Military Standard Products

UT69161 ASCENT™ Dual-Rate 1773A Protocol Device

Product Brief

UNITED TECHNOLOGIES MICROELECTRONICS CENTER

February 1995

FEATURES

- ☐ *A*dvanced *S*peed and *C*omputing on an *E*nhanced *N*etworking *T*erminal (ASCENT)
- ☐ Comprehensive MIL-STD-1773A dual redundant Bus Controller (BC) and Remote Terminal (RT)
  - Supports both 1 and 20Mbps operation
  - Programmable message preamble
  - Supports MIL-STD-1553B and MIL-STD-1773A
- ☐ Flexible bus controller operation
  - Powerful 16 opcode instruction set
  - User-selectable auto-start capability
- ☐ Double buffered remote terminal architecture
  - Simple mailbox architecture with message logging
- ☐ 8- or 16-bit host interface
  - Multiplexed and non-multiplexed address and data
  - Dual-port memory interface
  - Requires no arbitration logic
  - User-defined control signals
- ☐ Programmable message interrupts
  -Pulsed or level outputs
- ☐ Single 5 volt supply
- ☐ QML Q and V compliant
  - Radiation-hardened option available
    - Total dose: 1.0E6 rads(Si)
    - SEL immune
- ☐ Flexible packaging offering:
  - 144-pin pingrid array (PGA)
  - 132-lead flatpack

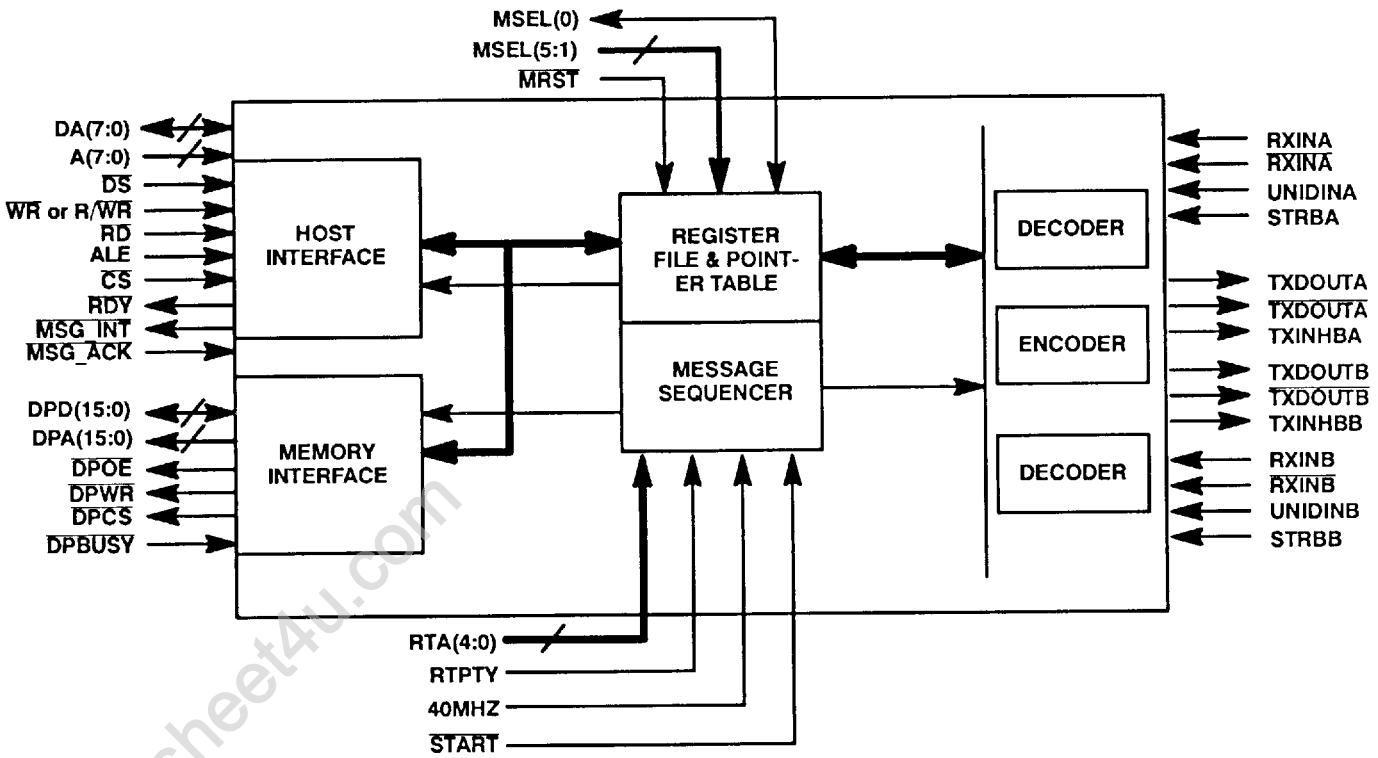


Figure 1. UT69161 ASCENT Block Diagram

www.datasheet4u.com

www.DataSheet4U.com

ASCENT is a trademark of United Technologies Microelectronics Center, Inc.

1.0 INTRODUCTION

The monolithic CMOS ASCENT provides the system designer with an intelligent solution to MIL-STD-1773A multiplexed serial data bus design problems. The ASCENT is a single-chip device that implements both RT and BC functions. Operating either autonomously or with a tightly coupled host, the ASCENT will solve a wide range of MIL-STD-1773A interface problems. Powerful state machines provide automatic message handling, message status, general status, and interrupt information. The register based interface architecture provides many programmable functions as well as extensive information pertinent to device maintenance. In either operating mode, the ASCENT can access up to 64K x 16 of external memory (65,536 x 16).

The ASCENT is a powerful asset to a system designer solving the MIL-STD-1773A interface problem.

2.0 REMOTE TERMINAL FEATURE

The ASCENT RT conforms to the requirements of MIL-STD-1773A. In addition to meeting the standard's requirements the RT has an extensive list of flexible features to meet MIL-STD-1773A interface requirements.

2.1 Subaddress Data Mapping

The ASCENT locates subaddress data pointers from an internal look-up table by decoding the command word's T/$\overline{R}$ bit and subaddress field. The decode selects one of thirty-two possible combinations (16 receive, 16 transmit). The subaddress look-up table resides internally in the ASCENT at address location 0000 (hex) to 007F (hex) . In the eight bit mode of operation, the subaddress look-up table resides at locations 0000 (hex) to 00FF (hex).

The host processor loads two data pointers per subaddress in the look-up table prior to starting RT operation. The ASCENT determines the active subaddress buffer by decoding and reading the subaddress' Pointer A at the beginning of message processing. If Pointer A's LSB is 0 then Pointer A is active. If Pointer A's LSB is 1 then Pointer B is active. The ASCENT updates Pointer A's LSB at the end of message processing.

2.2 Buffer Ping-Pong

To support the transfer of periodic data, double buffering schemes are often incorporated into RT designs. Periodic data transfer incorporates the use of two data buffers per subaddress. The RT processes messages (receive or transmit) via the designated primary buffer. The host or subsystem uses the alternate buffer to collect new data for transmission or to process data received during a defined time interval. When the defined interval ends, the RT will switch the primary and alternate data buffers (i.e., ping-pong).

2.3 Illegal and Broadcast Commands

The ASCENT treats all subaddress and defined mode code as legal commands. Considered illegal, the ASCENT responds with only a status word to undefined and reserved mode code commands. In response to illegal commands, the status word has the message error bit set to a logical one. Upon reception of a broadcast command, the ASCENT suppresses status word transmission and executes the command. The ASCENT treats transmit broadcast commands as undefined by asserting the message error bit to a logical one. The host enables and disables the broadcast option via the Control Register. The ASCENT suppresses status word transmission for all broadcast commands.

2.4 Mode Code and Busy Operation

The ASCENT supports all mode codes defined in MIL-STD-1773A. The BC designates mode code commands by assigning a mode field of 00000 or 11111. Except for Selected Transmitter Shutdown, Override Selected Transmitter Shutdown, Reset RT, Initiate Self-Test, and Dynamic Bus Control the ASCENT automatically executes all mode code commands. The synchronize without data mode code commands results in the Time Tag Register being reset to a logical zero. The synchronize with data mode code commands results in the Time Tag Register being set to the contents of the data word. When busy, the ASCENT continues to execute all mode code commands. For non-mode code commands, the ASCENT responds with only a status word when busy; the status word's busy bit is a logical one.

2.5 Message Information

The ASCENT stores message information words into a Status Buffer in conjunction with each command. The Status Buffer resides in external memory. The data written with a receive message includes the following: command word, time-tag value, and data pointer. For transmit commands only a data pointer is stored. The ASCENT stores additional message information in the Interrupt Completion Code Register and Completion Code Register. The ASCENT uses a "mailbox" storage architecture to implement the Status Buffer.

The Status Buffer resides in external memory at location 0000 (hex) to 007F (hex). The Status Buffer divides into four blocks of 32, three for receive and one for transmit. The system reserves location 0000 (hex) to 001F (hex) for receive command words, locations 0020 (hex) to 003F (hex) for receive subaddress data pointers, and 0040 (hex) to 005F (hex) for message time tag words. The final 32 locations, 0060 (hex) to 007F (hex), are reserved for transmit pointers. The command word's subaddress field decodes to selects one of 32 possible locations in each block. The command word's T/$\overline{R}$ selects between transmit and receive mail boxes. The ASCENT automatically inserts a 32 address off-set when storing receive data pointers into the Status Buffer and 64 address off-set when storing receive time tag words into the Status Buffer.

9343947 0004948 223

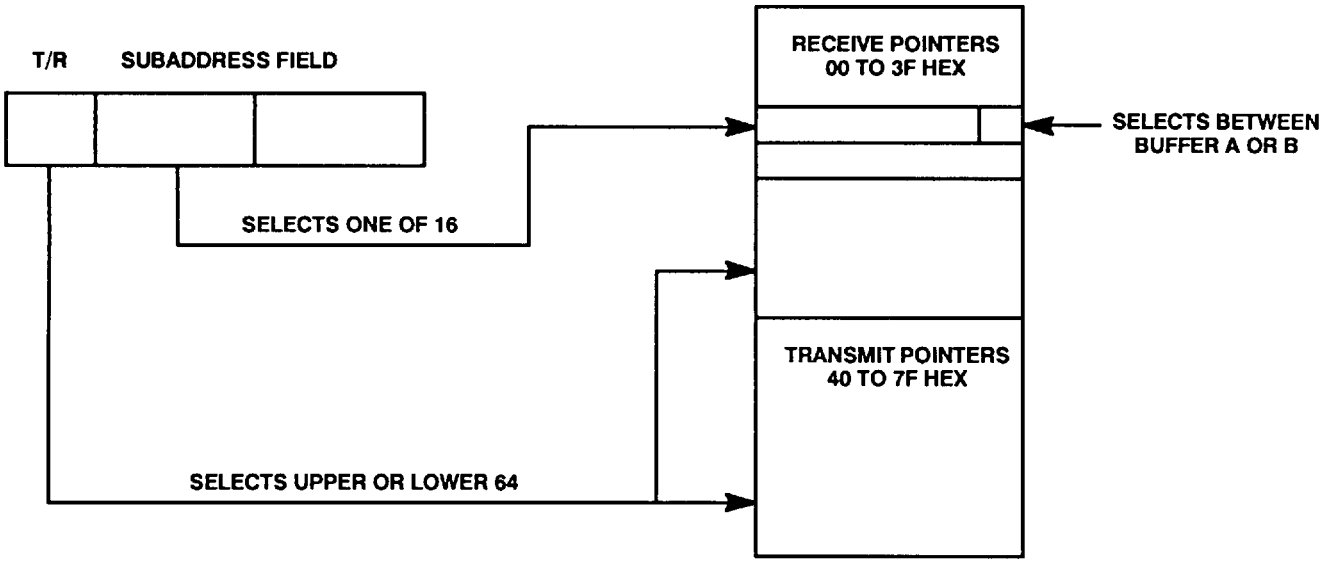


Figure 2. Subaddress Data Mapping

3.0 BUS CONTROLLER FEATURES

The ASCENT BC is a powerful dual-speed BC developed to meet the requirements of multi-frame processing with low host overhead. User-defined decision making and message flow instructions allows the BC to operate autonomously from the host until a designated event or series of events takes place. The BC architecture includes the following features:

- Auto-Repeat
- Auto-Start
- Auto-Stop
- Graceful Halt
- Stop On Condition
- Message Retry
- Programmable Response Time-Out
- Command determination of 1 or 20Mbps operation

A simple data structure of three words comprises a BC instruction. Instructions control three major functions: BC communication, instruction sequencing, and serial bus operations. For serial bus control operations a single bit determines if the command bandwidth is 1Mbps or 20Mbps. A pin programmable auto-start feature allows the BC to commence operation after the application of power and a reset pulse. The host can terminate serial bus operation gracefully by writing to the Control Register. The graceful-halt feature finishes a current command and then halts. If the serial bus is inactive, the BC immediately halts. For long bus applications, the ASCENT allows for programmable no response time-out values. At 1Mbps the no response time-out value varies from 14μs to 408μs, during 20Mbps operation the time-out value varies from 7μs to 204μs.

3.1 Bus Controller Instruction Set

The ASCENT's BC instruction set divides into three types: controller communication, instruction sequencing, and serial bus operation. Communication instructions allow the BC and host to communicate via the external memory. A major benefit of this architecture is that the host communicates and monitors the BC without interrupting the BC.

To control, alter, or monitor BC operation execute instruction sequencing commands. Instruction sequencing commands allow the host to insert delays, modify registers, branch on status word responses, branch on register contents and halt operation.

Serial bus instructions initiate commands on the bus. The instruction consists of a control word, a command word or words, and a data pointer for the storage or retrieval of message data. A single bit in the control word specifies whether the command executes at 1Mbps or 20Mbps.

3.2 Opcode Definition

OPCODE	NAME	DESCRIPTION
0000	Write Flag.	BC writes a host specified data pattern (i.e., flag) into a memory location during execution of this instruction. Host monitors the flag location for a pattern match.
0001	Wait.	BC inserts a 16-bit delay into command instruction processing. The instruction's second word contains the delay value. Delay resolution is 100ns/bit.
0010	Load Register.	Instructs BC to load an internal register. Host stores data for the register write in external memory. The instruction's second word specifies the data address. The Load Register instruction loads the following registers: Control, Instruction Counter, Real Time Clock, Retry, and Mask.
0011	Write Completion Status.	BC saves command status information into external memory. Information consists of status word, Completion Code Register contents, and Time Tag Register contents.
0100	Completion Code Branch.	Branch is taken if the Completion Code Register matches a compare value. The instructions second word contains the branch address while the third word contains the compare value.
0101	Receive Status Branch.	The BC branches if the receive command's status word response matches a compare value. The instructions second word contains the branch address. The third word contains the compare value.
0110	Transmit Status Branch.	The BC branches if the transmit command's status word response matches an compare value. The instructions second word contains the branch address. The third word contains the compare value.
0111	BC to RT Command.	BC executes a BC to RT transfers on the serial bus.
1000	RT to BC Command.	BC executes a RT to BC transfers on the serial bus.
1001	RT to RT Command.	BC executes a RT to RT transfers on the serial bus.
1010	Mode Code Without Data Command.	BC executes a Mode Code Without Data Command on the serial bus.
1011	Receive Mode Code With Data Command.	BC executes a Receive Mode Code With Data Command on the serial bus.
1100	Transmit Mode Code With Data Command.	BC executes a Transmit Mode Code With Data Command on the serial bus.
1101	Real Time Clock Wait.	BC inserts a delay into command instruction processing. Instruction processing stops until the Real Time Clock/Time Tag Register matches a compare value. The instruction's second word contains the compare value.
1110	Branch on Flag.	Instructs BC to sample a flag residing in memory. Branch is taken when the flag is non-zero. The second word of the instruction contains the Flag's address. The third word contains a branch address.
1111	Halt.	BC stops instruction execution.

9343947 0004950 981

3.3 Automatic Retry

The BC will automatically retry a message if a status exception or error occurs. The host specifies number of retries in the Retry/Delay Register. The Completion Code Register contains information regarding retry activity and the last active bus. The BC alternates buses during retries scenario. The BC allows for a maximum of sixteen retries.

3.4 Message Scheduling

The BC allows the host to enter data controlling the time between messages. This feature is useful when the BC has to perform periodic message transactions to RTs. The Wait and Real Time Clock Wait instructions allow for efficient timing of minor frames. For message scheduling examples, refer to figures 3a, 3b, and 3c.

3.5 Polling

The host instructs the BC to interrogate the status word response of an RT to determine if any BC action is required. The BC can detect the assertion of status word bits and generate interrupts or branch to a new instruction string. Polling is useful if the application requires control of instruction flow as a function of RT response. Receive Status Branch and Transmit Status Branch instructions allow for the comparison of status word response to a compare value. When a match occurs the branch is taken. In addition, the Completion Code Register can also modify instruction flow via the Completion Code Branch Instruction.

4.0 SYSTEM CONFIGURATION

The ASCENT supplies hardware designers with a flexible interface to meet the needs of state-of-the-art MIL-STD-1773A interfaces. The ASCENT contains internal registers, interfaces to either 8-bit or 16-bit subsystems, supports multiplexed and non-multiplexed buses, and has user selectable control signals.

4.1 Pointer Map

The ASCENT contains a 128 x16 RT look-up table. The RT accesses the look-up table to locate subaddress data buffers. The host treats the lookup table as memory along with the internal registers. When the ASCENT is organized x8 the look-up table is 256 x 8. Table 1 shows the look-up table organization for either 8-bit or 16-bit operation.

4.2 Internal Registers

The ASCENT contains 11 internal registers that control and report on message activity and operation. The 11 registers are memory mapped into the subsystem memory. Tables 2a and 2b review the registers and identify the mode of operation they are applicable. The host reads or writes these registers using the timing diagrams shown in figures 4a - 4d.

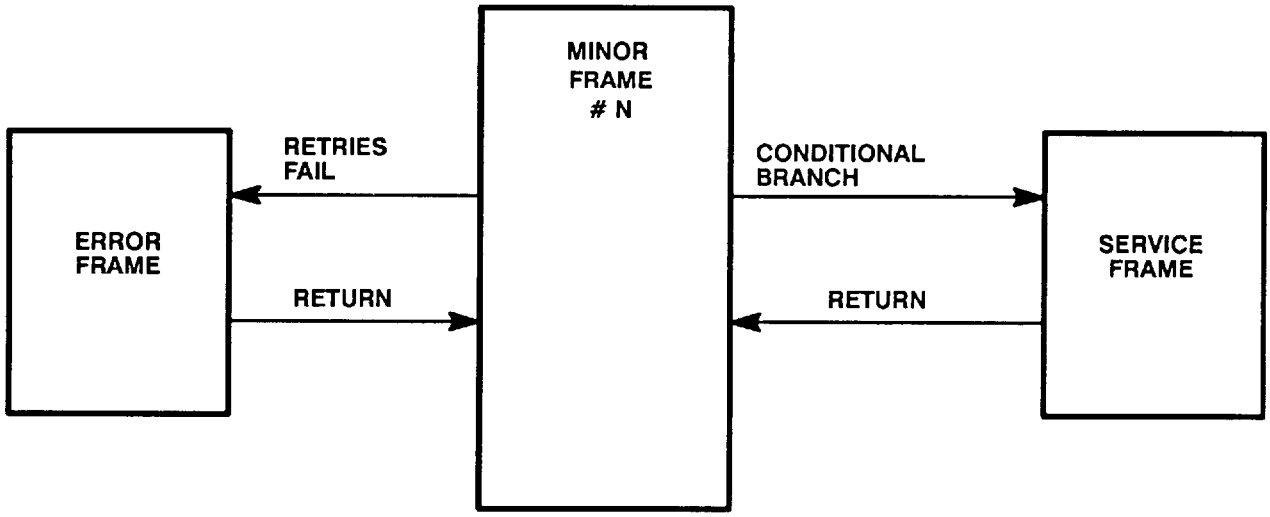


Figure 3a. Minor Frame Branching

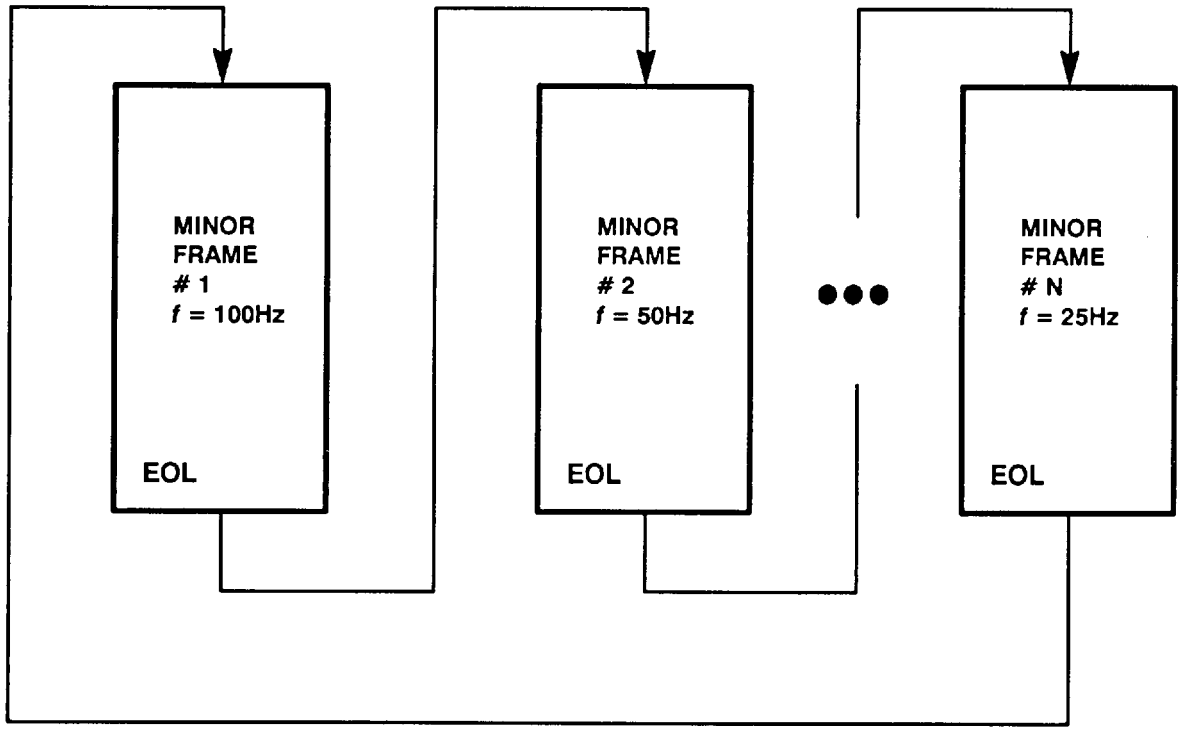


Figure 3b. Minor Frame Sequencing

9343947 0004952 754

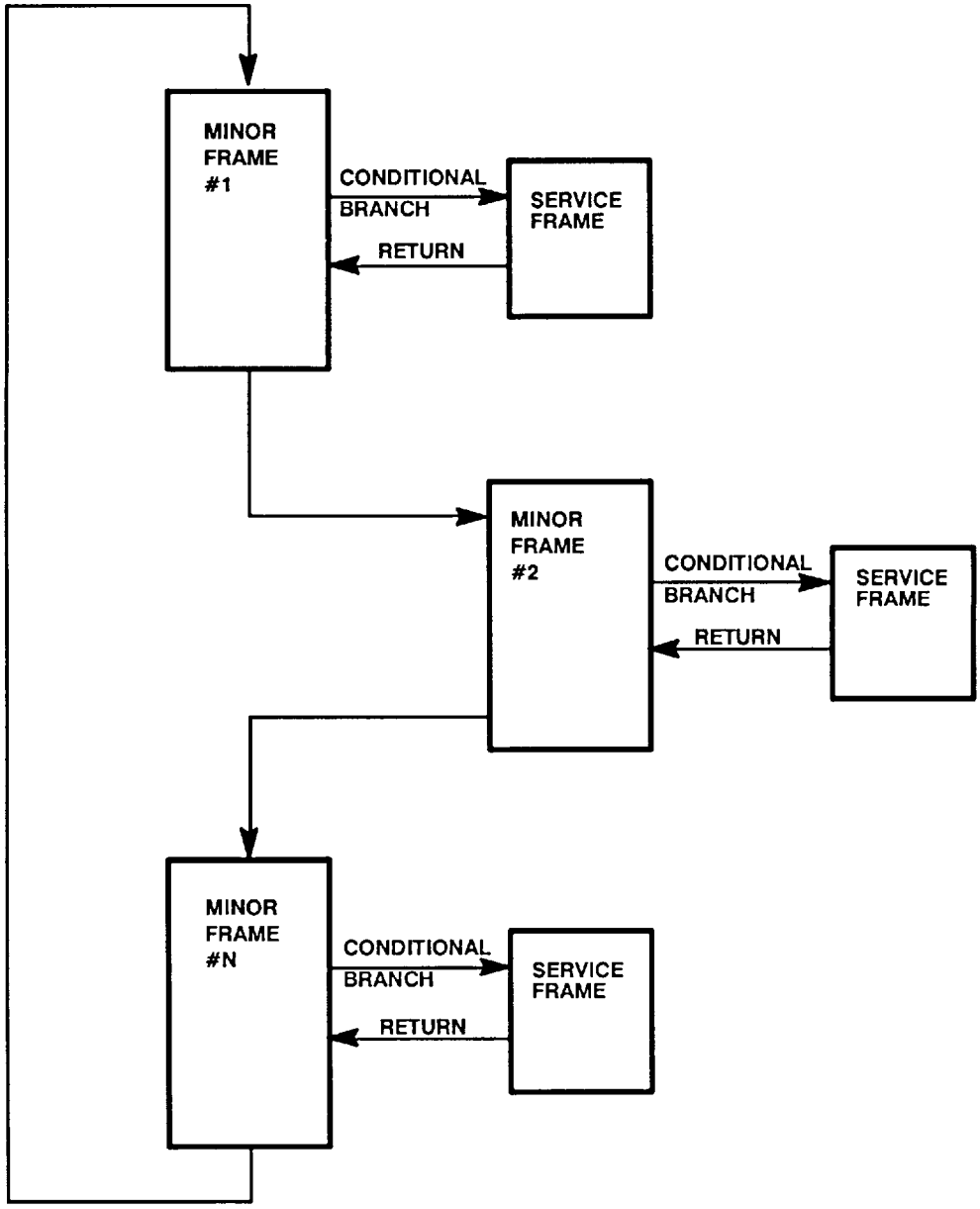


Figure 3c. Major Frame Sequencing

Table 1. Memory Organization

Mode	Dual-Port Memory Organization	Register Location	Pointer Range
8-bit	64K x 8	0100 (hex) to 0115 (hex)	0000 (hex) to 00FF (hex)
16-bit	64K x 16	0080 (hex) to 008A (hex)	0000 (hex) to 007F (hex)

Table 2a. Remote Terminal Register Utilization

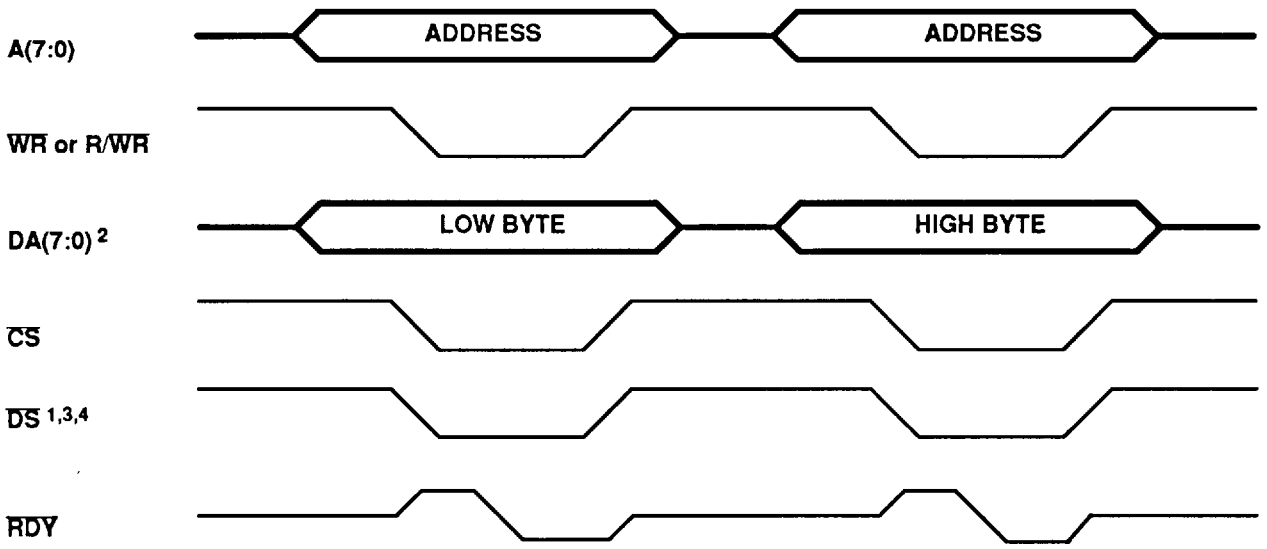
Register Name	Address (hex)	Host Access
Control Register	80	R/W
Vector Word Register	81	R/W
Time Tag Register	82	R/W
Status Register	83	R/W
Status Mask Register	84	R/W
Completion Code Register	85	R
Last Command Register	86	R
Interrupt Completion Code Register	87	R
Interrupt Last Command Register	88	R
Auxiliary Register	89	R

Table 2b. Bus Controller Register Utilization

Register Name	Address (hex)	Host Access
Control Register	80	R/W
Instruction Counter Register	81	R/W
Time Tag/Real Time Clock Register	82	R/W
Retry/Delay Register	83	R/W
Mask Register	84	R/W
Completion Code Register	85	R
Instruction Register 1	86	R
Instruction Register 2	87	R
Instruction Register 3	88	R
Receive Status	89	R
Transmit Status	8A	R

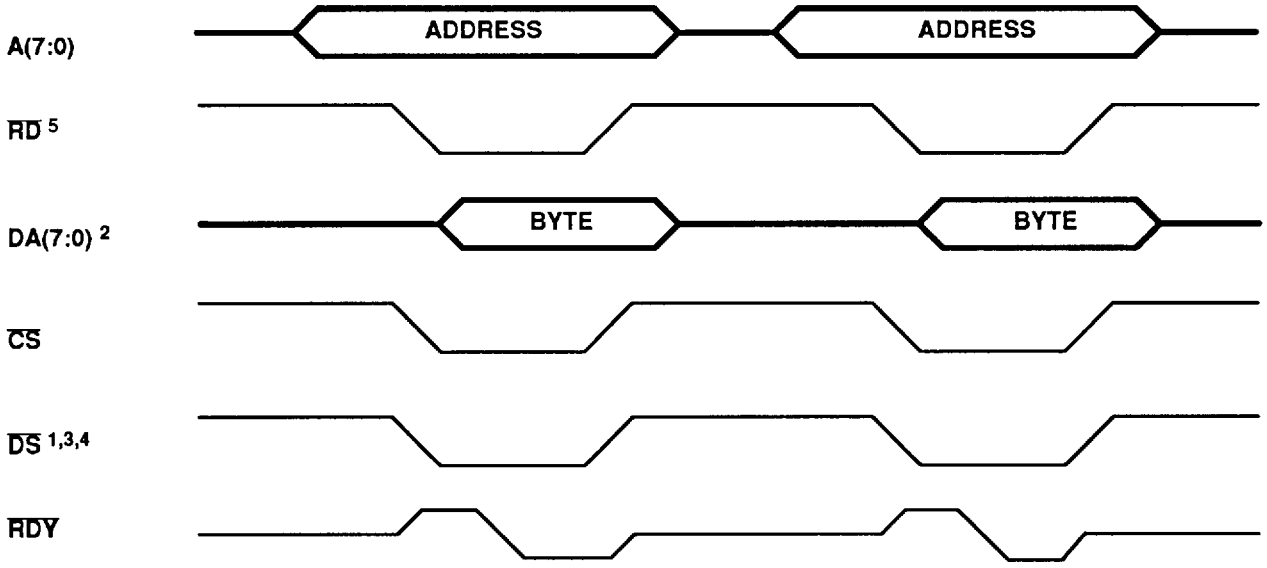
9343947 0004954 527

Non-Multiplexed Register Write Access (8-bit Mode)



Notes:
1. DS asserts to signal that data is valid on the bus.
2. Tie DA(15:8) to V_{DD} via a 10K resistor.
3. ALE must be tied high.
4. Logical "and" of CS, DS, WR starts memory cycle.

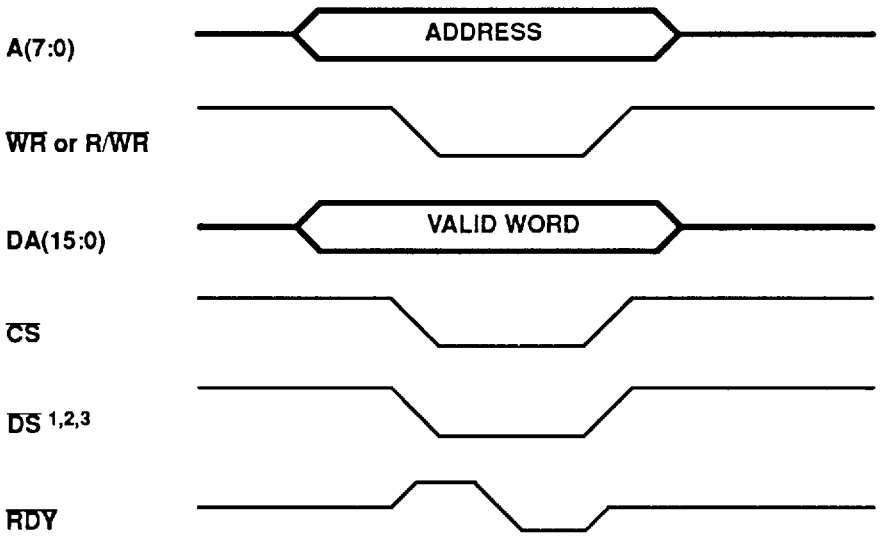
Non-Multiplexed Register Read Access (8-bit Mode)



Notes:
1. DS asserts to signal the ASCENT to place data on the bus.
2. Tie DA(15:8) to V_{DD} via a 10K resistor.
3. ALE must be tied high.
4 Logical "and" of CS, DS, RD starts memory cycle.
5. When using R/WR as an input signal, tie RD to a logical one. During the read cycle R/WR remains a logical one.

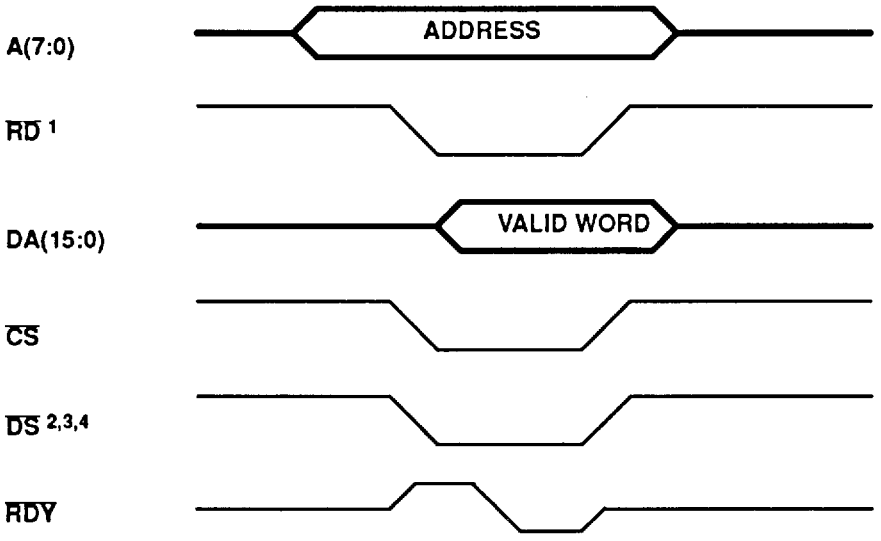
Figure 4a. 8-bit Register Access

Non-Multiplexed Register Write Access (16-bit Mode)



Notes:
1. DS asserts to signal that data is valid on the bus.
2. ALE must be tied high.
3. Logical "and" of CS, DS, WR starts memory cycle.

Non-Multiplexed Register Read Access (16-bit Mode)

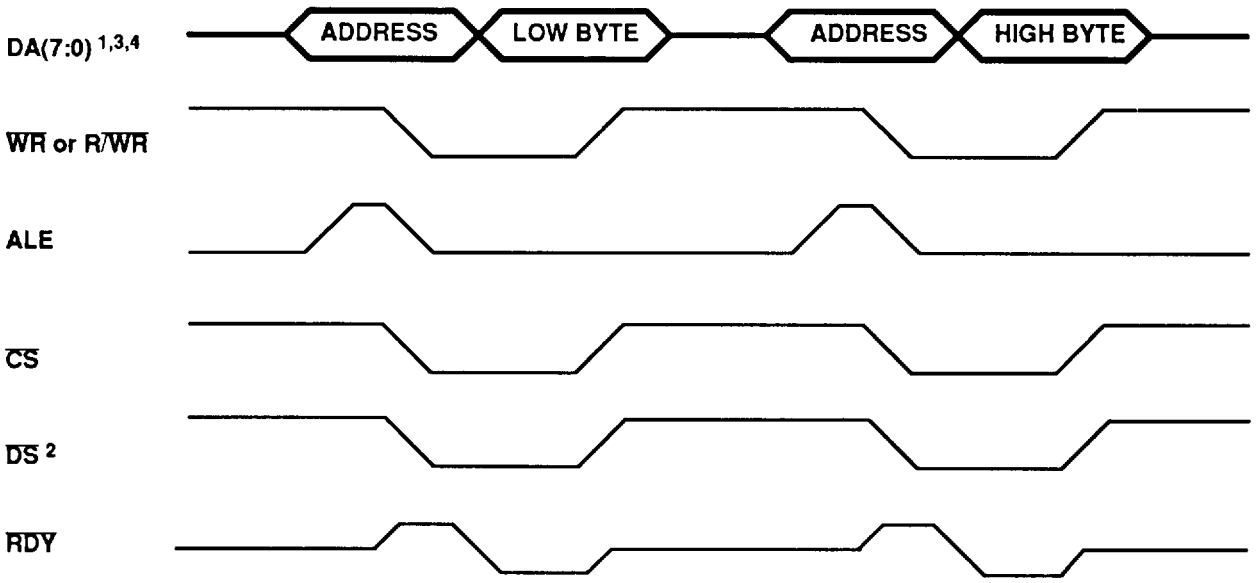


Notes:
1. When using R/WR as an input signal, tie RD to a logical one. During the read cycle R/WR remains a logical one.
2. DS asserts to signal the ASCENT to place data on the bus.
3. ALE must be tied high.
4. Logical "and" of CS, DS, RD starts memory cycle.

Figure 4b. 16-bit Register Access

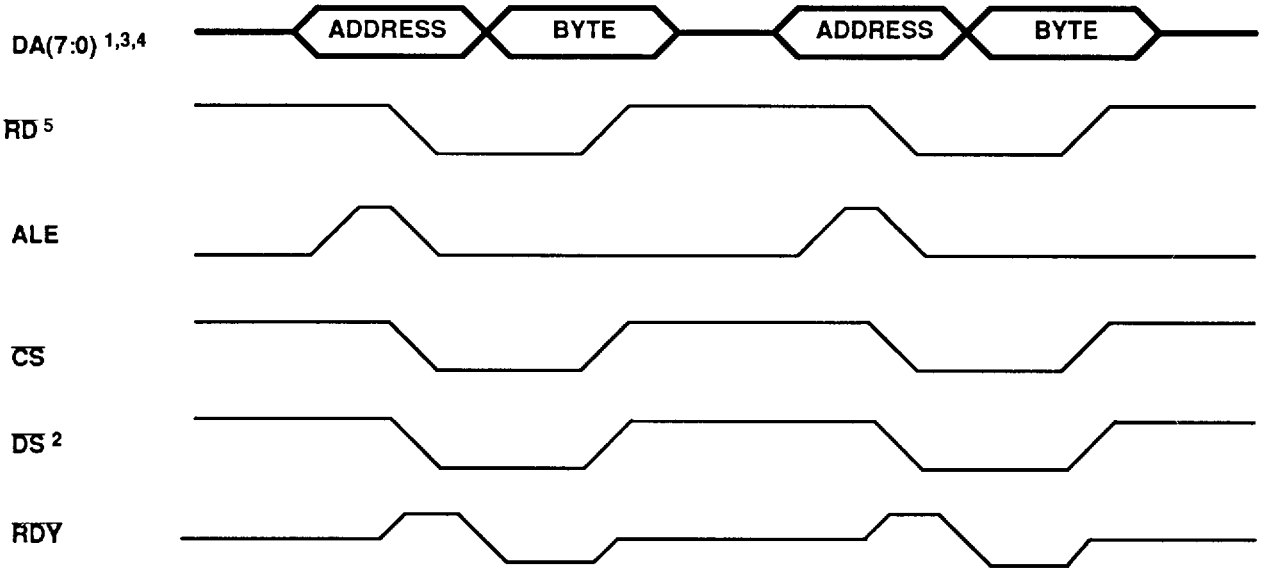
9343947 0004956 3TT

Multiplexed Register Write Access (8-bit Mode)



Notes:
1. For multiplexed address and data interfaces ALE latches the address into the ASCENT. Data is applied to inputs DA(7:0), tie A(15:0) to either a logical one or zero.
2. $\overline{DS}$ asserts to signal that data is valid on the bus.
3. Logical "and" of $\overline{CS}$, $\overline{DS}$, $\overline{WR}$ starts memory cycle.
4. Tie DA(15:7) to V_{SS} via a 10K resistor.

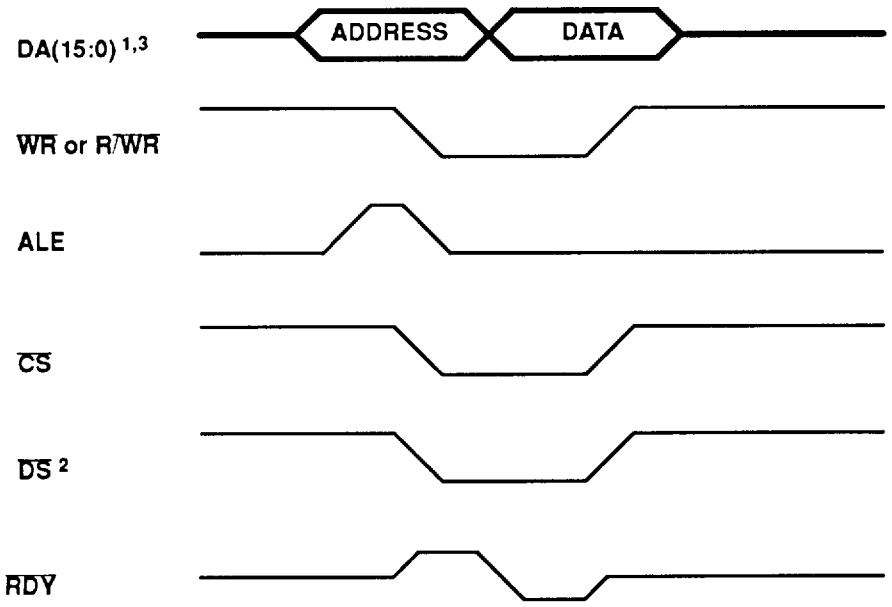
Multiplexed Register Read Access (8-bit Mode)



Notes:
1. For multiplexed address and data interfaces ALE latches the address into the ASCENT. Data is read from outputs DA(7:0), tie A(15:0) to either a logical one or zero.
2. $\overline{DS}$ asserts to signal the ASCENT to place data on the bus.
3. Logical "and" of $\overline{CS}$, $\overline{DS}$, $\overline{RD}$ starts memory cycle.
4. Tie DA(15:7) to V_{SS} via a 10K resistor.
5. When using R/$\overline{WR}$ as an input signal, tie $\overline{RD}$ to a logical one. During the read cycle R/$\overline{WR}$ remains a logical one.

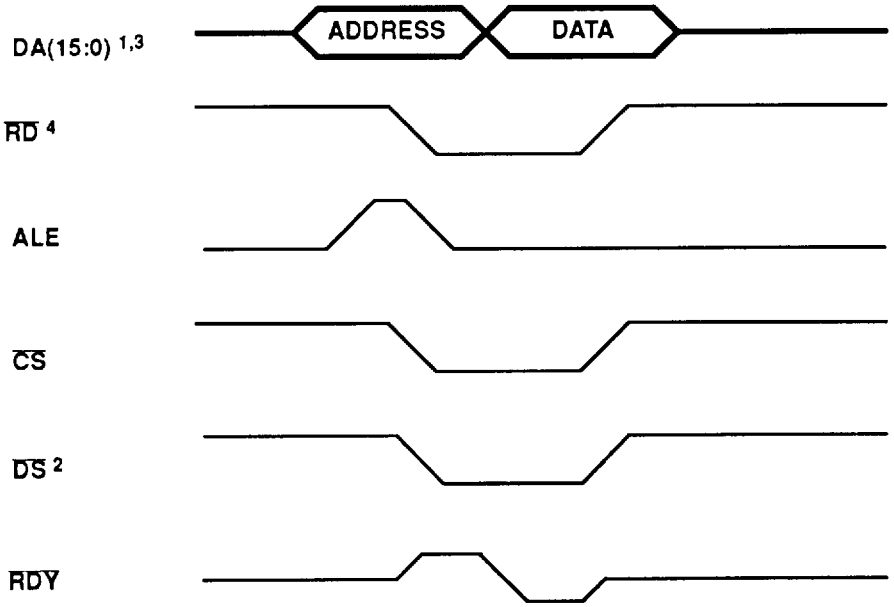
Figure 4c. Multiplexed 8-bit Register Access

Multiplexed Register Write Access (16-bit Mode)



Notes:
1. For multiplexed address and data interfaces ALE latches the address into the ASCENT. Data is applied to inputs DA(15:0), tie A(15:0) to either a logical one or zero.
2. $\overline{DS}$ asserts to signal that data is valid on the bus.
3. Logical "and" of $\overline{CS}$, $\overline{DS}$, $\overline{WR}$ starts memory cycle.

Multiplexed Register Read Access (16-bit Mode)



Notes:
1. For multiplexed address and data interfaces ALE latches the address into the ASCENT. Data is read from outputs DA(15:0), tie A(15:0) to either a logical one or zero.
2. $\overline{DS}$ asserts to signal the ASCENT to place data on the bus.
3. Logical "and" of $\overline{CS}$, $\overline{DS}$, $\overline{RD}$ starts memory cycle.
4. When using R/$\overline{WR}$ as an input signal, tie $\overline{RD}$ to a logical one. During the read cycle R/$\overline{WR}$ remains a logical one.

Figure 4d. Multiplexed 16-bit Register Access

9343947 0004958 172

4.3 Hardware Interface

The ASCENT offers hardware designers a flexible interface to commonly found embedded computers. The hardware designer selects control signals, bus widths, and bus functionality (i.e., non-multiplexed versus multiplexed) to meet their interface requirements. Table 3 reviews how the user selects the ASCENT's operation that best meets their interface requirements.

5.0 MIL-STD-1773A INTERFACE

The ASCENT's internal encoder/decoder interfaces directly with bus transceivers. Figures 5 and 6 show a block diagram of a complete MIL-STD-1773A interface.

The ASCENT protocol device supplies RT fail-safe timer stimulus which enables and disables the bus transceivers as required by most standards. Bus transmitters are enabled prior to transmission on bus, otherwise the transmitters are in an idle state. Bus receivers are enabled at all times.

Table 3. User Selectable Control Signals

Function	Input Pin	Logical 1	Logical 0
Rate Select	MSEL(0) [1]	20Mbps	1Mbps
Mode Select	MSEL(1)	BC	RT
Control Signal Select	MSEL(2)	R/$\overline{WR}$, $\overline{CS}$, $\overline{DS}$, $\overline{RDY}$	$\overline{RD}$, $\overline{WR}$, $\overline{CS}$, $\overline{DS}$, $\overline{RDY}$,
Bus Functionality Select	MSEL(3)	Non-Multiplexed Address and Data	Multiplexed Address and Data, ALE
Interrupt Select	MSEL(4)	Level ($\overline{MSG\_INT}$ and $\overline{MSG\_ACK}$)	Pulse ($\overline{MSG\_INT}$)
Bus Width Select	MSEL(5)	8-bit	16-bit

Note:
1. RT mode of operation MSEL(0) is an input and BC mode of operation MSEL (0) is an output pin.

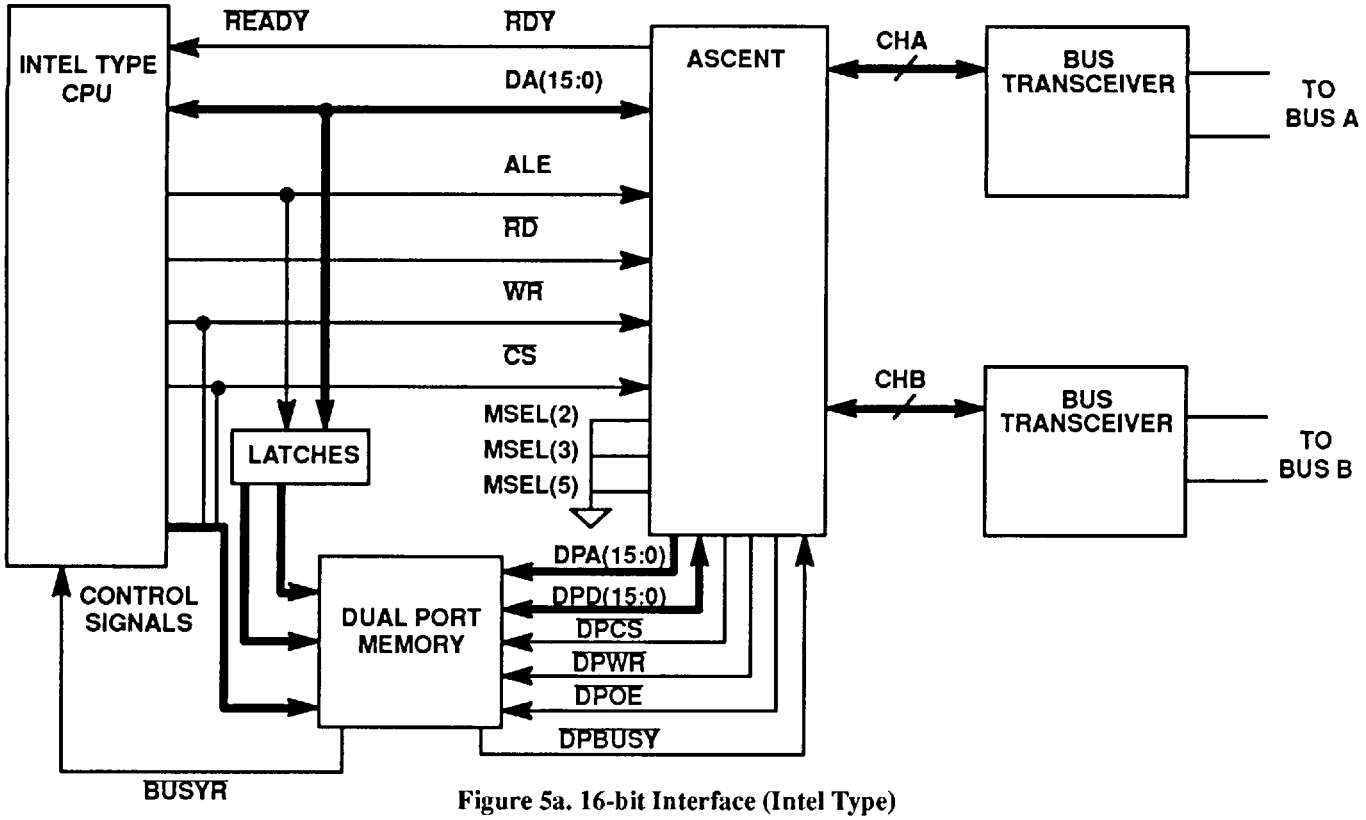


Figure 5a. 16-bit Interface (Intel Type)

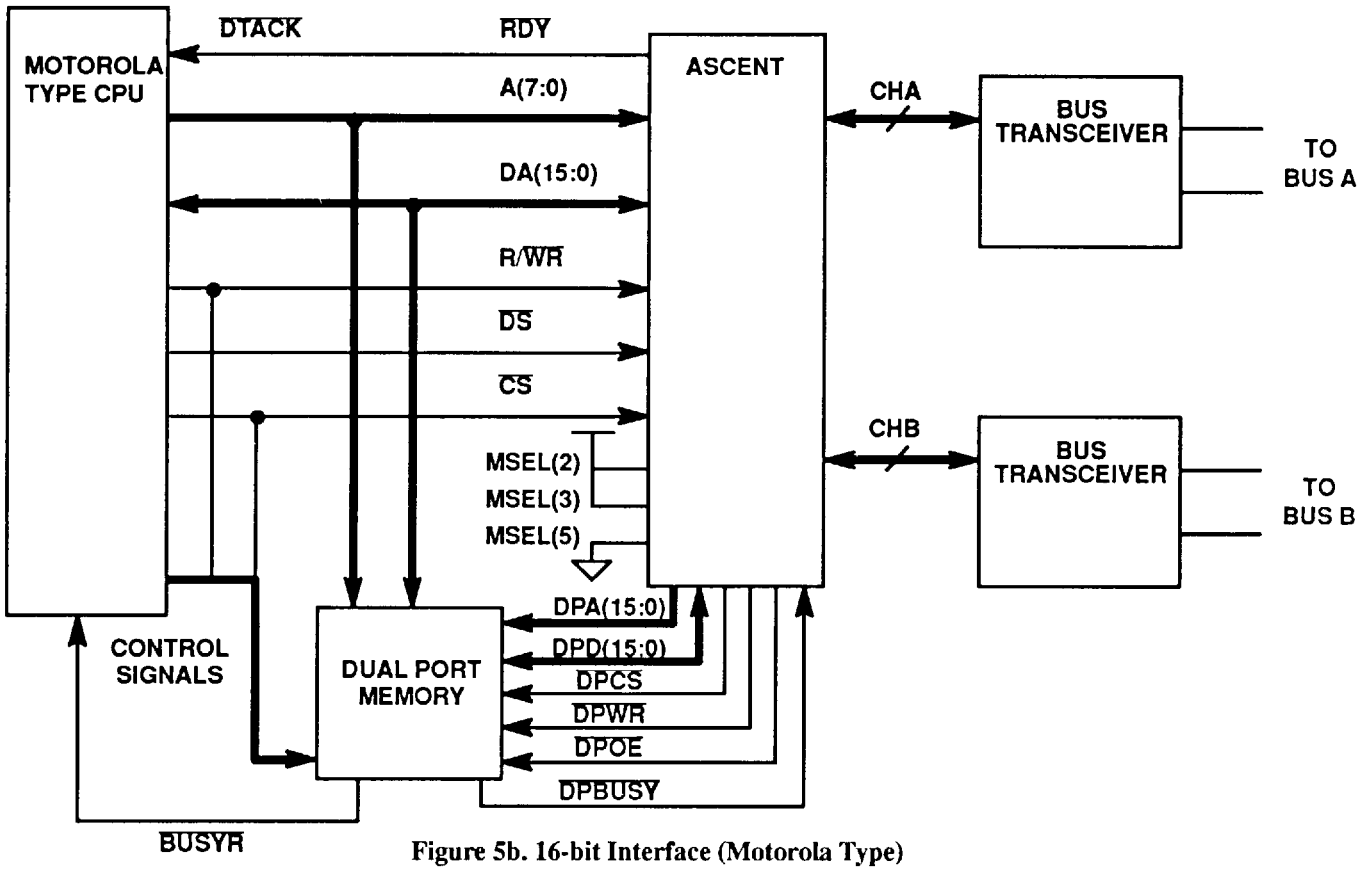


Figure 5b. 16-bit Interface (Motorola Type)

9343947 0004960 820

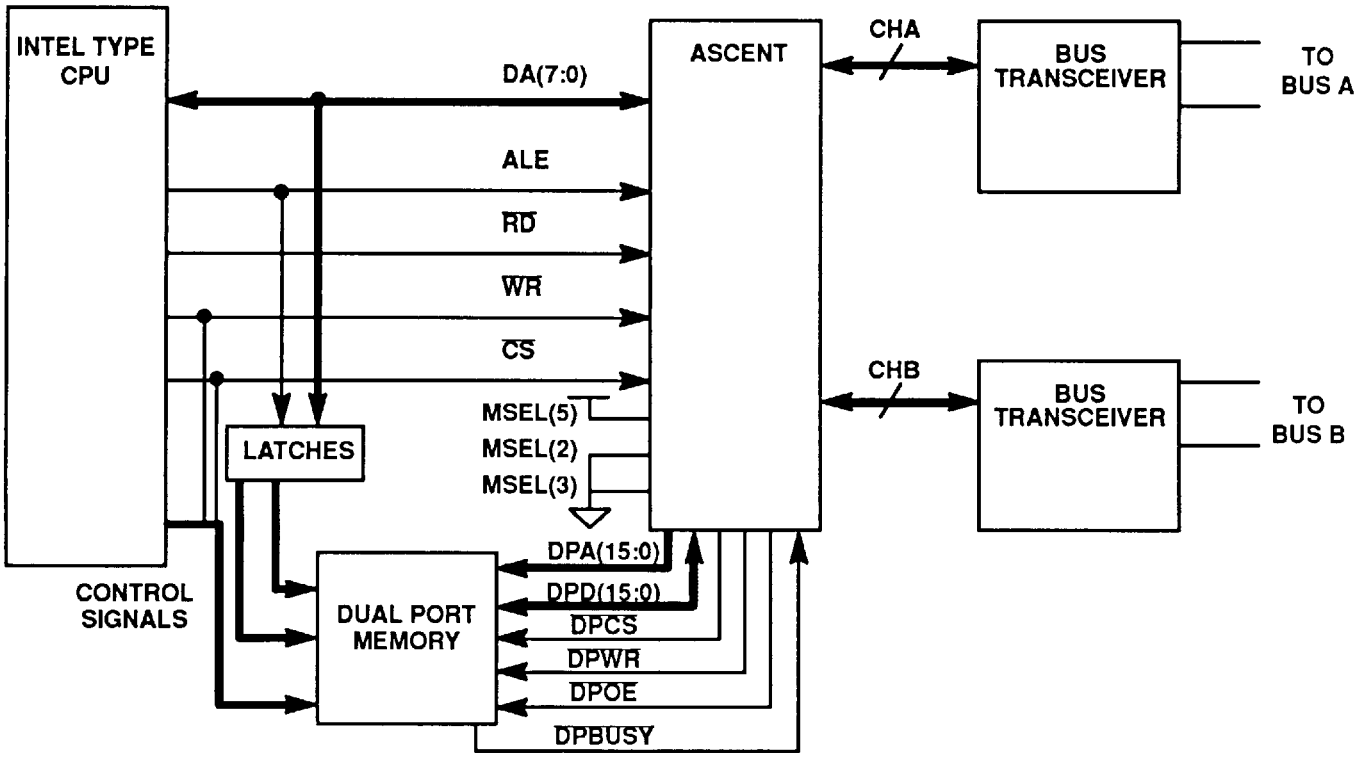


Figure 6a. 8-bit Interface (Intel Type)

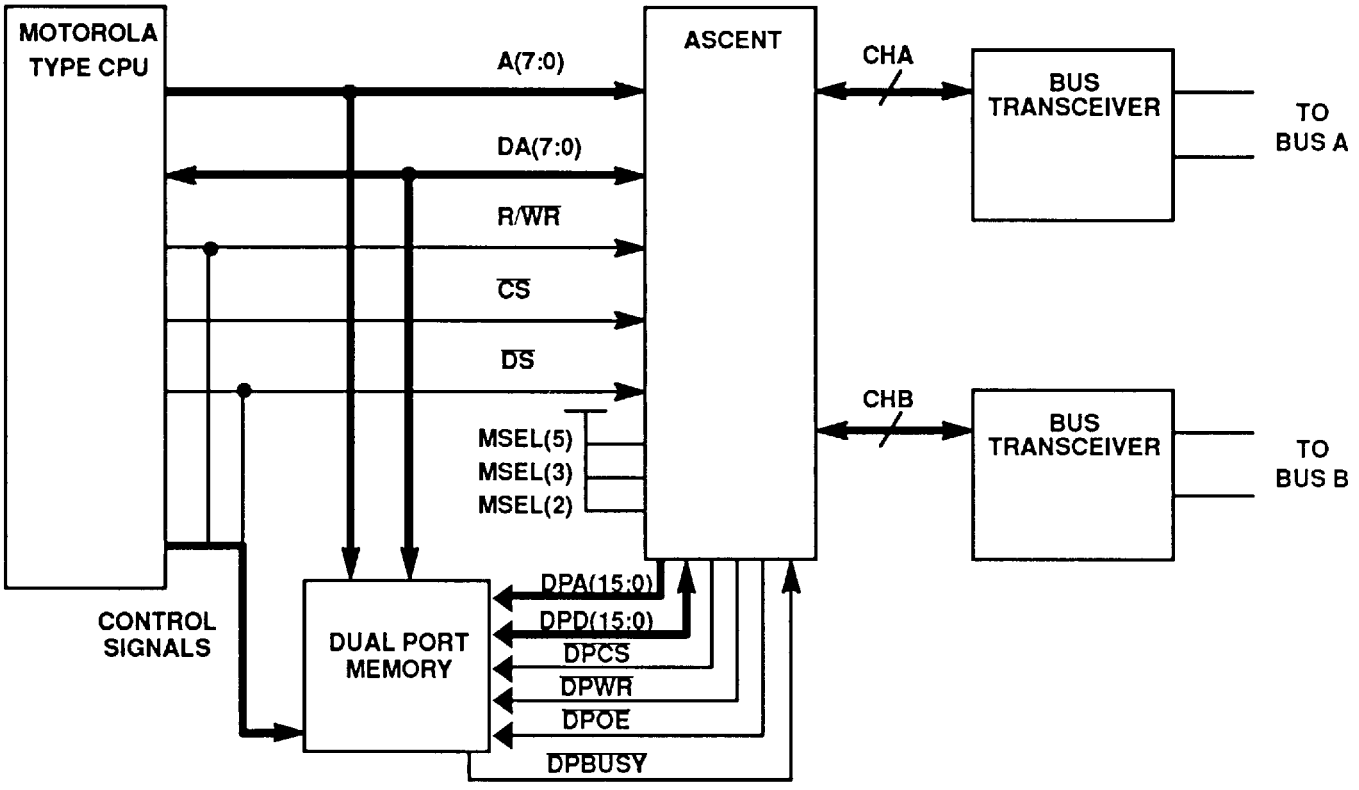


Figure 6b. 8-bit Interface (Motorola Type)

11.0 PIN DESCRIPTION

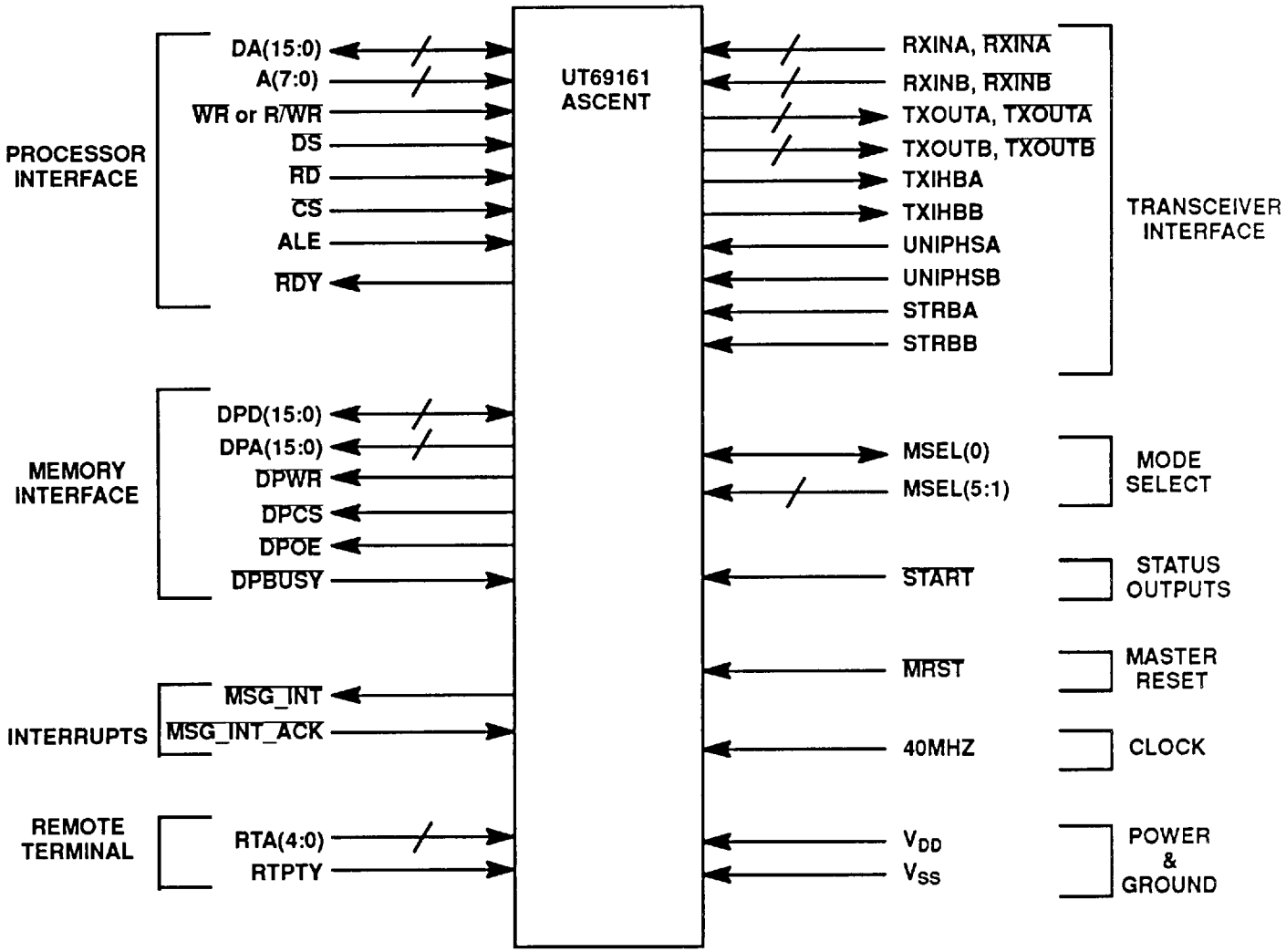


Figure 10. ASCENT Functional Pin Diagram

UTMC Main Office
1575 Garden of the Gods Road
Colorado Springs, CO 80907-3486
1-800-722-1575

Western Sales Office
101 Columbia Street, Suite 130
Aliso Viejo, CA 92636
1-714-362-2260

Eastern Sales Office
1901 S. Harbor City Blvd., Suite 802
Melbourne, FL 32901
1-407-951-4164

UNITED TECHNOLOGIES MICROELECTRONICS CENTER

United Technologies Microelectronics Center, Inc. (UTMC) reserves the right to make changes to any products and services herein at any time without notice. Consult UTMC or an authorized sales representative to verify that the information in this data sheet is current before using this product. UTMC does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by UTMC; nor does the purchase, lease, or use of a product or service from UTMC convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual rights of UTMC or of third parties.

Copyright 1995 by United Technologies Microelectronics Center, Inc.

All rights reserved

9343947 0004962 6T3

49228