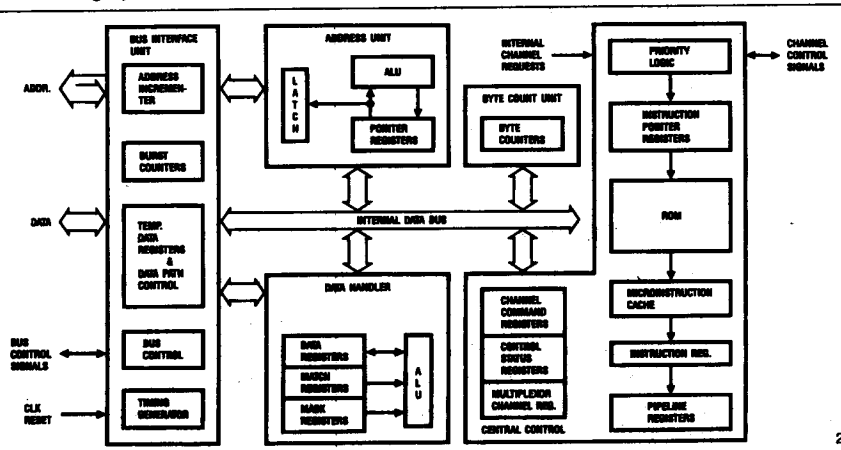
82258
ADVANCED DIRECT MEMORY ACCESS COPROCESSOR (ADMA)

- **High Performance 16 Bit DMA Coprocessor for the 80386, 80286 and 80186 Families**
  - **— 8 MByte/sec Maximum Transfer Rate in 8 MHz 80286 Systems**
- **Four Independently Programmable Channels**
- **Multiplexor Channel Capability to Support Up to 32 Subchannels**
- **On Chip Bus Interface for the Whole 8086 Architecture**
  - **— 80286**
  - **— 80186/188**
  - **— 8086/88**
- **Command Chaining for CPU Independent Processing**
- **Automatic Data Chaining for Gathering and Scattering of Data Blocks**
- **16 MByte Addressing Range**
- **16 MByte Block Transfer Capability**
- **"On the Fly" Compare, Translate and Verify Operations**
- **Automatic Assembly/Disassembly of Data**
- **Programmable Bus Loading**
- **6 and 8 MHz Speed Selections**
- **Available in 68-Pin LCC and PGA Packages**

(See Packaging Spec. Order #231369)

INTRODUCTION

Intel's 82258, Advanced Direct Memory Access Coprocessor is a high performance, 16 bit DMA processor optimized for the 80286, 80186 and the 8086 families of CPUs and compatible with 80386 CPU. It has on-chip bus interface for the whole 8086 family architecture. Four high speed, independently programmable DMA channels can achieve a maximum cumulative transfer rate of 8 MByte/sec in an 8 MHz 80286 system and 4 MByte/sec in 8 MHz 8086/80186 systems. Channel 3 can be used as a Multiplexor channel, whereby, it supports 32 subchannels. This flexibility allows one to use a single DMA channel to handle a large number of slow and medium speed I/O devices. Advanced capabilities like Command and Data chaining and "On the fly" operations allow the 82258 to remove the I/O management load from the processor. The 82258 addresses the full 80286 CPU memory (16 MB for 80286), thus simplifying the system design. Automatic assembly/disassembly of data allows 16 bit processors to interface with common 8 bit peripherals and vice-versa. Remote mode of operation, where the 82258 has its own resident bus, allows modular system design. The 82258 complements the high performance, multitasking capabilities of the 80286.



231263-1

Figure 1. 82258 Internal Block Diagram

November 1987
Order Number: 231263-004

FABRICATION

The 82258 is a 68 pin device, fabricated in Intel HMOS II technology. It is packaged in JEDEC type A hermetic leadless chip carrier and pin grid array.

PIN DEFINITIONS AND FUNCTIONS

The 82258 has four operational modes

— 286
— 186—for the 80186/88 and the 8086/88 (Min. mode) CPUs
— 8086—for the 8086/88 (Max. mode) CPUs
— Remote

Pins of the 82258 have different definitions for different modes. 286 and remote modes have the same non-multiplexed bus structure and similar pin descriptions. Similarly, the 186 and the 8086 modes have multiplexed bus and similar pin description.

PINNING IN THE 286 MODE

In the 286 mode, the bus signals and the bus timings of the 82258 are the same as those of the 80286 processor. The processor can access the internal registers of the 82258 and these accesses must be supported by the bus signals. Therfore, some of the bus control signals are bidirectional and some additional bus control signals are necessary.

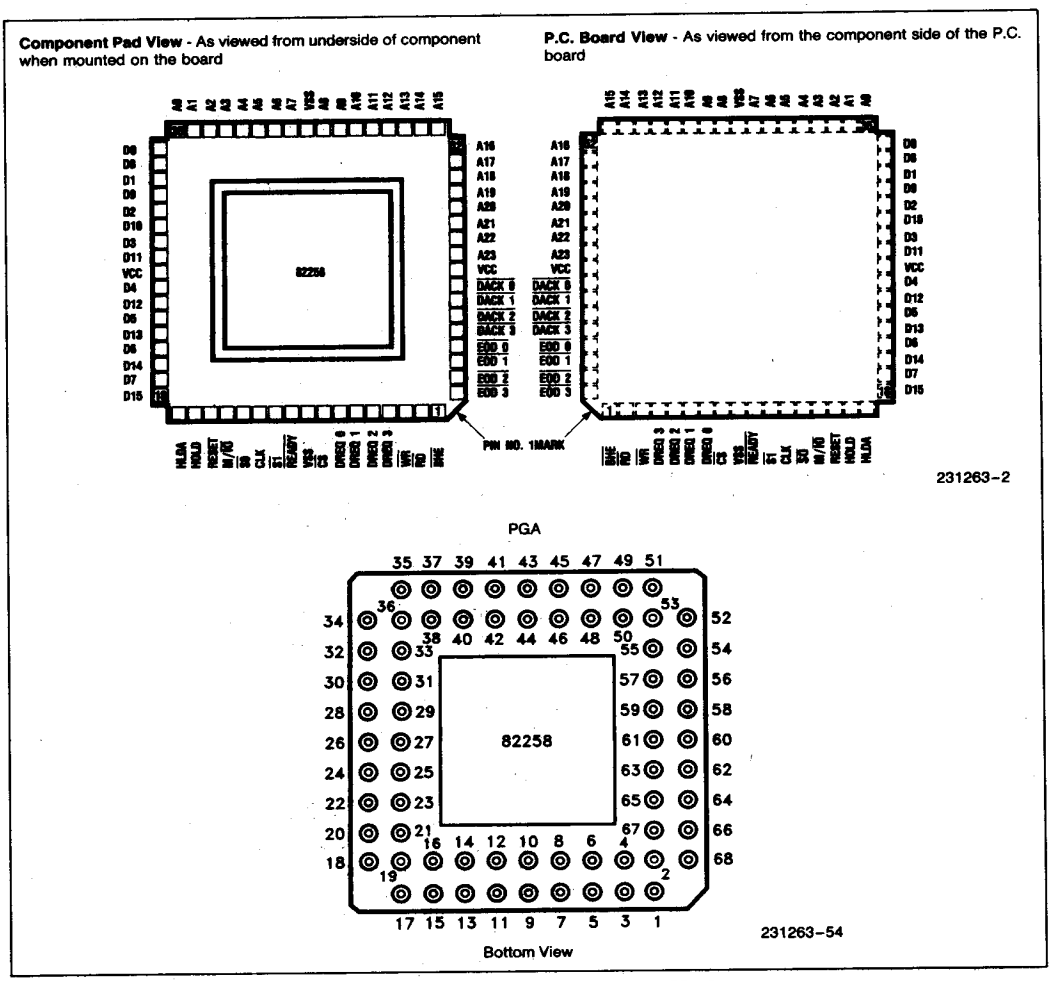


Figure 2. Pin Configuration in 286 Mode

Table 1. Pin Description for the 286 Mode (Also Contains Pins Identical in Other Modes)

Symbol	Pin Type Input (I) Output (O)	Pin Number	Identical In All Modes	Functions
$\overline{BHE}$	I/O	1	YES	**BUS HIGH ENABLE** indicates transfer of data on the upper byte of the data bus, D15–D8. Eight bit devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select function. $\overline{BHE}$ is active LOW and floats to Tri-State OFF when the 82258 does not own the bus.

$\overline{BHE}$ and A0 Encoding

$\overline{BHE}$ Value	A0 Value	Function
0	0	Word Transfer (D15–D0)
0	1	Byte Transfer on upper half of data bus (D15–D8)
1	0	Byte Transfer on lower half of data bus (D7–D0)
1	1	Odd addressed byte on 8 bit bus (D7–D0)

$\overline{RD}$	I	2	NO	**READ** command in conjunction with chip select ($\overline{CS}$) enables reading out of the 82258 register, addressed by the address lines A7-A0. $\overline{RD}$ is an active LOW signal and is asynchronous to the 82258 clock.
$\overline{WR}$	I	3	NO	**WRITE** command along with $\overline{CS}$ is used for writing into the 82258 registers. $\overline{WR}$ is an active LOW signal and is asynchronous to the 82258 clock.
DREQ3-DREQ0	I	4—7	YES	**DMA REQUEST** input signals are used for externally synchronized DMA transfers. If channel 3 is used as a Multiplexor channel, DREQ3 is defined as I/O Request (IOREQ) signal. These signals are active HIGH signals and are asynchronous to the 82258 clock. Unused DREQn lines should not be left floating, but should be tied inactive to V_{SS}.
$\overline{CS}$	I	8	NO	**CHIP SELECT** is used to enable a processor to access the 82258 registers. This access is additionally controlled either by bus status signals or by the Read or Write command signals. $\overline{CS}$ is an active LOW signal, asynchronous to the 82258 clock.
$\overline{READY}$	I	10	NO	**BUS READY** terminates a bus cycle. Bus cycles are extended without limit until terminated by an active $\overline{READY}$. $\overline{READY}$ is an active LOW, synchronous input, requiring set up and hold times relative to system clock to be met for correct operation.
$\overline{S1}$, $\overline{S0}$	I/O	11,13	YES	**BUS CYCLE STATUS** signals control the support circuitry. The beginning of a bus cycle is indicated by $\overline{S1}$, or $\overline{S0}$, or both going active. The termination of a bus cycle is indicated by all the status signals going inactive in the 186 mode or the bus ready ($\overline{READY}$) going active in the 286 mode. Both $\overline{S0}$ & $\overline{S1}$ are active LOW signals. $\overline{S0}$, $\overline{S1}$ along with $\overline{S2}$ (in the 186 mode) or M/$\overline{IO}$ (in the 286 mode) define the type of bus cycle. $\overline{S2}$ and M/$\overline{IO}$ have the same meaning but, in the 186 mode $\overline{S2}$ signal can be active only when at least one of $\overline{S1}$ and $\overline{S0}$ is active, whereas in the 286 mode the M/$\overline{IO}$ signal is valid with the address on address lines.

Table 1. Pin Description for the 286 Mode (Also Contains Pins Identical in Other Modes) (Continued)

<table>
<tr><th rowspan="2">Symbol</th><th colspan="2">Pin</th><th rowspan="2">Identical In All Modes</th><th rowspan="2">Functions</th></tr>
<tr><th>Type Input (I) Output (O)</th><th>Number</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
The 82258 Bus Cycle Status Definitions (82258 Local Bus Master, All Signals (O))
<table>
<tr><th>M/$\overline{IO}$ or $\overline{S2}$</th><th>$\overline{S1}$</th><th>$\overline{S0}$</th><th>Bus Cycle Initiated</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Read I/O-Vector (For Multiplexor channel)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Read from I/O space</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Write into I/O space</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>None. (Does not occur in the 186 mode).</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>None. (Does not occur)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Read from memory space</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Write into memory space</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>None; not a bus cycle</td></tr>
</table>
When the 82258 is not a bus master of the local bus, the status signals are used as inputs for detection of synchronous accesses to the 82258.
Interpretation of the Status and $\overline{CS}$ Signals by the 82258 (82258 Slave, All Signals (I))
<table>
<tr><th>$\overline{CS}$</th><th>$\overline{S1}$</th><th>$\overline{S0}$</th><th>Interpretation</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>82258 not selected (No action)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No 82258 access (No action)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Read from an 82258 register</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Write into an 82258 register</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Not a bus cycle*</td></tr>
</table>
*: The 82258 is selected but no synchronous access is activated. The 82258 monitors $\overline{RD}$ and $\overline{WR}$ signals for detection of an asynchronous access.
</td></tr>
<tr><td>CLK</td><td>I</td><td>12</td><td>NO</td><td>SYSTEM CLOCK provides the fundamental system timing. It is divided by two to generate the 82258 internal clock. CLK is an active HIGH signal which can be connected directly to the 82284 CLK output. The internal divide-by-two circuitry is synchronized to the external clock generator by a LOW to HIGH transition on the RESET input, or by first HIGH to LOW transition on the Status Input $\overline{S0}$ or $\overline{S1}$ after RESET.</td></tr>
<tr><td>M/$\overline{IO}$</td><td>O</td><td>14</td><td>NO</td><td>MEMORY/$\overline{IO}$ SELECT distinguishes between memory and I/O space addresses.</td></tr>
<tr><td>RESET</td><td>I</td><td>15</td><td>YES</td><td>SYSTEM RESET forces the 82258 to the initial state. RESET is an active HIGH signal and must be synchronous to the system clock. Reset must be activated for at least 16 CLK cycles.</td></tr>
</table>

Table 1. Pin Description for the 286 Mode (Also Contains Pins Identical in Other Modes) (Continued)

Symbol	Pin		Identical In All Modes	Functions
	Type Input (I) Output (O)	Number		
HOLD HLDA	O I	16 17	NO	**BUS HOLD REQUEST AND HOLD ACKNOWLEDGE** control ownership of the local 82258 bus. When active, HOLD indicates a request for the control of the local bus. HOLD goes inactive when the 82258 relinquishes the bus. HLDA, when active, indicates that the 82258 can acquire the control of the bus. When HLDA goes inactive, the 82258 must relinquish the bus at the end of its current cycle. HLDA may be asynchronous to the system clock. Both HOLD and HLDA are active HIGH signals.
D15-D0	I/O	18-25, 27-34	NO	**DATA BUS** is the bidirectional 16 bit bus. For use with an 8 bit bus, only the lower 8 data lines D0–D7 are relevant. The data bus is active HIGH.
A0–A7	I/O	35-42	NO	**ADDRESS LINES** A0–A7 are the lower 8 address lines for DMA transfers. They are also used to input the register address when the processor accesses an 82258 register. All lines are active HIGH.
A8–A23	O	44-59	NO	**ADDRESS LINES** A8–A23 form the remainder of the 82258 address bus. Address bus is active HIGH. ***Pin A21 must have a pull-up resistor (n 10k Ω) connected to it to ensure that it is high during reset.***
$\overline{DACK0}$–$\overline{DACK3}$	O	61-64	YES	**DMA ACKNOWLEDGE** signal acknowledges the requests of the corresponding DREQ signal. $\overline{DACK}$i goes active when the requested transfers are performed on the channel i in response to a DREQi. If channel 3 is in the multiplexor mode, $\overline{DACK}$3 is defined as I/O acknowledge (IOACK). These signals are active LOW.
$\overline{EOD0}$–$\overline{EOD3}$	I/O	65-68	YES	**END OF DMA** signals are open drain drivers with internal high impedance pull up resistors (an external pull-up resistor is required) and can be used as quasi-bi-directional lines. These signals are active LOW. As OUTPUTs the signals are activated (if enabled) for two T-STATE cycles at the end of the DMA transfer of the corresponding channel or they are activated under program control (End of DMA output or interrupt output). $\overline{EOD}$s acts as "End of DMA" level triggered INPUTs if the signals are held high internally but forced low by the external circuitry for at least 250 ns. The current transfer is aborted and the 82258 continues with the next command. $\overline{EOD2}$ can also be used as a common active high interrupt signal (INTOUT) for all four channels. In this mode, this signal is a push-pull output and not an open drain output. Other EODi pins may still be used in their regular I/O mode.
V_{SS}	I	9, 43	YES	**SYSTEM GROUND:** 0 Volt.
V_{CC}	I	26, 60	YES	**SYSTEM POWER:** + 5V Power Supply Pin.

PINNING IN THE 186 MODE

The 80186 has a multiplexed bus structure. Therefore, many 82258 pins have different meaning in the 186 mode than in the 286 mode. Since the 80186 has 20 address lines compared to 24 for the 80286, the 4 extra lines are used to generate additional bus control signals. The following table gives the details of pins having different meaning in the 186 mode compared to the 286 mode:

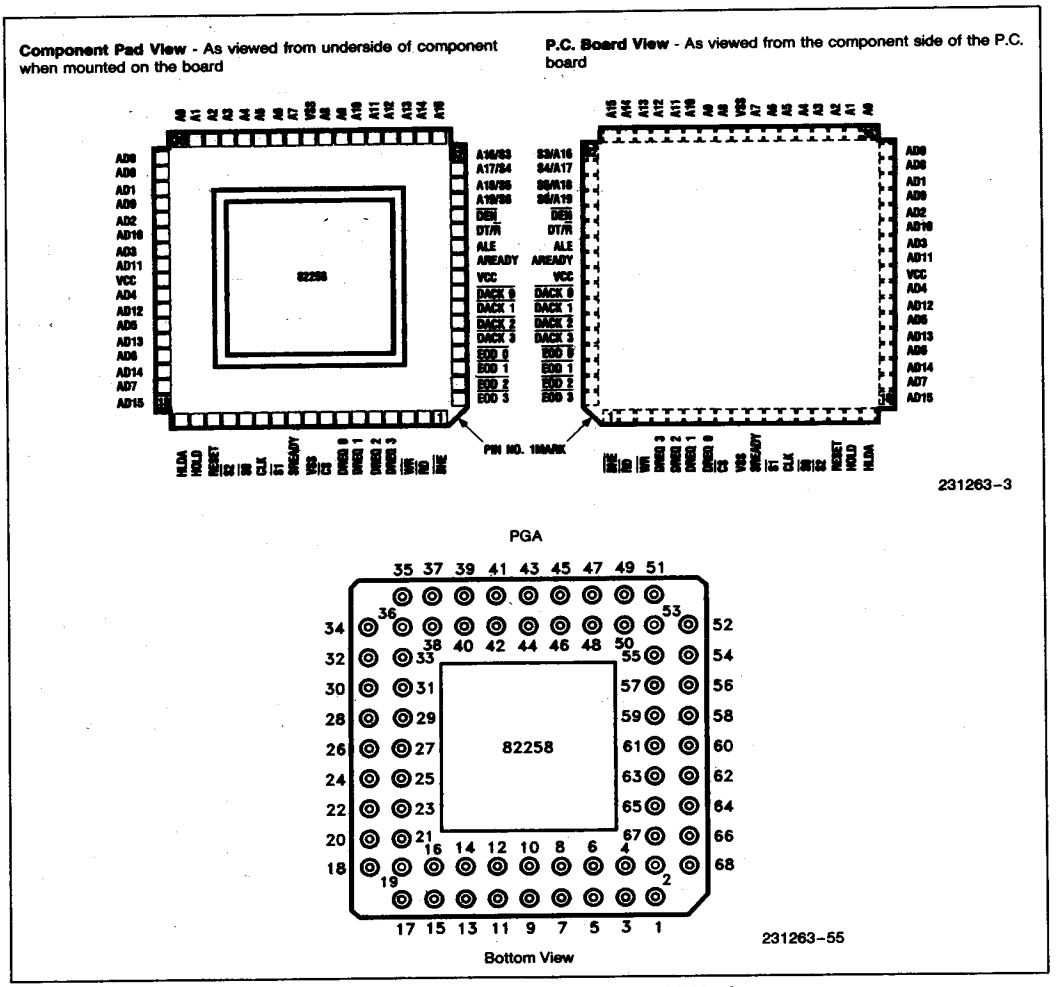


Figure 3. Pin Configuration in the 186 Mode

Table 2. Changes in Pin Description in the 186 Mode: (Compared to the 286 Mode)

Symbol	Pin Type Input (I) Output (O)	Pin Number	Functions
$\overline{RD}$, $\overline{WR}$	I/O	2, 3	**READ, WRITE** In the 186 mode, the $\overline{RD}$ & $\overline{WR}$ pins are used additionally as output pins to support the 80186 or the 8086 minimum systems. These signals are active LOW.
ALE	O	58	**ADDRESS LATCH ENABLE** signal provides a strobe to separate the address information on the multiplexed address-data lines. ALE is an active HIGH signal.
$\overline{DEN}$	O	56	**DATA ENABLE** signal is used for enabling the data transceiver, 8286/8287. $\overline{DEN}$ is an active LOW signal.
DT/$\overline{R}$	O	57	**DATA TRANSMIT/RECEIVE** signal controls the direction of data flow through the external data bus transceiver, depending on whether a read, or a write bus cycle is performed. ***This pin must have a pullup resistor connected to it to ensure that it is high during reset.***
SREADY	I	10	**SYNCHRONOUS READY** input signal must be synchronized externally. Use of this pin permits a relaxed system and timing specification by eliminating the clock phase, required for resolving the signal level, when using AREADY input. SREADY is an active HIGH signal.
CLK	I	12	**SYSTEM CLOCK** input gets a prescaled signal from the 186 clock (CLKOUT) or the 8086 clock (50% duty cycle for 186 and 33% duty cycle for 8086). No internal prescaling is done. CLK is an active HIGH signal.
$\overline{S2}$	O	14	**STATUS SIGNAL** along with $\overline{S0}$ and $\overline{S1}$ provides the bus cycle description (for details see 286 mode pin description of $\overline{S0}$ and $\overline{S1}$).
AD0–AD15 A0–A7 A8–A15	I/O I/O O	18–25 27–34 35–42 44–51	**ADDRESS/DATA BUS** signals AD0–AD15 contain multiplexed lower address and data information. Also, the demultiplexed address information is available on address pins A0–A15.
A16/S3–A19/S6	O	52–55	**ADDRESS PINS** A16–A19 are multiplexed with additional status information on the bus cycle. These pins are active HIGH. Signals S5 and S6 provide information on the status of the bus cycle. During an active bus cycle, S6 is always high and S5 always low. Low S6 implies a processor bus cycle. Signals S4 and S3 give the channel number for the running bus cycle as follows:

S4	S3	Channel Number
0	0	0
0	1	1
1	0	2
1	1	3

AREADY	I	59	**ASYNCHRONOUS READY** is an asynchronous bus ready signal. The rising edge is internally synchronised. During reset, AREADY must be low to enter the 82258 into the 186 mode. AREADY is an active HIGH signal.

PINNING FOR THE 8086 MODE

For the 8086 MIN configuration the pinning is identical to the 186 mode. For the 8086 MAX configuration, the bus arbitration is done via the $\overline{RQ}/\overline{GT}$ protocol. Otherwise, the function of pins is identical to the 186 mode.

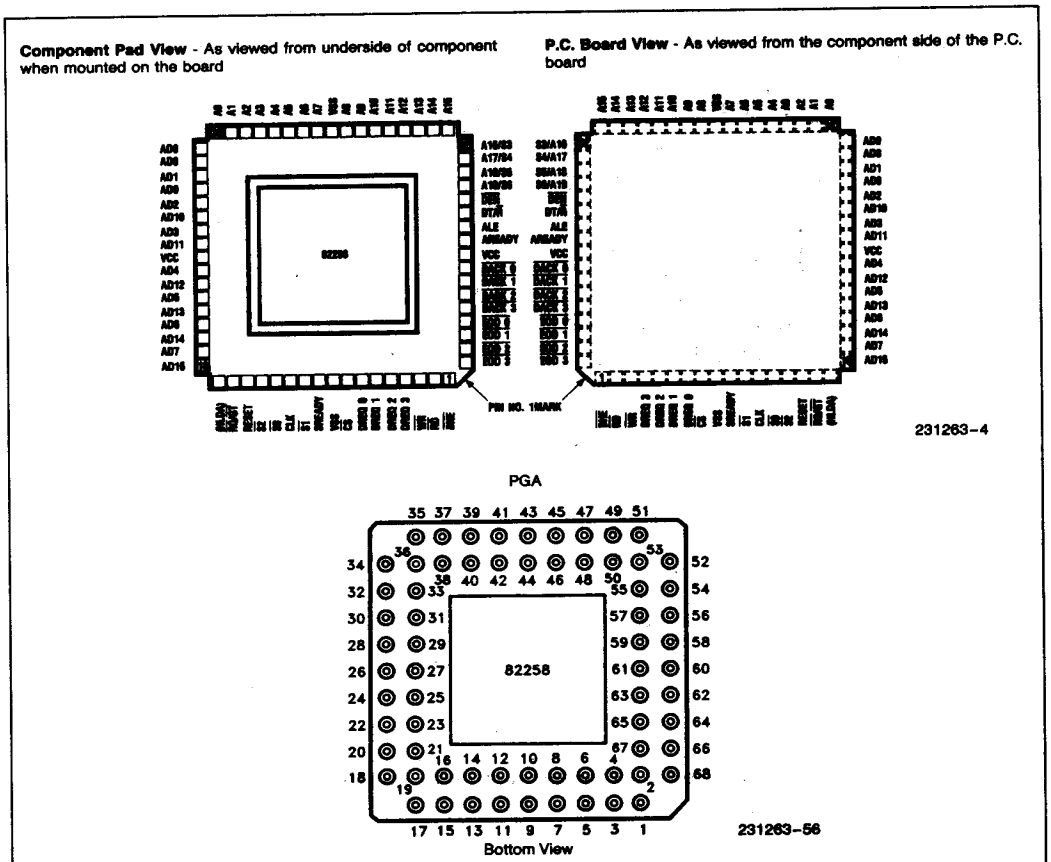


Figure 4. Pin Configuration in the 8086 (Max) Mode

Table 3. Changes in Pin Description in the 8086 (Max) Mode
(Compared to the 186 Mode)

Symbol	Pin		Functions
	Type Input (I) Output (O)	Number	
$\overline{RQ}/\overline{GT}$	I/O	16	**REQUEST/GRANT** implements a one line communication protocol to arbitrate the use of the system bus; normally done via HOLD/HLDA. $\overline{RQ}/\overline{GT}$ is an active LOW signal having an internal pull-up resistor.
HLDA	I	17	**HOLD ACKNOWLEDGE** has no meaning in the 8086 (Max) mode. It should be tied high for mode recognition during reset.

PINNING IN THE REMOTE MODE

In the remote mode, most of the signals have the same function as in the 286 mode. Exceptions are noted in the following table:

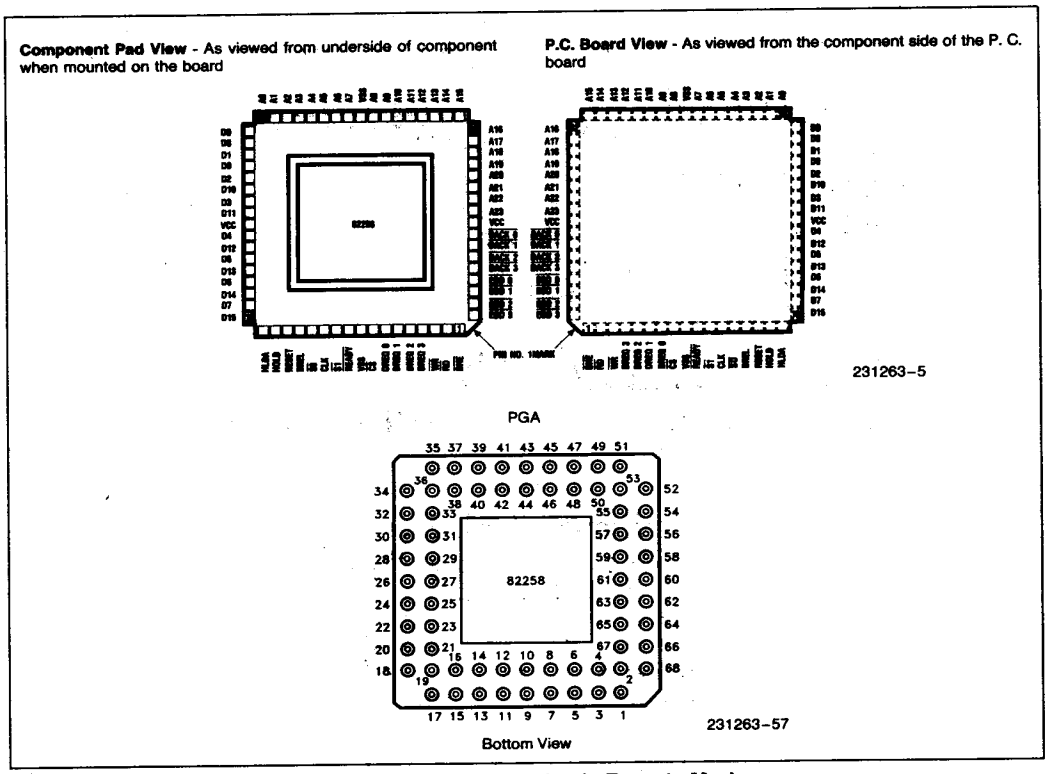


Figure 5. Pin Configuration in Remote Mode

Table 4. Changes in Pin Description in the Remote Mode (Compared to the 286 Mode)

Symbol	Pin		Functions
	Type Input (I) Output (O)	Number	
$\overline{CS}$	I	8	**CHIP SELECT** has two functions in the remote mode. As in the 286 mode, $\overline{CS}$ enables access to the 82258 internal registers. In addition $\overline{CS}$ works as an Access Request Input. When forced LOW, it signals to the 82258 that another bus master needs access to the local bus of the 82258. The 82258 releases the bus as soon as possible and signals it to the CPU by activating BREL (Bus Release) output. $\overline{CS}$ is an active LOW signal.
BREL	O	14	**BUS RELEASE** signal is used to indicate when the 82258 releases control of the resident bus.
HOLD HLDA	O I	16 17	**HOLD & HOLD ACKNOWLEDGE** signals are used only for access to the system bus. They are connected to the bus arbiter (i.e., 82289). Resident bus accesses are directly executed without the HOLD/HLDA sequence.

FUNCTIONAL DESCRIPTION

The 82258 is an advanced DMA coprocessor for the 8086 family architecture. In addition to providing high speed DMA transfers (8 MByte/sec in an 8 MHz 80286 and 4 MByte/sec in 8 MHz 80186/86 systems), the 82258 takes I/O processing load off the CPU, thus improving overall system performance. The 82258 has advanced features not found in the previous generation DMA controllers: multiple-xor channel, command & data chaining and 'on the fly' data manipulation operations.

MODES OF OPERATION

The 82258 has a number of different modes of operation based upon its coupling with the CPU (tight or loose) and its adaptive on-chip bus interface (the 286 bus or the 186 bus).

Figure 6 shows the different operating modes of the 82258 and the CPUs it can interface with in those modes. Figure 7 shows how to configure the 82258 into these different modes.

LOCAL MODE

In this mode the 82258 shares the local bus and all the support/control devices with the CPU. Because of its on-chip bus interface, the 82258 can be directly coupled to the whole 8086 family of microprocessors.

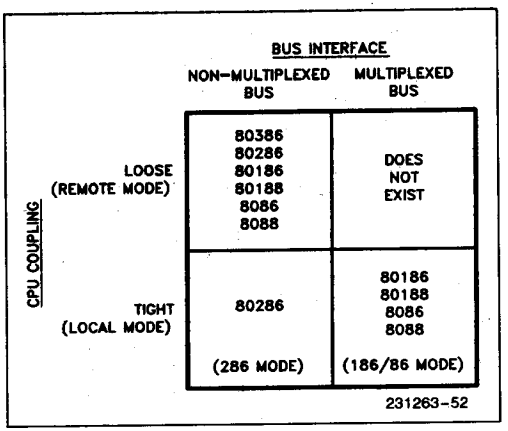


Figure 6. Operating Modes for the 82258

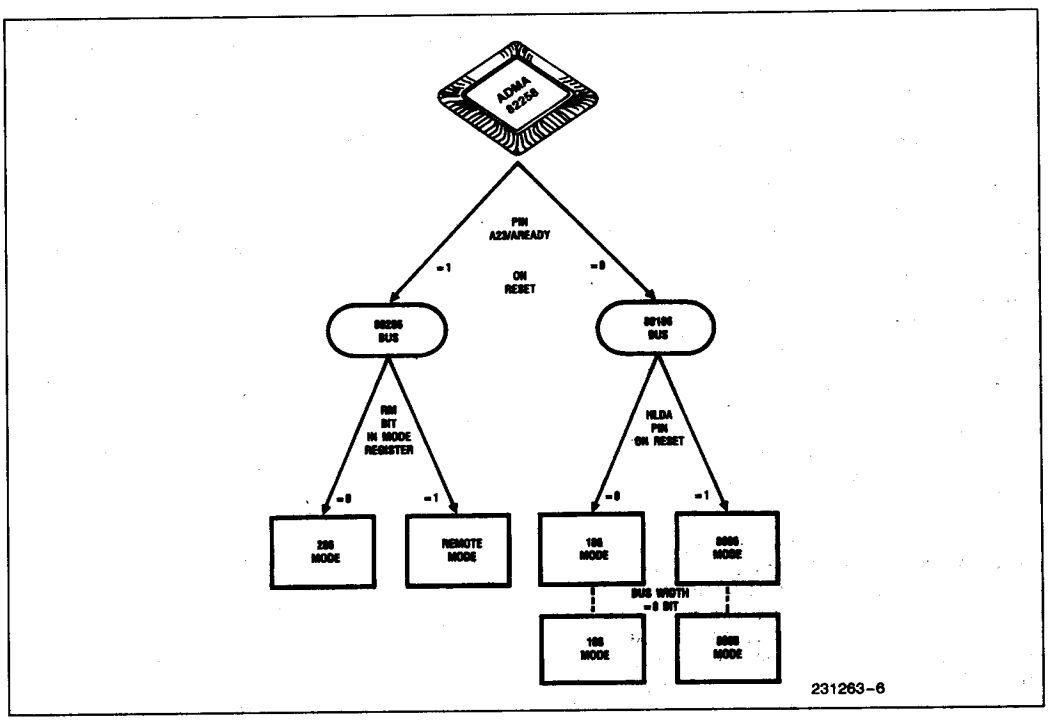


Figure 7. Selecting Modes of Operation

286 System

The configuration in Figure 8 shows the 82258 in the local mode (286 mode) in an 80286 system which includes the Numeric Processor Extension, 80287. The 286 mode is selected during reset (Figure 7). In this mode the 82258 supports the non-multiplexed, pipelined 286 bus. The DMA coprocessor resides on the processor's local bus (physical pins of the 80286) and shares all the support circuits: latches, tranceivers, bus controller and arbiter, clock generator etc. By residing on the 286 bus, the 82258 achieves maximum data transfer rate; up to 8 MByte/sec at 8 MHz for single cycle transfer. HOLD/ HLDA protocol is used for bus exchange between the 80286 and the 82258. The 82258 can be programmed to handle both internal and external terminate conditions. Internal termination is programmed in the command block (in type 2 command as explained later). External termination is handled by the $\overline{EOD}$ (end of DMA) pins if they are enabled. Interrupts for the CPU are handled by an interrupt controller (e.g. 8259A) which receives the end of DMA pins ($\overline{EOD}$ 0-3) as interrupts. The multiplexor channel uses external 8259As to prioritize and arbitrate service requests between peripherals (Figure 13).

To link this system to the MULTIBUS® bus architecture another set of latches, tranceivers, bus controllers and a bus arbiter (i.e., 82289) as shown in Figure 11 (for remote mode configuration) are needed.

186/188 (8086/8088 Min) Systems

The 82258 can be configured into the 186 mode during reset (Figure 7). In this mode it supports the 80186 and the 8086 (Min) processors. It can be programmed to support the 80188 and the 8088 (Min) by programming the bus width in General Mode Register (GMR). Figure 9 shows the 82258 used in an 80186 system containing the 8087 numeric coprocessor. This system uses the 8086 bipolar support components: latches, transceivers and the bus controller (8288). The Integrated Bus Controller (82188) links the 80186 to the 8087. The 82188 is also used to support the 82258, since the 80186 has only one set of bus exchange signals (HOLD/ HLDA). An interrupt controller (8259A) processes the $\overline{EOD}$ signals for the CPU.

In the 186 mode, the 82258 directly supports the 80186/ 8086 bus with 16 address bits internally multiplexed into the data lines (AD15-AD0). The address pins A19-A16 are multiplexed with the status lines S6-S3. The address pins A22-A20 (in the 286 mode) are used to generate the control signals ALE, $\overline{DEN}$ and DT/$\overline{R}$ (in the 186 mode). The A23 pin (in the 286 mode) serves as an asynchronous ready input AREADY (in the 186 mode). As a master in the 186 mode, the 82258 offers address lines A15-A0 as latched outputs and shares all the 186/8086 support components with the processor.

8086/88 Systems

The 82258 is configured into the 8086 mode during reset (Figure 7). In this mode the 82258 supports 8086/88 in the maximum mode and uses the $\overline{RQ}/\overline{GT}$ protocol for the processor - DMA coprocessor bus exchange. The 8087 can be supported in the system without requiring the integrated bus controller, 82188. To support the 8088 system in the maximum mode, the General Mode Register is programmed for 8 bit bus width. Figure 10 shows the 82258 in an 8086 system containing the 8087. The system configuration is very similar to the 80186 system in Figure 9.

REMOTE MODE

The 82258 is configured to be in the Remote Mode (Figure 7) by programming the General Mode Register (RM bit), after putting the 82258 in the 286 mode during the reset. The 82258 has the bus timings and signals compatible to the 286 bus.

In the remote mode, the ADMA can access two 16 MByte address spaces normally called the resident space and the system space. The ADMA does not distinguish between accessing an I/O device and accessing a memory in the remote mode, so either peripheral or memory can belong to either of the two spaces.

In the remote mode, the 82258 is the sole local bus (resident bus) master and interfaces to the processor through the system bus (using a bus arbiter). Therefore, the 82258 can work in parallel with the processor. The remote mode is useful for a modular I/O subsystem.

Figure 11 shows the 82258 configured in the remote mode of operation. The peripherals interface to the 82258 on the resident bus. The resident bus components are similar to the ones used for the 286 system. Additional support components are used to interface the 82258 to a system bus e.g. the MULTIBUS. The 82258 communicates with the CPU (80286) over the system bus.

Since the 82258 is the only master of the local/resident bus, it can start the local bus cycles without any bus arbitration. For system bus accesses, a deadlock can arise if:

— The 82258 occupies the local bus to gain access to the system bus and

— The CPU (80286) occupies the system bus to gain access to the 82258 (through its local bus)

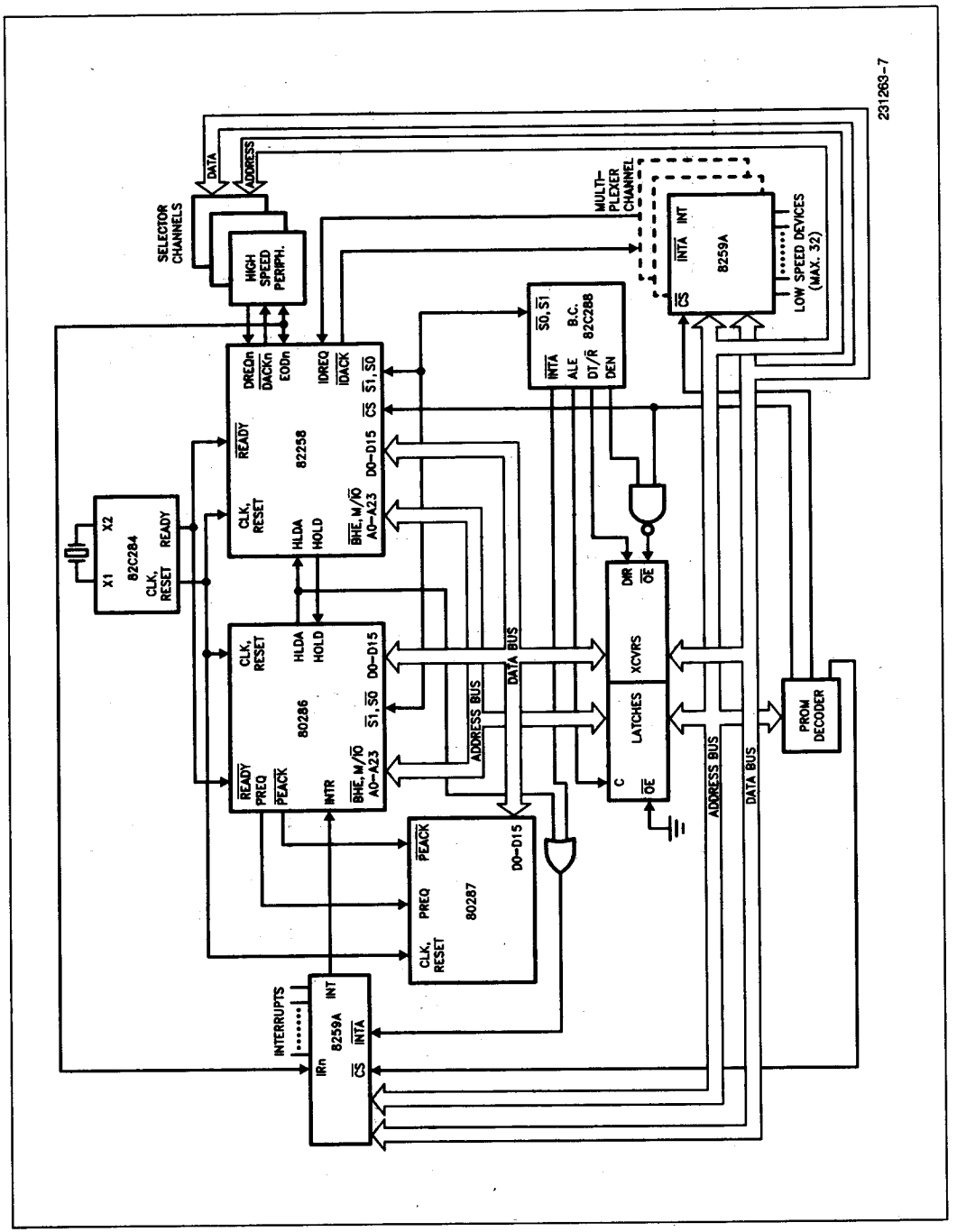


Figure 8. 80286 in an 80286 System

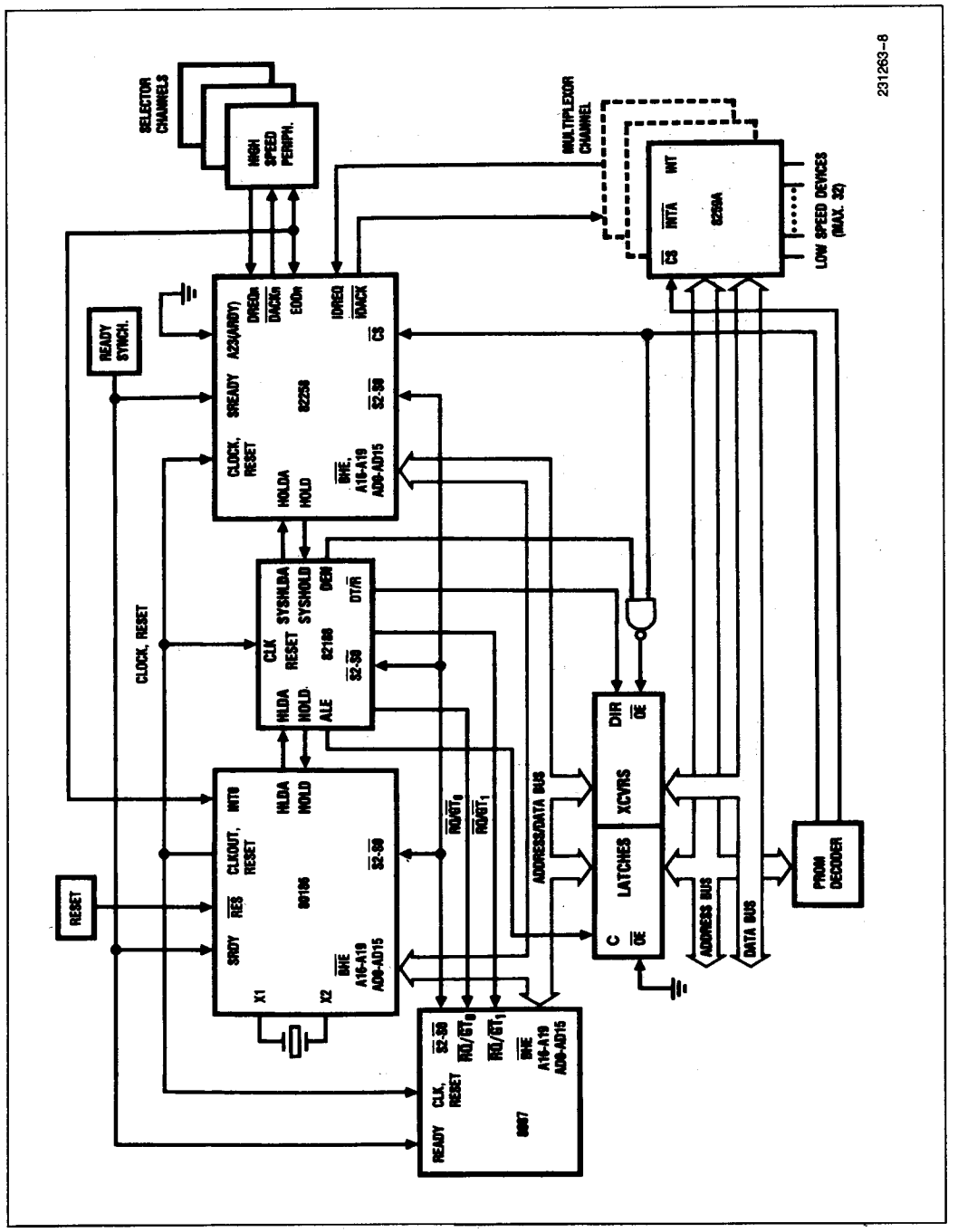


Figure 9. 82258 in an 80186 System

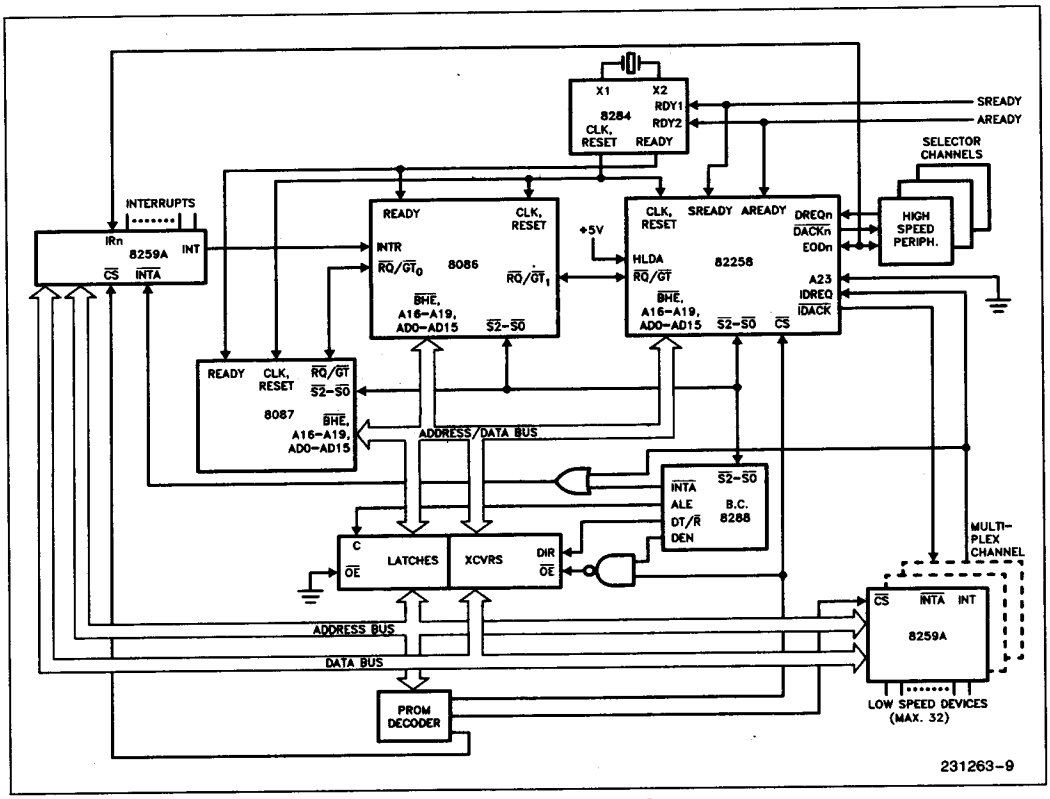


Figure 10. 82258 in an iAPX 86 System

To prevent this deadlock, for the system bus accesses the 82258 does not occupy the local bus until it has the system bus. Therefore, in the remote mode, the 82258 initiates all system bus accesses (and only these) through the HOLD/HLDA protocol. The local bus arbitration (for the CPU) is done through the $\overline{CS}$ and the BREL lines.

COMMUNICATION MECHANISMS

CPU → 82258 COMMUNICATION

Communication from the CPU to the 82258 is twofold:

— Some 82258 registers receive the main commands from the CPU, through the slave interface of the 82258. Access to the 82258 is either synchronous (using $\overline{CS}$, $\overline{S1}$, $\overline{S0}$) or asynchronous (using $\overline{CS}$, $\overline{RD}$, $\overline{WR}$; $\overline{S1}$ = $\overline{S2}$ = 1).

— Most of the data is transferred via the control space in the memory in terms of organization blocks e.g. command blocks and multiplexor table. Control space can lie in the memory space or the memory mapped I/O space (system or resident space for the remote mode) and can be dynamically changed with every start channel command.

The CPU communicates with the 82258 by depositing data in the memory and into the on-chip registers of the 82258. The CPU can access the 82258 general registers and status registers, and can start a channel by writing the proper command to the general command register (GCR). The 82258 will then read the data from the memory command block and set itself up.

Slave Interface

The slave interface of the 82258 is used by the CPU to access the 82258 internal registers. Although most of the CPU to 82258 communication is done through memory based data blocks, some direct accesses to the 82258 registers are necessary. For example, during the initialization phase the general

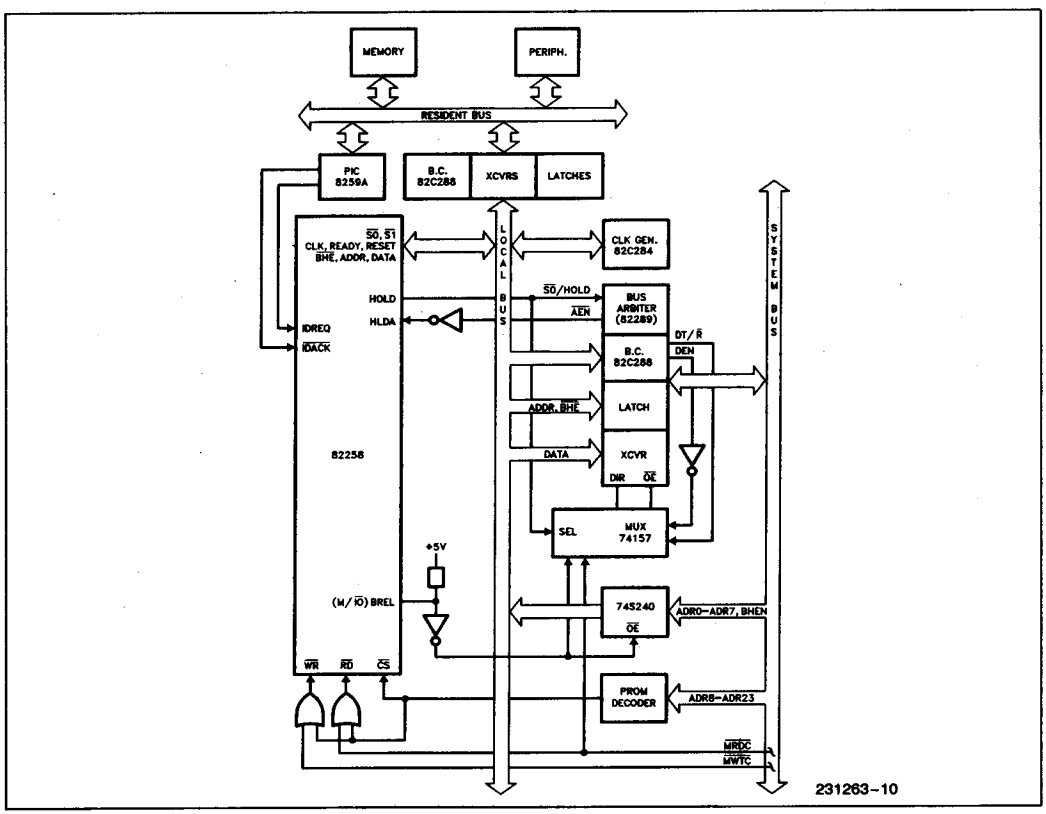


Figure 11. 82258 In Remote Mode

mode register (GMR) must be written to set up the 82258 or, to start a channel the command pointer registor (CPR) and the general command register (GCR) must be loaded. During the system debugging phase, access to the 82258 internal registers is very important.

The slave interface is enabled by the $\overline{CS}$ input and consists of the following lines:

- $\overline{S1}$, $\overline{S0}$ —Status Lines (inputs)
- $\overline{RD}$, $\overline{WR}$ —Control Lines (inputs)
- A7–A0 —Register Address (inputs)
- D15–D0 —Data Lines (inputs/outputs)-(for the 286 and the remote modes)
- AD15–AD0 —Address/Data Lines (inputs/outputs)-(for the 186 and 8086 modes)

In the 286 mode and the 186/86 mode, two types of accesses are possible:

— synchronous access through the status lines $\overline{SI}$ and $\overline{S0}$

— Asynchronous access using $\overline{RD}$ and $\overline{WR}$

The register address must be supplied on the address pins A7–A0, except for the synchronous access in the 186/86 mode. Address data lines AD7–AD0 are used for the register address information in case of a synchronous access in the 186/86 mode.

In the remote mode, a synchronous access is not possible as the 82258 has to release its local bus to enable the CPU to access its registers. On receiving an access request ($\overline{CS}$ input asserted), the 82258 releases the local bus as soon as possible and signals it by asserting the BREL line. Only then, can the CPU access the 82258 registers.

82258 → CPU COMMUNICATION

The 82258 to the CPU communication is also twofold:

— Hardware based communication, using one or more $\overline{EOD}$ lines as interrupt request lines to the CPU. The CPU can then read the status registers

(and the interrupt vector register for the multiplexor channel) and service the interrupt.

— Control space based communication: At the end of a DMA transfer, the 82258 writes the contents of the appropriate channel status register into the channel command block. Additionally, it may transfer some other information (e.g. the updated source pointer) into the command status block.

The 82258 updates its internal registers (e.g. the channel command pointer, the general status register etc.) for any CPU access.

82258 — PERIPHERAL COMMUNICATION

The DMA interface of the 82258 is used for its communication with the peripherals. It consists of three signal lines:

DREQ —DMA Request
$\overline{DACK}$ —DMA Acknowledge
$\overline{EOD}$ —End of DMA

DREQ and $\overline{DACK}$ control the externally synchronized DMA transfers. A burst of data is transferred for a continuous DMA request, as long as the request signal is active.

$\overline{EOD}$ lines, which are quasi-bidirectional, enhance the 82258—Peripheral communication link. First these can be used as inputs to the 82258 to receive an asynchronous external terminate signal to terminate a running DMA. As outputs, they can be used to interrupt the CPU and/or to signal a specific status to the peripheral (e.g. transfer aborted or, end of a block or, send/receive next block..). In addition, the $\overline{EOD}$ output of channel 2 can be used as a collective interrupt output (INTOUT) for all the DMA channels while the other three $\overline{EOD}$ lines retain their normal function.

An $\overline{EOD}$ output signal can be generated synchronous to a synchronising device at the last data transfer or, synchronous to the internal clock at the last destination cycle. An $\overline{EOD}$ can also be generated asynchronously through a Type 2 command.

BUS ARBITRATION

HOLD/HLDA Sequence

These signals are used for the bus arbitration in the 286 mode and the 186/88 (8086/88 Min.) mode. Whenever the 82258 needs the bus, it activates the HOLD signal and the processor surrenders the local bus as soon as possible by asserting HLDA. The 82258 performs the transfer and switches the HOLD to low. The processor takes the bus and switches the HLDA to low. To force the 82258 to surrender the bus, the HLDA must be set to low. The 82258 will release the bus after the currently running bus cycle or the unseparable bus cycles. Unseparable bus cycles are:

— The two IO acknowledge bus cycles for the 8259A PIC.
— Word transfers on odd boundary addresses, realised by two bus cycles where each transfer is a byte.
— Fetch of 24 bit address pointers out of the memory or restore of the pointers.
— Read- modify- write the 8259A mask registers.

The 82258 signals the surrendering of the bus by floating the bus and removing the HOLD signal. If requests for bus cycles are present, the HOLD will go active after a delay of two T-states.

$\overline{RQ}/\overline{GT}$ Sequence

$\overline{RQ}/\overline{GT}$ protocol is used for the 8086/88 (Max.) Mode. The 82258 requests the bus by sending a request pulse of one CLK period length, via the $\overline{RQ}/\overline{GT}$ signal, to the processor. The processor acknowledges it with a pulse on the same line. Then the 82258 controls the bus. When surrendering the bus, it sends a release pulse on the $\overline{RQ}/\overline{GT}$ line.

$\overline{CS}$/BREL Sequence

This is used in the remote mode along with the HOLD/HLDA signals. HOLD/HLDA are used for system bus arbitration and $\overline{CS}$/BREL for local bus arbitration (to allow the CPU to access the 82258 registers or the resident bus). The CPU asserts the $\overline{CS}$ signal to ask for the local bus and the 82258 releases the bus as soon as possible by activating BREL. After the CPU has completed its access, it should set $\overline{CS}$ high. The 82258 deactivates BREL and proceeds with its own bus cycles on the local bus.

NOTE:
When the 82258 is not in possession of the bus, all output signals are tristated except the following:
HOLD (except in the $\overline{RQ}/\overline{GT}$ protocol), $\overline{DACK0}$–$\overline{DACK3}$, $\overline{EOD0}$–$\overline{EOD3}$,
BREL (remote mode) and ALE (186 mode)

CHANNEL CONFIGURATION

The 82258 has four independently programmable DMA channels with their own register sets. All channels can be used as high speed selector channels for achieving maximum transfer rate or channel 3 can be used as a multiplexor channel to allow the 82258 to interface to a large number of I/O devices.

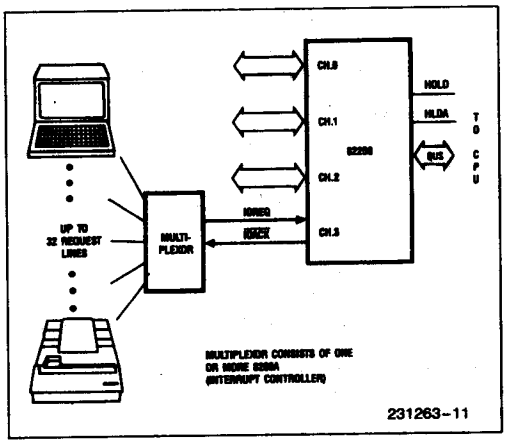


Figure 12. 82258 Channel Configuration

The selector channels support synchronised and non synchronised transfers as well as advanced features like single cycle transfer, command and data chaining. Channel switching imposes no performance penalty on the 82258. Programmable priority schemes allow flexible multiple channel processing.

MULTIPLEXOR CHANNEL

Channel 3 of the 82258 can also be operated as a multiplexor channel supporting up to 32 subchannels. External 8259As are used to arbitrate and prioritize channel requests (Figure 13). Multiplexor channel allows command chaining but data chaining is not supported.

As a multiplexor channel, channel 3 uses an external multiplexor table (MT) in the memory to store separate command pointers and, the PIC (8259A) mask register locations for each device in that channel. Each entry in the MT consists of 8 bytes; the first 4 give the command pointer for the subchannel and the second 4 the address of the mask register of the 8259A for that subchannel (Figure 14).

After an I/O request from the 8259A, the 82258 fetches an 8 bit vector (device number) from the interrupt controller (by the INT/INTA mechanism), left shifts it by three and, uses that as an offset into the multiplexor table with that entry pointing to the current subchannel command block. The 8259A should be programmed for AEOI mode.

Each subchannel can have a subchannel program or a command chain. The command chain must be terminated by a stop and mask command (as opposed to a stop command for a selector channel). Three kinds of data transfers are possible:

Byte/Word Multiplex: One byte/word is transferred per request. The source/destination pointer and the byte count fields of the command block are updated. The command pointer is not advanced until the block transfer is terminated. Maximum cumulative data transfer rate of 275K Bytes/sec can be achieved for the channel.

Single Transfer: Similar to the byte/word multplex. But, the command pointer is advanced after each transfer, thus, executing command chaining.

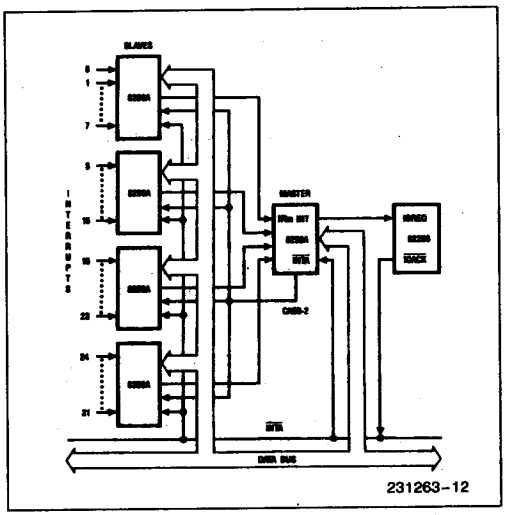


Figure 13. Multiplexor Configuration

Block Multiplex Transfer: The whole command block is executed and a block transfer made upon receiving a request. Such transfer is necessarily free running or non-synchronized and is carried out at a maximum speed of 4 MByte/sec in an 8 MHz 80286 system. After termination, the command pointer is advanced (command chaining).

The type 2 commands have the same function as for the selector channels (Table 6). A subchannel is stopped with a stop and mask command which must occur at the end of a command block chain. The 82258 generates the interrupt (INTOUT) or $\overline{EOD}$, if programmed. The 82258 automatically masks the request line on the 8259A by setting its mask bit. Thus no further requests can come from this subchannel until it is enabled by the CPU. The 82258 indicates the interrupted subchannel (vector) in the Multiplexor Channel Interrupt Vector Register (MIVR). The MIVR can be accessed by the CPU and, after reading the MIVR, the stop bit of the indicated subchannel is reset. If no channel 3 interrupt (EOD or programmed INTOUT) is enabled, the internal interrupt flag is set by the stop and mask command. Then the CPU checks the MIVR by polling, i.e., with each reference of this register, the CPU can read off the stopped subchannel vector that has the highest priority in queue until the NV (vector is not valid) bit in MIVR is set.

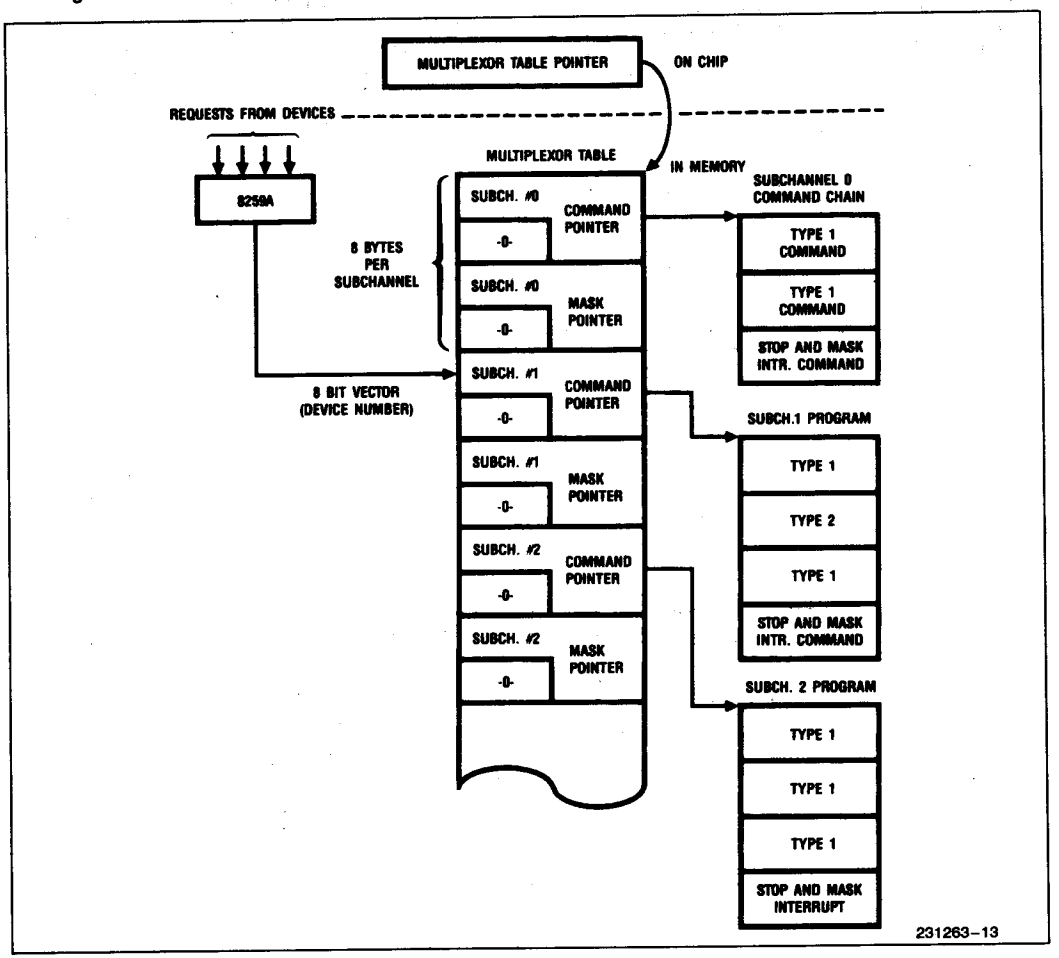


Figure 14. Multiplexor Table

DATA TRANSFER AND MANIPULATION CONTROL

SINGLE CYCLE AND TWO CYCLE TRANSFERS

The 82258 provides the flexibility to optimize the system design by allowing:

— Highest speed DMA transfers in the single cycle transfer mode. In this mode bytes or words (16 bits) are transferred directly from the source to the destination without storing the data in the 82258 registers (Figure 15). The single cycle transfer mode does not, necessarily, mean one bus cycle for transfer (though most of the transfers require either a source or a destination data cycle only). Maximum single channel or multiple channel transfer rate of 8 MByte/sec. in an 8 MHz 80286 system (4 MByte/sec in 8 MHz 80186 systems) is achieved in this mode.

In the single cycle transfer mode, while the requesting device is serviced (and addressed) using $\overline{DACK}$ signal, the pointer to the other location (memory or I/O) is issued and its bus cycle executed by the 82258. It is the duty of the I/O device to know whether the cycle is a read cycle or a write cycle and, to generate its command signal out of the bus command signals.

Single cycle transfers mode is not allowed for the multiplexor channel. All single cycle transfer are externally synchronised and "On the fly" operations are restricted (see Table 5).

— Maximum data manipulation operations in the two cycle transfer mode. The two cycle transfer mode does not, necessarily, imply two bus cycles, though most of the transfers consist of a fetch cycle from the source and a store cycle to the destination location. In this mode the source data is always stored in the 82258 registers before being sent out to the destination. Although half as

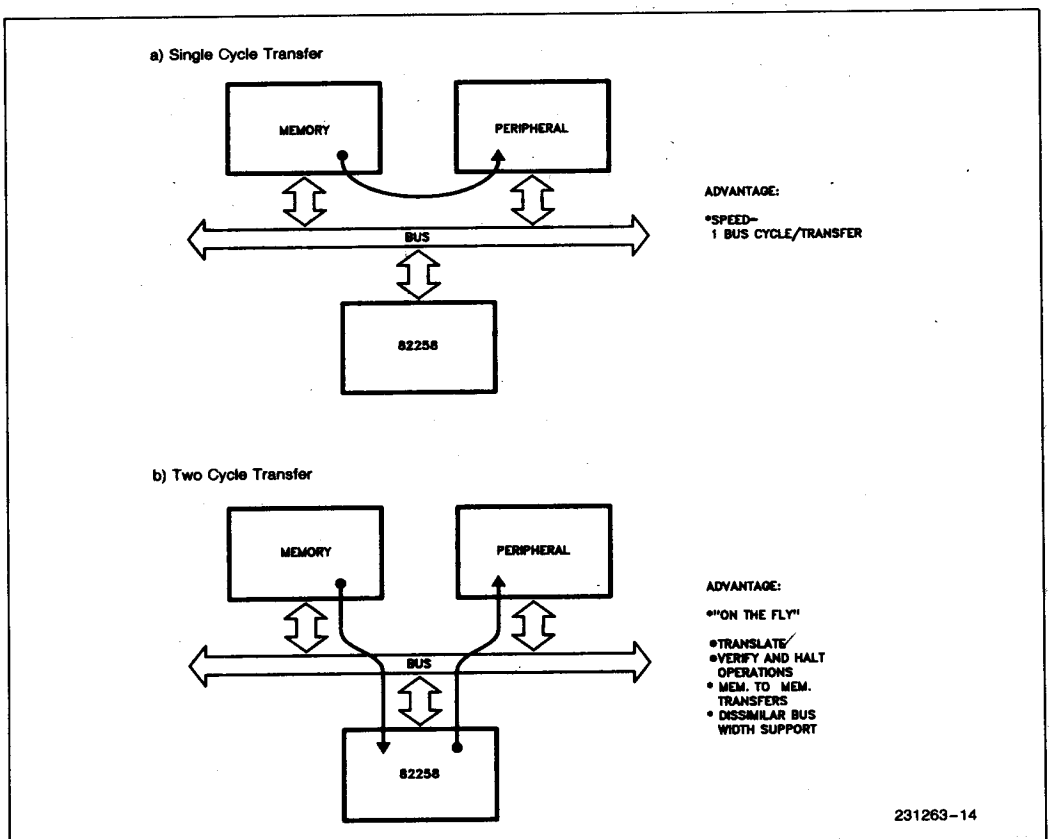


Figure 15. Single/Two Cycle Transfer

fast as the single cycle mode, a number of "On the fly" operations e.g., translation, make this mode extremely versatile. The two cycle transfer mode also allows automatic assembly and disassembly of the data, i.e., the data can be read as one 16 bit word and written as 2 bytes or vice-versa. It is useful for linking the 8 bit peripherals to a 16 bit system and vice-versa.

The two cycle transfer mode allows multiplexor channel operation and memory to memory transfers. Two special cases of two cycle data transfer are:

Read Operation or, data transfer without a destination address (the data assembly register of the 82258 itself is the destination of the source data). Compare operations on the source data are possible (e.g. to test the status of a disk controller).

Write Operation or, data transfer with no source address i.e., the source data is a byte or word constant (literal) in the data assembly register of the 82258 (loaded during the setup routine with a low word out of the source pointer field). The write operation can be used to erase a memory/peripheral data block (or peripheral register) or to load it with a certain constant.

CHANNEL COMMANDS AND COMMAND BLOCKS

The 82258 controls the data transfer, with all its modifications, through the channel command blocks. These contain the channel command word and all the initial parameters for the data transfer execution. The channel start command from the CPU causes the 82258 to read the channel command block, with all its parameters from the memory and, to load them into the internal channel registers. The channel registers that are loaded via the command blocks are: CCR, SPR, DPR, BCR, TTPR,

Table 5. Data Manipulation Operations

Operation	Single Cycle	Two Cycle	Byte/Word Multiplex*	Block Multiplex*
	Bus Cycles Required**			
Masked Compare (Byte/Word)	2	2	2	2
Verify	N/A	2	N/A	2
Verify and Halt	N/A	2	N/A	2
Verify and Save	2	F	F	F
Translate	F	3	3	3
Transfer w/o Source or Destination	F	1	1	1
	Operation Allowed			
Command Chaining	Yes	Yes	Yes	Yes
List Data Chaining	Yes	Yes	No	No
Linked List Data Chaining	Yes	Yes	No	No
Assembly/Disassembly	No	Yes	Yes	Yes
Source Synchronization	Yes	Yes	Yes	Yes
Destination Synchronization	Yes	Yes	Yes	Yes
Free Running	No	Yes	Yes	Yes

* : The multiplexor channel can only run in the two cycle transfer mode.

** : Actual number of bus cycles may vary depending upon address boundary, hardware wait state number, pointer modification direction etc.

F : Fatal error is generated.

N/A : Not Allowed

LPR/MTPR, MASKR and COMPR (see the register description for details on these registers). After examining the channel command for programming errors, the data block transfer is executed if no errors are detected. After the transfer termination, the reason for the termination is displayed in a word in the channel command block (channel status). Optionally, the last values of the source and the destination pointers and the byte count register may also be written out to the command block (constituting a status block if enabled). The CPU should not access the channel's control space while the channel is active (not stopped).

There are two basic types of channel commands:

Type 1 Channel Command—Data transfer Operation (Transfer Channel Command).

Type 2 Channel Command—Control Operation (Organizational Channel Command).

A complete channel program consists of at least one channel command block with a type one command and one type 2 command (stop).

Type 1 Channel Commands And Command Blocks

A command block always specifies a data transfer operation. The type 1 channel command defines the task to be performed by the channel (see the channel command register for details). Simple block transfer is specified by the short channel command block (Figure 16), which also allows data chaining. For more complex operations, the standard block is expanded by a command and a block extension, forming a long channel command block (Figure 16). The command block is always pointed at by the command pointer. Each channel has its own command pointer. Enabling of the status block (a bit in the channel command extension) extends the long channel command block by a status field of 12 byte length. This status field is loaded by the 82258 after the termination of the block transfer (Figure 16).

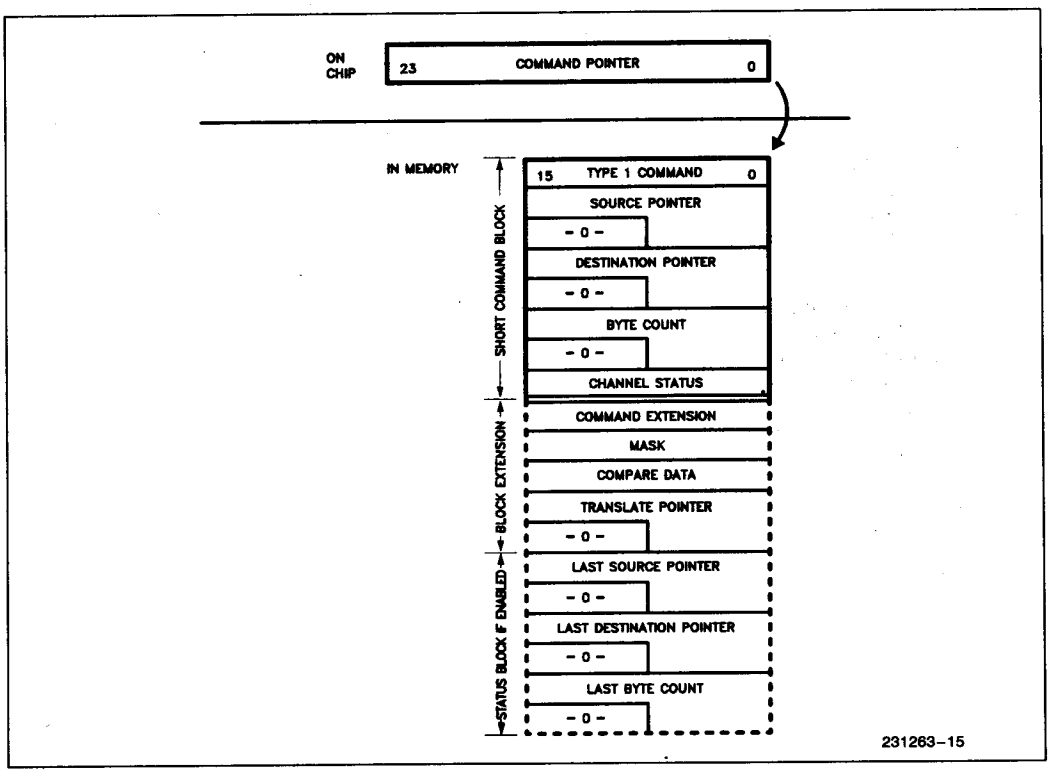


Figure 16. Type 1 Command Block

Type 2 Channel Commands and Command Blocks

The type 2 channel commands support the construction of channel programs by allowing operations such as auto-initialization, conditional chaining or program controlled interrupts. Figure 17 shows the structure of the type 2 channel command blocks.

The first word of the type 2 command block is the command and the second and the third may be an address.

Most of the type 2 commands can be executed conditionally; only exception being the unconditional stop which on the multiplexor channel functions as the Stop and Mask command. The 4 termination conditions are given in the CSR. If more than one condition is specified, the conditions are ORed. A special flag in the command word (I flag) allows to invert the channel status register bits before they are compared with the termination conditions. Table 6 gives the list of the different type 2 channel commands.

The type 2 commands can also activate a program controlled interrupt (INTOUT) and/or an $\overline{EOD}$ signal during the execution of a command (controlled by the ED and the IT flags). In the type 2 command the $\overline{EOD}$ is an asynchronous $\overline{EOD}$ (compared to the type 1 $\overline{EOD}$ which is synchronous to the last data transfer). If the ED or the IT flag is set, the signal generation is unconditional, independent of the condition code.

Table 6. Type 2 Channel Commands

Command
Relative Jump*
Absolute Jump*
Unconditional Stop
(Stop and Mask Subchannel for multiplexor channel)
Conditional Stop**

* : Both conditional or unconditional

** : The 82258 does not check if a selector channel only type 2 command is used on the multiplexor channel, but its execution will lead to erroneous channel processing.

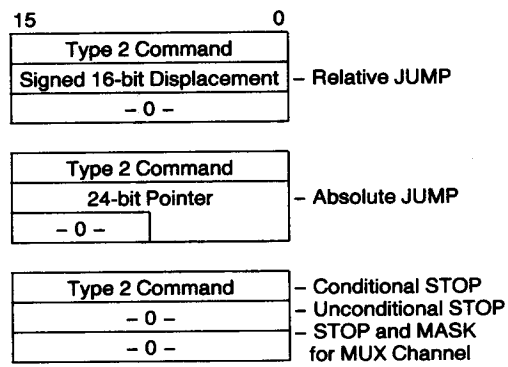


Figure 17. Type 2 Command Block

COMMAND AND DATA CHAINING

Command Chaining

The 82258 allows chaining of the command blocks in the memory, for any channel, for sequential execution. Figures 16 and 17 show channel command blocks and Figure 18 shows the examples of command chaining. The 82258 gets the address of the command block from its on-chip command pointer (initialized by the CPU) and starts executing. When it comes to the end of one command, it automatically starts to fetch and execute the next command block until a stop command is found. Conditional and unconditional STOP and JUMP commands allow complex sequences of DMAs to be performed.

Command chaining allows the 82258 to do CPU indepedent I/O processing, thus, saving valuable CPU time.

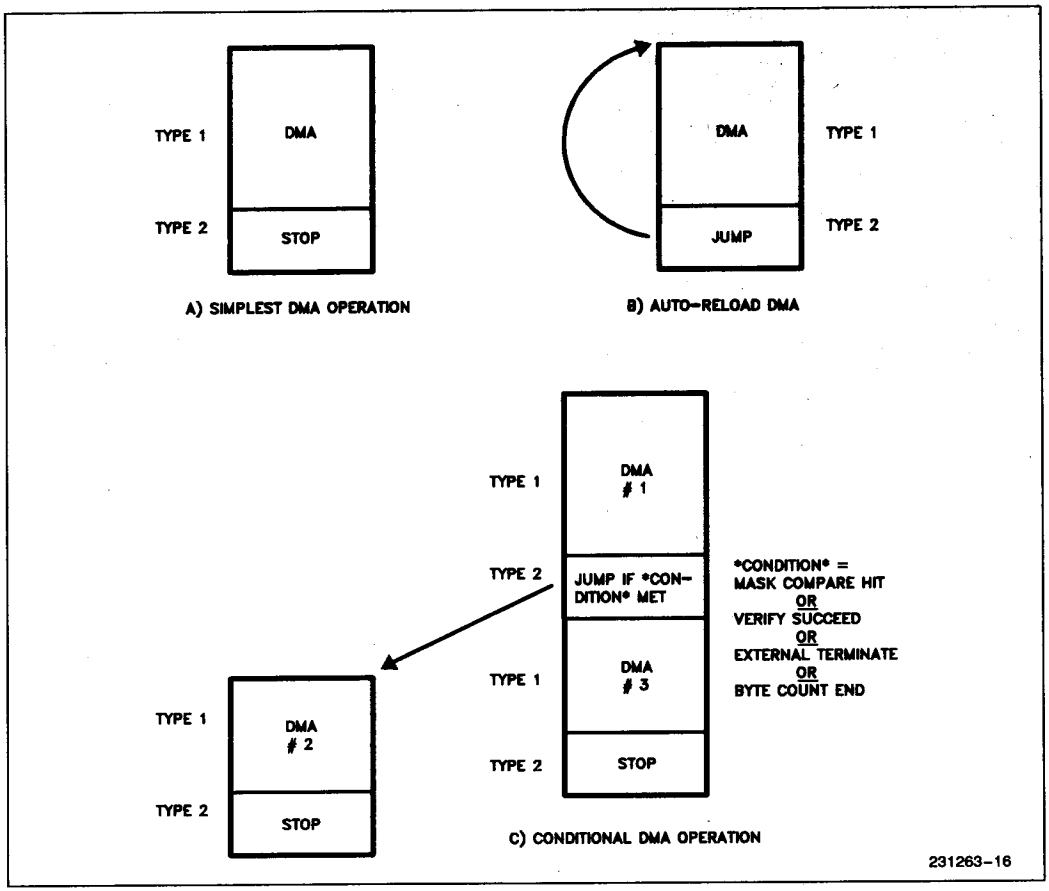


Figure 18. Command Chaining

Data Chaining

Data chaining allows gathering and scattering of data blocks. The 82258 permits automatic, dynamic linking of the data blocks scattered in the memory. Each data block in a chain can be up to 64K bytes. Two types of data chaining are allowed:

List Chaining: The chained data block descriptors are contiguous in a block which forms the data chain list (Figure 19). End of the chain is indicated by making the byte count field zero in the data chain list. List chaining is fast (1 microsecond between completion of one block transfer and going to the next element in the list, in an 8 MHz 80286 system) but not very flexible.

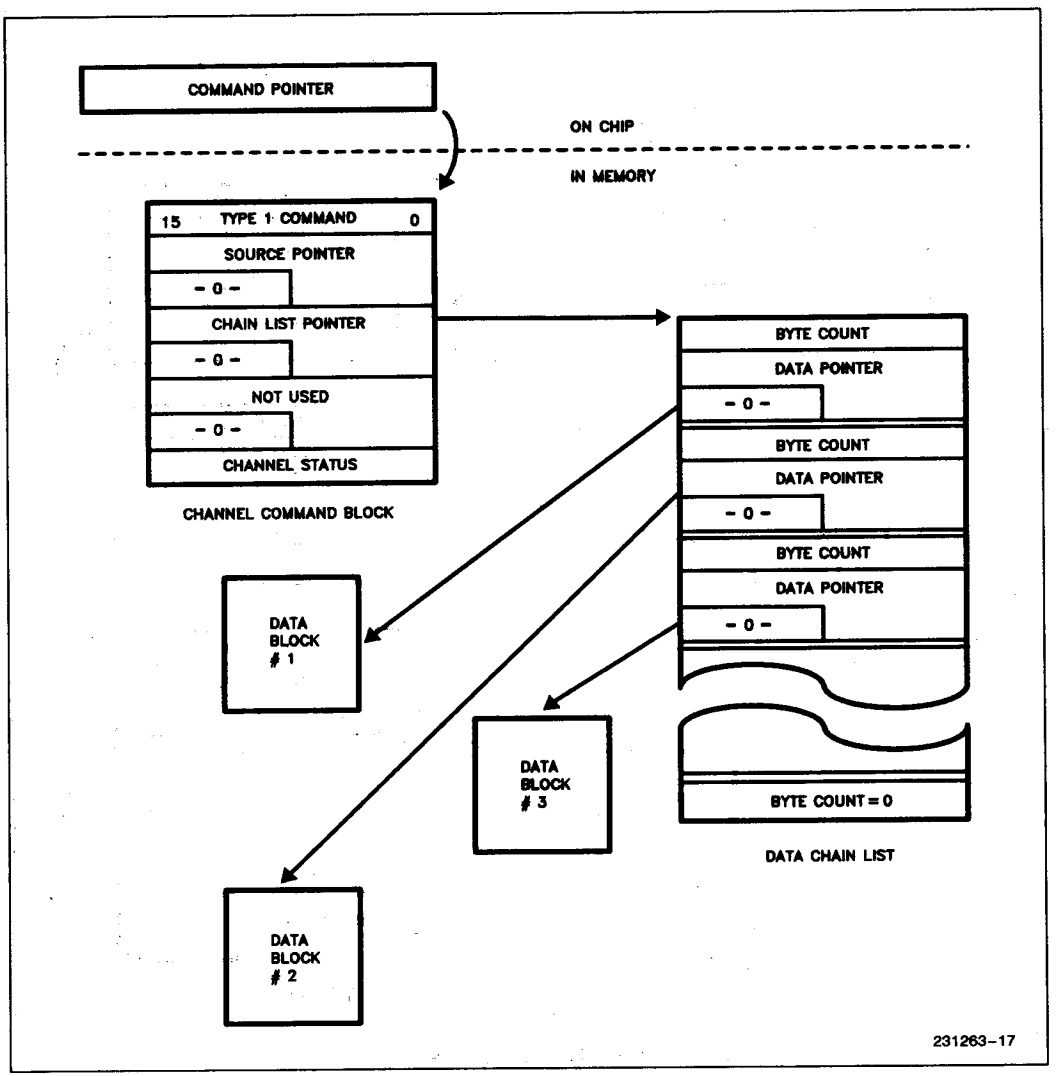


Figure 19. Destination List Chaining of Data

Linked List Chaining: Each list element which describes a particular data block (location and length) also holds a pointer to the next list element to be processed (Figure 20). End of the chain is indicated by making the byte count field zero in the linked list.

Linked list chaining is slower than the list chaining but the data blocks can be included, removed or, their sequence altered dynamically, through the link pointer manipulation by the CPU.

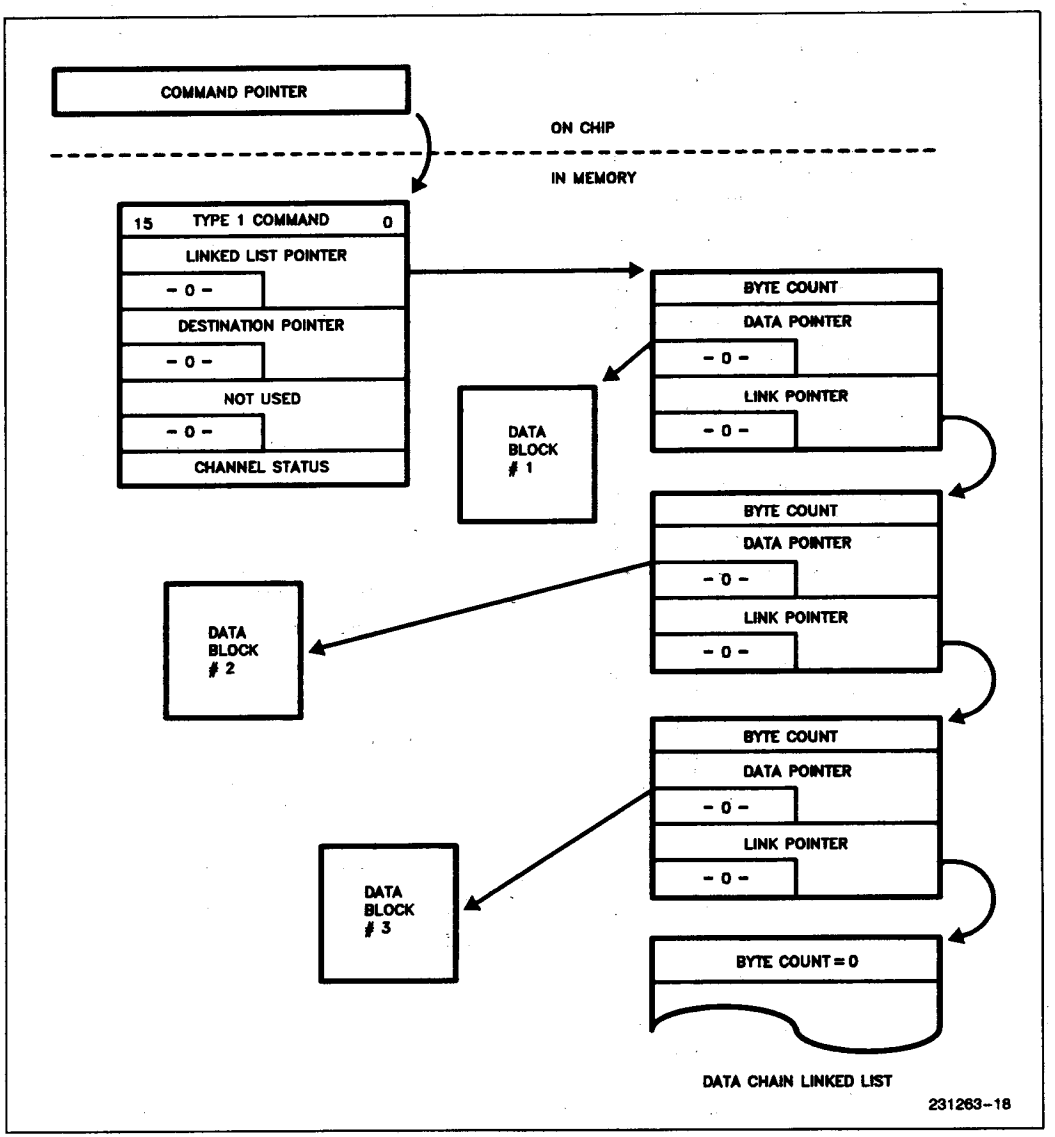


Figure 20. Source Linked List Chaining of Data

"ON THE FLY" OPERATIONS

The 82258 allows various data manipulation operations during the transfer:

Mask and Compare

Allows comparison of each byte, word or bit field (masking) in source data with some given pattern. Data transfer can be terminated on a match or a mismatch depending upon the program. This is possible both for the single and the two cycle transfer modes but, the transfer rate is halved in the single cycle mode.

Verify

No data transfer is performed, but the complete source data block is compared with a given data block. The data conversion can be terminated on mismatch (Verify and Halt). Supported only for the two cycle transfer mode.

Verify and Save

The data block is transferred from source to destination and in parallel compared with a given data block. The data transfer is not stopped on a mismatch. This operation is supported only for the single cycle transfer.

Translate

The source data (bytes) is translated with the aid of a translation table (Figure 21) before being sent to the destination. Translation is supported for the two cycle transfer mode only. If the destination is 16 bits, the two translated source bytes are assembled in the DAR before the destination cycle is executed.

Various 'on the fly' operations can be combined to allow the 82258 to perform versatile DMA operations.

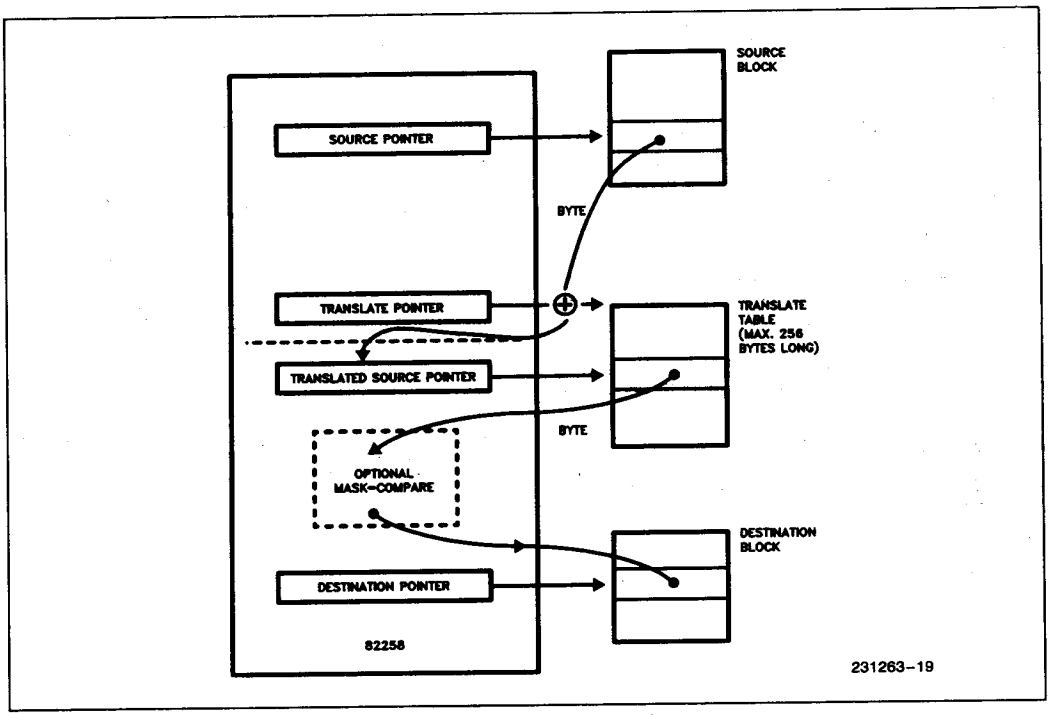


Figure 21. Translate Operation

PRIORITY CONTROL

The 82258 controls concurrent processing of its different channels (and subchannels) and, the internal and the external requests through a flexible priority scheme.

The PRI bits in the GMR are used to select the priority scheme which can be fixed or variable or a combination of the two (see the GMR description for the details). The unseparable bus cycles (e.g., 24 bit pointers) are not affected by the priority rotation. External 8259As determine the priorities for the multiplexed subchannels.

The processing of the internal or the external requests is controlled by a fully nested priority system including all four channels. Since more than one request can compete for the same channel, the requests are also prioritised in relation to their types as follows (in descending order of priority).

— Channel Stop (Command from the CPU out of the GCR)
— External asynchronous termination request (through $\overline{EOD}$)
— Internal continue request on previously interrupted sequence
— Start or stop subchannel or multiplexor channel
— Internal (without synchronization) or external (with synchronization) data service request or IO request for the multiplexor channel
— Channel wait (idle)

Data chaining and internal termination belong to the data service request processing, command chaining belongs to the termination processing.

Slave operations, where the 82258 is addressed by the CPU, have the highest priority of all the activities.

ADDRESSABILITY

The 82258 has two address spaces like the 80286, the 80186/188 and the 8086/88 processors:

— Memory space
— I/O space

Both the spaces are 16 MByte large for the 286/remote mode and 1 Mbyte for the 186/8086 mode. All types of transfers are possible:

— Memory/Memory
— I/O / I/O
— Memory/I/O
— I/O / Memory

Either of the memory or the peripheral can lie in either of the two spaces. Each space can be independently 8 bit or 16 bit wide. All possible Even-Odd boundary address combinations are supported for the data transfer from source (8 bit or 16 bit) to destination (8 bit or 16 bit) in the two cycle transfer mode. The source and the destination pointers can be incremented, decremented or not modified at all (INC/DEC bits of type 1 channel command in the CCR) after the corresponding data bus cycle. The 82258 does not indicate or check an 'address out of range' condition. Address overflow and underflow during a block transfer results in an address wrap around. Maximum length of the data block can be 16 MBytes in an 80286 system. In the 186/86 mode the maximum byte count is (1M-1). This is not checked by the 82258.

SYNCHRONIZATION OF DATA TRANSFER

The 82258 allows both the external synchronization of a DMA transfer (from a source or a destination device) or a free running DMA (internally synchronized).

The external synchronization allows control of input/output operations in the cycle of the peripheral device, hence occupying the bus only when the peripheral is able to receive or transmit data.

Free running DMA (no external synchronization) is used for the memory to memory transfers, during a continuous DMA request or, in the block multiplex subchannel after the channel start. It is not supported for the single cycle transfer mode.

286 PROTECTION

The 82258 needs special consideration to operate in an 80286 system in the protected mode. The 82258 works only with the real addresses but it supports a protected mode 80286 system if the following conditions are fulfilled:

— The 286 kernel software must check all the protection rules during the set up routine for the 82258 and perform the limit checks for the block transfers. This is supported by the 80286 instructions e.g. VERR (verify Read Access), VERW (verify Write Access), LSL (load Segment Limit).

— The 286 kernel has to translate the logical addresses into the physical addresses.

— All the 82258 registers should be memory mapped and access to them should be allowed only for a 286 kernel routine (task isolation).

Normally an I/O utility routine is provided by the operating system to service the 82258. No direct user access should be allowed to the 82258 from the lower privilege levels. The real addresses can be generated only by using the 286 protection mechanism and are so checked against any protection violation.

82258 REGISTER MODEL

The 82258 has three sets of registers (Figure 22):

General Registers
Channel Registers
Multiplexor Channel Registers

All registers can be read or written into by the CPU but, most are accessed only for the test purposes. The CPU loads some registers (e.g. General Mode Register) during the initialization after the reset, and others during the invocation of a channel (General Command Register). Some of the channel registers are programmed or read by the CPU but most of them are loaded by the 82258 itself during the setup routine after a channel start. All accessible registers can be accessed bytewise or wordwise by the CPU.

Figure 23 gives a layout of the registers. Note that all registers lie on even addresses.

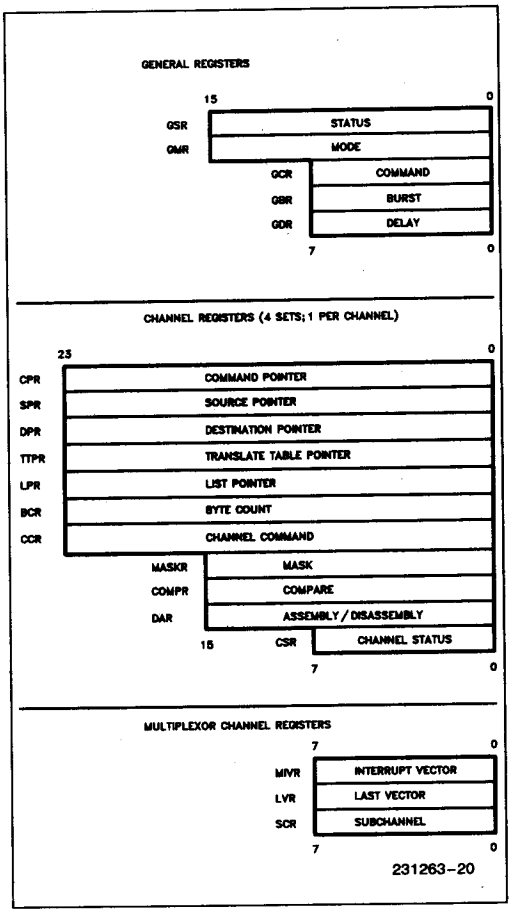


Figure 22. 82258 Register Set

Address Bits 5-0 (hexadecimal)	Address Bits 7,6				Address Bits 5-0 (binary)
	00	01	10	11	
0	GCR	RESERVED	RESERVED	RESERVED	000000
2	SCR				000010
4	GSR				000100
6	RESERVED				000110
8	GMR				001000
A	GBR				001010
C	GDR				001100
E	RESERVED				001110
10	CSR0	CSR1	CSR2	CSR3	010000
12	DAR0	DAR1	DAR2	DAR3	010010
14	MASKR0	MASKR1	MASKR2	MASKR3	010100
16	COMPR0	COMPR1	COMPR2	COMPR3	010110
18	RESERVED	RESERVED	RESERVED	MIVR	011000
1A				LVR	011010
1C				RESERVED	011100
1E					011110
20	CPRL0	CPRL1	CPRL2	CPRL3	100000
22	CPRH0	CPRH1	CPRH2	CPRH3	100010
24	SPRL0	SPRL1	SPRL2	SPRL3	100100
26	SPRH0	SPRH1	SPRH2	SPRH3	100110
28	DPRL0	DPRL1	DPRL2	DPRL3	101000
2A	DPRH0	DPRH1	DPRH2	DPRH3	101010
2C	TTPRL0	TTPRL1	TTPRL2	TTPRL3	101100
2E	TTPRH0	TTPRH1	TTPRH2	TTPRH3	101110
30	LPRL0	LPRL1	LPRL2	LPRL3/MTPRL	110000
32	LPRH0	LPRH1	LPRH2	LPRH3/MTPRH	110010
34	RESERVED	RESERVED	RESERVED	RESERVED	110100
36	RESERVED	RESERVED	RESERVED	RESERVED	110110
38	BCRL0	BCRL1	BCRL2	BCRL3	111000
3A	BCRH0	BCRH1	BCRH2	BCRH3	111010
3C	CCRL0	CCRL1	CCRL2	CCRL3	111100
3E	CCRH0	CCRH1	CCRH2	CCRH3	111110

GCR = General Command Register
SCR = Subchannel Register
GSR = General Status Register
GMR = General Mode Register
GBR = General Burst Register
GDR = General Delay Register
CSR = Channel Status Register
DAR = Data Assembly Register
MASKR = Mask Register
COMPR = Compare Register
L = Low Word
H = High Byte

MIVR = Multiplexor Interrupt Vector Register
LVR = Last Vector Register
CPR = Command Pointer Register
SPR = Source Pointer Register
DPR = Destination Pointer Register
TTPR = Translate Table Pointer Register
LPR = List Pointer Register
MTPR = Multiplexor Table Pointer Register
BCR = Byte Count Register
CCR = Channel Command Register
0, 1, 2, 3 = Channel Number

Figure 23. Layout of Register Addresses

GENERAL REGISTERS

These registers are common to all the channels.

General Mode Register (GMR)

This is the first register to be programmed after the reset since it describes the 82258 environment. Here the system wide parameters are specified. The 16 bit register is loaded bytewise with the low byte being programmed first.

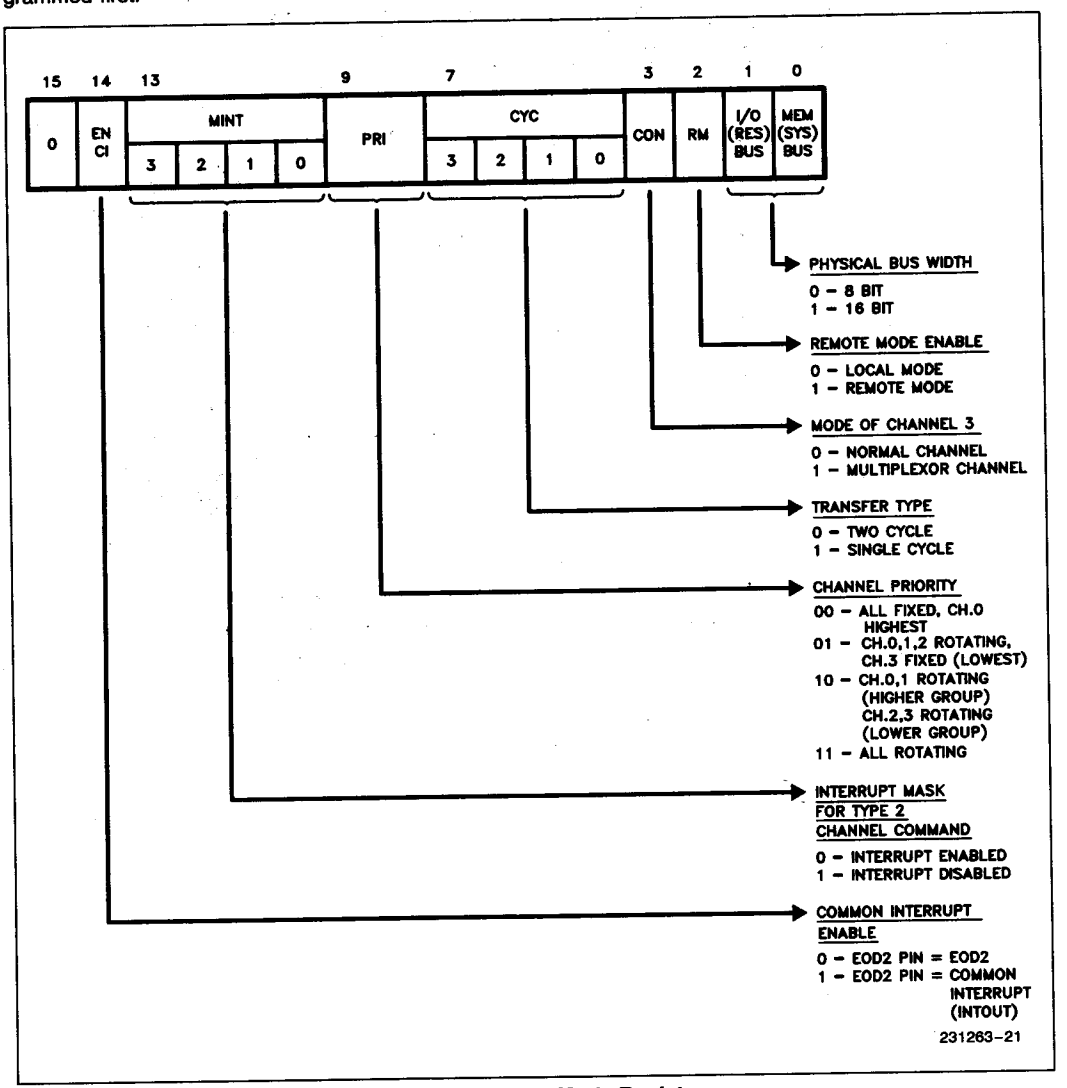


Figure 24. General Mode Register

General Status Register (GSR)

This register provides the status information for all the channels. It also shows which channels have interrupts pending and, where the channel control space lies. It is a 16 bit register.

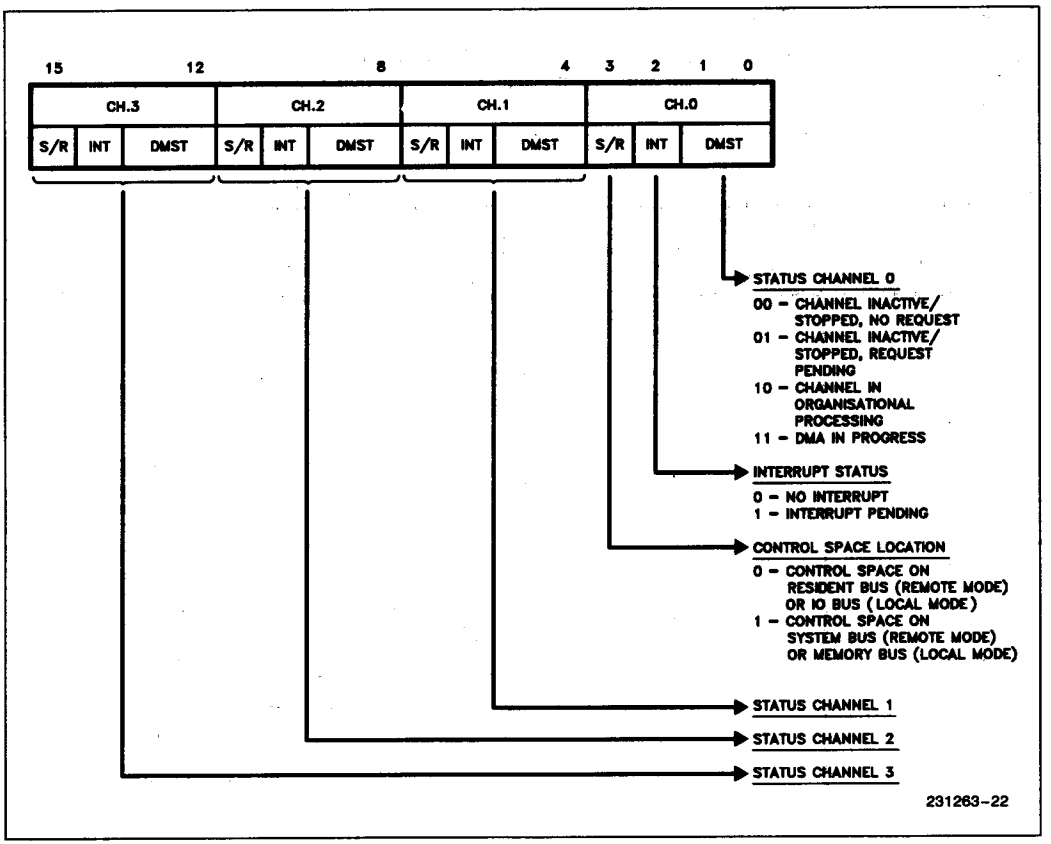


Figure 25. General Status Register

General Command Register (GCR)

GCR is an 8 bit register directly loaded by the CPU to start or stop a channel. The START command also defines the control space assignment. The pending interrupt from any channel is also cleared through the GCR. Any combination of channels can be addressed simultaneously. To start/stop a multiplexor subchannel, the subchannel number must be first loaded in the Subchannel Register (SCR). The Halt/single step command is useful for the system debugging.

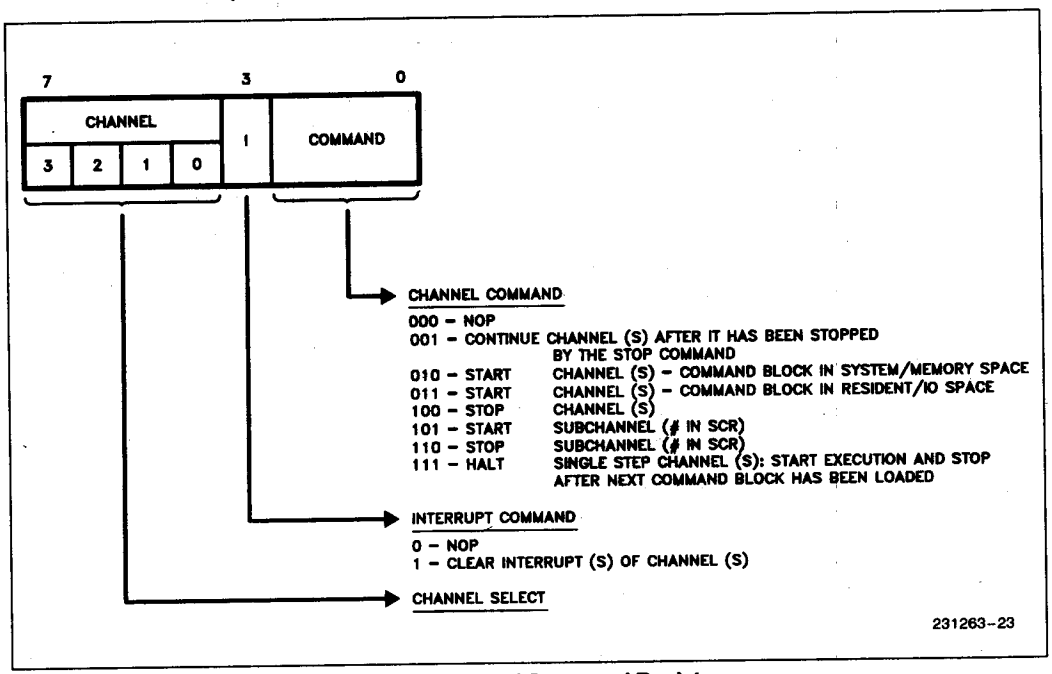


Figure 26. General Command Register

General Burst Register (GBR)

This 8 bit register determines the maximum number of contiguous bus cycles that can be requested by the 82258. GBR = 0 means unlimited contiguous bus cycles for the 82258. The GBR must be directly loaded by the CPU.

General Delay Register (GDR)

GDR is an 8 bit register which determines the minimum number of clocks between the 82258 burst accesses. GDR = 0 means no minimum delay between the HOLD request.

Burst/Delay Algorithm

Both the GBR and the GDR do their actual counting through their respective counters the GBC and the GDC. For the burst and delay counters, the following rules apply:

— Whenever the 82258 controls a bus cycle the burst counter is decremented by one but not beyond zero.

— Whenever the 82258, in the local mode, does not have the bus, the delay counter is decremented by one: every second T-state in the 286 mode or, every fourth T-state in the 186 mode.

— Whenever the delay counter is zero, the burst and the delay counters are loaded from the burst and the delay registers.

— If the burst counter is zero (and no exception occurs), the 82258 releases the bus and the delay counter counts until it is zero. Then both counters are loaded from their corresponding registers and the 82258 can again request the bus by activating HOLD signal. Unseparable bus cycles are the exception to this rule. Counting of the burst is not prevented but surrendering of the bus is.

— In the remote mode the burst and the delay are relevant only for the system bus cycles. The GBC is only decremented while the 82258 performs the system bus cycles and the GDC decrements when the 82258 does not control the system bus (idling or the resident bus cycles).

CHANNEL REGISTERS

Each of the four 82258 channels has these registers. All the channel registers are loaded by the 82258 from the memory except the Command Pointer (CPR) [Multiplexor Table Pointer (MTPR) & Subchannel Register (SCR) for the channel 3 in the multiplexor mode]. The initial contents of the registers are specified, by the CPU in the command blocks in the memory.

Command Pointer Register (CPR)

This 24 bit register contains the physical address of the command block. It must be loaded by the CPU before starting the channel. For the channel 3 in the multiplexor mode, the CPR is loaded by the 82258 from the multiplexor table (MT) in the memory.

Source Pointer Register (SPR)

SPR is 24 bits and contains the physical address of the source (memory or I/O, system or resident space) in a DMA transfer. In the single cycle transfer mode, it contains the only address pointer (source or destination).

Destination Pointer Register (DPR)

DPR contains the physical address of the destination (memory or I/O, system or resident space) in a DMA transfer. During Verify operations it contains the verify pointer (pointer to compare the data block). For the single cycle transfer mode, it is only used for the verify and save operation. It is a 24 bit register.

Translate Table Pointer Register (TTPR)

This 24 bit register is used to reference the translate table in the memory when the translate function is enabled in the channel command register extension (CCRX).

List Pointer Register (LPR)

LPR is used for data chaining (list and linked list) operation. It is a 24 bit register and points to the list element. In the multiplexor mode for the channel 3, it is used as the Multiplexor Table Pointer Register (MTPR). (Multiplexor mode does not support data chaining).

Byte Count Register (BCR)

BCR is a 24 bit register and contains the byte count for the DMA transfer.

Channel Command Register (CCR)

CCR specifies the type of DMA transfer or the type of internal operation. The channel commands are contained in a channel command block. The 82258 has two types of channel commands:

— Type 1 for data movement

— Type 2 for command chaining control

The channel command register has three configurations:

— Short Type 1 command: SYN field NE. 00 and ECX = 0. Upper 8 bits, i.e., Channel Command Register Extension (CCRX field), are not valid.

— Long Type 1 command: SYN field. NE. 00 and ECX = 1. All 24 bits are valid.

— Type 2 command: SYN field = 00, Upper 8 bits (CCRX field) are not valid.

Figure 27 shows CCR for Type 1 command and Figure 28 has the CCRX (Channel Command Register Extension). Figure 29 shows CCR for type 2 command.

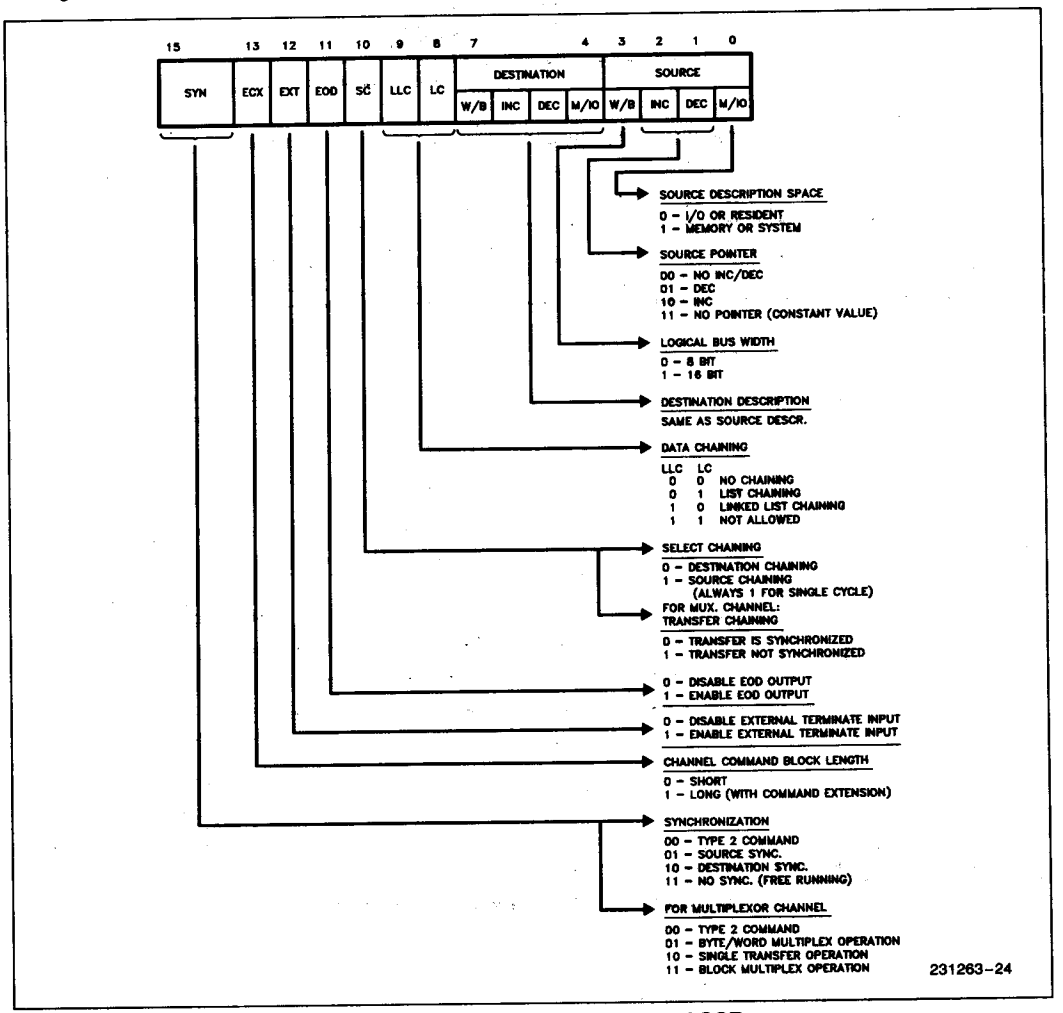


Figure 27. Type 1 Channel Command CCR

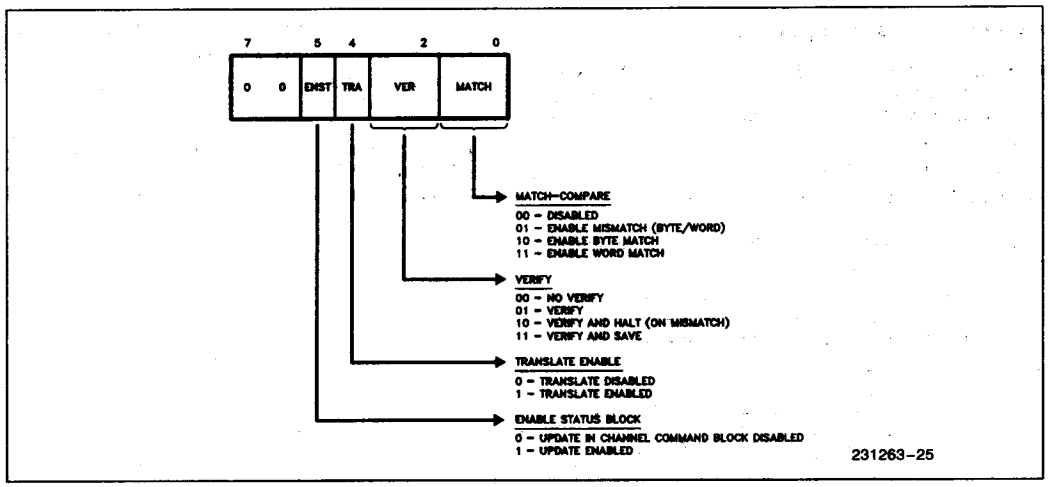
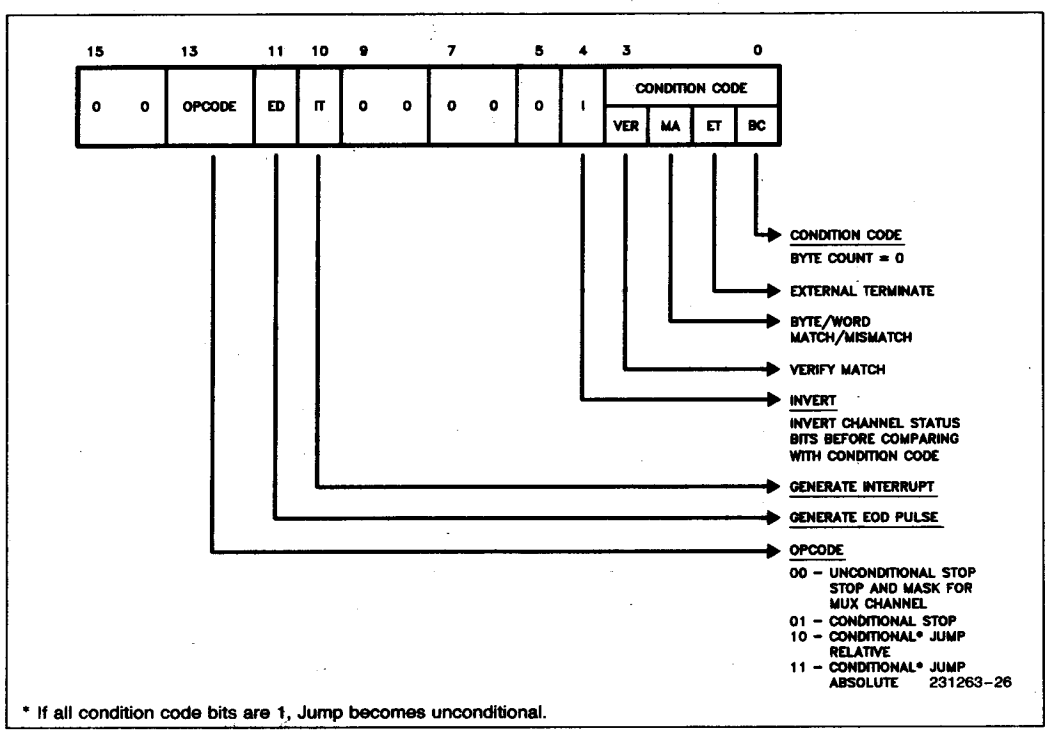


Figure 28. Channel Command Register Extension CCRX



* If all condition code bits are 1, Jump becomes unconditional.

Figure 29. Type 2 Channel Command CCR

Mask Register (MASKR) and Compare Register (COMPR)

Both of these registers are 16 bit and are used during the match/mismatch operation. For comparison with the transferred data, only those bit positions in the Compare Register which are not masked with 1's in the Mask register are considered. These two registers together allow byte, word or bit level comparisons. MASKR is also used during the verify operations. MASKR and COMPR each should contain two identical bytes for Byte Match/Mismatch operations.

Channel Status Register (CSR)

CSR, an 8 bit register, reflects the status of the channel. The least significant half byte is the termination condition and the most significant half byte indicates fatal error, busy state and halted state.

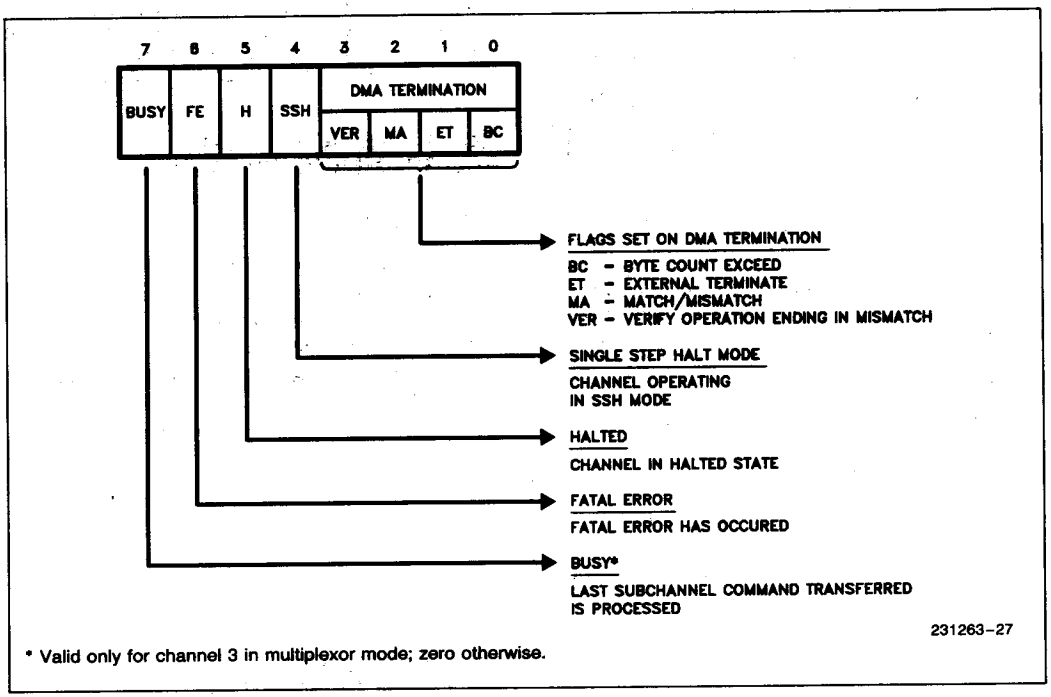


Figure 30. Channel Status Register

Data Assembly Register (DAR)

This 16 bit register is used for automatic assembly/disassembly of data.

Mutiplexor Channel Registers

These registers are valid only for channel 3, when used as a multiplexor channel.

Multiplexor Table Pointer (MTPR)

This register is used to reference the multiplexor table in the memory when channel 3 is programmed as a multiplexor channel. Since data chaining is not allowed for the multiplexor channel, the List Pointer Register (LPR) is used as the MTPR. MTPR is 24 bit and must be loaded by the CPU.

Multiplexor Interrupt Vector Register (MIVR)

This 8 bit register is used by the CPU to determine which channels are stopped. The vectors of the stopped subchannels are output in the priority order (0 has the highest priority) upon each reference of this register, until the NV bit is set. A maximum of 32 vectors can be distinguished.

Last Vector Register (LVR)

LVR gives the last vector read by the 82258 (from the 8259A). In case of a fatal error stop of channel 3, LVR determines the guilty subchannel. LVR is an 8 bit register.

Subchannel Register (SCR)

This register gives the 8 bit subchannel number for the general commands START/STOP Subchannel. It must be loaded by the CPU before a subchannel command is written into the GCR. MIVR limits the number of subchannels supported to 32 (5 bits).

82258 OPERATION AND PROGRAMMING OVERVIEW

INITIAL STATE

Upon activation of the RESET signal:

— all channels are disabled (by clearing the DMA status bits in the General Status Register)

— all bus activities are stopped

— all tristate signals are tristated and the others enter the inactive state

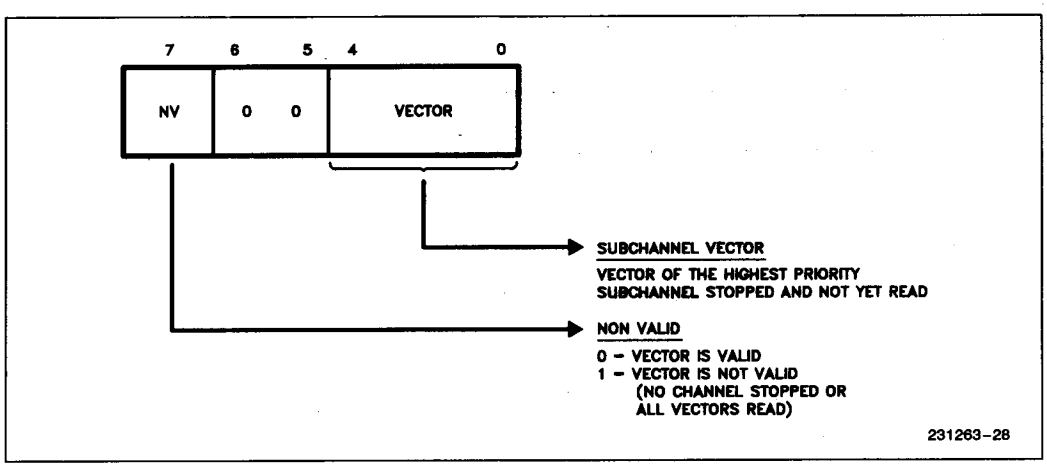


Figure 31. Multiplexor Interrupt Vector Register

After the RESET signal becomes inactive, the 82258 state gets defined:

— it is in the 186 mode if A23 pin was low at the falling edge of RESET; otherwise it is in the 286 mode

— it is in the 8086 max (Request/Grant) mode if the 186 mode is detected and HLDA pin was high at the falling edge of RESET; otherwise it is in the 186/8086 Min. (HOLD/HLDA) mode.

— The contents of the 82258 registers are as follows:

- GMR: All bits are zero
- GBR: Zero value
- GDR: Zero value
- GSR: — DMST bits for channels: 0X (Stopped)
  — INT for all channels: 0 (no interrupt pending)
  — S/R = 0 (I/O or resident space)
- All Channel Status Registers (CSR): Zero Values
- MIVR: NV = 1 (Vector not valid)
  — Vector is all 1, rest zero
  — All stop bits in matrix are reset
- All other registers (GCR, LVR, SCR, CPRn, SPRn, DPRn, TTPRn, LPRn, BCRn, CCRn, COMPRn, MASKRn, MTPR) are undefined

INITIALIZATION AND CHANNEL INVOCATION

After RESET, the 82258 has to be initialized by the CPU. The General Mode Register (GMR) should be loaded first in the 16 bit systems; the lower byte of the GMR (which gives main configuration information) in the 8 bit systems.

SYSBUS (MEMBUS) bit of the GMR determines the physical bus width of the CPU-82258 communication. All register write and read operations are executed:

— Bytewise on the lower half of the data bus (D7–D0), if SYSBUS (MEMBUS) = 0

— wordwise on D15–D0 if SYSBUS (MEMBUS) = 1. Byte transfers are also possible here with the bytes being transferred on that half of the data bus which is addressed by the least significant bit of the register address.

Internally the 82258 uses $\overline{BHE}$ and A0 to detect the effective transfer width of the 82258—CPU communications. After the GMR, the General Burst Register (GBR) and the General Delay Register (GDR) should be programmed, if needed (Initial state = 0 for both), by the CPU.

Before a channel is invoked, the control space in the memory and the channel registers in the 82258 have to be initialized:

Selector Channel Start

Following conditions should be met:

— channel program in the control space

— if data chaining enabled, the chaining list or the linked lists in the control space

— if translate enabled, the translate table in the control space

— load the CPR with the start address of the channel program

Multiplexor Channel Start

For the multiplexor channel operation, the following is essential:

— the multiplexor table MT in the control space with the subchannel command pointer and the mask register pointer of the associated 8259A for each subchannel

— initialization of the 8259A's mask registers by masking off all the request inputs. In the remote mode, this can also be done by the data transfer operation on the selector channel (or by stop subchannel commands)

— load MTPR with the base address of the multiplexor table (MT)

For the subchannel start

— the subchannel program should be in the control space

— if translate enabled, the translate table should be in the control space

— the subchannel command pointer should be in the multiplexor table

— read the multiplexor channel status register CSR3. Write a new subchannel number into the SCR only if BUSY bit = 0.

In case of a normal channel start, the last CPU operation is to write the general command into the GCR. Then the start will be processed by the 82258 according to the requested channel's priority, with the highest priority being processed first. If the addressed channel is already active, the start command is ignored. If I = 1 in GCR, the INT bit(s) of the indicated channel(s) will be erased in the GSR.

COMMAND EXECUTION

Selector Channel: The command bits in the GCR give the commands available to a selector channel. Execution of the continue and the start commands is prioritized; the stop commands are executed immediately. The stop command forces the DMA status bit (DMST) in the GSR to channel inactive (stopped) without any additional routine. The continue command works directly with internal stored register parameters and continues a previously stopped channel operation. The start commands define the location of the control space and initiate the set up routine. The halt command has multiple functions:

— It forces the channel into the single step and halt mode, indicated by the SSH bit in the CSR

— If the channel is running, it will be halted after the completion of the current command block execution; the halted data is shown by the H bit of the CSR; the DMST bits of the GSR are not changed

— If the channel is halted (or stopped) the halt/single step command starts the channel, and the channel will again be halted after the completion of the next command block execution (type 1 or 2)

The single step and halt mode is finished by a start or a continue command. After a channel start, first the general status reflected in the GSR is changed into 'DMA in organizational processing'. GSR also indicates the location of the control space (S/R bit). After the prioritization of the start command, the channel's set up routine is executed.

After the set up routine execution, all the transfer parameters are accessible in the 82258 internal registers. The SYN bits in the CCR decide:

— if the channel activity is continued by an immediate start of the data transfer (i.e., free running mode or an internal data transfer service request)

— or the channel is waiting for a DMA request i.e., external synchronization mode.

Multiplexor Channel: On the multiplexor channel, there are two cases:

a. The whole channel has to be treated by a general command

b. Only the addressed subchannel has to be treated by a general command

a. In case of the whole channel, the commands are the same as the selector channel commands. Execution of the continue and the stop (stops whole channel) is the same. The channel 3 start command has only two functions:

— specify whether the system/memory or the resident/IO control space has to be used on the multiplexor channel (S/R bit in GSR)

— change of the general status of the channel 3 (DMST bits in GSR) into "Channel started but idling" thus, enabling the IOREQs and the Subchannel commands.

The general channel command "Halt/Single Step" has a slightly different interpretation for the multiplexor channel. While the selector channel can only be halted during the chaining of the command blocks, the multiplexor channel in the single step/halt mode will also be halted when it takes the idle state. In that case, a new halt/single step command will only be executed if an IOREQ or a subchannel start/stop command is pending.

b. With the start subchannel command, the 82258 unmasks the corresponding bit in the 8259A mask register for the addressed subchannel, thus enabling the subchannel. The BUSY bit in the CSR is set indicating the state: "subchannel command pending". After prioritization, the subchannel routine is executed. When an I/O request is received on the subchannel, the command pointer is fetched from the MT and the channel's set up routine is executed. After the reset of the BUSY bit, a new start/stop subchannel command can be accepted by the multiplexor channel.

Only distinction between the stop subchannel command and the start subchannel command is the handling of the mask bit in the 8259A. For the STOP command, the vector specific mask bit is set by the 82258. As the start command, the stop command has also to be prioritized before execution.

For the multiplexor channel the following rules are observed:

— Before any IOREQ can be processed, the whole channel 3 has to be started and the channel 3 must be in the idle state

— In any state a subchannel command can be accepted and transferred into the state "subchannel command pending"

— A pending subchannel command can be processed only in the idle state

— In the idle state, a subchannel command has a higher priority than an IOREQ

— In case of a fatal error stop of a subchannel, the whole channel 3 is stopped. LVR identifies the guilty subchannel. To stop (mask) this subchannel, the CPU at first has to issue a START CH3 command and then stop the affected subchannel.

TERMINATION CONDITIONS

The 82258 distinguishes the following conditions for termination of a block transfer:

— byte count is zero and the data chaining not enabled; a standard termination condition

— data chaining enabled and the new fetched byte count is zero

— external termination via the channel's $\overline{EOD}$ line if enabled by the EXT bit in the CCR

— match/mismatch during the masked byte or word compare, as specified and enabled in the command extension CCRX

— mismatch during a verify & halt operation, as specified and enabled in the command extension CCRX

— The CPU loading the GCR with a stop command, though the channel is not really terminated.

INTERRUPT CONTROL

The 82258 has four programmable $\overline{EOD}$ pins (one for each channel) for the CPU interruption and for communication with the system environment. As inputs, the $\overline{EOD}$ pins are used for external termination, enabled by the EXT bit of the type 1 channel command in the CCR. When used as output, the $\overline{EOD}$ pins provide two basic functions:

$\overline{EOD}$ (end of DMA), a channel specific active LOW pulse signal of 2 T-states length, always enabled by the software. With a type 1 channel command, $\overline{EOD}$s, if enabled, are synchronous and always controlled by the byte count. If data chaining is enabled, type 1 $\overline{EOD}$s should not be used for interrupts since multiple $\overline{EOD}$s (with every exceeding byte count) are issued. With a type 2 command, the $\overline{EOD}$, if enabled (ED = 1 in the CCR), is an asynchronous signal generated after a command execution.

INTOUT (interrupt output) is a hardware generated (error detection) or a software enabled static active HIGH signal on the $\overline{EOD}$2 pin, if programmed (ENCI = 1 in the GMR). The channel generating the INTOUT is indicated by the INT bit in the GSR. Hardware generated interrupt occurs in case of a fatal error (INTOUT issued if not masked by the MINT bit in the GMR). Type 2 channel command allows software generated INTOUT if programmed (IT = 1 in the CCR and not masked by the MINT bit in the GMR). A channel's INT bit in the GSR is activated independent of the MINT (in GMR). INTOUT remains active until all INT bits in the CSR are reset by the CPU with the general command CLEAR INTERRUPT.

Multiplexor Channel Interrupts

Interrupts from the multiplexor channel belong to a certain subchannel. For program controlled interrupts, the status and the context information cannot be fetched from the internal 82258 registers (since the multiplexor channel is not stopped). Hence, the CPU can only investigate the interrupt via the MIVR register. After the MIVR read from the CPU, the valid bit and matrix stop bit (the vector of which was indicated in the MIVR) are erased. For multiple stop conditions in the stop matrix, the stopped subchannels get their vectors in the MIVR in the priority order (highest for vector zero). The MIVR is activated independent of the programming of $\overline{EOD}$ or INTOUT. Therefore, the CPU can sample the MIVR in a polling mode when neither $\overline{EOD}$ nor INTOUT is used. With the interrupt vector out of the MIVR, the CPU finds the related command pointer (in MT) which points to the last executed channel command (stop and mask). For status information of last block transfer, the CPU has to find the last type 1 command block in the channel program. Programmable intermediate interrupt messages should not be used on the multiplexor subchannels (MIVR is activated only for the stopped subchannel).

For hardware generated INTOUT the whole channel 3 is stopped with the LVR indicating the last (guilty) vector. After the error investigation the CPU should start the channel 3 and then stop the affected subchannel.

FAULT DETECTION

On detecting a fatal error, the 82258 does the following:

immediately stops the affected channel
sets error bit in the channel's status register
sets channel specific INT bit in the GSR
sends interrupt if not masked (in GMR)

For error investigation, the CPU should:

read GSR (what channel?, channel stopped?)
read CSR (error?)
read CPR and investigate the channel command (type 1 command)
read LVR for multiplexor channel, if affected (what subchannel?)

The 82258 recognizes only type 1 command errors. Other error types are defaulted into non-fatal errors and not identified. The FE bit in the CSR indicates the fatal errors.

Fatal Errors: Fatal errors are detected during the decoding of a type 1 channel command with the GMR. Six conditions are used for detection and the allowed six combinations of them lead to six different transfer executions (Table 7). All other combinations of the six conditions generate a fatal error.

Table 7. Fatal Error Detection

Valid Combination	Conditions Decoded						Operation Performed
	Single Cycle	No Dst. Ptr.	No Src. Ptr.	Verify & Save	Trans-late	Sync. Error	
1	False	False	False	False	False	—	Two Cycle DMA
2	False	False	False	False	True	—	Translate
3	False	False	True	False	False	False	No Source Ptr. DMA
4	False	True	False	False	False	False	No Dest. Ptr. DMA
5	True	False	False	False	False	False	Single Cyc. DMA
6	True	False	False	True	False	False	Verify & Save

The synchronization error is predecoded and activated in the following cases:

Single cycle combined with free running

No source pointer mode combined with the source synchronization on a selector channel

No destination pointer combined with the destination synchronization on a selector channel

Non Fatal Errors and Undetected Fatal Errors

A non fatal error is not indicated in the channel status register. It is only defaulted. Channel processing is not interrupted. Following are some examples of non fatal errors and the undetected fatal errors:

Fault	Action
Remote mode + 186 mode	RM not inhibited but read/write pins are also used as outputs
Both list chaining and linked list chaining enabled	Linked list data chaining executed
Start/Stop subchannel and BUSY active	New command overwrites old command (Fatal Error)
Data chaining enabled on the multiplexor channel	MTPR is overwritten with the list pointer (Fatal Error)

TRANSFER RATES

Selector Channel

Table 8 illustrates the different transfer rates (in MBytes/sec) for the 286 mode of operation. These transfer rates are not affected by switching channels and are halved for both 186 and 86 modes of operation.

Table 8. Cummulative Selector Channel Transfer Rates (8 MHz 286 System)

Transfer	Single Cycle	Two Cycle
Word → Word	8	4
Word → Byte	not possible	2.66
Byte → Word	not possible	2.66
Byte → Byte	4	2
Byte → Byte w/ Translate	not possible	800 KBytes

Multiplexor Channel

The transfer rates on the multiplexor channel are different from the selector channel and depend on the mode of operation and the size of the command block.

Table 9. Cummulative Multiplexor Channel Transfer Rates

Mode	Command Block	Word Transfers	Byte Transfers
Byte/ Word	short	275 KBytes/sec	138 KBytes/sec
Multiplex	long	240 KBytes/sec	120 KBytes/sec
Block Multiplex	short	4 MBytes/sec	2 MBytes/sec
	long	4 MBytes/sec	2 MBytes/sec

Data Chaining

The transfer rate for data chaining depends on the block length of each chained data block, the number of blocks in the chain and also the type of chaining that is being done. See the section on data chaining latencies.

LATENCIES

The latency calculations do not take into account set up, hold and output delay times which are specified in the A.C. Characteristics section. These should be added to get the final latency figures. All timings are in units of T-states (125 ns in an 8 MHz system). If bus cycles are involved then the following abbreviations are used:

T = time for one bus transfer

W = wait time during bus cycles for a slow device

In case of various influences affecting the timing, the most typical case is mentioned in the table and explained in notes.

Assumptions:

1. The channel for which latencies are calculated currently has the highest priority and will not be blocked by other still higher priority requests.
2. In remote mode delays due to CPU accesses to the 82258 are not taken into account for latencies.
3. All control space accesses are on a 16 bit bus and command blocks and data chain lists are addressed on even boundaries.
4. Organizational and other unsynchronized transfers (e.g. prefetch) have been completed before the processing of DREQ starts.

DMA Request Processing:

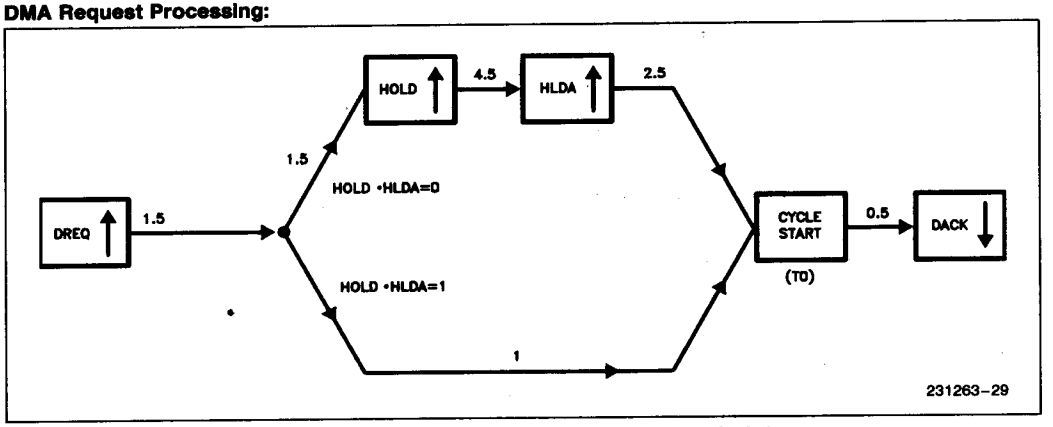


Figure 32. DREQ to DACK Latency in Local Mode*

Table 10. DREQ to DACK in Local Mode*

	Minimum	Typical	Maximum
DREQ to HOLD	2.5	3	3 + W (1) (2)
HOLD to HLDA	1	4.5	(3)
HLDA to CYCLE START	1.5	2.5	2.5
DREQ to CYCLE START (without bus arbitration)	2	2.5	4 + W (1)
CYCLE START to DACK	0.5	0.5	0.5

Notes are indicated in parenthesis

*All timings are in units of T-states (125 ns in an 8 MHz system). If bus cycles are involved then the following abbreviations are used:
T = Time for one bus transfer
W = Wait time during bus cycles for a slow device

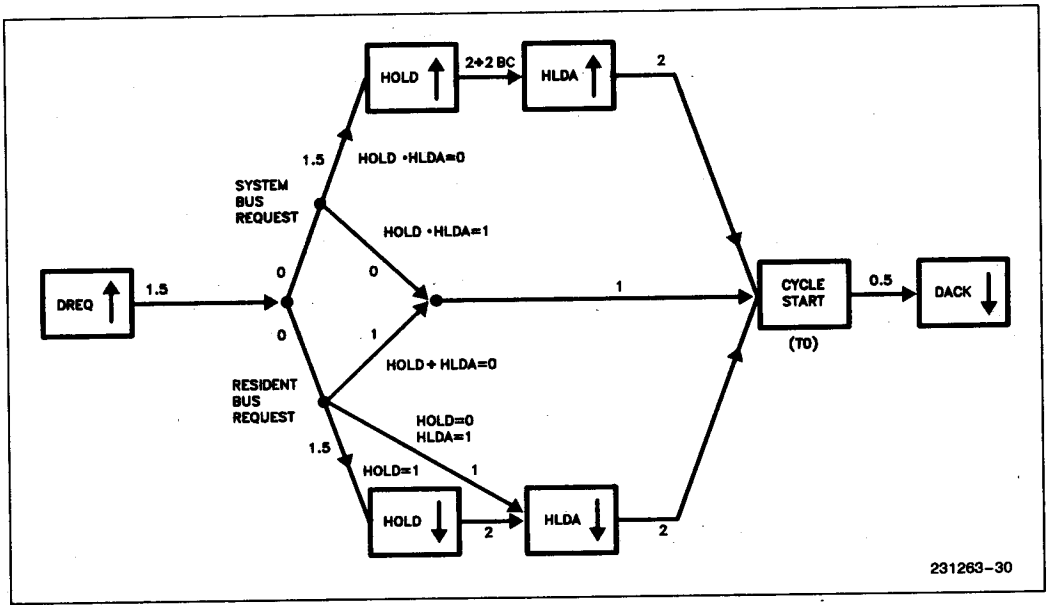


Figure 33. DREQ to DACK Latency in Remote Mode*

Table 11. DREQ to DACK in Remote Mode*

	Minimum	Typical	Maximum
DREQ to HOLDset	2.5	3	3 + W (1) (2)
HOLDset to HLDAset	2 BC	2 + 2 BC	(4)
HOLDAset TO CYCLE START	1.5	2	2.5
DREQ to HOLDreset	1.5	3	5.5 + W (1)
HOLDreset to HLDAreset	1	2	2
HLDAreset to CYCLE START	1.5	2	2.5
DREQ to CYCLE START (without bus change)	2	3.5	5 + W (1)
CYCLE START to DACK	0.5	0.5	0.5

Notes:
(1) Single bus cycle running: 1 + W
unseparable bus cycles running:
—word access at odd addresses (and pointer transfers): 3 + 2W
—IOACK cycle (only multiplexor channel): 7 + 2W
(2) General Burst Counter = 0: 2 × GDR
HLDA = 1, HOLD = 0: Wait for HLDA = 0
HLDA lost: 2
(3) 16 + 15W (from the 286 manual, assumed repeat and lock prefix not combined)
(4) Bus arbitration + currently running bus transfers.
BC = Multibus clock cycle.
* All timings are in units of T-states (125 ns in an 8 MHz system).
If bus cycles are involved then the following abbreviations are used:
T = Time for one bus transfer
W = Wait time during bus cycles for a slow device

General Command Processing:*

	Minimum	Typical	Maximum
WRITE to Set Up	6.5	8	9.5
	+ HOLD/HOLDA sequence		

At this point the start command is ready for the start of the channel set up routine

Set Up Processing:*

Standard command block	: 7T + 4
additional for long command block	: 5T
additional for list data chaining	: 1T + 2
additional for linked list data chaining	: 3T + 2

Type 1 Command Processing:*

Chaining : same as the set up processing

Termination :

store CSR and calculate next command pointer	: 1T + 6
store status block (if programmed)	: 6T

Type 2 Command Processing:*

Standard :

CCR load	: 1T
CCR decode and execution	: 2T + 2
additional for jump	: 4

START/STOP Subchannel:*

(see General Command Processing for set up)

Execution : 4T + 6

Multiplexor Channel:*

(see General Command Processing for set up)

IOREQ to IOACK : identical to DREQ to DACK timing

First IOACK to second IOACK	: 1T + 2
Second IOACK to vector in LVR	: 1T + 2
Calculate MT address and read command pointer into CPR	: 2T + 4
Data transfer	: 2T + 2
Restore pointers	: 4T + 4
Restore byte count	: 2T

Data Chaining:*

Latencies in data chaining occur when transfers are changed between data blocks.

List Chaining	: 3T + 6
Linked List Chaining	: 5T + 6

* All timings are in units of T-states (125 ns in an 8 MHz system).

If bus cycles are involved then the following abbreviations are used:

T = Time for one bus transfer
W = Wait time during bus cycles for a slow device

Absolute Maximum Ratings

Ambient Temperature Under Bias	0°C to 55°C
Case Temperature	0°C to 85°C
Storage Temperature	−65°C to +150°C
Voltage on Any Pin with Respect to Ground	−1.0V to +7V
Power Dissipation	3.6 Watt

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. Characteristics V_{CC} = 5V ±5%; T_A = 0°C to +55°C, or T_{CASE} = 0°C to +85°C

Symbol	Parameter	Limit Values		Units	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage (except CLK)	−0.5	+0.8	V	—
V_{IH}	Input High Voltage (except CLK)	2.0	V_{CC} + 0.5	V	—
V_{OL}	Output Low Voltage	—	0.45	V	I_{OL} = 3.00 mA
V_{OH}	Output High Voltage	2.4	—	V	I_{OH} = −400 µA
I_{CC}	Power Supply Current	—	475 370	mA	T_A = 0°C, T_A = 55° all outputs open
I_{LI}	Input Leakage Current	—	±10	µA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current	—	−200	µA	0.45V ≤ V_{OUT} = V_{CC}
	$\overline{S0}$, $\overline{S1}$, $\overline{S2}$, $\overline{BHE}$, $\overline{RD}$, $\overline{WR}$, M/$\overline{IO}$				
	HOLD (RQ/GT mode), $\overline{EOD}$		−1.5	mA	
	other pins		± 10	µA	
V_{CL}	Clock Input Low Voltage	−0.5	+0.6	V	—
V_{CH}	Clock Input High Voltage	3.8	V_{CC}+1.0	V	—
C_{IN}	Capacitance of Inputs (except CLK)	—	10	pF	fc = 1 MHz
C_O	Capacitance of I/O or Outputs	—	20	pF	fc = 1 MHz
CCLK	Capacitance of CLK Input	—	12	pF	fc = 1 MHz

A.C. Characteristics $V_{CC} = 5V \pm 5\%$; $T_A = 0°C$ to $+55°C$, or $T_{CASE} = 0°C$ to $+85°C$

AC timings are referenced to 0.8V and 2.0V points of signals as illustrated in datasheet waveforms, unless otherwise noted.

Sym	Parameter	6 MHz		8 MHz		Unit	Test Conditions
		Min	Max	Min	Max		
1	CLK Cycle Period (286 Mode)	83	250	62	250	ns	
2	CLK Low Time (286 Mode)	20	225	15	230	ns	at 1.0V
3	CLK High Time (286 Mode)	25	230	20	235	ns	at 3.6V
4	Output Valid Delay	1 –	80	1 –	60	ns	CL = 125 pF
5	Output Valid Delay	1 –	55	1 –	40	ns	CL = 125 pF
6	Data Setup Time	15		10		ns	
6a	Address Input Setup (186 Mode)	20		15		ns	
7	Data Hold Time	8		5		ns	
8	READY Setup Time	50		38		ns	
9	READY Hold Time	35		25		ns	
10	Input Setup Time	25		20		ns	
10a	Status Setup Time (186 Mode)	30		30		ns	
11	Input Hold Time	25		20		ns	
11a	BHE Hold Time (186 Mode)	15		10		ns	
12	Address Setup Time	3		2		ns	
13	Data Valid Delay	0	60	0	50	ns	
14	Data Float Delay	8	80	5	60	ns	
15	Chip Select Setup	30		20		ns	
16	Command Length	320		290		ns	
17	Data Setup Time	185		165		ns	
18	Address Setup Time	30		20		ns	
19	Command Inactive	320		290		ns	
19a	Access Time		420		380	ns	
20	CLK Period (186 Mode)	166	500	125	500	ns	
21	CLK Low Time (186 Mode)	76		55		ns	
22	CLK High Time (186 Mode)	76		55		ns	
23	CLK Rise Time (186 Mode)		15		15	ns	
24	CLK Fall Time (186 Mode)		15		15	ns	
25	READY Active Setup Time	20		20		ns	
26	READY Hold Time	10		10		ns	
26a	SREADY Hold Time (186 Mode)	15		15		ns	
27	READY Inactive Setup Time	35		35		ns	
28	Control Reset Setup Time	25		20		ns	
29	Control Reset Hold Time	0		0		ns	
30	Address/Data Valid Delay	10	55	10	50	ns	
31	Status Delay	10	75	10	55	ns	
32	Address/Data Float Delay	10	50	10	50	ns	
33	DT/$\overline{R}$ Delay (186 Mode)	10	76	10	55	ns	
34	DEN Delay (186 Mode)	10	80	10	60	ns	

A.C. MEASUREMENT POINT DESCRIPTION

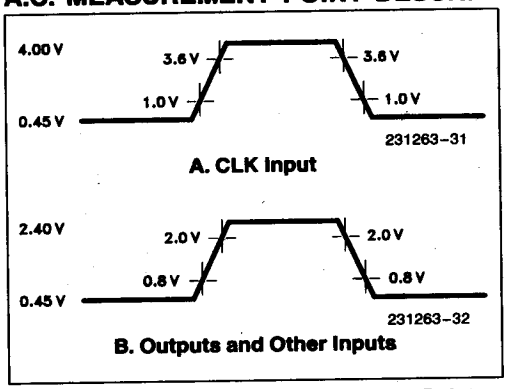


Figure 33a. AC Drive and Measurement Points

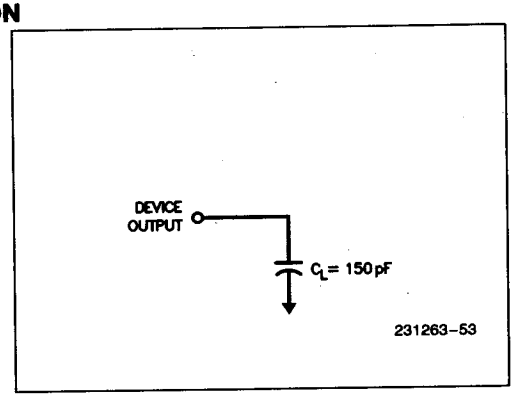


Figure 33b. AC Test Loading on Outputs

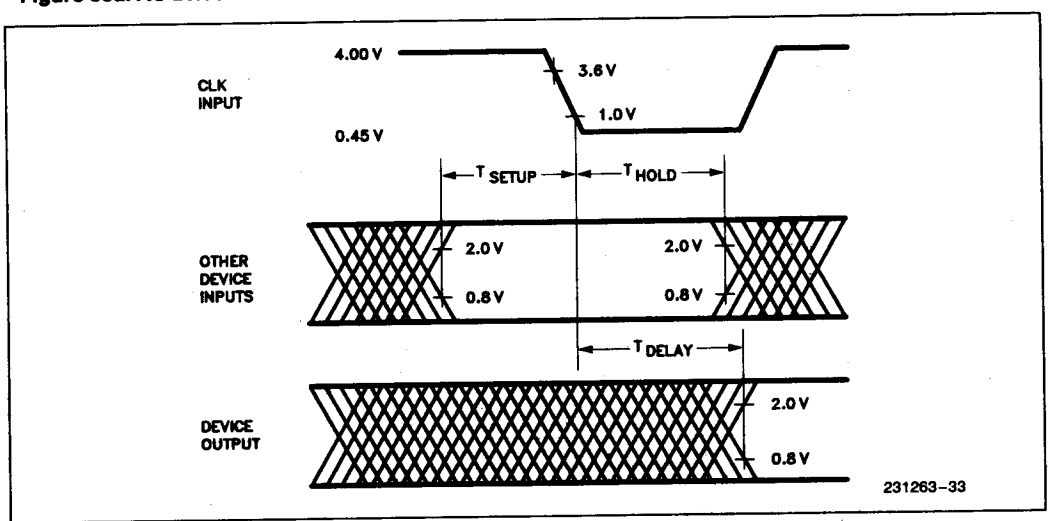


Figure 34. AC Setup, Hold and Delay Time Measurement - General

BUS CYCLE T-STATES:

The bus cycles are subdivided into T-states which are interpreted differently depending on whether the 82258 is in the 286 mode or the 186 mode.

286 Mode T-states: Each T-state is two clock cycles long and starts in the middle of a processor cycle and ends in the middle of the succeeding processor cycle.

TI: [The bus is idle] This state will occur if the 82258 cannot start the next bus cycle.

T0: [A new bus cycle is beginning] When the address and status of a new bus cycle is to be sent as output, this state is used.

T1: [A bus cycle is proceeding] This state is used to allow the bus controller commands to become active and, to output data during a write cycle.

T2I: [A bus cycle is prepared for termination with no new cycle ready to begin] If the $\overline{\text{READY}}$ signal is active and no new bus cycle is ready to begin, this will be the state used. Input data will be accepted during this state if the $\overline{\text{READY}}$ signal is active and if the bus cycle is an input cycle.

T20: [A bus cycle is prepared for termination with a new cycle ready to begin] This state terminates a bus cycle if the $\overline{\text{READY}}$ signal is active and if a new bus cycle is ready to

begin. As with the T2I state, input data will be accepted during this state if the cycle is an input cycle and if the READY signal is active. This state will also output the address of the new bus cycle, and if READY is active, the status also.

186 Mode T-states: The T-states are one CLK period long, beginning and ending with the falling edge of the CLK signal.

TI: [The bus is idle] This state occurs if the 82258 cannot start the following bus cycle.

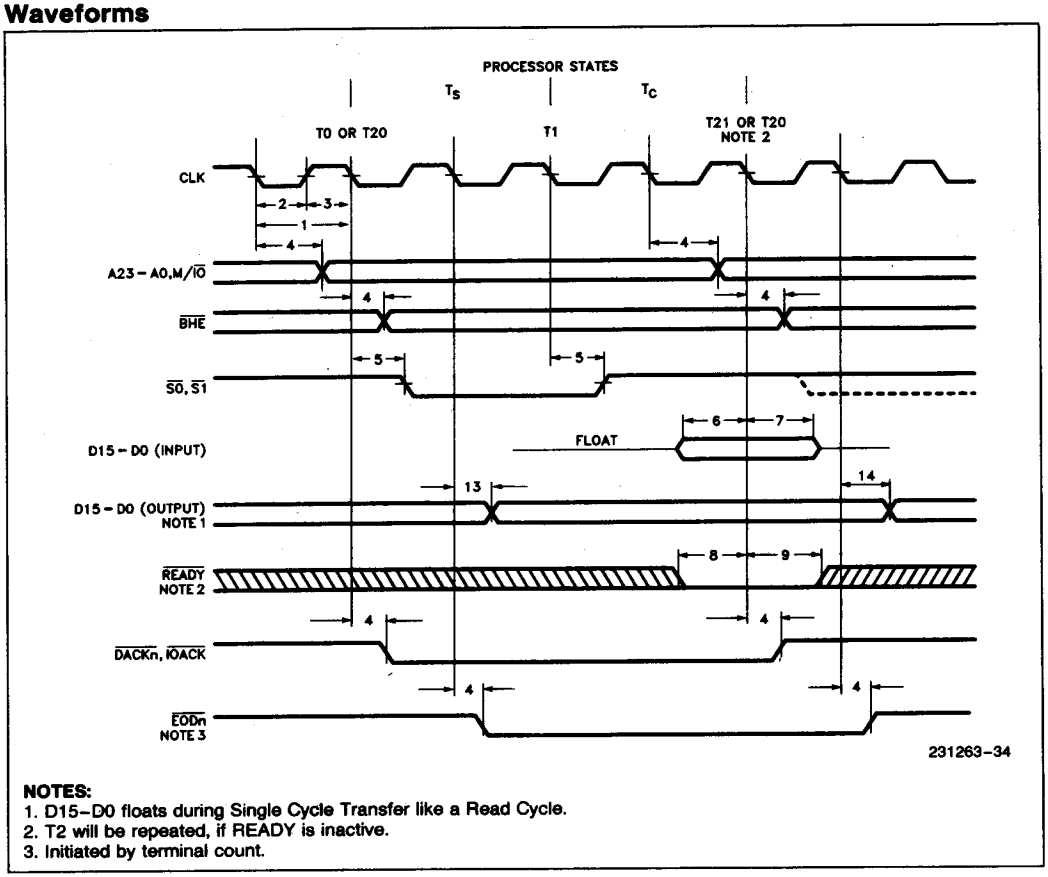
T1: [The first bus cycle T-state] During this state, address information is output to the A19/S6–A16/S3 and AD15–AD0 pins. The status is activated with the rising edge of the CLK previous to this state.

T2: [The second bus cycle T-state] This state allows the bus controller and the 82258 commands to become active and outputs data if the cycle is a write cycle.

T3: [The third bus cycle T-state] This state is used to synchronize the ready signals. If the bus is not ready, then the bus cycle is extended by repeating this state, with the status lines going inactive during the last T3-state.

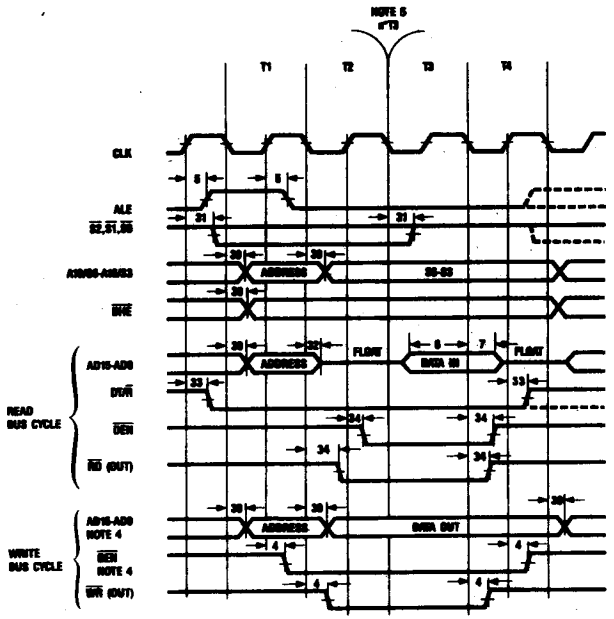
T4: [The last bus cycle T-state] During this cycle, data is input for input cycles and the bus controller and the 82258 commands are disactivated. If the following state is T1, then the status is activated during this state.

Waveforms



NOTES:
1. D15–D0 floats during Single Cycle Transfer like a Read Cycle.
2. T2 will be repeated, if READY is inactive.
3. Initiated by terminal count.

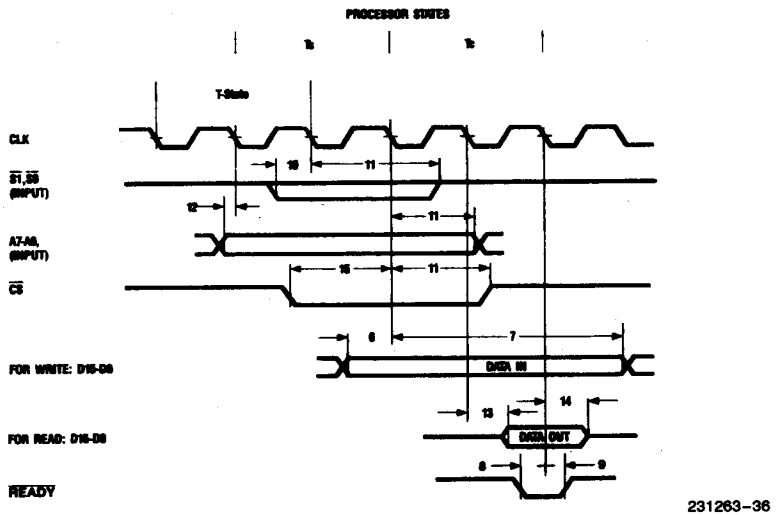
Figure 35. Timing of an Active Bus Cycle (286 and Remote Modes)



231263-35

NOTES:
4. For a Single Cycle Transfer the timings of AD15–AD0, $\overline{DEN}$ and DT/$\overline{R}$ are the same as in a Read Bus Cycle.
5. Additional T3 cycles will be inserted if bus is not ready (see Figure 40).

Figure 36. Timing of an Active Bus Cycle (186 and 8086 Modes)



231263-36

Figure 37. Timing of a Synchronous Access to the 82258 (286 Mode)

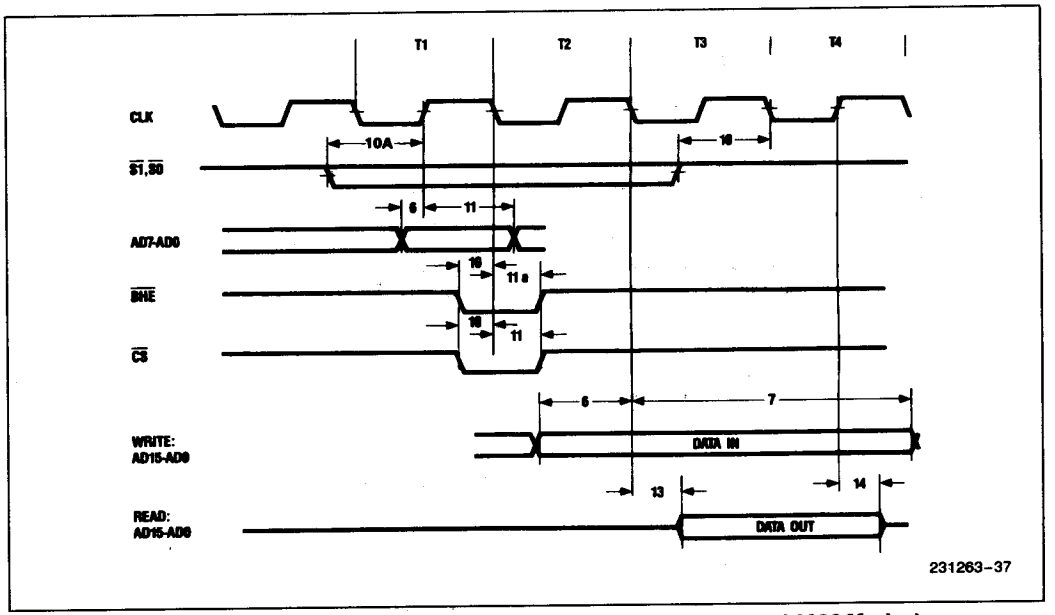


Figure 38. Timing of a Synchronous Access to the 82258 (186 and 8086 Modes)

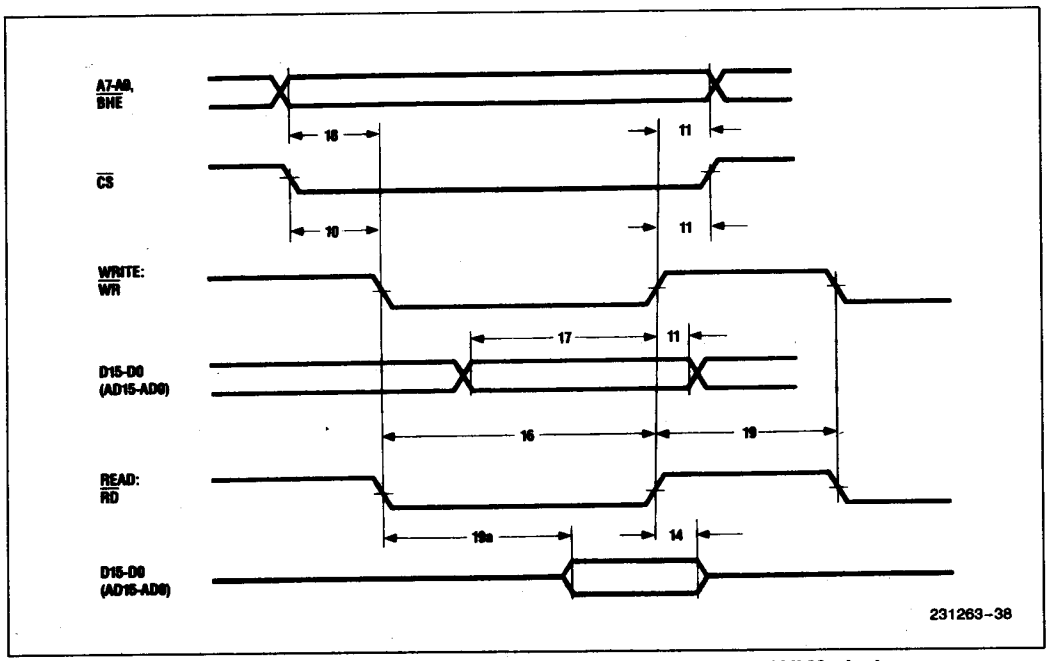


Figure 39. Timing of an Asynchronous Access to the 82258 (All Modes)

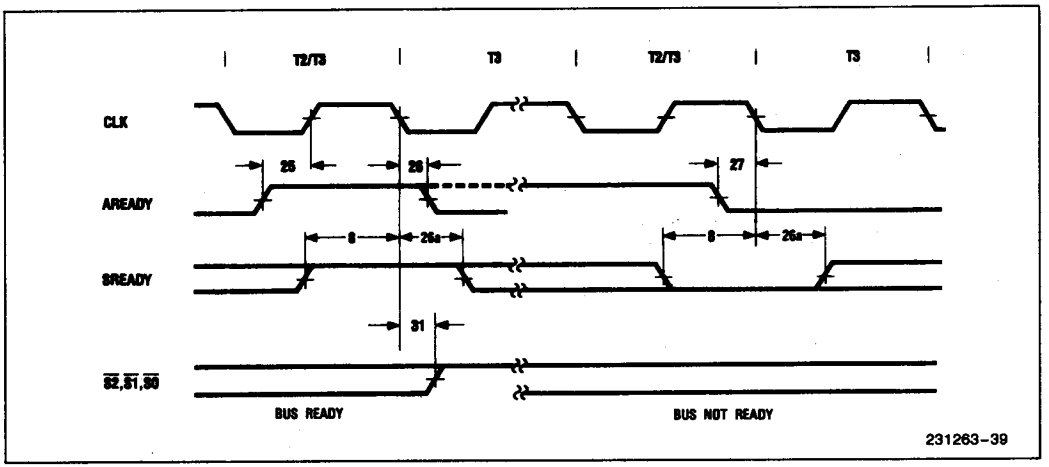
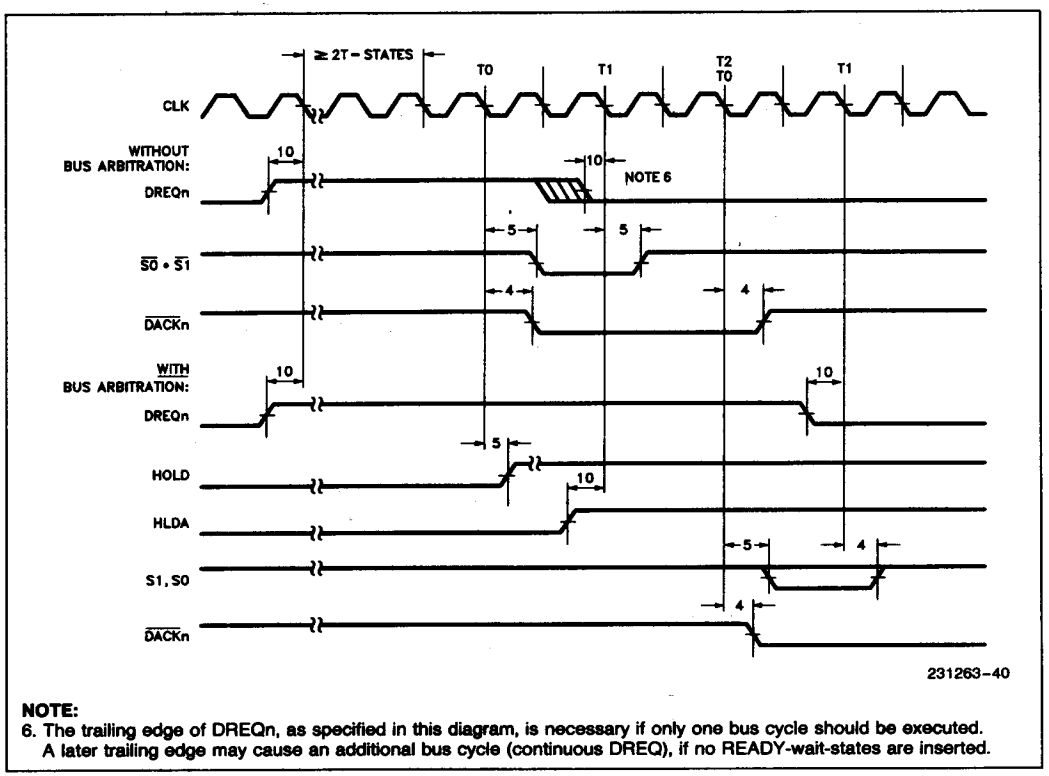
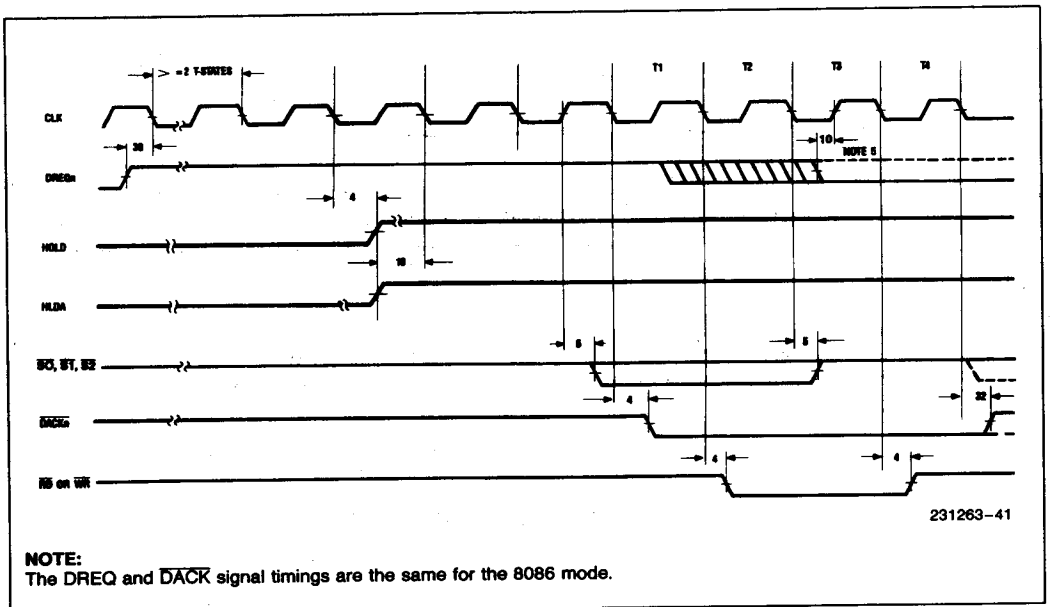


Figure 40. READY Timing (186 Mode)



NOTE:
6. The trailing edge of DREQn, as specified in this diagram, is necessary if only one bus cycle should be executed. A later trailing edge may cause an additional bus cycle (continuous DREQ), if no READY-wait-states are inserted.

Figure 41. DREQ, $\overline{\text{DACK}}$ Timing (286 and Remote Modes)



NOTE:
The DREQ and $\overline{\text{DACK}}$ signal timings are the same for the 8086 mode.

Figure 42. DREQ, $\overline{\text{DACK}}$ Timing (186 Mode)

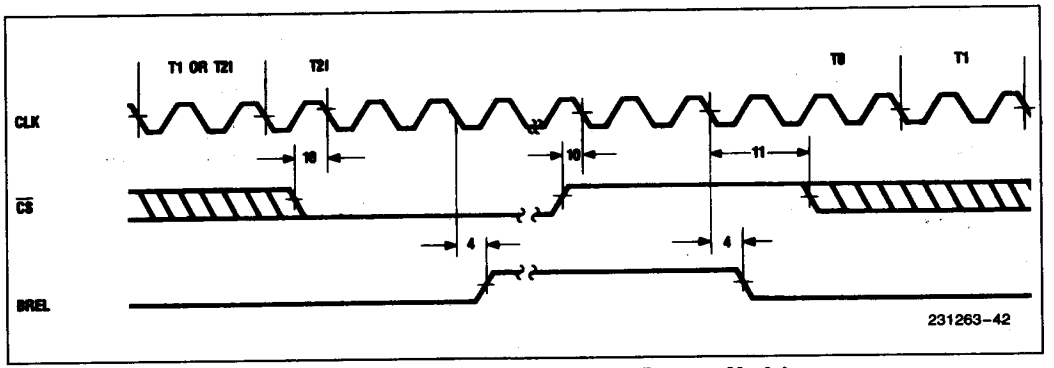


Figure 43. BREL, Bus Tristate Timing (Remote Mode)

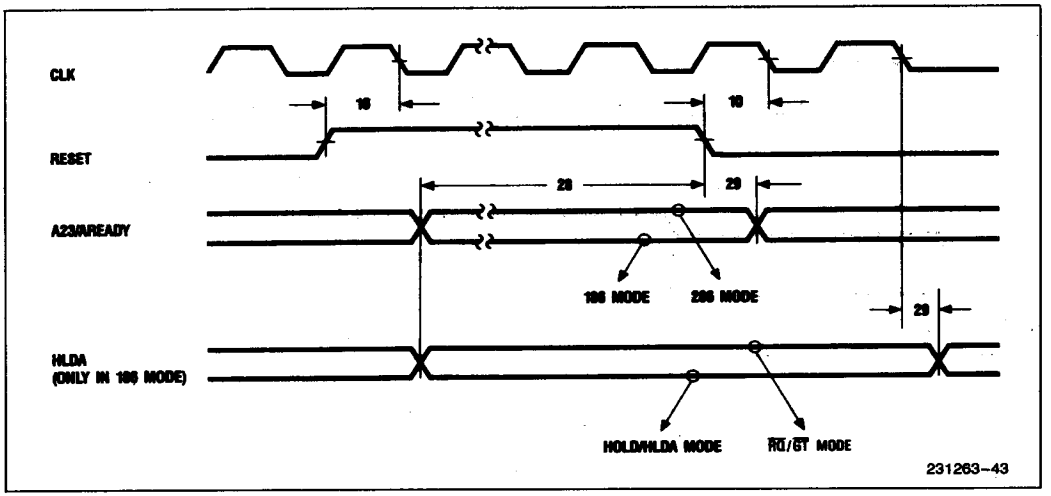
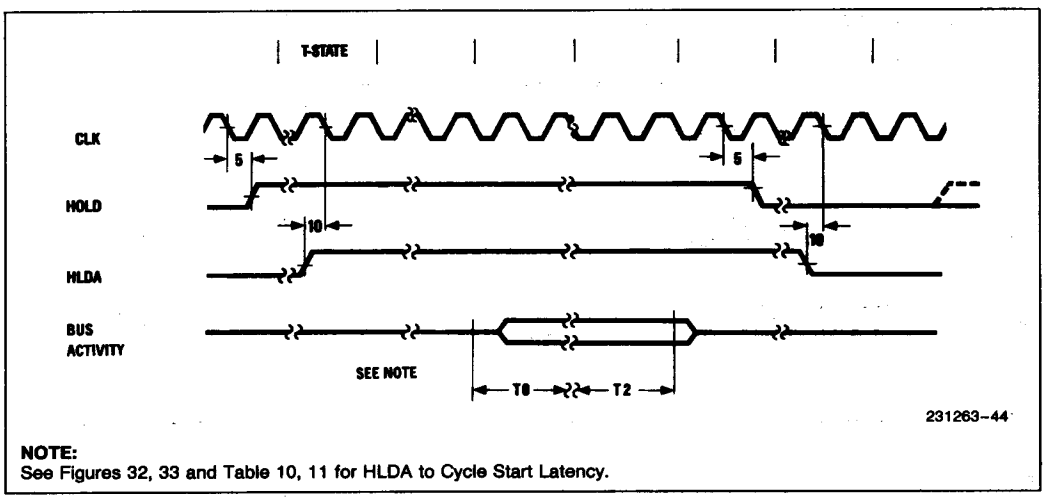
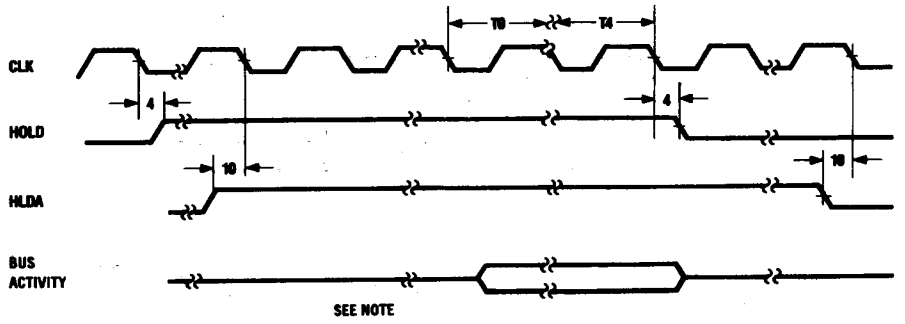


Figure 44. RESET Timing (All Modes)



NOTE:
See Figures 32, 33 and Table 10, 11 for HLDA to Cycle Start Latency.

Figure 45. HOLD, HLDA Timing (286 and Remote Modes)



NOTE:
See Figure 32 and Table 10 for HLDA to Cycle Start Latency.

Figure 46. HOLD, HLDA Timing (186 Mode)

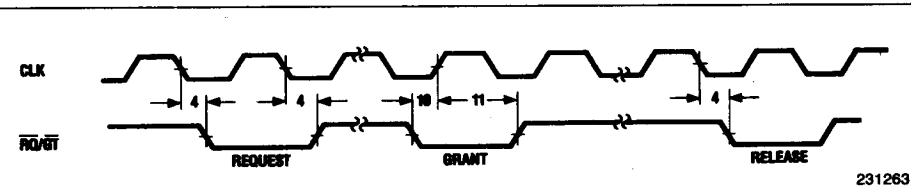


Figure 47. $\overline{RQ}/\overline{GT}$ Timing (8086 Mode)

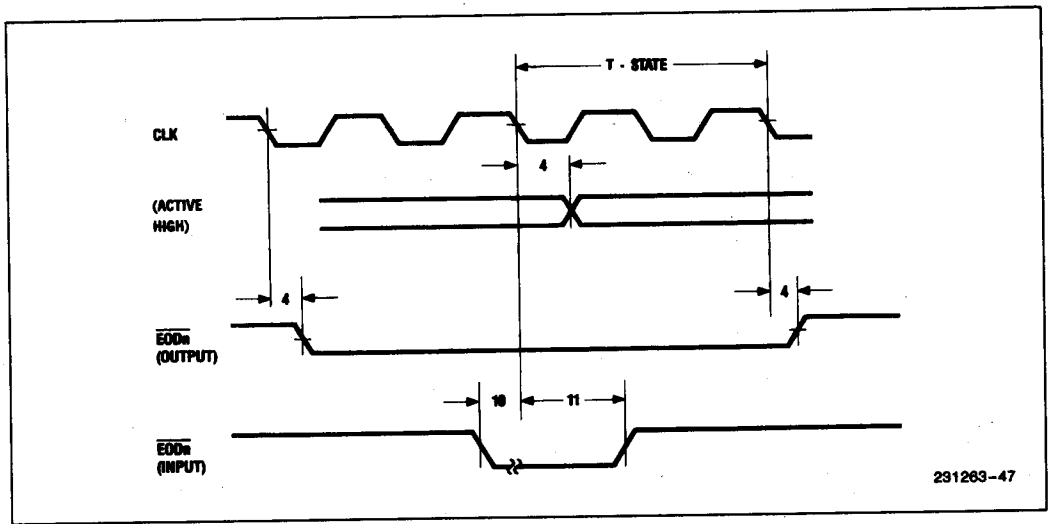


Figure 48. INTOUT, $\overline{EOD}$ Timing (286 and Remote Modes)

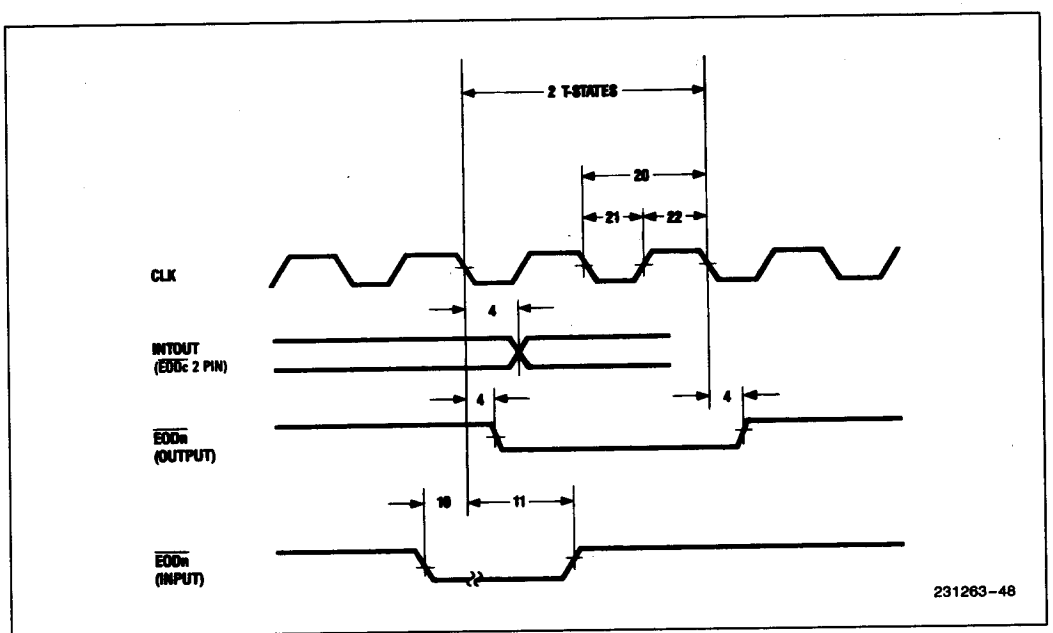


Figure 49. INTOUT, $\overline{EOD}$ Timing (186 and 8086 Modes)

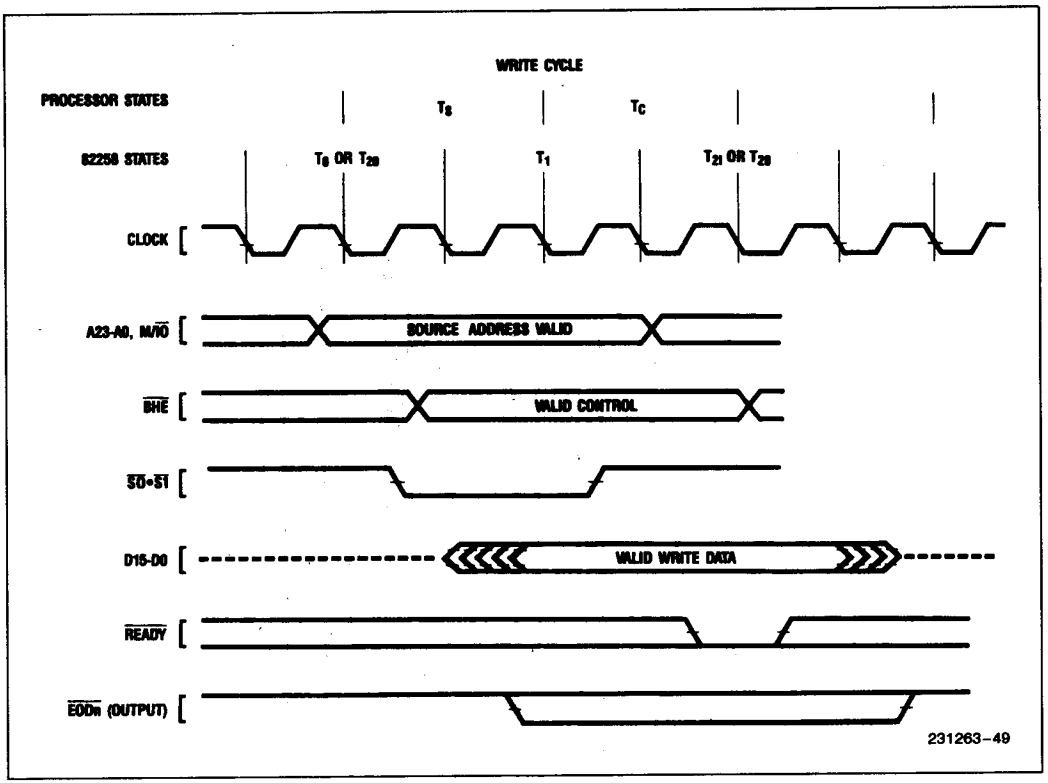


Figure 50. Single Cycle Transfer (286 Mode)

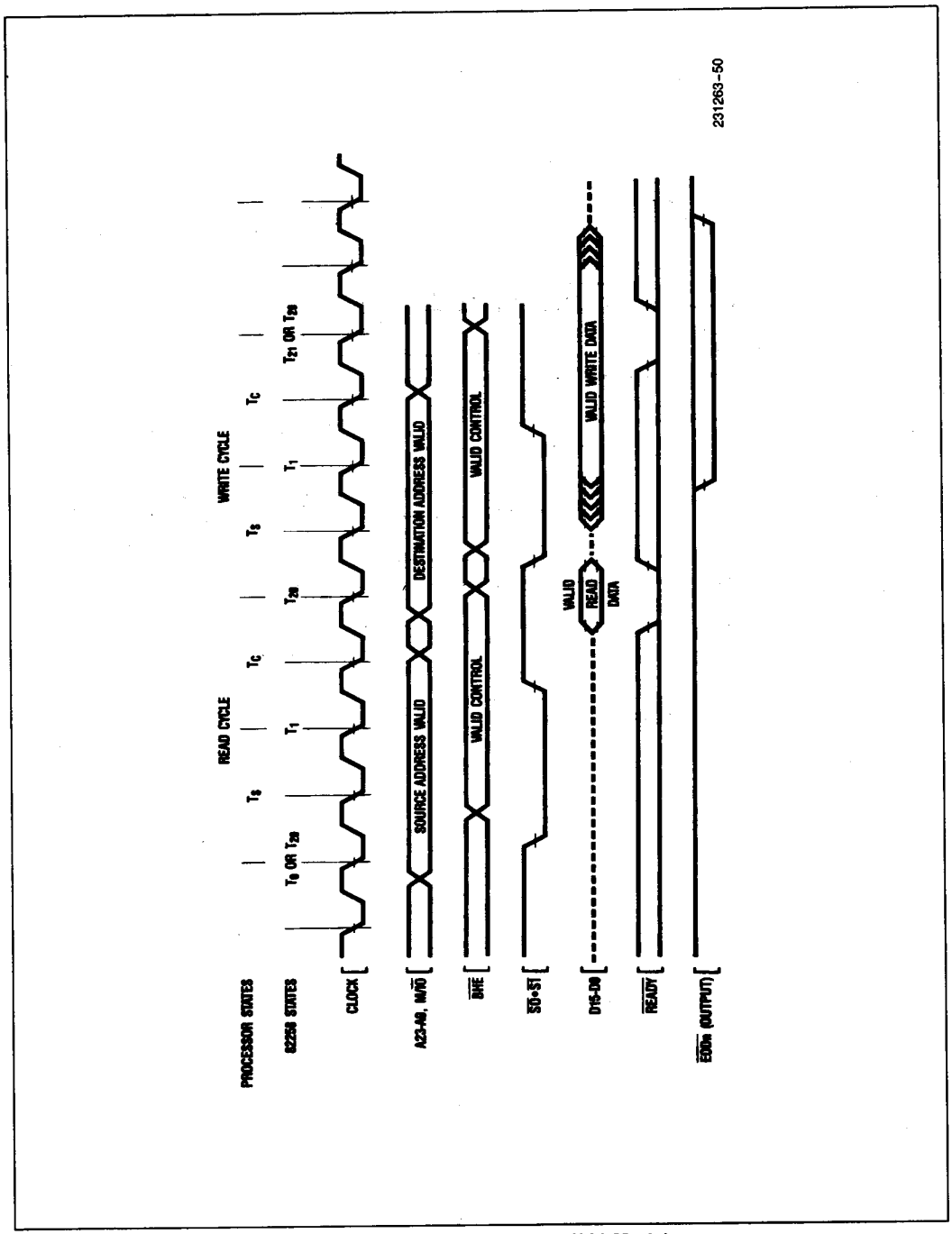
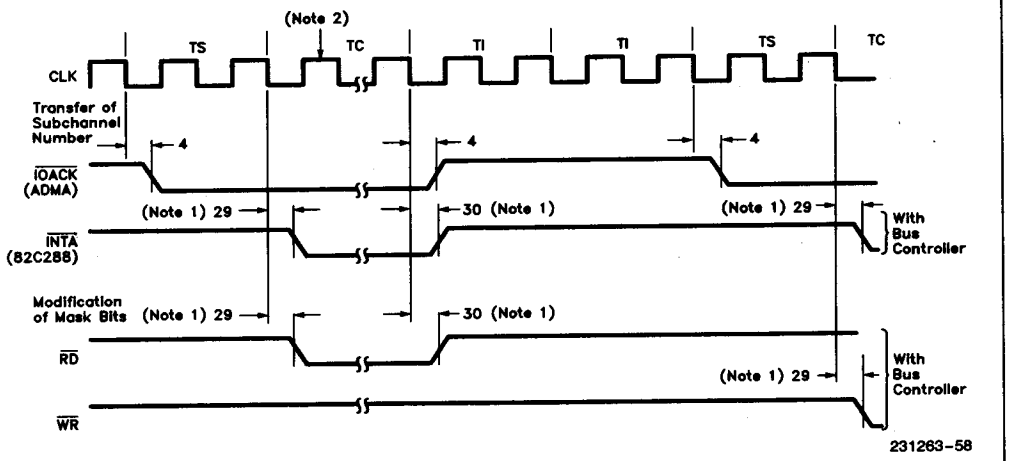
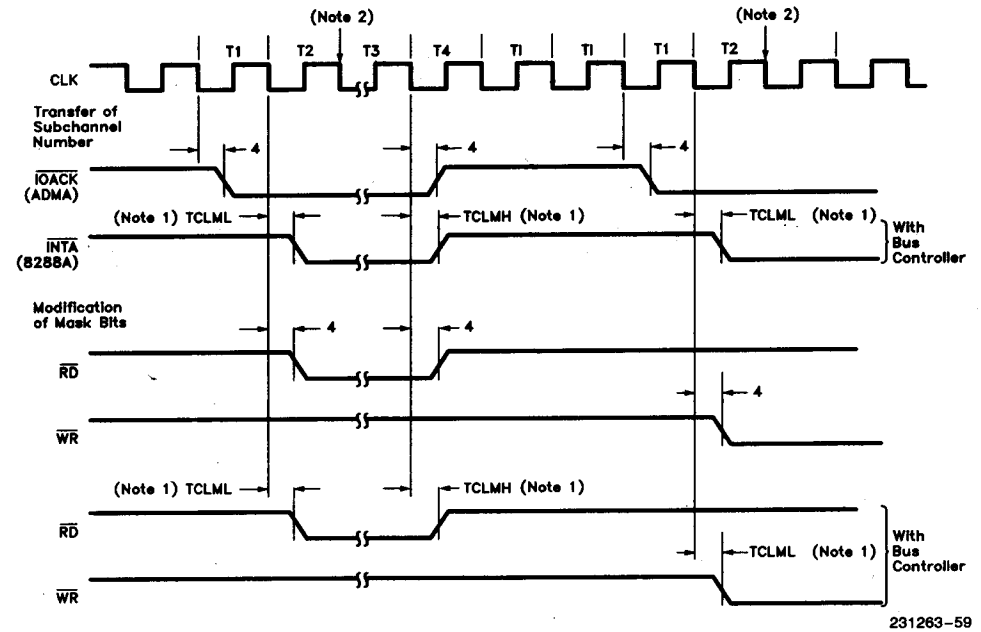


Figure 51. Two Cycle Transfer (286 Mode)



NOTES:
1. These timings are 82C288 timings.
2. Additional wait states may be inserted.

Figure 52. Access to 8259A in 80286 and Remote Modes.



NOTES:
1. These timings are 8288A timings.
2. Additional wait states may be inserted.

Figure 53. Access to 8259A in 8086 and 80186 Modes

DATA SHEET REVISION REVIEW

The following list represents key differences between this and the -003 82258 data sheet. Please review this summary carefully.

1. Figure 35 was updated. The new timing diagram now illustrates DACKN#, IOACK#, and EODn# timings during active bus cycles in the 80286 and remote modes.
2. Figure 37 was updated. The new timing diagram now illustrates the READY# signal during a synchronous access to the 82258.
3. Figure 41 was updated. The new timing diagram completely separates the DREQ, DACK# timings from "without bus arbitration" and "with bus arbitration".
4. Two new timing illustrations were added to the 82258 data sheet. Figure 52 illustrates bus accesses to the 8259A in 80286 and remote modes, and Figure 53 illustrates bus accesses to the 8259A in 80186 and 8086 modes.
5. A note to the DREQ pin description was added to advise designers to leave unused DREQn inputs left floating.