS320C25 I/O 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 21. 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.

The design of Figure 21 consists of two sections: the analog-to-digital conversion and the interface to the TMS320C25. 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.

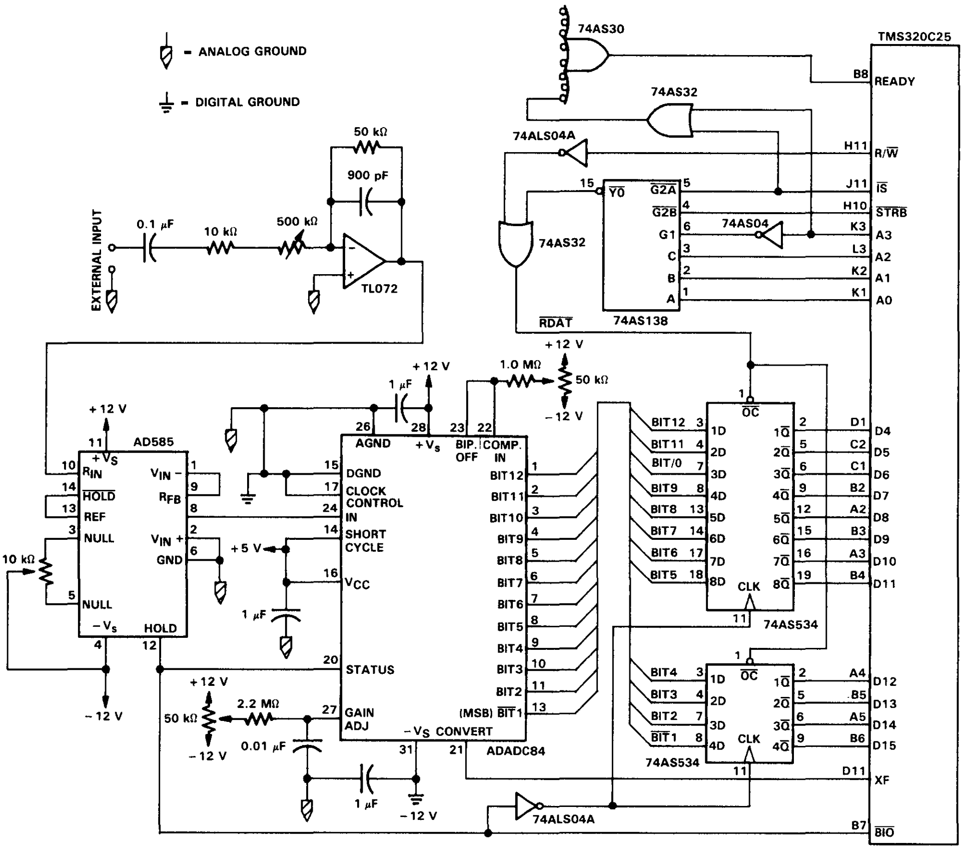


Figure 21. Interface of the ADADC84 to the TMS320C25

In the design of Figure 21, the antialiasing filter is implemented using a TL072 opamp connected in the inverting configuration. A 500-kΩ potentiometer is used to vary the level of the analog input signal. The gain varies from 50 kΩ/510 kΩ = 0.098 when the resistance of the potentiometer is 500 kΩ, to 50kΩ/10kΩ = 5 when the potentiometer is turned down 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 21, the 900-pF capacitor in the feedback path introduces a pole at the frequency f defined by

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

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. complementary two's-complement form. A conversion begins when the CONVERT input goes high. The XF (external flag) output of the TMS320C25 is used to drive the CONVERT input. Since the XF pin is software-controlled, the TMS320C25 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 21, the $\overline{\text{BIO}}$ pin of the TMS320C25 is connected to STATUS. By polling $\overline{\text{BIO}}$, the TMS320C25 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 TMS320C25 data bus until the output buffers of the 74AS534s are enabled.

Interface to the TMS320C25

The interface logic in Figure 21 is used to perform the following functions:

- Generate READY, and
- Enable the output buffers of the 74AS534s so that the TMS320C25 can read the data from the A/D conversion.

To meet the TMS320C25 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 scheme is used to enable the I/O-mapped devices.

The address decoding for READY is implemented with the 74AS32 positive-OR gate. The output of the 74AS32 goes low when both $\overline{\text{IS}}$ 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 require one or more wait states can utilize ports 8 through 15.

To enable the I/O devices, a 74AS138 is used. Outputs $\overline{\text{Y0}}$ through $\overline{\text{Y7}}$ of the 74AS138 can be used to enable the devices on ports 0 through 7, respectively. In Figure 21, $\overline{\text{Y0}}$ is used to enable a read from the A/D converter. Note that $\overline{\text{Y0}}$ is ORed with the inverted R/$\overline{\text{W}}$. This prevents the bus conflict that occurs if the TMS320C25 writes to port 0.

The timing diagram of a TMS320C25 read from port 0 is shown in Figure 22.

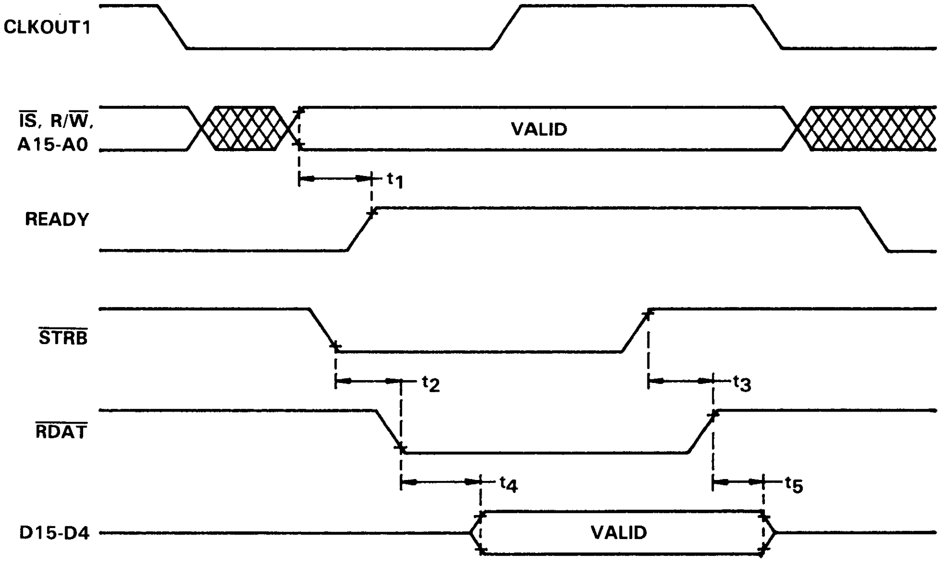


Figure 22. Interface Timing of the ADADC84 to the TMS320C25

Time t_1 in Figure 22 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.

$\overline{\text{RDAT}}$ in Figure 22 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 TMS320C25 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 TMS320C25.

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 8 summarizes the most critical timing parameters of the ADADC84 interface to the TMS320C25.

Table 8. Timing Parameters of the ADADC84 Interface to the TMS320C25

Description	Symbol Used in Figure 22	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)
Propagation delay through the 74AS534 ($\overline{\text{OC}}$ to $\overline{\text{Q}}$)	t_4	10 ns (max)
74AS534 disable time	t_5	6 ns (max)

Controlling A/D Conversions with the TMS320C25 Timer

The TMS320C25 timer can generate periodic interrupts that may be used to set the A/D sampling frequency. The TMS320C25 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 cycles of CLKOUT1, and the frequency of these interrupts is CLKOUT1/(PRD + 1).

As an example, consider a TMS320C25 operating at 40 MHz. The design of Figure 21 is utilized to interface the A/D converter to the TMS320C25. 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

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

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 TMS320C25 source code:

```
LDPK    0       ; Point to Data Page #0
LALK    999     ; ACC ← 999
SACL    DMA3    ; Period Register ← ACC
LACK    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:

```
ISR     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 ADC84.

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 describes the interface of an A/D conversion subsystem to the TMS320C25. This subsystem contains antialiasing filters, a sample-and-hold circuit, and a 12-bit A/D converter. Communication with the TMS320C25 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 TMS320C25's internal timer is used to establish the A/D sample periods, 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 TMS320C25. The design, shown in Figure 23, 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.

This design consists of three sections: the interface to the TMS320C25, the D/A converter, and the smoothing filter. Each of these sections is considered separately.

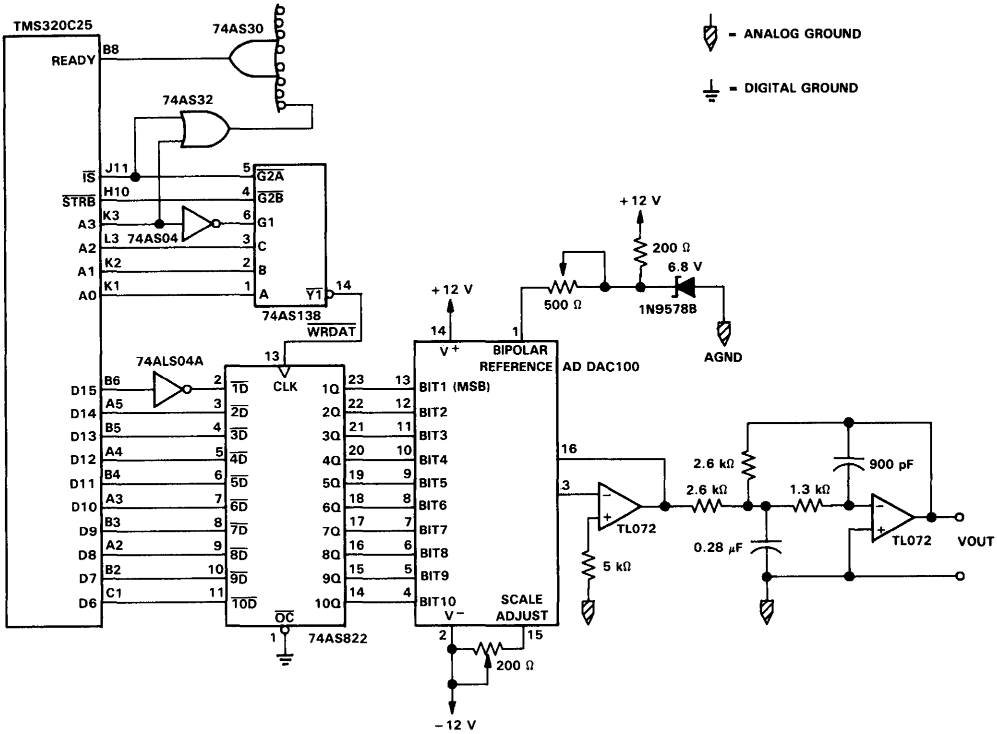


Figure 23. Interface of the ADDAC100 to the TMS320C25

Interface to the TMS320C25

The 74AS822 is used to latch the data from the TMS320C25. 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 TMS320C25. In bipolar mode, the DAC100 accepts data in complementary offset binary form. By inverting the MSB of the two's-complement data from the TMS320C25, 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 DAC100 becomes complementary offset binary form.

The circuit shown in Figure 23 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 TMS320C25 I/O space. The timing diagram for a write to the D/A is shown in Figure 24.

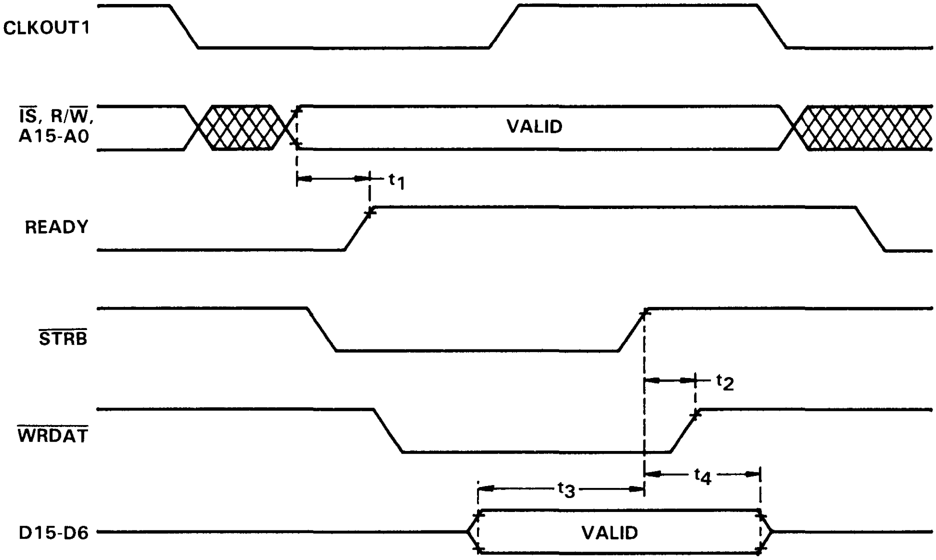

CLKOUT1
$\overline{\text{IS}}$, R/$\overline{\text{W}}$, A15-A0
VALID
t_1
READY
$\overline{\text{STRB}}$
t_2
$\overline{\text{WRDAT}}$
t_3
t_4
D15-D6
VALID

Figure 24. Interface Timing of the ADDAC100 to the TMS320C25

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 24). Time t_3 min is the minimum setup time for the TMS320C25 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 9 summarizes the most critical timing parameters of the ADDAC100 interface to the TMS320C25.

Table 9. Timing Parameters of the ADDAC100 Interface to the TMS320C25

Description	Symbol Used in Figure 19	Value
Address valid to READY valid	t_1	10.8 ns (max)
$\overline{\text{STRB}}$ high to $\overline{\text{WRDAT}}$ high	t_2	7.5 ns (max)
Data setup time before $\overline{\text{STRB}}$ high	t_3	19 ns (min)
Data setup time before $\overline{\text{WRDAT}}$ high	$t_3 + t_2$	21 ns (min)
Data hold time from $\overline{\text{STRB}}$ high	t_4	15 ns (min)
Data hold time from $\overline{\text{WRDAT}}$ high	$t_4 - t_2$	7.5 ns (min)

D/A Converter

The DAC100 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 DAC100 in Figure 23. The offset and gain adjustments are implemented with the 500-Ω and 200-Ω potentiometers, respectively.

Smoothing Filter

The output of the DAC100 contains high-frequency components to be removed by the smoothing filter. In the design of Figure 23, 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 25. This design is a cascade of three opamps, each implementing a second-order section.

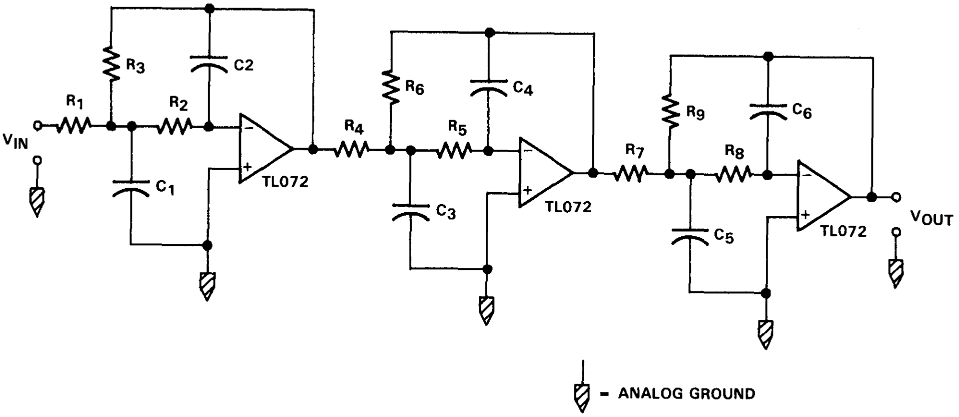


Figure 25. Sixth-Order Lowpass Filter Used for Antialiasing and Smoothing Filtering Operations

The design of Figure 25 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 10.[3]

Table 10. 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
R_1	2.588	2.588	2.588	2.588	2.588	2.588	2.588
C_1	0.280	0.101	0.0617	0.0475	0.0396	0.0297	0.0238
R_2	1.294	1.294	1.294	1.294	1.294	1.294	1.294
R_3	2.588	2.588	2.588	2.588	2.588	2.588	2.588
C_2	0.00936	0.00339	0.00207	0.00160	0.00133	0.000995	0.000796
R_4	7.071	7.071	7.071	7.071	7.071	7.071	7.071
C_3	0.0375	0.0136	0.00827	0.00637	0.00531	0.00398	0.00318
R_5	3.536	3.536	3.536	3.536	3.536	3.536	3.536
R_6	7.071	7.071	7.071	7.071	7.071	7.071	7.071
C_4	0.00936	0.00339	0.00207	0.00160	0.00133	0.000995	0.000796
R_7	9.659	9.659	9.659	9.659	9.659	9.659	9.659
C_5	0.0201	0.00726	0.00443	0.00341	0.00284	0.00213	0.00171
R_8	4.830	4.830	4.830	4.830	4.830	4.830	4.830
R_9	9.659	9.659	9.659	9.659	9.659	9.659	9.659
C_6	0.00936	0.00339	0.00207	0.00160	0.00133	0.000995	0.000796

NOTE: The unit for resistance is kΩ.
The unit for capacitor is μF.
The above values are not industry-standard values.

In summary, the 10-bit linear D/A converter provides analog output for the TMS320C25. 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 TMS320C25 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 TMS320C25.

References

1. *TMS320C25 User's Guide*, Texas Instruments (1986).

2. *Digital Signal Processing Applications with the TMS320 Family*, Texas Instruments (1986).

3. *TMS32010 Analog Interface Board User's Guide*, Texas Instruments.

TI Worldwide Sales Offices

ALABAMA: Huntsville: 500 Wynn Drive, Suite 514, Huntsville, AL 35805, (205) 837-7530.

ARIZONA: Phoenix: 8825 N. 23rd Ave., Phoenix, AZ 85021, (602) 995-1007.

CALIFORNIA: Irvine: 17891 Cartwright Rd., Irvine, CA 92714, (714) 660-8187; **Sacramento:** 1900 Point West Way, Suite 171, Sacramento, CA 95815, (916) 929-1521; **San Diego:** 4333 View Ridge Ave., Suite B., San Diego, CA 92123, (619) 278-9601; **Santa Clara:** 5353 Betsy Ross Dr., Santa Clara, CA 95054, (408) 980-9000; **Torrance:** 690 Knox St., Torrance, CA 90502, (213) 217-7010; **Woodland Hills:** 21220 Erwin St., Woodland Hills, CA 91367, (818) 704-7759.

COLORADO: Aurora: 1400 S. Potomac Ave., Suite 101, Aurora, CO 80012, (303) 368-8000.

CONNECTICUT: Wallingford: 9 Barnes Industrial Park Rd., Barnes Industrial Park, Wallingford, CT 06492, (203) 269-0074.

FLORIDA: Ft. Lauderdale: 2765 N.W. 62nd St., Ft. Lauderdale, FL 33309, (305) 973-8502; **Maitland:** 2601 Maitland Center Parkway, Maitland, FL 32751, (305) 660-4600; **Tampa:** 5010 W. Kennedy Blvd., Suite 101, Tampa, FL 33609, (813) 870-6420.

GEORGIA: Norcross: 5515 Spalding Drive, Norcross, GA 30092, (404) 662-7900

ILLINOIS: Arlington Heights: 515 W. Algonquin, Arlington Heights, IL 60005, (312) 640-2925.

INDIANA: Ft. Wayne: 2020 Inwood Dr., Ft. Wayne, IN 46815, (219) 424-5174; **Indianapolis:** 2346 S. Lynhurst, Suite J-400, Indianapolis, IN 46241, (317) 248-8555.

IOWA: Cedar Rapids: 373 Collins Rd. NE, Suite 200, Cedar Rapids, IA 52402, (319) 395-9550.

MARYLAND: Baltimore: 1 Rutherford Pl., 7133 Rutherford Rd., Baltimore, MD 21207, (301) 944-8600.

MASSACHUSETTS: Waltham: 504 Totten Pond Rd., Waltham, MA 02154, (617) 895-9100.

MICHIGAN: Farmington Hills: 33737 W. 12 Mile Rd., Farmington Hills, MI 48018, (313) 553-1500.

MINNESOTA: Eden Prairie: 11000 W. 78th St., Eden Prairie, MN 55344 (612) 828-9300.

MISSOURI: Kansas City: 8080 Ward Pkwy., Kansas City, MO 64114, (816) 523-2500; **St. Louis:** 11816 Borman Drive, St. Louis, MO 63146, (314) 569-7600.

NEW JERSEY: Iselin: 485E U.S. Route 1 South, Parkway Towers, Iselin, NJ 08830 (201) 750-1050

NEW MEXICO: Albuquerque: 2820-D Broadbent Pkwy NE, Albuquerque, NM 87107, (505) 345-2555.

NEW YORK: East Syracuse: 6365 Collamer Dr., East Syracuse, NY 13057, (315) 463-9291; **Endicott:** 112 Nanticoke Ave., P.O. Box 618, Endicott, NY 13760, (607) 754-3900; **Melville:** 1 Huntington Quadrangle, Suite 3C10, P.O. Box 2936, Melville, NY 11747, (516) 454-6600; **Pittsford:** 2851 Clover St., Pittsford, NY 14534, (716) 385-6770; **Poughkeepsie:** 385 South Rd., Poughkeepsie, NY 12601, (914) 473-2900.

NORTH CAROLINA: Charlotte: 8 Woodlawn Green, Woodlawn Rd., Charlotte, NC 28210, (704) 527-0930; **Raleigh:** 2809 Highwoods Blvd., Suite 100, Raleigh, NC 27625, (919) 876-2725.

OHIO: Beachwood: 23408 Commerce Park Rd., Beachwood, OH 44122, (216) 464-6100; **Dayton:** Kingsley Bldg., 4124 Linden Ave., Dayton, OH 45432, (513) 258-3877.

OREGON: Beaverton: 6700 SW 105th St., Suite 110, Beaverton, OR 97005, (503) 643-6758.

PENNSYLVANIA: Ft. Washington: 260 New York Dr., Ft. Washington, PA 19034, (215) 643-6450; **Coraopolis:** 420 Rouser Rd., 3 Airport Office Park, Coraopolis, PA 15108, (412) 771-8550.

PUERTO RICO: Hato Rey: Mercantil Plaza Bldg., Suite 505, Hato Rey, PR 00919, (809) 753-8700.

TEXAS: Austin: P.O. Box 2909, Austin, TX 78769, (512) 250-7655; **Richardson:** 1001 E. Campbell Rd., Richardson, TX 75080, (214) 680-5082; **Houston:** 9100 Southwest Frwy., Suite 237, Houston, TX 77036, (713) 778-6592; **San Antonio:** 1000 Central Parkway South, San Antonio, TX 78232, (512) 496-1779.

UTAH: Murray: 5201 South Green SE, Suite 200, Murray, UT 84107, (801) 266-8972.

VIRGINIA: Fairfax: 2750 Prosperity, Fairfax, VA 22031, (703) 849-1400.

WASHINGTON: Redmond: 5010 148th NE, Bldg B, Suite 107, Redmond, WA 98052, (206) 881-3080.

WISCONSIN: Brookfield: 450 N. Sunny Slope, Suite 150, Brookfield, WI 53005, (414) 785-7140.

CANADA: Nepean: 301 Moodie Drive, Mallorn Center, Nepean, Ontario, Canada, K2H9C4, (613) 726-1970. **Richmond Hill:** 280 Centre St. E., Richmond Hill L4C1B1, Ontario, Canada (416) 884-9181; **St. Laurent:** Ville St. Laurent Quebec, 9460 Trans Canada Hwy., St. Laurent, Quebec, Canada H4S1R7, (514) 335-8392.

ARGENTINA: Texas Instruments Argentina S.A.I.C.F.: Esmeralda 130, 15th Floor, 1035 Buenos Aires, Argentina, 1 + 394-3008.

AUSTRALIA (& NEW ZEALAND): Texas Instruments Australia Ltd.: 6-10 Talavera Rd., North Ryde (Sydney), New South Wales, Australia 2113, 2 + 887-1122; 5th Floor, 418 St. Kilda Road, Melbourne, Victoria, Australia 3004, 3 + 267-4677; 171 Philip Highway, Elizabeth, South Australia 5112, 8 + 255-2066.

AUSTRIA: Texas Instruments Ges.m.b.H.: Industriestrabe B/16, A-2345 Brunn/Gebirge, 2236-846210.

BELGIUM: Texas Instruments N.V. Belgium S.A.: Mercure Centre, Raketstraat 100, Rue de la Fusee, 1130 Brussels, Belgium, 2/720.80.00.

BRAZIL: Texas Instruments Electronicos do Brasil Ltda.: Rua Paes Leme, 524-7 Andar Pinheiros, 05424 Sao Paulo, Brazil, 0815-6166.

DENMARK: Texas Instruments A/S, Mairelundvej 46E, DK-2730 Herlev, Denmark, 2 - 91 74 00.

FINLAND: Texas Instruments Finland OY: Teollisuuskatu 19D 00511 Helsinki 51, Finland, (90) 701-3133.

FRANCE: Texas Instruments France: Headquarters and Prod. Plant, BP 05, 06270 Villeneuve-Loubet, (93) 20-01-01; Paris Office, BP 67 8-10 Avenue Morane-Saulnier, 78141 Velizy-Villacoublay, (3) 946-97-12; Lyon Sales Office, L'Oree D'Ecully, Batiment B, Chemin de la Forestiere, 69130 Ecully, (7) 833-04-40; Strasbourg Sales Office, Le Sebastopol 3, Quai Kleber, 67055 Strasbourg Cedex, (88) 22-12-66; Rennes, 23-25 Rue du Puits Mauger, 35100 Rennes, (99) 31-54-86; Toulouse Sales Office, Le Peripole—2, Chemin du Pigeonnier de la Cepiere, 31100 Toulouse, (61) 44-18-19; Marseille Sales Office, Noilly Paradis—146 Rue Paradis, 13006 Marseille, (91) 37-25-30.

TEXAS INSTRUMENTS

GERMANY (Fed. Republic of Germany): Texas Instruments Deutschland GmbH: Haggertystrasse 1, D-8050 Freising, 8161 + 80-4591; Kurfuerstendamm 195/196, D-1000 Berlin 15, 30 + 882-7365; III, Hagen 43/Kibbelstrasse, .19, D-4300 Essen, 201-24250; Frankfurter Allee 6-8, D-6236 Eschborm 1, 06196 + 8070; Hamburgerstrasse 11, D-2000 Hamburg 76, 040 + 220-1154, Kirchhorsterstrasse 2, D-3000 Hannover 51, 511 + 648021; Maybachstrabe 11, D-7302 Ostfildern 2-Nelingen, 711 + 547001; Mixikoring 19, D-2000 Hamburg 60, 40 + 637 + 0061; Postfach 1309, Roonstrasse 16, D-5400 Koblenz, 261 + 35044.

HONG KONG (+ PEOPLES REPUBLIC OF CHINA): Texas Instruments Asia Ltd., 8th Floor, World Shipping Ctr., Harbour City, 7 Canton Rd., Kowloon, Hong Kong, 3 + 722-1223.

IRELAND: Texas Instruments (Ireland) Limited: Brewery Rd., Stillorgan, County Dublin, Eire, 1 831311.

ITALY: Texas Instruments Semiconduttori Italia Spa: Viale Delle Scienze, 1, 02015 Cittaducale (Rieti), Italy, 746 694.1; Via Salaria KM 24 (Palazzo Cosma), Monterotondo Scalo (Rome), Italy, 6 + 9003241; Viale Europa, 38-44, 20093 Cologno Monzese (Milano), 2 2532541; Corso Svizzera, 185, 10100 Torino, Italy, 11 774545; Via J. Barozzi 6, 40100 Bologna, Italy, 51 355851.

JAPAN: Texas Instruments Asia Ltd.: 4F Aoyama Fuji Bldg., 6-12, Kita Aoyama 3-Chome, Minato-ku, Tokyo, Japan 107, 3-498-2111; Osaka Branch, 5F, Nissho Iwai Bldg., 30 Imabashi 3- Chome, Higashi-ku, Osaka, Japan 541, 06-204-1881; Nagoya Branch, 7F Daini Toyota West Bldg., 10-27, Meieki 4-Chome, Nakamura-ku Nagoya, Japan 450, 52-583-8691.

KOREA: Texas Instruments Supply Co.: 3rd Floor, Samon Bldg., Yuksam-Dong, Gangnam-ku, 135 Seoul, Korea, 2 + 462-8001.

MEXICO: Texas Instruments de Mexico S.A.: Mexico City, AV Reforma No. 450 — 10th Floor, Mexico, D.F., 06600, 5 + 514-3003.

MIDDLE EAST: Texas Instruments: No. 13, 1st Floor Mannai Bldg., Diplomatic Area, P.O. Box 26335, Manama Bahrain, Arabian Gulf, 973 + 274681.

NETHERLANDS: Texas Instruments Holland B.V., P.O. Box 12995, (Bullewijk) 1100 CB Amsterdam, Zuid-Oost, Holland 20 + 5602911.

NORWAY: Texas Instruments Norway A/S: PB106, Refstad 131, Oslo 1, Norway, (2) 155090.

PHILIPPINES: Texas Instruments Asia Ltd.: 14th Floor, Ba- Lepanto Bldg., 8747 Paseo de Roxas, Makati, Metro Manila, Philippines, 2 + 8188987.

PORTUGAL: Texas Instruments Equipamento Electronico (Portugal), Lda.: Rua Eng. Frederico Ulrich, 2650 Moreira Da Maia, 4470 Maia, Portugal, 2-948-1003.

SINGAPORE (+ INDIA, INDONESIA, MALAYSIA, THAILAND): Texas Instruments Asia Ltd.: 12 Lorong Bakar Batu, Unit 01-02, Kolam Ayer Industrial Estate, Republic of Singapore, 747-2255.

SPAIN: Texas Instruments Espana, S.A.: C/Jose Lazaro Galdiano No. 6, Madrid 16, 1/458.14.58.

SWEDEN: Texas Instruments International Trade Corporation (Sverigefilialen): Box 39103, 10054 Stockholm, Sweden, 8 - 235480.

SWITZERLAND: Texas Instruments, Inc., Reidstrasse 6, CH-8953 Dietikon (Zuerich) Switzerland, 1-740 2220.

TAIWAN: Texas Instruments Supply Co.: Room 903, 205 Tun Hwan Rd., 71 Sung-Kiang Road, Taipei, Taiwan, Republic of China, 2 + 521-9321.

UNITED KINGDOM: Texas Instruments Limited: Manton Lane, Bedford, MK41 7PA, England, 0234 67466; St. James House, Wellington Road North, Stockport, SK4 2RT, England, 61 + 442-7162.

BM

TI Sales Offices

ALABAMA: Huntsville (205) 837-7530.

ARIZONA: Phoenix (602) 995-1007;
Tucson (602) 624-3276.

CALIFORNIA: Irvine (714) 660-8187;
Sacramento (916) 929-0192;
San Diego (619) 278-9601;
Santa Clara (408) 980-9000;
Torrance (213) 217-7010;
Woodland Hills (818) 704-7759.

COLORADO: Aurora (303) 368-8000.

CONNECTICUT: Wallingford (203) 269-0074.

FLORIDA: Ft. Lauderdale (305) 973-8502;
Altamonte Springs (305) 260-2116;
Tampa (813) 870-6420.

GEORGIA: Norcross (404) 662-7900.

ILLINOIS: Arlington Heights (312) 640-2925.

INDIANA: Ft. Wayne (219) 424-5174;
Indianapolis (317) 248-8555.

IOWA: Cedar Rapids (319) 395-9550.

MARYLAND: Baltimore (301) 944-8600.

MASSACHUSETTS: Waltham (617) 895-9100.

MICHIGAN: Farmington Hills (313) 553-1500;
Grand Rapids (616) 957-4200.

MINNESOTA: Eden Prairie (612) 828-9300.

MISSOURI: Kansas City (816) 523-2500;
St. Louis (314) 569-7600.

NEW JERSEY: Iselin (201) 750-1050.

NEW MEXICO: Albuquerque (505) 345-2555.

NEW YORK: East Syracuse (315) 463-9291;
Melville (516) 454-6600; **Pittsford** (716) 385-6770;
Poughkeepsie (914) 473-2900.

NORTH CAROLINA: Charlotte (704) 527-0930;
Raleigh (919) 876-2725.

OHIO: Beachwood (216) 464-6100;
Dayton (513) 258-3877.

OREGON: Beaverton (503) 643-6758.

PENNSYLVANIA: Blue Bell (215) 825-9500.

PUERTO RICO: Hato Rey (809) 753-8700

TEXAS: Austin (512) 250-7655;
Houston (713) 778-6592; **Richardson** (214) 680-5082;
San Antonio (512) 496-1779.

UTAH: Murray (801) 266-8972.

VIRGINIA: Fairfax (703) 849-1400.

WASHINGTON: Redmond (206) 881-3080.

WISCONSIN: Brookfield (414) 785-7140.

CANADA: Nepean, Ontario (613) 726-1970;
Richmond Hill, Ontario (416) 884-9181;
St. Laurent, Quebec (514) 335-8392.

TI Regional Technology Centers

CALIFORNIA: Irvine (714) 660-8140,
Santa Clara (408) 748-2220.

GEORGIA: Norcross (404) 662-7945.

ILLINOIS: Arlington Heights (312) 640-2909.

MASSACHUSETTS: Waltham (617) 895-9197.

TEXAS: Richardson (214) 680-5066.

CANADA: Nepean, Ontario (613) 726-1970

Customer Response Center

TOLL FREE: (800) 232-3200

OUTSIDE USA: (214) 995-6611
(8:00 a.m. — 5:00 p.m. CST)

TI Distributors

TI AUTHORIZED DISTRIBUTORS IN USA

Arrow Electronics
General Radio Supply Company
Graham Electronics
Hall-Mark Electronics
Kierulff Electronics
Marshall Industries
Newark Electronics
Schweber Electronics
Time Electronics
Wyle Laboratories
Zeus Component, Inc. (Military Only)

TI AUTHORIZED DISTRIBUTORS IN CANADA

Arrow Electronics Canada
Future Electronics

TI AUTHORIZED DISTRIBUTORS IN USA

—OBSOLETE PRODUCT ONLY—
Rochester Electronics, Inc.
Newburyport, Massachusetts
(617) 462-9332

ALABAMA: Arrow (205) 837-6955;
Hall-Mark (205) 837-8700; Kierulff (205) 883-6070;
Marshall (205) 881-9235; Schweber (205) 895-0480.

ARIZONA: Arrow (602) 968-4800;
Hall-Mark (602) 437-1200; Kierulff (602) 437-0750;
Marshall (602) 968-6181; Schweber (602) 997-4874;
Wyle (602) 866-2888.

CALIFORNIA: Los Angeles/Orange County:
Arrow (818) 701-7500, (714) 838-5422;
Hall-Mark (818) 716-7300, (714) 669-4700,
(213) 217-8400; Kierulff (213) 725-0325, (714) 731-5711,
(714) 220-6300; (818) 407-2500;
Marshall (818) 407-0101, (818) 459-5500,
(714) 458-5395; Schweber (818) 999-4702;
(714) 863-0200; (213) 327-8409; Wyle (213) 322-8100,
(818) 880-9001, (714) 863-9953; Zeus (714) 921-9000;
Sacramento: Hall-Mark (916) 722-8600;
Marshall (916) 635-9700; Schweber (916) 929-9732;
Wyle (916) 638-5282;
San Diego: Arrow (619) 565-4800;
Hall-Mark (619) 268-1201; Kierulff (619) 278-2112;
Marshall (619) 578-9600; Schweber (619) 450-0454;
Wyle (619) 565-9171;
San Francisco Bay Area: Arrow (408) 745-6600;
(415) 487-4600; Hall-Mark (408) 946-0900;
Kierulff (408) 971-2600; Marshall (408) 942-4600;
Schweber (408) 946-7171; Wyle (408) 727-2500;
Zeus (408) 998-5121.

COLORADO: Arrow (303) 696-1111;
Hall-Mark (303) 790-1662; Kierulff (303) 790-4444;
Marshall (303) 451-8444; Schweber (303) 799-0258;
Wyle (303) 457-9953.

CONNECTICUT: Arrow (203) 265-7741;
Hall-Mark (203) 269-0100; Kierulff (203) 265-1115;
Marshall (203) 265-3822; Schweber (203) 748-7080.

FLORIDA: Ft. Lauderdale: Arrow (305) 429-8200;
Hall-Mark (305) 971-9280; Kierulff (305) 486-4004;
Marshall (305) 977-4880; Schweber (305) 977-7511;
Orlando: Arrow (305) 725-1480;
Hall-Mark (305) 855-4020; Kierulff (305) 682-6923;
Marshall (305) 841-1878; Schweber (305) 331-7555;
Zeus (305) 365-3000;
Tampa: Hall-Mark (813) 530-4543;
Marshall (813) 576-1399.

GEORGIA: Arrow (404) 449-8252;
Hall-Mark (404) 447-8000; Kierulff (404) 447-5252;
Marshall (404) 923-5750; Schweber (404) 449-9170.

ILLINOIS: Arrow (312) 397-3440;
Hall-Mark (312) 860-3800; Kierulff (312) 250-0500;
Marshall (312) 490-0155; Newark (312) 784-5100;
Schweber (312) 364-3750.

TEXAS INSTRUMENTS

INDIANA: Indianapolis: Arrow (317) 243-9353;
Graham (317) 634-8202; Hall-Mark (317) 872-8875;
Marshall (317) 297-0483;
Ft. Wayne: Graham (219) 423-3422.

IOWA: Arrow (319) 395-7230;
Schweber (319) 373-1417.

KANSAS: Kansas City: Arrow (913) 541-9542;
Hall-Mark (913) 888-4747; Marshall (913) 492-3121;
Schweber (913) 492-2921.

MARYLAND: Arrow (301) 995-0003;
Hall-Mark (301) 988-9800; Kierulff (301) 840-1155;
Marshall (301) 840-9450; Schweber (301) 840-5900;
Zeus (301) 997-1118.

MASSACHUSETTS: Arrow (617) 933-8130;
Hall-Mark (617) 667-0902; Kierulff (617) 667-8331;
Marshall (617) 658-0810; Schweber (617) 275-5100,
(617) 657-0760; Time (617) 532-6200;
Zeus (617) 863-8800.

MICHIGAN: Detroit: Arrow (313) 971-8220;
Marshall (313) 525-5850; Newark (313) 967-0600;
Schweber (313) 525-8100;
Grand Rapids: Arrow (616) 243-0912.

MINNESOTA: Arrow (612) 830-1800;
Hall-Mark (612) 941-2600; Kierulff (612) 941-7500;
Marshall (612) 559-2211; Schweber (612) 941-5280.

MISSOURI: St. Louis: Arrow (314) 567-6888;
Hall-Mark (314) 291-5350; Kierulff (314) 997-4956;
Schweber (314) 739-0526.

NEW HAMPSHIRE: Arrow (603) 668-6968;
Schweber (603) 625-2250.

NEW JERSEY: Arrow (201) 575-5300,
(609) 596-8000; General Radio (609) 964-8560;
Hall-Mark (201) 575-4415, (609) 235-1900;
Kierulff (201) 575-6750, (609) 235-1444;
Marshall (201) 882-0320, (609) 234-9100;
Schweber (201) 227-7880.

NEW MEXICO: Arrow (505) 243-4566.

NEW YORK: Long Island: Arrow (516) 231-1000;
Hall-Mark (516) 737-0600; Marshall (516) 273-2053;
Schweber (516) 334-7555; Zeus (914) 937-7400.
Rochester: Arrow (716) 427-0300;
Marshall (716) 235-7620; Schweber (716) 424-2222.
Syracuse: Marshall (607) 798-1611.

NORTH CAROLINA: Arrow (919) 876-3132,
(919) 725-8711; Hall-Mark (919) 872-0712;
Kierulff (919) 872-8410; Marshall (919) 878-9882;
Schweber (919) 876-0000.

OHIO: Cleveland: Arrow (216) 248-3990;
Hall-Mark (216) 349-4632; Kierulff (216) 831-5222;
Marshall (216) 248-1788; Schweber (216) 464-2970.
Columbus: Arrow (614) 885-8362;
Hall-Mark (614) 888-3313;
Dayton: Arrow (513) 435-5563;
Kierulff (513) 439-0045; Marshall (513) 236-8088;
Schweber (513) 439-1800.

OKLAHOMA: Arrow (918) 665-7700;
Kierulff (918) 252-7537; Schweber (918) 622-8000.

OREGON: Arrow (503) 684-1690;
Kierulff (503) 641-9153; Wyle (503) 640-6000;
Marshall (503) 644-5050.

PENNSYLVANIA: Arrow (412) 856-7000,
(215) 928-1800; General Radio (215) 922-7037;
Schweber (215) 441-0600, (412) 782-1600.

TEXAS: Austin: Arrow (512) 835-4180;
Hall-Mark (512) 258-8848; Kierulff (512) 835-2090;
Marshall (512) 837-1991; Schweber (512) 458-8253;
Wyle (512) 834-9957;
Dallas: Arrow (214) 380-6464;
Hall-Mark (214) 553-4300; Kierulff (214) 840-0110;
Marshall (214) 233-5200; Schweber (214) 661-5010;
Wyle (214) 235-9953; Zeus (214) 783-7010;
Houston: Arrow (713) 530-4700;
Hall-Mark (713) 781-6100; Kierulff (713) 530-7030;
Marshall (713) 895-9200; Schweber (713) 784-3600;
Wyle (713) 879-9953.

UTAH: Arrow (801) 972-0404;
Hall-Mark (801) 972-1008; Kierulff (801) 973-6913;
Marshall (801) 485-1551; Wyle (801) 974-9953; .

WASHINGTON: Arrow (206) 643-4800;
Kierulff (206) 575-4420; Wyle (206) 453-8300;
Marshall (206) 747-9100.

WISCONSIN: Arrow (414) 792-0150;
Hall-Mark (414) 797-7844; Kierulff (414) 784-8160;
Marshall (414) 797-8400; Schweber (414) 784-9020.

CANADA: Calgary: Future (403) 235-5325;
Edmonton: Future (403) 438-2858;
Montreal: Arrow Canada (514) 735-5511;
Future (514) 694-7710;
Ottawa: Arrow Canada (613) 226-6903;
Future (613) 820-8313;
Quebec City: Arrow Canada (418) 687-4231;
Toronto: Arrow Canada (416) 661-0220;
Future (416) 638-4771;
Vancouver: Future (604) 294-1166
Winnipeg: Future (204) 339-0554

BS