

Interfacing External Memory to the TMS32010

APPLICATION REPORT: SPRA125

*Author: Jon Bradley
Digital Signal Processing – Semiconductor Group*

*Digital Signal Processing Solutions
1989*



IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Interfacing External Memory to the TMS32010

Abstract

This report describes interface circuits for the TMS32010 to asynchronous inputs and to external memory devices, such as external ROM or RAM. A description of hardware peripheral devices interface device produced by Pacific Microcircuits is also included, which eases the TMS32010 interface to both external memory and codec devices.



Product Support on the World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

INTRODUCTION

External ROM or RAM can be interfaced to the memory bus of the TMS32010 microprocessor in applications that require additional RAM or program memory.

The purpose of this application report is to describe two basic low-cost methods for expanding the TMS32010's memory configuration:

1. Direct expansion, utilizing a standard memory cycle for memory access
2. Extended memory interface, utilizing an address (latched using a standard memory cycle) that automatically increments or decrements after each access.

Of the two methods, the first method is very efficient for program and small data memory expansion, whereas the second method is more useful for large data memory expansion.

The design techniques presented here can easily be extended to encompass interface of other devices to the TMS32010. Each of the circuits discussed in this application report has been built and tested to verify its operation.

RAM/ROM PROGRAM MEMORY EXPANSION

For systems requiring additional program memory or small amounts of external data memory, the direct expansion circuit described in this section provides a straightforward approach.

The TMS32010 program memory can be expanded beyond the 1.5K-word internal capacity in two ways:

1. Implementing an additional 2.5K words externally ($MC/\overline{MP} = 1$), or
2. Implementing the full 4K-word program memory space externally ($MC/\overline{MP} = 0$).

In either case, the memory is accessed in a single memory cycle and appears no different to the TMS32010 than internal program memory. The circuit described in this section uses the full 4K-word program memory space implemented externally using $MC/\overline{MP} = 0$. This configuration is useful in applications where, perhaps for cost reasons, using a masked ROM version of the TMS32010 is impractical.

Design Considerations

An important consideration in the design of the direct expansion circuit is the desire to minimize chip count in order to reduce cost. This is an important factor in digital signal processing (DSP) systems since their cost must compete with that of analog approaches. In this circuit, as little additional logic as possible is used without sacrificing performance.

The memories used in the circuit are chosen to provide minimum chip count using currently available devices. The circuit is configured with half of the address space implemented as RAM and the other half as PROM. The RAMs used are Advanced Micro Devices Am9128-70 2K x 8 static NMOS RAMs, and the PROMs used are Texas Instruments TBP28S166 2K x 8 TTL PROMs. This memory configuration results in a minimum chip count and provides an even mix between RAM and PROM; however, other RAM/ROM mixes may be used. Note that if PROMs or ROMs are the only external memory required in the system, no additional logic is needed since the interface to most of these devices allows direct connection to the TMS32010.

Using RAM as program memory allows downloading into the address space from slower (possibly EPROM) memory or a host system if required, and also allows for communication between program and data memory spaces using the Table Read (TBLR) and Table Write (TBLW) instructions. If internal program ROM and external program RAM are required, the same external memory configuration may be used with MC/\overline{MP} set to 1. Note that in this case, external RAM located at addresses coincident with those of the internal program ROM cannot be used without some modification of the address decoding scheme.

Functional Description

The direct memory expansion circuit, shown in Figure 1, consists of the four memory chips, a set of address bus buffers, and logic that controls the address bus buffers and enables the memories' three-state outputs.

The 2K words of PROM are located at addresses 0-7FF, preventing any conflict with I/O ports if present in the system, and RAM is located at addresses 800-FFF. PROM or RAM is selected on the basis of the most significant address line using a single inverter (U8:8,9).

Circuit Operation

Read operations are performed from the PROM/RAM memory space either during opcode or operand fetches or during TBLR instructions. Write operations occur to RAM only during TBLW instructions; write operations to PROM have no effect on the circuit.

The TMS32010 initiates a read operation from either the RAM (U2,3) or the PROM (U4,5) by presenting an address and driving \overline{MEN} low (see Figure 2). Since the control line to the address latches (buffered \overline{WE}) is high, these latches (U6,7) are transparent, and the address is presented to the memories after a short propagation delay. After the memory-access time delay, data is available internally but is not driven out of the chip since the memories' three-state outputs (controlled by \overline{RDEN}) are not enabled. \overline{RDEN} is generated so that the memories' outputs are not enabled on a read operation until CLKOUT goes high. This

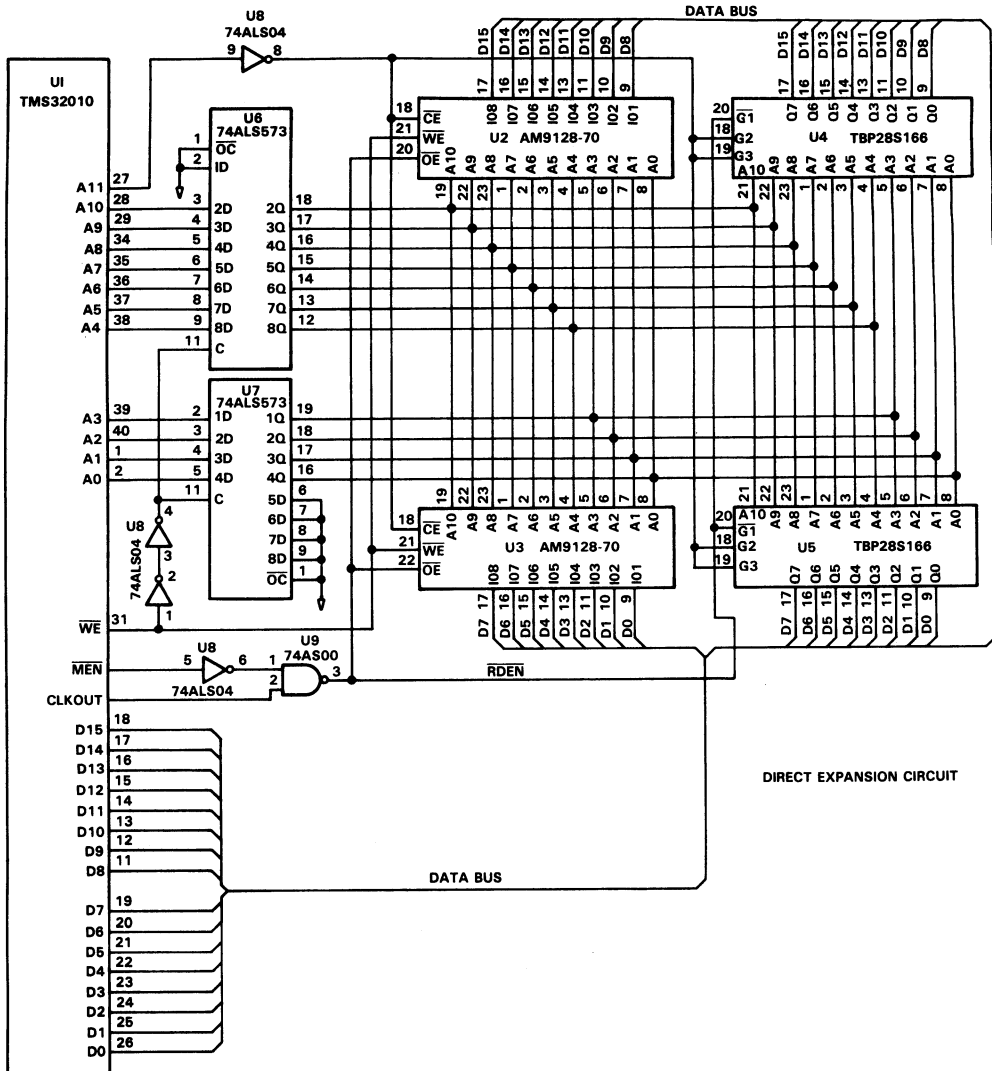


Figure 1. Direct Memory Expansion Circuit

ensures that with fast memories, if $\overline{\text{MEN}}$ occurs early, a data bus conflict will not occur when the read follows a write operation. $\overline{\text{MEN}}$ can be used directly to control the output drivers of the memories if it can be guaranteed that the memory will not drive the data bus immediately following a write operation or if buffers are used on the data bus. Buffers are not used in this design in order to minimize chip count.

After CLKOUT goes high, the memory outputs are enabled. The TMS3201C processor's 50-ns read data setup time to CLKOUT falling is met from CLKOUT rising by an 8-ns propagation delay through the 74AS00 NAND gate

(U9) and the memories' 40-ns maximum output turn-on delay (with a 100-ns high portion of CLKOUT). At the end of the cycle, the TMS3201C requires that read data be held at least until CLKOUT falls or $\overline{\text{MEN}}$ goes high, whichever occurs first. This is guaranteed since $\overline{\text{RDEN}}$ will not go high until at least one of these signals changes state.

Write operations begin in a similar manner to read operations, with the exception that the $\overline{\text{WE}}$ signal is active instead of $\overline{\text{MEN}}$ (see Figure 3). Since the address buffers are controlled by $\overline{\text{WEB}}$ (buffered $\overline{\text{WE}}$), which is high for the first half of the cycle, the buffers are transparent when the address becomes valid. Then, when $\overline{\text{WEB}}$ goes low, the address remains latched until after $\overline{\text{WE}}$ goes high.

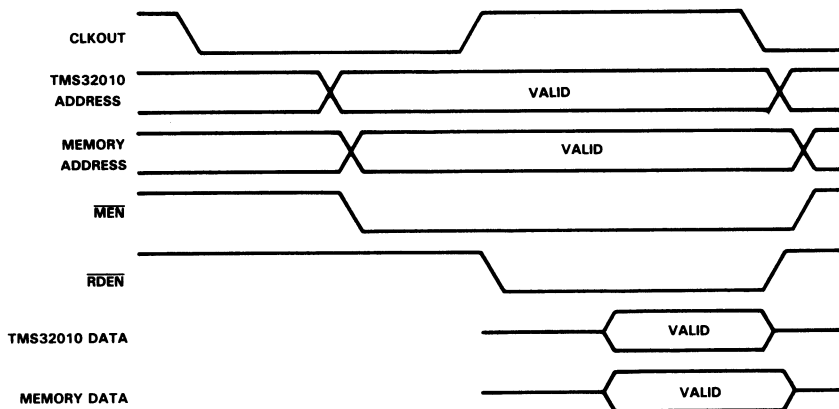


Figure 2. Memory Read Timing

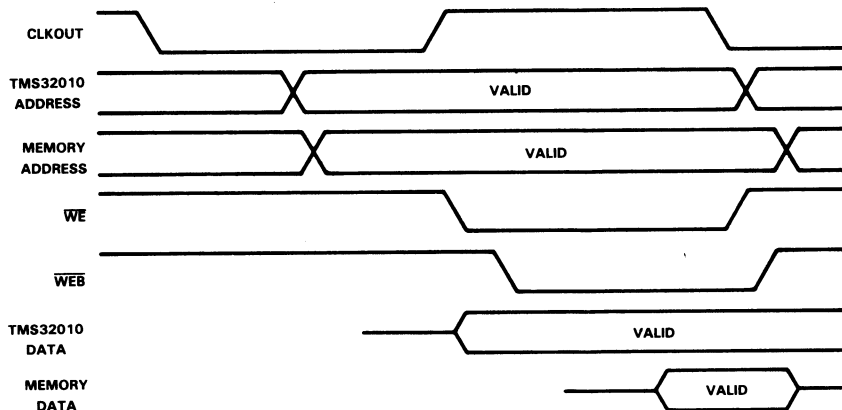


Figure 3. Memory Write Timing

The address bus buffers provide the 5-ns address hold time required by the RAMs following \overline{WE} going high. This hold time is provided by the propagation delay of \overline{WE} resulting from the two 74ALS04s (U8, 1-4) and the buffer propagation delay. While it is not generally good design practice to rely on propagation delays for timing, in this case the technique can be used to eliminate the need for more cumbersome design approaches, such as delay lines, since the minimum propagation delays for these ALS devices are specified. Note that the address bus buffers are not required if memories with 0-ns write-address hold times (such as the INMOS 4K x 4 devices) are used. These devices are not used here because their organization does not suit the desired configuration of this design.

The remaining memory bus timing requirements for write operations are also easily met by this circuit design. The RAMs require a 30-ns data setup time with respect to \overline{WE} rising, of which the TMS32010 provides about 80 ns. The data hold time required with respect to \overline{WE} going high is 5 ns, of which the TMS32010 provides a minimum of 20 ns.

EXTENDED MEMORY INTERFACE

If large program memory expansion is required, bank switching techniques can be employed with the direct expansion scheme to allow greater program memory space, some of which can still be used for small segments of data. These segments of data, however, can only be accessed using TBLR/TBLW instructions. For this reason, the direct expansion scheme is quite inefficient when large amounts of external data storage are required.

To implement large data memory expansion, the extended memory interface can be used. With this approach, memory can be accessed in two cycles once an address has been loaded, making this technique preferable to the direct memory expansion scheme for data storage. Note that the primary savings in cycles required to access the memory result from loading the address only once and having this address increment or decrement with each access. Thus, for the most efficient use of this memory, data should be stored sequentially to avoid having to reload an address for each

access. If data is not saved sequentially, four cycles are required for each access, making the direct expansion scheme the preferred approach.

The extended memory interface is more efficient for data storage, but may be used to access instructions even though they cannot be executed directly. Instructions are accessed by using an IN instruction, followed by a TBLW instruction, thereby placing the instruction in program memory. Because the transfer requires a minimum of five cycles, this technique should only be used to store instructions that need to be accessed infrequently. This feature is useful, for example, for implementing downloads from slow memory or a host system.

The extended memory expansion approach may also be used in conjunction with one of the direct memory expansion schemes to expand both program and data memory efficiently.

Design Considerations

A primary consideration of the extended memory expansion design is to implement an efficient interface to large amounts of data memory. The program interface to this memory uses the I/O ports. These ports are accessed in two cycles, whereas three cycles are required to access program memory via the TBLR/TBLW instructions.

This interface is mapped into three port locations:

1. Port 0, which receives the starting address for the memory access
2. Port 1, which decrements the address following each access
3. Port 2, which increments the address following each access.

Functional Description

The extended memory interface circuit, shown in Figure 4, contains the minimum amount of logic required to efficiently communicate with larger amounts of memory at relatively high speeds. Due to the nature of the interface, the devices used for the memory space are not required to be as fast as those used in the direct expansion circuit. The devices used are Synertek SY2128-1 4K x 4 NMOS static RAMs, chosen on the basis of their organization and availability. The RAM organization provides a 4K x 16 memory space using only four chips.

TMS32010

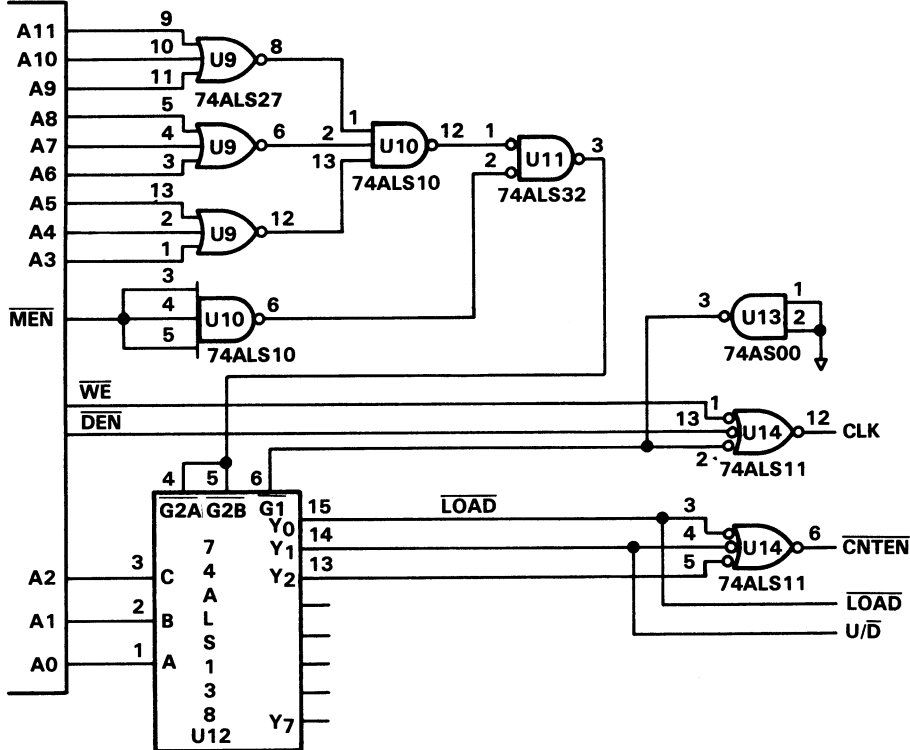


Figure 4. Extended Memory Interface Circuit (page 2 of 2)

The address used to access these RAMs is derived from a 12-bit up/down counter, implemented using three 74LS169A 4-bit counters (U6-8) cascaded together. An address is loaded into the counters, using an OUT instruction to port address 0. Then, with each access to port 1 or 2, this address is decremented or incremented, respectively.

The logic controlling this interface consists of a 74ALS138 decoder (U12), which decodes the three port addresses and some miscellaneous gating that generates strobe and enable signals (U9-11,13,14).

Circuit Operation

The memory in the extended memory interface circuit is accessed using three types of memory cycles:

1. An address load cycle
2. A read cycle to RAM
3. A write cycle to RAM.

Addresses are loaded by writing the desired address to port 0. In order to simplify the design, logic to allow reading of the address counters was not included. Therefore, port 0 should not be read or improper loading of the counters

will occur. Note that although reading port 0 may corrupt the counters' contents, succeeding loads will function properly.

After an address is loaded, each access to either port 1 or 2 decrements or increments, respectively, the memory address after the completion of the cycle. Since the effective address for the next memory cycle becomes valid shortly after the end of the current cycle, the RAMs used can be quite slow. Their speed is limited only by the output enable time, which is generally significantly faster than the address access time. The memories used in this circuit must have an output enable time of no more than 42 ns, but their address access time can be as slow as 150 ns or more. For this reason, less expensive memories can be used.

Figure 5 shows an address load operation. The address presented by the TMS32010 is decoded by the 74ALS138 and some random logic, consisting of AND and OR gates and an inverter, to detect an access to port 0. This decode results in the LOAD and CNTEN signals going active (low). The LOAD signal indicates that this is a load operation, and CNTEN enables the counters. CLK, the clock signal to the

counters (in this case, derived from \overline{WE}), goes high at the end of the cycle when both the \overline{LOAD} and a \overline{CNTEN} decodes are stable. The rising edge of CLK synchronously clocks the

address from the TMS32010 data lines into the counters. Shortly after the rising edge of CLK, the loaded address is available to the memories at the output of the counters.

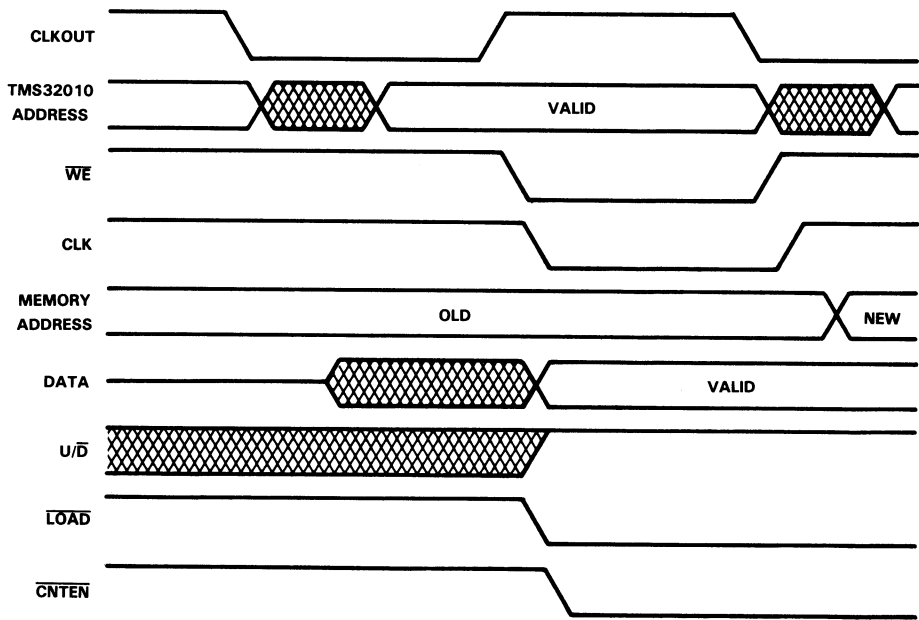


Figure 5. Address Load Timing

In a read operation, as shown in Figure 6, the TMS32010 address is decoded to detect a port address in the same manner as in an address load operation. In this case, however, accessing port addresses 1 and 2 results in the $\overline{\text{CNTEN}}$ signal being active and the $\overline{\text{LOAD}}$ signal inactive. In addition, the $\text{U}/\overline{\text{D}}$ signal asserts the correct state of the up/down control input to the counters depending on whether the cycle is an incrementing or decrementing access. In an I/O read cycle from the TMS32010, the $\overline{\text{DEN}}$ signal is active, and the interface uses this signal both to enable the memories'

output buffers and to clock the counters at the end of the buffer access. Since inverted $\overline{\text{DEN}}$ is gated with CLKOUT to enable the memories' output buffers, the buffers will not be enabled until CLKOUT goes high. As in the direct memory interface, this feature is included to avoid any bus conflicts that might occur between the TMS32010 and the memories following a write operation. Note that if the system reads other port addresses, $\overline{\text{DEN}}$ must be further gated to ensure that only the accessed port's output buffers are enabled.

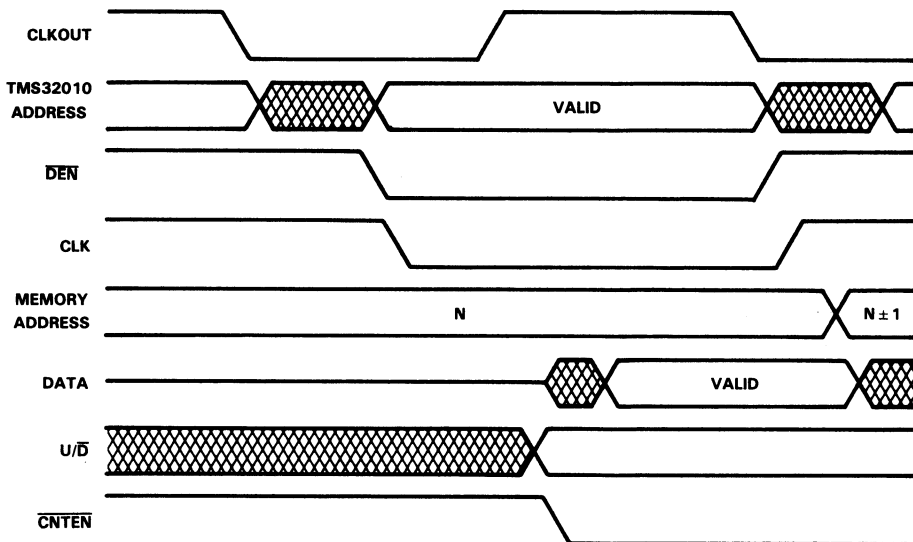


Figure 6. RAM Read Timing

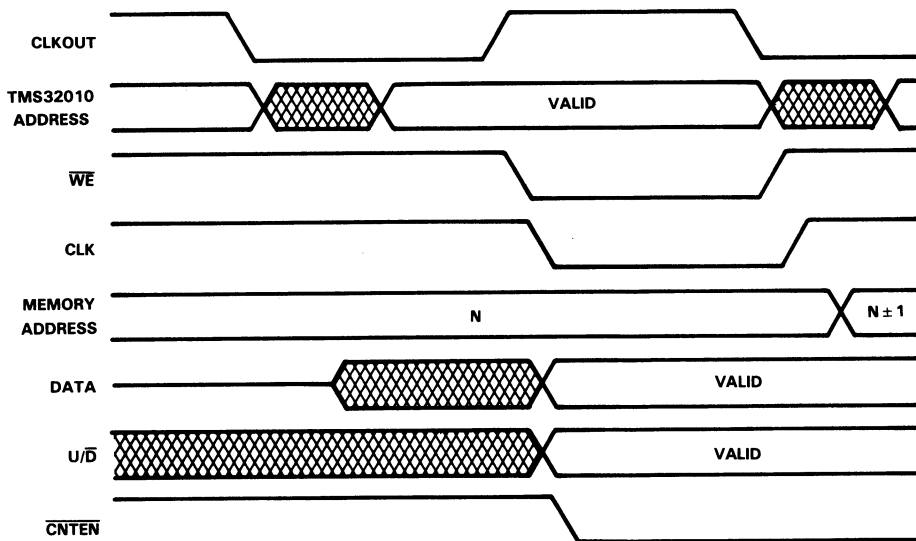


Figure 7. RAM Write Timing

The 35-ns output enable time of the SY2128-1 RAMs (U2-5), added to the delay due to the 74AS00 (U15), acceptably meets the 50-ns data setup time required by the TMS32010. At the end of the cycle, $\overline{\text{DEN}}$ going high causes CLK to go high which either increments or decrements the memory address contained in the counters depending on the state of U/D. Thus, the following access is made from the next sequential location.

A write cycle, as shown in Figure 7, occurs in much the same manner as a read cycle. The TMS32010 address is decoded to activate $\overline{\text{CNTEN}}$ and produce the correct state of U/D. At the end of the cycle, when $\overline{\text{WE}}$ goes high, the CLK signal generated from $\overline{\text{WE}}$ strobes the data into the

memories and increments or decrements the address in the counters.

SUMMARY

Two basic low-cost methods for expanding the TMS32010's memory configuration have been described in this application report. The direct memory expansion scheme provides program and small data memory expansion, and the extended memory interface provides large data memory expansion. The design techniques used in these interfaces may be extended to encompass interface of other devices to the TMS32010.

APPENDIX

Pacific Microcircuits Ltd. in British Columbia, Canada, has introduced a peripheral chip to support the TMS32010. The preliminary specification is included in the appendix of this application report to facilitate minimum chip-count design in TMS32010-based systems. In addition to the electrical specification, a schematic for an audio-processor board and an application note for PD32HC01 interrupt handling are provided.

The PD32HC01 is a digital signal processor interface circuit intended for use in voice-band signal processing applications. This CMOS single chip offers an efficient interface between the TMS32010 and external RAM, ROM, and a serial codec (see Figure A-1).

For further information on price, availability, and support, please refer to the list of Pacific Microcircuits Ltd. representatives on the last page of this appendix.

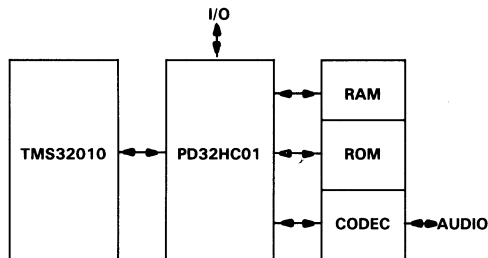


Figure A-1. Voice-Band Signal Processing Interface

PD32HC01 SIGNAL PROCESSOR INTERFACE for the T.I. TMS32010

Preliminary - September 1985

Features

- Single-chip solution to TMS32010 interfacing
- Serial Codec port
- Serial Data comm. port
- I/O and Interrupt control
- Decoding for external RAM and ROM memory
- I/O expansion interface
- 2400 Hz bit rate generator
- Low-power CMOS technology

Description

The PD32HC01 is a DIGITAL SIGNAL PROCESSOR INTERFACE circuit, intended for use in voice band signal processing circuits. **It provides an optimized interface between the Texas Instruments TMS32010 digital signal processor and external RAM, ROM, and Codec.**

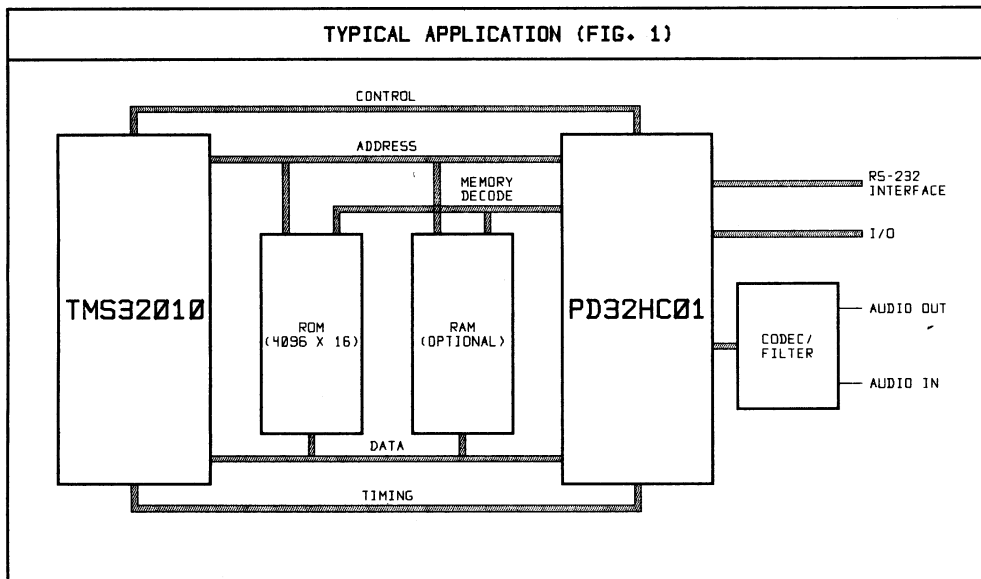
Applications

- Voice coders/decoders
- Speech synthesis
- Speech recognition
- Digital telephony
- Data communications
- Digital radio

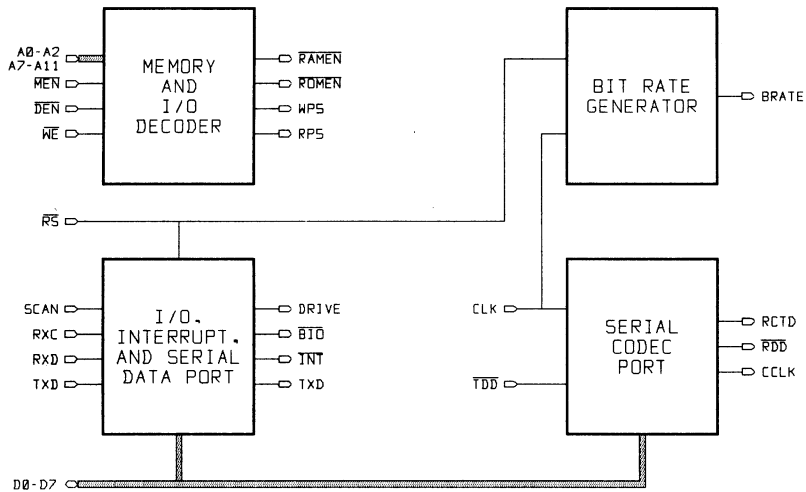
Package Availability

- 40 Lead DIL Ceramic (PD32HC01C)
- 40 Lead DIL Plastic (PD32HC01E)
- 44 Lead Surface Mounted Plastic (PD32HC01P)

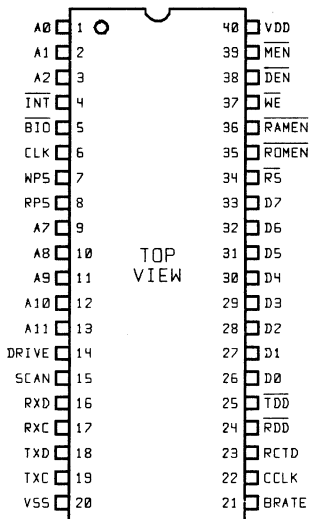
TYPICAL APPLICATION (FIG. 1)



FUNCTIONAL BLOCK DIAGRAM (FIG. 2)

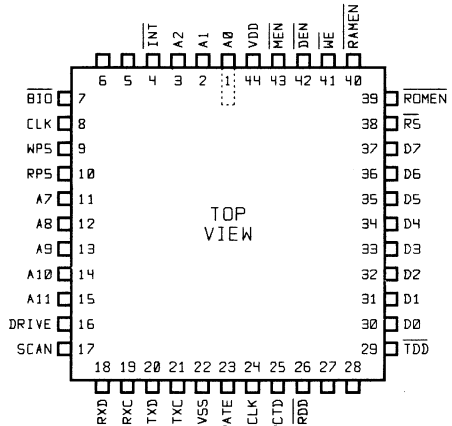


PIN ASSIGNMENTS



40 PIN DUAL-IN-LINE PACKAGE

PD32HC01C (CERAMIC)
PD32HC01E (PLASTIC)



44 PIN SURFACE-MOUNT PACKAGE

PD32HC01P (PLASTIC)

PD32HC01 Pin Description

Pin	Name	I/O	Description
1-3	A0-A2	Inputs	Address bus from processor.
4	$\overline{\text{INT}}$	Output	Interrupt request to processor. Responds to RXC, TXC, or Codec A/D interrupts.
5	$\overline{\text{BIO}}$	Output	Polled output port bit to processor. Data source to be polled is specified in the Peripheral Status Register.
6	CLK	Input	4.128 MHz (nominal) clock, derived from processor clock. Drives the bit rate generator and Codec interface timing.
7,8	WP5, RP5	Output	Decoded I/O port write and read pulses for I/O expansion.
9-13	A7-A11	Inputs	Address bus from processor.
14	DRIVE	Output	Output bit controlled from the Peripheral Status Register.
15	SCAN	Input	Input bit selected from the Peripheral Status Register to appear on $\overline{\text{BIO}}$.
16	RXD	Input	Serial data input. Must be stable on the rising edge of RXC. Selected from the Peripheral Status Register to appear on $\overline{\text{BIO}}$.
17	RXC	Input	Serial data receive clock. Rising edge retimes RXD, and raises an RX clock interrupt.
18	TXD	Output	Serial data output. Programmed from the Peripheral Status Register. Edges of TXD are synchronized to the rising edge of TXC.
19	TXC	Input	Serial data transmit clock. Rising edge clocks out data onto TXD from the Peripheral Status Register, and raises a TX clock interrupt.
20	VSS	Power	Negative supply (ground).
21	BRATE	Output	2400 Hz square wave (CLK / 1720—mask programmable).
22	CCLK	Output	2.064 MHz (nominal) Codec clock.
23	RCTD	Output	Codec framing pulse for Codec synchronization. Codec A/D interrupt occurs 16 CLK cycles after RCTD goes high.
24	$\overline{\text{RDD}}$	Output	Serial data output to Codec. PCM data is shifted out on the rising edges of the first 8 CCLK cycles after the rising edge of RCTD.
25	$\overline{\text{TDD}}$	Input	Serial data input from Codec. PCM data is sampled on the first 8 CCLK falling edges after the rising edge of RCTD.
26-33	D0-D7	In/Out	Data bus to chip.
34	RS	Input	Master reset to chip. A low on this input will reset the $\overline{\text{INT}}$ signal, and initialize the bit rate timer. This is a Schmitt trigger input.
35	$\overline{\text{ROMEN}}$	Output	ROM enable output. This signal goes low during a valid read from memory locations >000 - >F7F ($\overline{\text{FEN}}$ low).
36	$\overline{\text{RAMEN}}$	Output	RAM enable output. This signal goes low during a valid read or write to memory locations >F80 - >FFF ($\overline{\text{FEN}}$ low or $\overline{\text{WE}}$ low).
37	$\overline{\text{WE}}$	Input	Write enable to chip. Goes low for I/O or RAM write operations.
38	$\overline{\text{DEN}}$	Input	Data enable to chip. Goes low for I/O read operations.
39	$\overline{\text{FEN}}$	Input	Memory enable to chip. Goes low for ROM or RAM reads.
40	VDD	Power	Positive supply (+5 Volts).

Detailed Description

The PD32HC01 consists of 4 functional blocks: a memory and I/O decoder; I/O, interrupt, and serial data port control; a serial Codec port; and a bit rate generator (see figures. 2 & 3).

Memory and I/O Decoder

The memory and I/O decoder segments the 4K word address space of the TMS32010 into 3 areas: a 3968 word ROM area inclusive of addresses >000 to >F7F; a 128 word RAM area inclusive of addresses >F80 to >FFF; and from addresses >XX0 to >XX7, an I/O expansion port, an I/O, interrupt, and serial data port control; and a serial Codec port.

Memory Decoding

The ROMEN signal is used for selecting external program ROM. It goes low during memory read or table read cycles (PIEN low), and the processor address is less than >F80.

The RAMEN signal is used for selecting external data RAM. It goes low during memory read, table read, or table write cycles (PIEN or WE low), and the processor address is above >F7F.

I/O Expansion Port

The RPS and WPS signals are used for I/O port expansion. RPS goes high during an I/O read cycle from port 5 (DEN low). WPS goes high during an I/O write cycle to port 5, or a table write cycle to address >XX5 (WE low).

I/O, Interrupt, and Serial Data Port

The Program Status Register (PSR) at port location 6 controls the DRIVE and TXD output signals; the INT output operation via the Codec A/D, TX clock, and RX clock interrupt mask bits; and selects inputs to be tested on BIO (interrupt flags; the SCAN input; or the retimed RXD input). The bit encoding of the PSR is shown below:

- bit 0: RXMSK, RX clock interrupt mask.
- bit 1: TXMSK, TX clock interrupt mask.
- bit 2: ADMSK, Codec interrupt mask.

Writing 1's to these bits will mask interrupts from the respective sources, and/or clear posted interrupts. Writing 0's will enable interrupts. By testing for the interrupting source on the BIO line, interrupt vectoring can be managed (see bits 3,4,5 description).

bits 3,4,5: BIO Source Select. These three bits select one of five input sources (interrupt flags or pin inputs) onto the BIO output:

bit 5	bit 4	bit 3	Selected Source
0	0	0	Codec A/D Int. status
0	0	1	TX Clock Int. status
0	1	0	RX Clock Int. status
0	1	1	SCAN bit input
1	X	X	Retimed RXD input

Whenever a posted interrupt is selected, BIO will go low. BIO will stay high if the selected interrupt is not posted. When the SCAN input or the retimed RXD input is selected, BIO follows the polarity of the respective signal.

bit6: TXD, the serial data port transmit data bit. This signal is retimed by the rising edge of the TXC clock, and appears on the TXD output pin.

bit 7: DRIVE, a general purpose output pin.

Serial Codec Port

The Serial Codec Port consists of 8-bit Transmit and Receive data registers, designed to directly interface to Motorola 14400 series PCM Monochips.

The Transmit Register forms incoming serial data on TDD into 8-bit parallel PCM samples, while the Receive Register forms 8-bit parallel PCM data samples into serial data on RDD. The operation of these registers is controlled by the Codec Timing Generator, which also generates the CCLK, the RCTD, and the internal A/D interrupt signals.

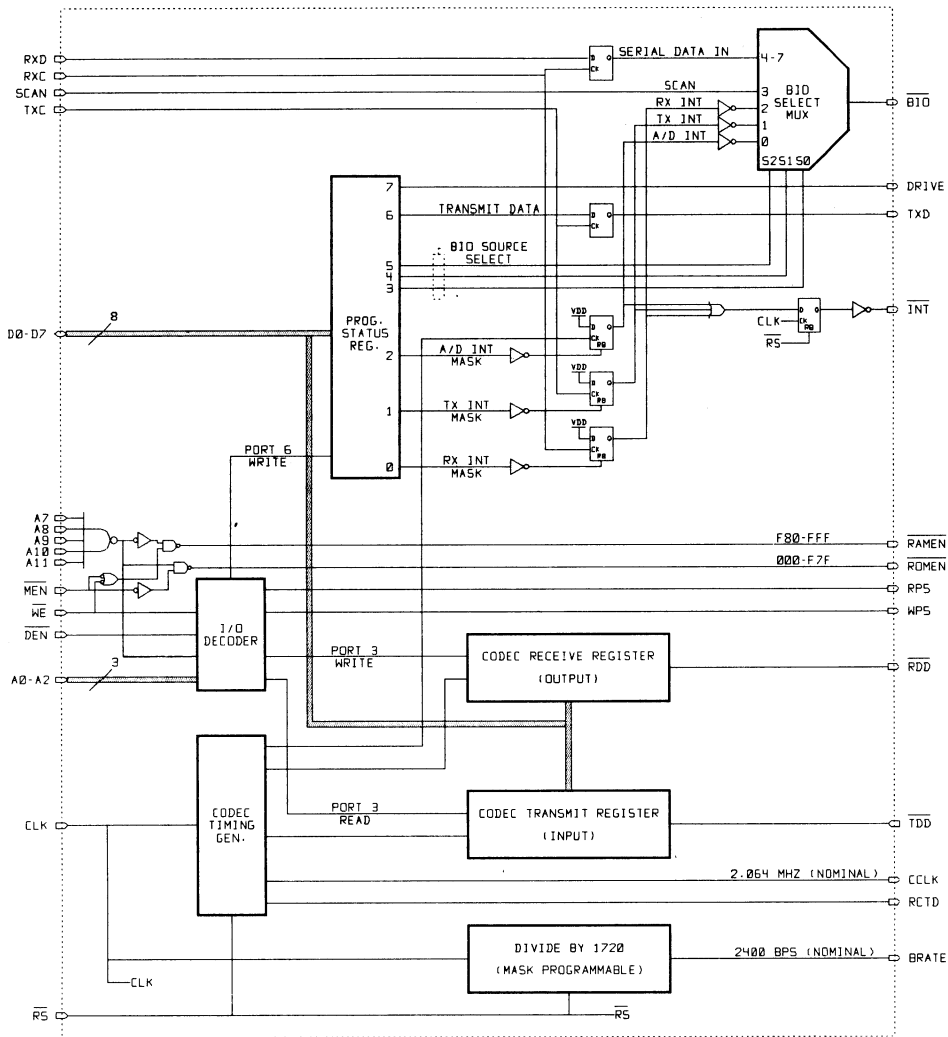
Data written to the Receive Register at I/O port location 3 is inverted, and sent MSB first on the RDD Pin. Data is shifted out on the 8 rising edges of CCLK following the rising edge of RCTD. To prevent Receive Register underflow, data must be available in the Receive Register within 248 CCLK cycles after an A/D interrupt (nominally 120 usec).

Serial PCM data on the TDD pin is inverted, and read into the Transmit Register, MSB first, at I/O port location 3. Data is shifted in on the 8 falling edges of CCLK following the rising edge of RCTD. To prevent Transmit Register overflow, data must be read within 248 CCLK cycles after an A/D interrupt (nominally 120 usec).

Bit Rate Generator

The BRATE output signal is nominally a 2400 Hz square wave, derived from the CLK input divided by 1720 (mask programmable). This signal may be used for bit rate generation, TXC or RXC clocking, or real-time interrupts. BRATE is reset whenever the RS signal is low.

PD32HC01 DETAILED BLOCK DIAGRAM (FIG. 3)



Address and Input/Output Map¹

Address/ Port ²	R/W ²	Function
>XX3 Port 3	Read ³	Codec Transmit Register. Valid for 248 CCLK cycles (nominally 120 usec) after an A/D interrupt. This 8-bit register contains the inverted version of the digitized serial PCM signal appearing on <u>TDD</u> . This register can only be read by using a IN from port 3 instruction.
>XX3/ Port 3	Write	Codec Receive Register. Must be valid within 248 CCLK cycles (nominally 120 usec) after an A/D interrupt. Data written to this 8-bit register will be inverted and shifted out on the <u>RDD</u> pin. This register can be written using a OUT to port 3 instruction, or by using a TBLW to address >XX3 (address must be less than >F80).
>XX5/ Port 5	Read ³	Input Port Expansion. The <u>RP5</u> signal will pulse high whenever an IN from port 5 instruction is executed.
>XX5/ Port 5	Write	Output Port Expansion. The <u>WP5</u> signal will pulse high whenever an OUT to port 5, or TBLW to >XX5 instruction is executed (address must be less than >F80).
>XX6/ Port 6	Write	Program Status Register. This register is used to: Select I/O and interrupt status bits onto the <u>BIO</u> pin; mask and reset interrupts; control the <u>DRIVE</u> pin; and to send data on <u>TXD</u> . This register can be written using an OUT to port 6 instruction, or by using a TBLW to address >XX6 (address must be less than >F80).
>000– >77F	Read	Program ROM space. Notice that the I/O space and the ROM space are mapped to overlapping addresses, but are distinguished by the <u>FIEN</u> and <u>DEN</u> signals. <u>FIEN</u> will go low for valid instruction reads, while <u>DEN</u> will go low for I/O reads.
>780– >7FF	R/W	Data/Program RAM space. For read cycles, <u>FIEN</u> goes low; for write cycles <u>WE</u> goes low.

- Notes: 1. When using TBLW to perform Output, many aliases of the I/O port locations exist due to incomplete address decoding. To maintain compatibility with future products, it is recommended that addresses >000 to >007 be used. When using OUT instructions, the TMS32010 always addresses >000 to >007, and the aliases are irrelevant.
2. Some I/O addresses are not used. To prevent data corruption, port locations 0, 1, 2, and 4 should not be written with TBLW or OUT.
3. The I/O read locations can only be accessed with a TMS32010 IN instruction. The actual address locations are shown, however, for applications using other than the TMS32010 processor.

Recommended operating conditions • $V_{SS} = 0$ Volts

Parameter	Symbol	Min	Typ	Max	Units	Notes
Supply voltage	V_{DD}	4.75	5.0	5.25	V	
Input high voltage	V_{IH}	2		$V_{DD} + 0.3$	V	
Input low voltage	V_{IL}	-0.3		0.8	V	
Output high current	I_{OH}			3	mA	
Output low current	I_{OL}			3	mA	
Operating Temperature	T_A	0		70	°C	

Electrical characteristics over recommended operating conditions ¹

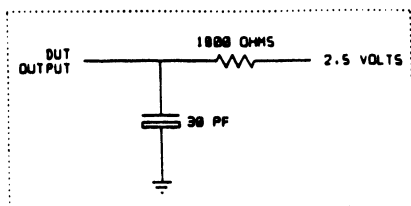
Parameter	Symbol	Min	Typ ²	Max	Units	Test Condition
High level output voltage	V_{OH}	3.5	4.5		V	$I_{OH} = 1$ ma
Low level output voltage	V_{OL}		0.3	0.5	V	$I_{OL} = 2$ ma
RS hysteresis voltage	V_{HYS}		200		mV	
Off-state leakage current	I_{OZ}		0.5	5	uA	
Input current	I_{IN}		0.5	5	uA	$V_{SS} < V_{IN} < V_{DD}$
Supply current ³	I_{DD}		2		mA	
Input capacitance	C_I		10		pF	• 1 MHz;
Output capacitance	C_O		10		pF	all other pins 0 V
CLK input frequency	F_{CLK}	0	4.128	6.0	MHz	

Note: 1. See Fig. 4 for DUT test loads.

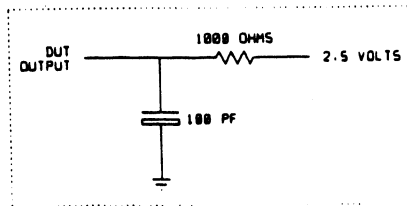
2. Typical specifications are valid at $T_A = 25$ °C, $V_{DD} = 5.0$ Volts.

3. I_{DD} is a function of V_{DD} , clock frequency, and output loading.

TEST LOADS (FIG. 4)



ALL OUTPUTS EXCEPT D0-D7



D0-D7

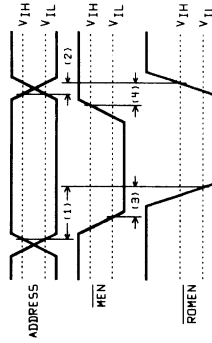
Timing Specifications over recommended operating conditions ¹

	Name	Description	Min	Typ ²	Max	Units
R	(1) t_{ROML1}	\overline{ROMEN} select time from addr.		28		nsec
O	(2) t_{ROMLH1}	\overline{ROMEN} deselect time from addr.		30		nsec
M	(3) t_{ROML2}	\overline{ROMEN} select time from \overline{MEN}		25		nsec
	(4) t_{ROMLH2}	\overline{ROMEN} deselect time from \overline{MEN}		18		nsec
	(5) t_{RAML1}	\overline{RAMEN} select time from addr.		27		nsec
	(6) t_{RAMLH1}	\overline{RAMEN} deselect time from addr.		15		nsec
R	(7) t_{RAML2}	\overline{RAMEN} select time from \overline{MEN}		24		nsec
A	(8) t_{RAMLH2}	\overline{RAMEN} deselect time from \overline{MEN}		14		nsec
M	(9) t_{RAML3}	\overline{RAMEN} select time from addr.		27		nsec
	(10) t_{RAMLH3}	\overline{RAMEN} deselect time from addr.		15		nsec
	(11) t_{RAML4}	\overline{RAMEN} select time from \overline{WE}		21		nsec
	(12) t_{RAMLH4}	\overline{RAMEN} deselect time from \overline{WE}		15		nsec
	(13) t_{DR1}	Data read access time from addr.		33		nsec
R	(14) t_{DRHLD1}	Data read hold time from addr.		60		nsec
E	(15) t_{DR2}	Data read access time from \overline{DEN}		29		nsec
G	(16) t_{DRHLD2}	Data read hold time from \overline{DEN}		56		nsec
I	(17) t_{RPLH1}	RPS select time from address		30		nsec
S	(18) t_{RPHL1}	RPS deselect time from address		32		nsec
T	(19) t_{RPLH2}	RPS select time from \overline{DEN}		26		nsec
E	(20) t_{RPHL2}	RPS deselect time from \overline{DEN}		26		nsec
R	(21) t_{ASUW}	Address set-up time to \overline{WE}		4		nsec
	(22) t_{AHLDW}	Address hold time from \overline{WE}		-5		nsec
&	(23) t_{DWSU}	Data write set-up time to \overline{WE}		-25		nsec
	(24) t_{DWHLD}	Data write hold time from \overline{WE}		0		nsec
I	(25) t_{WPLH1}	WPS select time from address		31		nsec
/	(26) t_{WPHL1}	WPS deselect time from address		32		nsec
O	(27) t_{WPLH2}	WPS select time from \overline{WE}		27		nsec
	(28) t_{WPHL2}	WPS deselect time from \overline{WE}		27		nsec
S	(29) t_{RXDSU}	RXD set-up time to RXC	20	3		nsec
I	(30) t_{RXDHLD}	RXD hold time from RXC	20	2		nsec
O	(31) t_{TXD}	TXD delay time from TXC		28		nsec
C	(32) t_{RCTD}	RCTD Delay time from CCLK	15	34	100	nsec
O	(33) t_{TDDSU}	TDD set-up time to CCLK	50	16		nsec
D	(34) t_{TDDHLD}	TDD hold time from CCLK	20	-14		nsec
E	(35) t_{RDD}	RDD delay time from CCLK		19	50	nsec
C	(36) t_{ADINT}	INT delay time from RCTD			Note 3	

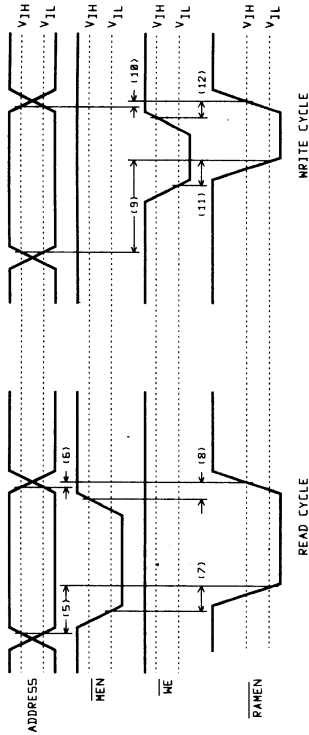
Note: 1. See Fig. 4 for DUT test loads.

2. Typical specifications are valid at $T_A = 25^\circ\text{C}$, $V_{DD} = 5.0\text{ Volts}$ 3. $t_{ADINT}(\text{max})$ is $17 \times t_{CLK} - 45\text{ nsec}$ (nominally 4.1 μsec).

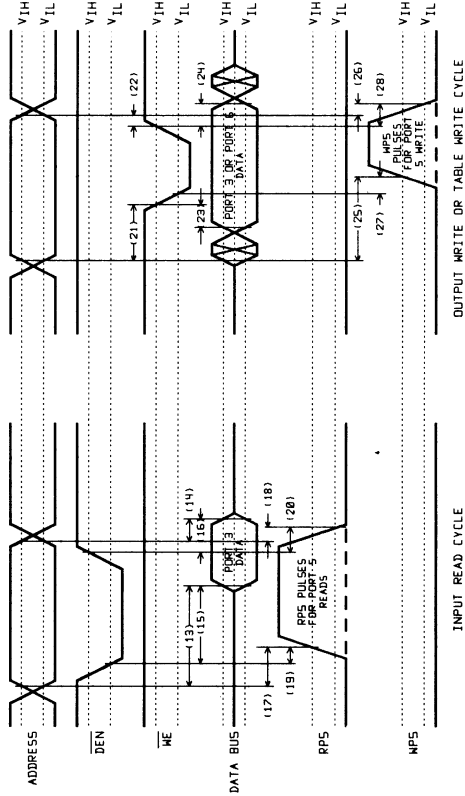
ROM READ TIMING (DEN AND WE HIGH)



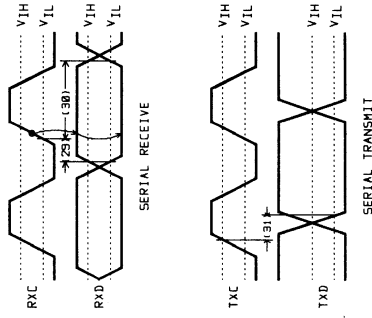
RAM READ/WRITE TIMING (DEN HIGH)



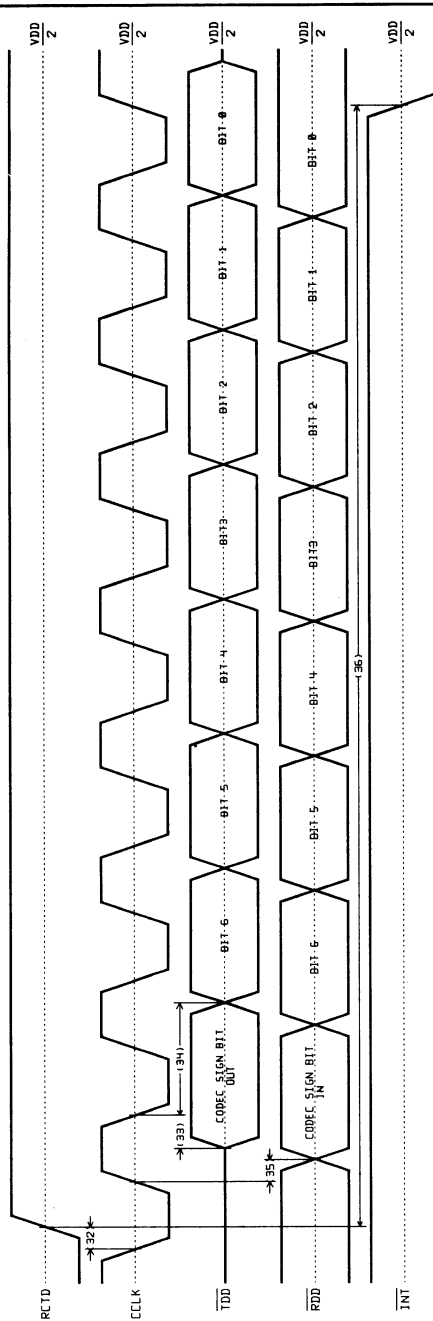
REGISTER AND I/O EXPANSION READ/WRITE TIMING (DEN HIGH)



SERIAL INTERFACE TIMING



CODEC TIMING (CMOS LEVELS)



To handle interrupts on the PD32HC01, the techniques illustrated below may be used. Not all applications may require the full implementation. Details on TMS32010 interrupts may be obtained from the publication "TMS32010 User's Guide", © 1983, Texas Instruments Ltd., Revision B, March 1985. For Mu-law to linear and linear to Mu-law conversion routines, see "Companding Routines for the TMS32010 -- Digital Signal Processing Application Report", © 1984, Texas Instruments Ltd.

Program Status Register

Set up an 8-bit Program Status Register (PSR) image in data RAM with the following definitions:

Bit	Name	Description
b7	DRIVE	General purpose output bit
b6	TXD	Serial data output
b5	\	3 bit field for selecting BIO input.
b4	---	0 = A/D interrupt status, 1 = TX interrupt status,
b3	/	2 = RX interrupt status, 3 = SCAN input, 4 = RXD input.
b2	\	Interrupt mask/acknowledge bits (active high). Used to clear posted interrupts.
b1	---	b2 = A/D interrupt mask, b1 = TX interrupt mask, b0 = RX interrupt mask.
b0	/	

The status of the PD32HC01 is defined by writing the contents of the PSR to port 6 with an OUT or a TBLW instruction. In addition to defining the state of the DRIVE and TXD pins, the PSR is used to respond to and acknowledge interrupts from the RX clock, TX clock, and a Codec A/D conversion. The following is an example of TMS32010 software which implements interrupt vectoring with the A/D interrupt at highest priority, and TX and RX interrupts at lower priority.

Interrupt Handling Program

```

*
* CONTEXT SAVE FOR INTERRUPT (INTRP stack and PSR on data page 1)
*
INT      SST      INTRP      Save_machine_state:
          SACH     INTRP+1,0    Save_Status;
          SACL     INTRP+2      Save_ACH;
          MPYK     1           Save_ACL;
          PAC      Save_T;
          SACL     INTRP+3      End.
*
* VECTORED INTERRUPT ARBITRATION (BIO Mux always initially points to A/D interrupt flag)
*
*                                     Vector:
          BIOZ     ADDA         IF A/D_interrupt THEN
*                                     AD_interrupt_service
*                                     END;
          LAC      PSR,0
          ADD      ONE,3        IF TX_interrupt THEN
          SACL     PSR          TX_interrupt_service
          OUT      PSR,PA6      END;
          BIOZ     TXDAT

```

```

*      ADD     ONE,3          IF RX_interrupt THEN
      SACL     PSR            RX_interrupt_service
      OUT      PSR,PA6        END;
      BIOZ     RXDAT

*
* INTERRUPT ERROR RECOVERY PROCEDURE (Should never need execution)
*
      LACK     >C0            IF No_interrupt THEN
      AND      PSR            Initialize_PSR;
      SACL     PSR            Restore_machine_state;
      OUT      PSR,PA6        Return_from_interrupt;
      ZALH     INTRP+1        END;
      ADDS     INTRP+2        END.
      LT       INTRP+3
      LST      INTRP
      EINT
      RET

*
* INTERRUPT SERVICE ROUTINES (A/D, TX DATA, RX DATA)
*
* ADDA: Clear A/D Interrupt, process, restore machine state, and return.
* TXDAT: Disable TX Interrupt, process, restore machine state (and enable TX interrupt), and return.
* RXDAT: Disable RX Interrupt, process, restore machine state (and enable RX interrupt), and return.
*
*      AD_interrupt_service:
ADDA  LAC      ONE,2          Clear_AD_interrupt; { highest priority interrupts }
      OR       PSR            { should have fast clear to allow }
      SACL     PSR            { reposting. }
      OUT      PSR,PA6
      SUB      ONE,2
      SACL     PSR
      OUT      PSR,PA6
      .
      .                      (Body of interrupt service routine)
      .
      ZALH     INTRP+1        Restore_machine_State;
      ADDS     INTRP+2        Return_from_interrupt;
      LT       INTRP+3        END.
      LST      INTRP
      EINT
      RET

*      TX_interrupt_service:
TXDAT LACK     >O2          Disable_TX_interrupt;
      OR       PSR            { lower priority interrupts can }
      SACL     PSR            { be left disabled until interrupt }
      OUT      PSR,PA6        { servicing is complete. }
      .
      .                      (Body of interrupt service routine)
      .
      LACK     >C0
      AND      PSR            Initialize_PSR;
      SACL     PSR            Restore_machine_state;
      OUT      PSR,PA6        Return_from_interrupt;
      ZALH     INTRP+1        END.
      ADDS     INTRP+2
      LT       INTRP+3
      LST      INTRP
      EINT
      RET

```

```

"
RXDAT LACK >21      RX_interrupt_service:
      OR   PSR      Disable_RX_interrupt_select_RXD;
      SACL PSR
      OUT  PSR,PA6
      BIOZ xxx

      (Body of interrupt service routine)

      LACK >C0
      AND  PSR      Initialize_PSR;
      SACL PSR      Restore_machine_state;
      OUT  PSR,PA6   Return_from_interrupt;
      ZALH INTRP+1   END.
      ADDS INTRP+2
      LT   INTRP+3
      LST  INTRP
      EINT
      RET

```

A/D Interrupt Handling

When responding to an A/D interrupt, the following TMS32010 code is appropriate:

```

      OUT  SMPL_PA3   (output a sample)
      IN   SMPL_PA3   (input a sample)

      (Remainder of Interrupt Service Routine)

```

Note that if the software is written with a one sample "look-ahead", almost 124 usec (one sample period) can elapse between the interrupt request and execution of the A/D interrupt service routine without loss of data. The above code should be executed early in the interrupt service routine to take maximum advantage of the hardware architecture. This feature is important when several foreground software routines are implemented, or when the A/D interrupt service routine execution time can occasionally exceed 124 usec.

Note: The information contained in this document is for illustrative purposes only. No guarantee as to its suitability for end-use applications is implied.



Interface Chip Simplifies TMS32010 Based Voice-band Processing

Vernon R. Little, Pacific Microcircuits Ltd., White Rock, B.C., Canada

The TMS32010 is well suited for voice band signal processing. Its powerful instruction set and simple architecture makes cost-effective digital signal processing a reality for modems, vocoders, smart telephones, and a host of other voice and data communications functions.

Pacific Microcircuits Ltd. of White Rock, B.C., Canada has introduced a high-speed, low power CMOS interface chip, the PD32HC01, specifically tailored to interface the TMS32010 to an external Codec, and RAM and ROM memory. With the addition of a few external components, a complete general purpose audio processor can be constructed without any logic 'glue'.

The PD32HC01, besides having high-speed memory decoding and a serial Codec interface, has SCAN and DRIVE input and output pins, a serial data communications port for USRT emulation, a mask-programmable 2400 Hz bit-rate generator or real time clock, multiple source interrupt control, and decoding for expanding I/O off chip. It is available in both 40 pin plastic and ceramic DIP and 44 pin surface-mount plastic packages.

General Purpose Audio Processor

The utility of the PD32HC01 is illustrated in the schematic diagram entitled "GENERAL PURPOSE AUDIO PROCESSOR". This is a complete DSP function capable of implementing a fully functional LPC vocoder, voice band modem, intelligent telephone, or a telecom test set.

In the schematic, the PD32HC01 (U8) ties all of the system resources to the TMS32010 (U7). ROM (U5; U6) is segmented as a 3968 word address space from address >000 to >7FF. An optional RAM (U1-U4) is segmented as a 128 word address space from >F80 to >FFF. This decoding can be arbitrarily modified to increase the RAM space at the expense of ROM (see detail on schematic diagram). This allows 'soft' programming by downloading program instructions into RAM by using a bootstrap loader in ROM. Downloading can occur either on the serial RS-232 interface, or, with additional hardware, the WP5 and RP5 signals on the PD32HC01 can be used to coordinate parallel data transfers with a host processor.

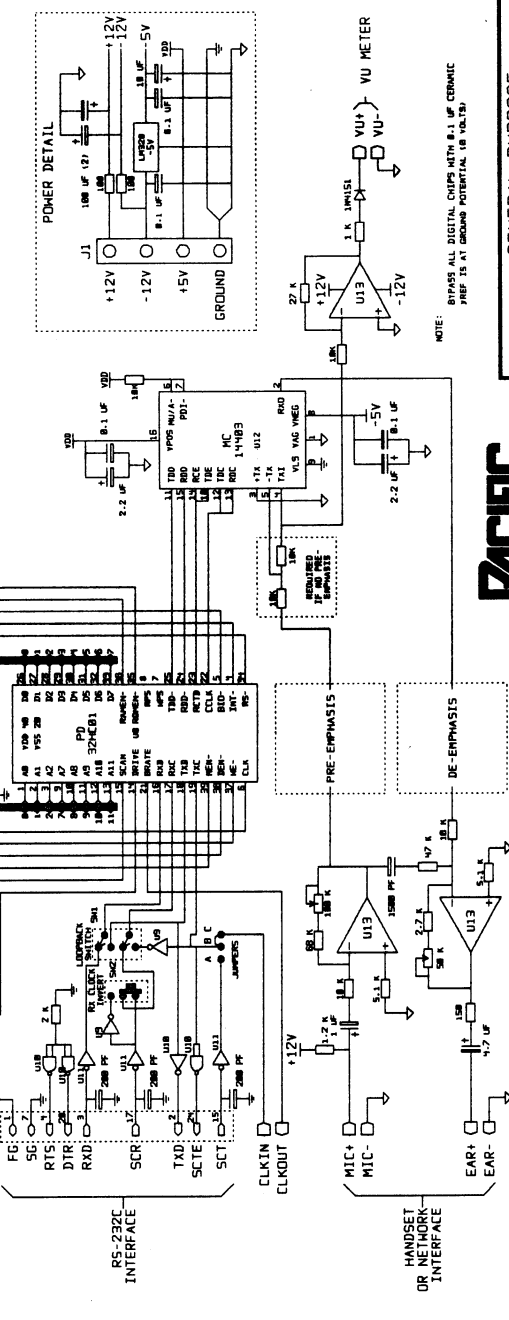
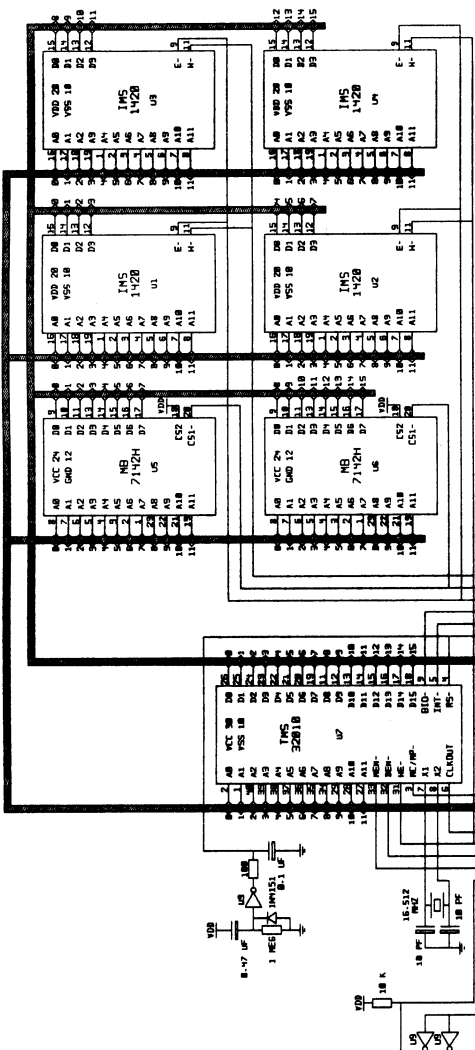
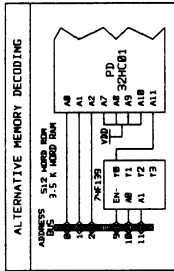
The 8-bit serial Codec interface connects directly to a μ -law or A-law PCM Codec (U12), and is accessible through parallel registers on the PD32HC01. Analog circuitry (U13 etc.), including pre-emphasis and de-emphasis filters, can be connected to the Codec as required. Also shown is a VU-meter circuit for level setting of incoming audio.

Provision for a synchronous RS-232 port is provided. Using the on-chip interrupt control and I/O logic, the TMS32010 can emulate a USRT at 2400 bps. This is very useful for modem and vocoder applications, and it may also be used for program downloading into RAM. With minor modifications in hardware, and appropriate software, asynchronous RS-232 communications can also be emulated at speeds up to 1200 bps. EIA drivers and receivers (U10, U11) are required to interface the RS-232 signals to the PD32HC01.

The SCAN and DRIVE pins are used for miscellaneous control functions. In the example given, SCAN is used to sense the off hook status of the handset, and DRIVE is buffered with a 74HC04 (U9), and used to activate a buzzer or lamp to indicate device operation or error states.

For more information on the PD32HC01 signal processor interface, and other members of the product family, please contact the Product Development group at Pacific Microcircuits Ltd.

REFERENCE	TYPE
U1-U4	IMS1428 4K x 4 RAM
U5-U6	MS7142 4K x 8 ROM
U7	74HC01 6V INVERTER
U8	74HC01 6V INVERTER
U9	74HC01 6V INVERTER
U10	MC1488 EIA DRIVER
U11	MC1493 COMPARATOR
U12	MC1493 COMPARATOR
U13	UAA135 QUAD OPAMP



NOTE:
BYPASS ALL DIGITAL CHIPS WITH 0.1 μ F CERAMIC
CAPS TO AT GROUND POTENTIAL FOR VLSI.



GENERAL PURPOSE
AUDIO PROCESSOR

AUDIO PRE-EMPHASIS & DE-EMPHASIS

PACIFIC MICROCIRCUITS LTD.

1645 140th Street
White Rock, B.C.
Canada, V4A 4H1
(604) 536-1886

North American Representatives

Astec Components Limited (416) 669-4022
101 Citation Drive, Unit #7
Concord, Ontario L4K 2S4

Carlisle Technical Sales, Inc. (617) 890-8800
391 Totten Pond Road
Waltham, Mass. 02154

Electro Sales Associates (513) 426-5551
1635 Mardon Drive
Dayton, Ohio 54532

Electro Sales Associates (216) 729-0190
Diplomat Building 729-0191
12575 Chillicothe Rd., Ste. #8
Chesterland Ohio, 44026

Electro Sales Associates (313) 474-7320
29200 Vassar Road, Suite 505
Livonia, Michigan 48152

Electro Sales Associates (412) 487-3801
3740 Mount Royal Blvd.
Allison Park, Pa. 15101

Electro Sales Associates (616) 323-2416
9816 Portage Road
Portage, Michigan 49002

L-Tec, Inc. (312) 773-2900
810 Arlington Heights Rd.
Itasca, Illinois 60143

/Mesa Technical Associates (609) 429-9531
P.O. Box 466
20 Kings Highway West
Haddonfield, N.J. 08033
