FEATURES/BENEFITS

- **Supports serial, parallel, and rotating priority resolving schemes**
- **Three modes of bus release operation**
- **Supports multi-master system bus arbitration protocol**
- **Compatible with IEEE 796 (MULTIBUS™) Standard**
- **Available in 20-pin plastic DIP**
- **8, 10, 12.5 and 16 MHz versions**
- **Low power CMOS**

DESCRIPTION

The Samsung KS82C289 20-pin CMOS Bus Arbiter signals to request, possess, and release the system bus. External logic determines which bus cycle requires the system bus and sets the priority of requests for control of the system bus.

The KS82C289 has processor-interface and Multibus state machines which support bus request and bus release logic.

The KS82C289 Bus Arbiter requires a Bus Controller, Clock Generator, and processor (bus master) to interface to the Multi-master System Bus.

Figure 1. KS82C289 Block Diagram

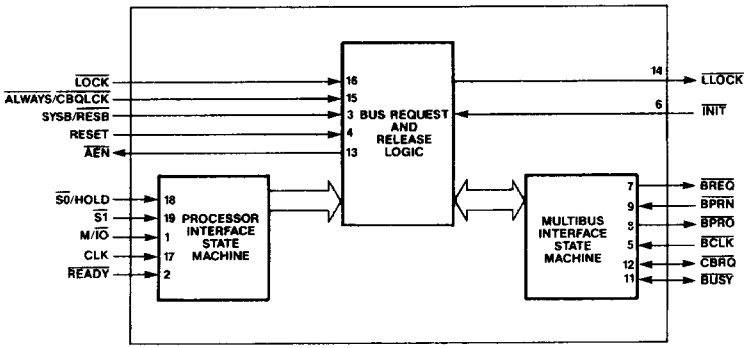


Figure 2a: 20-pin PLCC Configuration

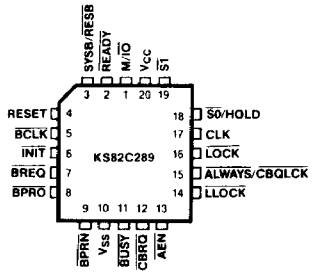
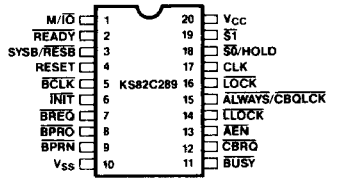


Figure 2b: 20-pin DIP Configuration



MULTIBUS is a trademark of Intel, Corp.

Table 1. KS82C289 Pin Allocations in a 20-pin Plastic DIP

Note on Conventions: A bar over the signal name is used to denote an active low signal (S0). Active high signals are shown with no bar (HOLD).

Pin No.	Signal Abbrev.	Signal Name
1	M/$\overline{IO}$	Memory or I/O Select
2	$\overline{READY}$	Ready
3	SYSB/$\overline{RESB}$	System Bus/Resident Bus
4	RESET	Reset
5	$\overline{BCLK}$	Bus Clock
6	$\overline{INIT}$	Initialize
7	$\overline{BREQ}$	Bus Request
8	$\overline{BPRO}$	Bus Priority Out
9	$\overline{BPRN}$	Bus Priority In
10	V_{SS}	Ground

Pin No.	Signal Abbrev.	Signal Name
11	$\overline{BUSY}$	Busy
12	$\overline{CBRQ}$	Common Bus Request
13	$\overline{AEN}$	Address Enable
14	$\overline{LLOCK}$	Level Lock
15	$\overline{ALWAYS}$/ $\overline{CBQLCK}$	Always Release/Common Bus Request Clock
16	$\overline{LOCK}$	Lock
17	CLK	System Clock
18	$\overline{S0}$/HOLD	Status Input S0/Hold
19	$\overline{S1}$	Status Input S1
20	V_{CC}	VCC

Table 2. KS82C289 Signal Descriptions

Note: I indicates that the signal is an input to the KS82C289 chip. O indicates that the signal is an output from the KS82C289 chip.

Symbol	Type	Description
CLK	I	**System Clock:** Receives the CLK signal from the clock generator as a timing reference. The processor interface state machine (see Figure 1) is synchronous to the falling edge of CLK.
$\overline{S0}$/HOLD	I	**Status Input S0 or Hold:** Becomes active if $\overline{S0}$ is received from the processor or HOLD is received from the bus master. HOLD is selected if the $\overline{S0}$/HOLD pin is high at the falling edge of the processor RESET. $\overline{S0}$ is selected if $\overline{S0}$/HOLD pin is low at the falling edge of the processor reset.
$\overline{S1}$, M/$\overline{IO}$	I	**Status Input 1, Memory or Input/Output Select:** $\overline{S0}$, $\overline{S1}$, and M/$\overline{IO}$ are the status input signals from the processor. These inputs are decoded, along with $\overline{S0}$/HOLD, to start a bus request or to release the bus. If either $\overline{S1}$ or $\overline{S0}$ is low at the falling edge of the clock, a bus cycle is started. Bus Cycle Status Encoding

M/$\overline{IO}$	$\overline{S0}$	$\overline{S0}$/HOLD	Type of Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	None; bus idle
1	0	0	Halt or shutdown
1	0	1	Memory read
1	1	0	Memory write
1	1	1	None; bus idle

SYSB/$\overline{RESB}$	I	**System Bus/Resident Bus:** Decides when the multi-master system bus is needed for the current bus cycle. If SYSB/$\overline{RESB}$ is high at the end of the T_S bus state, the arbiter will request or retain the multi-master system bus. SYSB/$\overline{RESB}$ is sampled at every falling edge of the CLK which starts at the end of the T_S bus state until the bus cycle is finished by the $\overline{READY}$ signal or SYSB/$\overline{RESB}$ becomes high (inactive).
$\overline{READY}$	I	**Ready:** Indicates the end of the bus cycle if $\overline{READY}$ is low (active). The processor does not require the $\overline{READY}$ signal to end the bus cycle.
$\overline{LOCK}$	I	**Lock:** If $\overline{LOCK}$ is active (low), the arbiter is prevented from releasing the multi-master system bus to any other arbiters having higher priority. $\overline{LOCK}$ is sampled at the end of every bus state.

Table 2. KS82C289 Signal Descriptions (Continued)

Symbol	Type	Description
$\overline{ALWAYS}$	I	**Always Release:** Must be programmed during the falling edge of the processor reset by setting this pin low. Arbiter will release the multi-master system bus after each bus transfer cycle. Arbiter will be in the Always Release mode until reprogrammed.
$\overline{CBQLCK}$	I	**Common Bus Request Lock:** Is programmed if this pin is set high during the falling edge of the processor reset. $\overline{CBQLCK}$ is active on and prevents the arbiter from releasing the multi-master system bus to any other arbiters.
$\overline{INIT}$	I	**Initialize:** If $\overline{INIT}$ is low (active), all the arbiters on the multi-master system bus are reset. Releases the multi-master system bus but, pending a bus request, it cannot be cleared. Hence, arbiters can regain the multi-master system bus immediately, if necessary. $\overline{INIT}$ is not synchronous to CLK. **Note:** $\overline{LLOCK}$ (Level Lock) is not affected by this signal.
RESET	I	**Reset:** If RESET is high (active), $\overline{BREQ}$, $\overline{BUSY}$, and $\overline{AEN}$ are cleared and become inactive. RESET will also stop any current bus cycle without waiting for it to end. The bus cycle terminated by RESET will not be completed when RESET becomes inactive.
$\overline{BCLK}$	I	**Bus Clock:** $\overline{BCLK}$ is the multi-master system bus clock. All of the multi-master bus interface signals are synchronized to $\overline{BCLK}$. $\overline{BCLK}$ may not be synchronous to CLK. The multi-master system bus interface state machine (see Figure 1) is asynchronous to the falling edge of $\overline{BCLK}$.
$\overline{BREQ}$	O	**Bus Request:** The arbiter keeps the $\overline{BREQ}$ low (active) until it releases the multi-master system bus. $\overline{BREQ}$ is essential in the rotating and parallel priority resolving technique.
$\overline{BPRN}$	I	**Bus Priority In:** When low (active), this arbiter has the highest priority. When high, another arbiter with higher priority is requesting the multi-master system bus.
$\overline{CBRQ}$	I/O	**Common Bus Request:** An open-drain input/output which requires an external pull-up resistor. As an input: Another arbiter is requesting the multi-master system bus. It is enabled by the $\overline{CBRQ}$. As an output: This arbiter is requesting the multi-master system bus. When $\overline{BREQ}$ (Bus Request) is issued, the $\overline{CBRQ}$ is pulled low. When the arbiter gains the multi-master system bus, the $\overline{CBRQ}$ is released.
$\overline{BPRO}$	O	**Bus Priority Out:** $\overline{BPRO}$ low (active) is used for the serial priority technique. $\overline{BPRO}$ is connected to the $\overline{BPRN}$ (Bus Priority In) of the immediately lower priority to decide the status of the priority for that arbiter.
$\overline{LLOCK}$	O	**Level Lock:** $\overline{LLOCK}$ cannot be cleared by the $\overline{INIT}$, but can be cleared by RESET. When buffered with a tri-state buffer enabled by the $\overline{AEN}$ (Address Enable), $\overline{LLOCK}$ can be used as a multi-master system bus lock. $\overline{LLOCK}$ is active low and it is decoded from the processor LOCK.

Table 2. KS82C289 Signal Descriptions (Continued)

Symbol	Type	Description
$\overline{AEN}$	O	**Address Enable:** Connected to the clock generator, bus controller, and the processor's address latches. When low (active), can be used as Hold ACK (Hold Acknowledge) to a bus master. When high, indicates to the bus master that the arbiter has released the system bus. $\overline{AEN}$ becomes active relative to $\overline{BCLK}$ (Bus Clock). $\overline{AEN}$ becomes inactive relative to $\overline{CLK}$ (System Clock).
$\overline{BUSY}$	I/O	**Busy:** An open-drain input/output which requires an external pull-up resistor. As an input: Low (active) indicates that the multi-master system bus is in use. As an output: When high, indicates that this arbiter has taken control of the multi-master system bus.
V_{SS}	—	**Ground.**
V_{CC}	—	+5 volts supply voltage.

OPERATIONAL DESCRIPTION

Arbitration Between Bus Masters

The KS82C289 Bus Arbiter is a priority controlling device which allows the multi-master system bus to be used for multi-processing. Both higher and lower priority bus masters are allowed to gain the system bus, depending on the release mode. Ordinarily, the higher priority master acquires the system bus immediately after any lower priority master finishes its present cycle. Therefore, at the end of each transfer cycle, the Arbiter can keep the system bus or release it depending on the bus arbitration inputs, arbiter strapping options, and the processor state.

Releasing the Multi-Master System Bus

The Bus Arbiter can retain or release control of the multi-master system bus following every transfer cycle. There are three modes in which the Arbiter can release the multi-master system bus.

These three modes cannot release the multi-master system bus if the cycles are $\overline{LOCK}$ed.

Mode 1

Always Release Mode

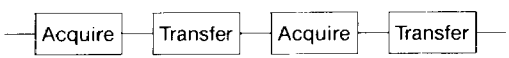


Figure 3. Always Release Mode

Releases the multi-master system bus at the end of each transfer cycle. Mode 1 must be programmed at the falling edge of the processor RESET.

Mode 2

Releases the multi-master system bus if either condition, below, is met:

a) a lower priority bus master demands the bus by pulling CBRQ low.

b) BPRN = 1, which indicates that the higher priority bus master is asking for the multi-master system bus.

Mode 3

Mode 3 is the same as Mode 2, only $\overline{CBRQ}$ has no effect.

Gaining Control of the Multi-Master System Bus

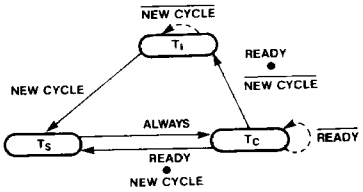
The $\overline{CBRQ}$ signal indicates whether or not another Arbiter wishes to gain control of the multi-master system bus. To perform this function, $\overline{CBRQ}$ must be connected to all other Arbiter $\overline{CBRQ}$ pins. Therefore, if any Bus Arbiter activates the $\overline{CBRQ}$ pin, it will pull down the $\overline{CBRQ}$ line to low.

Besides the $\overline{CBRQ}$ line, only the $\overline{BPRN}$ indicates if other, higher-priority, masters are requesting the bus.

A lower priority master can gain the bus in between the bus master's transfer cycles if the bus master has terminated its use of the bus. Then the bus must gain $\overline{BCLK}$ again at the beginning of the next transfer cycle.

This requires two $\overline{BCLK}$ periods if no other master demands the bus. This step of giving up and getting back the bus is wasteful and unnecessary. To bypass this problem $\overline{CBRQ}$ is useful. The Bus Arbiter does not need to release the bus if the $\overline{CBRQ}$ is not asserted. This alleviates the inefficient delay of getting back the multi-master system bus.

Figure 4. Bus States and the $\overline{READY}$ Signal



Bus States

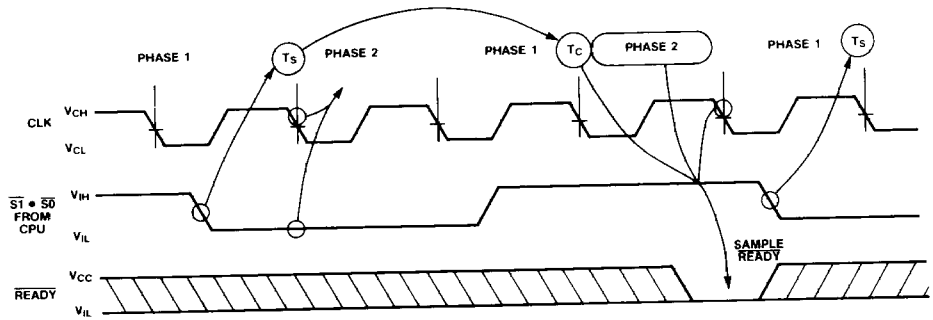
The Bus Arbiter has three processor bus states:

a) T_I (Idle)
b) T_S (Status)
c) T_C (Command)

Each bus cycle is two CLK cycles long.

Internal CPU processor clock phases correspond to the bus state phases.

Figure 5. 80286 Bus Cycle Definition (without wait states)



Bus Cycles

The $\overline{S1}$ and $\overline{S0}$ status inputs are sampled only at the falling edge of the CLK. $\overline{S1}$ and $\overline{S0}$ indicate the start of the bus cycle by going active (low).

The arbiter enters the T_S state if either the $\overline{S1}$ or $\overline{S0}$ is active (low) during the two CLK cycles.

The arbiter enters the T_C state after T_S is exited.

The shortest bus cycle is one T_S and one T_C. The longest bus cycle is one T_S followed by multiple T_C states. A repeated T_C bus state is referred to as a wait state.

The $\overline{READY}$ input determines whether the current T_C is to be repeated. It is sampled at the end of every T_C state if it is high. If it is high (1), then the T_C is repeated. When $\overline{READY}$ is sampled low, the current bus cycle is aborted.

If the $\overline{S1}$ and $\overline{S0}$ status lines are low at the next falling edge of the CLK, then the Bus Arbiter enters the T_S state immediately after the current bus cycle is aborted.

If none of the status lines are sampled active (low) at the next falling edge of the CLK, then the Bus Arbiter enters the T_I state. T_I is repeated until the status lines are sampled active (low).

Bus Masters

MULTIBUS protocols allow multiple processing elements to share access to common system resources. When a common system resource such as the system bus is "BUSY", local processors must wait for access.

The Bus Arbiter sets priorities and schedules access to the multi-master system bus. The bus arbiter supplies access to the system bus depending upon the release mode and the higher or lower priority of each bus master.

When the bus arbiter is used, higher priority bus masters access the system bus before lower priority bus masters or when the current lower priority bus master completes its transfer cycle. Lower priority bus masters access the system bus when there are no higher priority bus masters or when the proper surrender conditions exist.

The bus arbiter arranges scheduling and access transparently to the bus master.

The bus arbiter retains or releases the system bus at the end of each transfer cycle. The processor state, bus arbitration inputs, and arbiter strapping options are the factors used by the bus arbiter to determine release status. Refer to section "Release Modes" for more specific information.

Establishing Priority

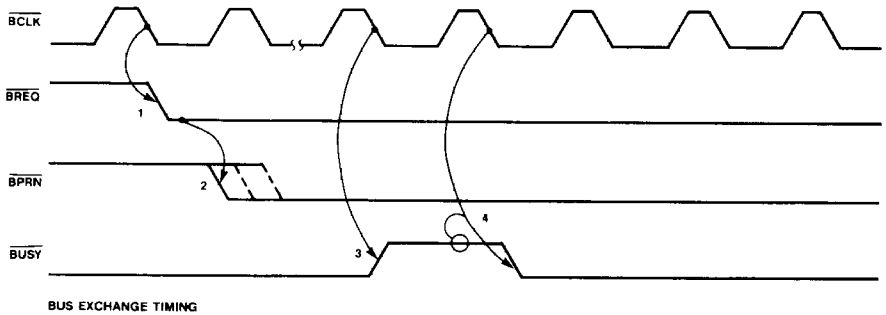
The Bus Arbiter establishes the priority level of the bus masters that are competing for access to a multi-master bus. To do this, the bus arbiter uses parallel, serial, and rotating techniques. Each of these techniques assumes that at any point in time, one bus master has priority over all other bus masters.

The highest priority arbiter is the arbiter with a $\overline{BPRN}$ input (low). The arbiter with the highest priority cannot access the system bus until the system bus is released from its current transaction.

When the system bus completes its current transaction, the present bus owner releases $\overline{BUSY}$. $\overline{BUSY}$ is an active-low 'Wired-OR' MULTIBUS signal which indicates that the system bus is inactive. This signal is sent to every bus arbiter in the system.

When the arbiter with the highest priority ($\overline{BPRN}$ low) receives the $\overline{BUSY}$ signal, it seizes the system bus by pulling $\overline{BUSY}$ (low). Figure 6 is a graphic representation of the Bus Exchange Timing.

Figure 6. Bus Exchange Timing for the MULTIBUS®



A multi-master bus request is initiated when two conditions occur. 1) a processor signals the status for memory read, memory write, I/O read, I/O write, or interrupt acknowledge. 2) an SYSB/$\overline{RESB}$ (high) at the end of T_S.

An interrupt acknowledge cycle does not always require the MULTIBUS each time the status input indicates. To determine when to request the MULTIBUS, the arbiter uses external logic, through the SYSB/$\overline{RESB}$ input.

When the arbiter samples SYSB/$\overline{RESB}$, and it is (high), the MULTIBUS is requested. When the arbiter samples SYSB/$\overline{RESB}$ and it is not (high), the arbiter continues to sample SYSB/$\overline{RESB}$ until either SYSB/$\overline{RESB}$ is (high) or the bus cycle is terminated. The arbiter does not request the MULTIBUS if the bus cycle is completed before SYSB/$\overline{RESB}$ returns (high). Figure 7 is an example of an SYSB/$\overline{RESB}$ sampled repeatedly.

The bus arbiter generates and uses only one $\overline{BREQ}$ from the time it requests the system bus through the entire time it has access to the system bus. The bus arbiter does not generate a separate $\overline{BREQ}$ for each bus cycle. All multi-master system bus requests using $\overline{BREQ}$ are synchronized to the system bus clock, $\overline{BCLK}$.

Parallel Priority Technique

In order to use the parallel technique for determining bus master priority, each bus arbiter on the multi-master system bus must have its own bus request line ($\overline{BREQ}$). Figure 8 is a representation of the parallel technique.

Each $\overline{BREQ}$ line is fed into a priority encoder. The encoder generates the binary address of the active $\overline{BREQ}$ line with the highest priority. Then a decoder uses the binary address to identify the $\overline{BPRN}$ line corresponding to the requesting bus arbiter with the highest priority. The $\overline{BPRO}$ output is not used with the parallel priority resolving technique.

When an arbiter receives the highest priority, $\overline{BPRN}$ (low) and the system bus is released, the arbiter's associated bus master is allowed onto the multi-master system.

The only limiting factor, for the number of bus masters that can be handled by the parallel technique, is the external circuitry. The external circuitry must be able to resolve the bus priorities within one $\overline{BCLK}$ period. Otherwise the parallel priority resolving technique can be used for any number of bus masters.

Serial Priority Technique

The serial priority technique does not require external circuitry. The arbiters are connected in a daisy chain fashion. The highest priority arbiter has its $\overline{BPRO}$ output connected to the $\overline{BPRN}$ input of the next lower priority arbiter. That next lower arbiter has its $\overline{BPRO}$ output connected to the $\overline{BPRN}$ input of the next lower priority arbiter after itself, etc. Figure 9 is a representation of serial technique connection.

This technique establishes a fixed position of priority. The highest priority bus arbiter has its $\overline{BPRN}$ tied (low), ensuring that it always receives highest priority when it requests the system bus. Figure 10 illustrates serial priority bus behavior.

A lower priority arbiter receives temporary higher priority status from the fixed higher priority arbiter. When the arbiter with the higher priority is not accessing or requesting the system bus, it asserts its $\overline{BPRO}$ signal (low). This asserts the $\overline{BPRN}$ signal of the fixed lower priority arbiter, allowing it to have the highest priority, temporarily.

When its $\overline{BPRO}$ goes inactive, a fixed higher priority arbiter retrieves its priority status from a fixed lower priority arbiter. The $\overline{BPRO}$ of an arbiter becomes inactive when it either requests access to the system bus or when its $\overline{BPRN}$ goes inactive because the $\overline{BPRO}$ from the next higher arbiter goes inactive. This allows for a trickle down effect from fixed higher priority arbiters down to the fixed lowest priority arbiter.

$\overline{BREQ}$ output is not used for the serial technique.

The number of bus arbiters connected in serial for priority resolution is limited by propagation delay between $\overline{BPRN}$ and $\overline{BPRO}$, 18ns, because priority must be established within one $\overline{BCLK}$ period. Therefore the maximum number of bus arbiters equals $\overline{BCLK}$ period divided by $\overline{BPRN}$ to $\overline{BPRO}$ delay.

$$\text{number of bus arbiters} = \frac{\overline{BCLK}\text{ period}}{\overline{BPRN}\text{ to }\overline{BPRO}\text{ delay}}$$

Figure 7. Bus Request Timing During an Interrupt Acknowledge Cycle

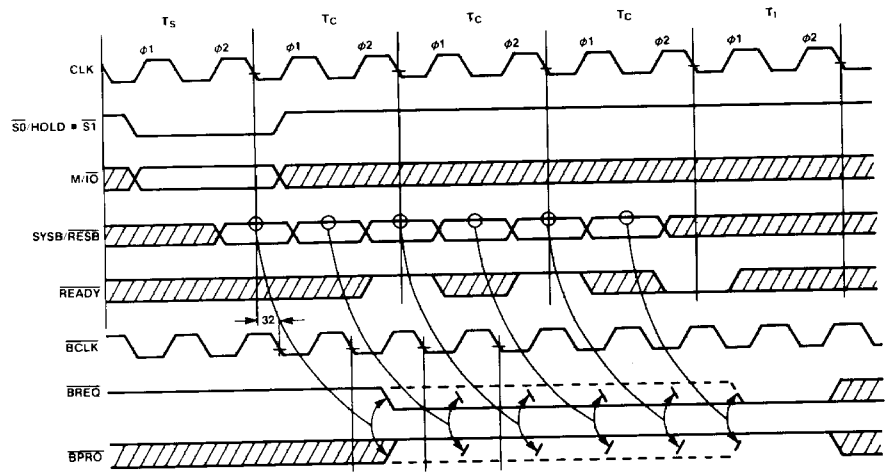


Figure 8. Parallel Priority Resolving Technique

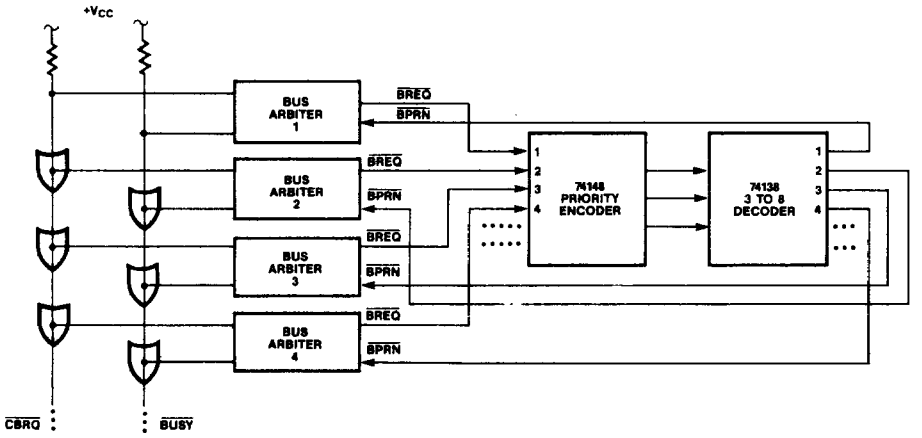


Figure 9. Connections for Serial Priority Resolving Technique

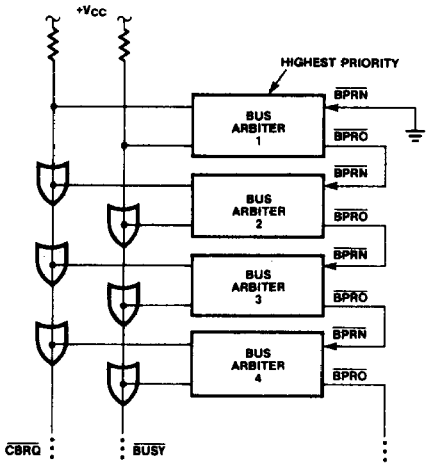
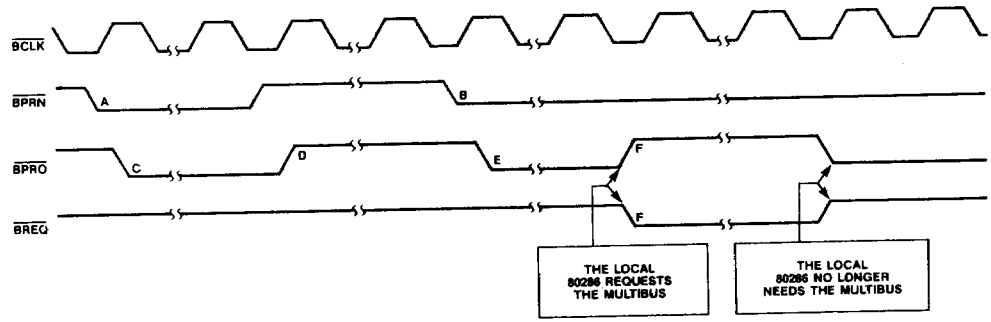


Figure 10. Serial Priority Bus Behavior



Note: Events A through F described above.

Rotating Priority Technique

The rotating priority technique requires external circuitry, similar to the parallel priority technique. The rotating technique assigns and re-assigns priority to the arbiters dynamically.

The priority encoder used in the rotating technique is a more complex circuit than the one used in the parallel technique. The circuit rotates priority between requesting arbiters. This provides each arbiter with an equal chance to use the multi-master system bus over a specified amount of time.

Choosing a Priority Technique

Each priority technique, parallel, serial, and rotating provides a trade-off between using complex external circuitry and allowing equal access to the system bus by each bus master.

The parallel priority technique does not require extensive external logic circuits, does allow for re-assignment of priority status for each bus master, and can accommodate a relatively large number of bus masters.

The serial priority technique does not require any external logic circuits but has fixed priority settings assigned to each bus master and can accommodate a limited number of bus masters.

The rotating priority technique does requires more complicated external logic circuits but does provide equal access between each of the bus masters and the system bus.

Releasing the MULTIBUS

The bus arbiter can either release or retain control of the system bus after the completion of a data transfer cycle on the MULTIBUS. Whether the bus arbiter releases control of the system bus depends upon the release mode selected and the priority settings in effect for the release mode selected.

There are three release modes. Table 3 describes the release modes and the mode settings which enable release of the system bus.

Table 3. Release Modes

Release Mode	Acceptable Release Conditions
Mode 1	The bus arbiter always releases the bus at the end of the transfer cycle.
Mode 2	The bus arbiter retains the system bus until: • a higher-priority bus master requests the system bus. This drives the $\overline{BPRN}$ (high) • a lower priority bus master requests the system by pulling $\overline{CBRQ}$ (low)
Mode 3	The bus arbiter retains the system bus until: • a higher priority bus master requests the system bus. This drives the $\overline{BPRN}$ (high) • $\overline{CBRQ}$ (low) is ignored **Note:** If the cycles are LOCKed, the bus arbiter does not release the system bus, even if the mode release conditions are met.

The arbiter will surrender the MULTIBUS after each complete transfer cycle if the "Always Release" mode 1 is programmed.

If the "Always Release" mode 1 is not programmed, the arbiter will not surrender the MULTIBUS until one of the following occur:

- the processor enters a halt state
- the arbiter is forced off because the $\overline{BPRN}$ becomes (high) and mode 2 or mode 3 is programmed into the arbiter
- the arbiter is forced off because a common bus request $\overline{CBRQ}$ input is enabled and mode 2 is programmed into the arbiter

$\overline{CBRQ}$ reduces bus exchanges. The present bus master retains the sytem bus as long as $\overline{CBRQ}$ is (high). $\overline{CBRQ}$ remains (high) until another master requests the system bus.

$\overline{BPRN}$ indicates if a bus master of higher priority is requesting the system bus. It does not indicate if a bus master of lower priority is requesting the system bus.

In order to allow lower priority bus masters access to the system bus, bus masters must release the system bus at the end of each transfer cycle and re-establish priority to access the system bus again and wait for a current transfer cycle opening. This release, re-establishing priority and re-accessing can take approximately two $\overline{BCLK}$ periods.

$\overline{CBRQ}$ eliminates unnecessary releasing of a bus master from the system bus. When a bus master requires the system bus it must assert $\overline{CBRQ}$ (low). If $\overline{CBRQ}$ remains (high), the current bus master does not have to release the system bus at the end of each transfer cycle.

$\overline{LOCK}$ overrides any of the release mode options. As long as $\overline{LOCK}$ is asserted, the arbiter will not release control of the MULTIBUS to any other requesting bus master.

$\overline{INIT}$ or RESET signals cause the arbiter to surrender the MULTIBUS. The release mode and arbiter input status are ignored.

The three bus release modes operate the same irregardless of the type of microprocessor used.

Choosing a Release Mode

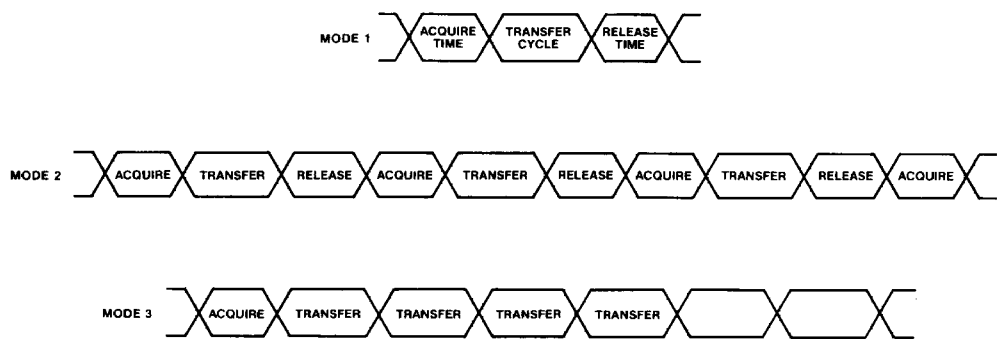
The release mode affects subsystem bus utilization and the system as a whole. The acquire and release times specified for each of the release modes impacts the system bus efficiency. Figure 11 illustrates the differences caused the release and acquisition times for each release mode.

Mode 1 requires a request and release phase for every transfer cycle. This allows lower priority bus masters to access the system bus, but it reduces the overall bus efficiency.

Modes 2 and 3 let the bus master retain the system bus for multiple transfer cycles. A bus master releases the system bus when it is forced off by another bus master's request.

Each release mode allows the designer to optimize the system use of the MULTIBUS.

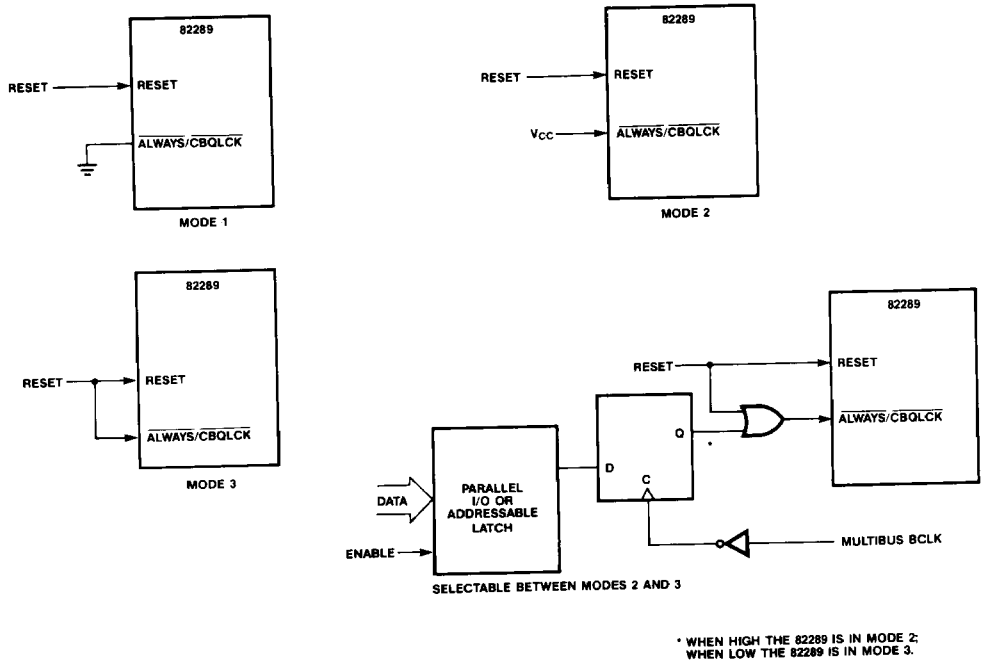
Figure 11. Effects of Bus Release Mode on Bus Efficiency



Configuring Release

The bus arbiter does not require any additional hardware to configure in any of the three release modes. In addition, the processor can be configured to switch between mode 2 and mode 3 by software control. This requires that a parallel port or addressable latch is used to drive the ALWAYS/CBQLCK input pin of the processor. Figure 12 illustrates the three release mode configurations.

Figure 12. 82289 Release Mode Configurations



LOCK and LLOCK

The three modes of releasing the multi-master system bus can be nulled by the LOCK input. But, LOCK will not surrender control of the multi-master system bus to any other Arbiter. The Bus Arbiter will surrender the multi-master system bus if RESET or INIT becomes active. RESET and INIT are independent of the states of the Arbiter inputs or the current release mode.

The LOCK signal can be asserted to the bus arbiter synchronous with the CLK and independent of the three release modes to prevent the release of the multi-master system bus to other bus masters regardless of their order of priority.

The LLOCK output signal can be asserted at all the bus cycles that are LOCKed. LLOCK is 1 if LOCK is 1, and 0 if LOCK is 0. Once LLOCK is active, it will wait until the end of the current transfer cycle before becoming inactive.

RESET and Initialization (INIT)

INIT (active low) is an asynchronous signal from the multi-master system bus. BREQ, BUSY, and AEN are cleared and become inactive when INIT is active (low). The Bus Arbiter will not clear any pending bus request from other bus masters while INIT is active. INIT can interrupt an active bus cycle, but, it will not prevent the Arbiter from requesting the multi-master system bus when it becomes inactive and completing the bus cycle.

RESET (active high) is synchronous to the CLK and can be synchronous to the processor. BREQ, BUSY, and AEN are cleared and become inactive when RESET is asserted. Also, RESET will clear the LLOCK signal and clear any pending bus request, unlike the INIT signal. RESET will stop any current bus cycle without waiting for the cycle to end. And, the bus cycle terminated by RESET will not be completed after the RESET becomes inactive.

DC ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Storage Temperature ..... -65°C to +150°C
Ambient Temperature Under Bias ..... 0°C to 70°C
Case Temperature ..... 0°C to 85°C

Note: Operation at absolute maximum ratings may cause permanent damage to the device.

Table 4. DC Electrical Characteristics

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage			0.8	V
V_{IH}	Input High Voltage		2.0		V
V_{ILC}	CLK Input Low Voltage			0.6	V
V_{IHC}	CLK Input High Voltage		3.0	V_{CC}	V
V_{OL}	Output Low Voltage: BUSY, CBRQ, BPRO BPRO, BREQ, AEN LLOCK	 I_{OL} = 32 mA I_{OL} = 16 mA I_{OL} = 5 mA		 0.45 0.45 0.45	 V V V
V_{OH}	Output High Voltage	I_{OH} = 400 μA	2.4		V
I_{LI}	Input Leakage Current	$V_{SS} < V_{IN} < V_{CC}$		+1	μA
I_O	Output Leakage Current	$V_{OUT} = V_{SS}$ or V_{CC}		+10	μA
I_{CC1}	Quiescent Current	CLK, $V_{IN} - V_{CC}$ or V_{SS}		+10	μA
I_{CC2}	Supply Current			+80	mA
C_{CLK}	CLK, $\overline{BCLK}$ Input Capacitance	FC = 1 MHz		12	pF
C_{IN}	Input Capacitance	FC = 1 MHz		10	pF
C_O	Input/Output Capacitance	FC = 1 MHz		20	pF

AC SWITCHING CHARACTERISTICS

Table 5. KS82C289 AC Switching Characteristics

No.	Parameter	Test Conditions	8.0 MHz		10.0 MHz		12.5 MHz		16.0 MHz (Preliminary)		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
01	CLK Cycle Period		60	BCLK+50	50	BCLK+50	40	BCLK+50	31	BCLK+50	ns
02	CLK Low Time	at 1.0V	15	230	15	230	10		8		ns
03	CLK High Time	at 3.6V	20	235	15	230	12		9		ns
04	CLK Rise/Fall Time	1 to 3.6V		10		10		9		7	ns
05	$\overline{BCLK}$ Cycle Time		100		100		100	∞	100	∞	ns
06	$\overline{BCLK}$ High/Low Time		30		25		20		16		ns

Table 5. KS82C289 AC Switching Characteristics (Continued)

No.	Parameter	Test Conditions	8.0 MHz		10.0 MHz		12.5 MHz		16.0 MHz (Preliminary)		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
07	$\overline{S0}$/HOLD, $\overline{S1}$, M/$\overline{IO}$ Setup Time		22		15		13		10		ns
08	$\overline{S0}$/HOLD, $\overline{S1}$, M/$\overline{IO}$ Hold Time		1		1		1		1		ns
09	$\overline{READY}$ Setup Time		38		30		24		19		ns
10	$\overline{READY}$ Hold Time		25		20		16		13		ns
11	$\overline{LOCK}$, SYSB/$\overline{RESB}$ Setup Time		20		15		12		10		ns
12	$\overline{LOCK}$, SYSB/$\overline{RESB}$ Hold Time		1		1		1		1		ns
13	RESET Setup Time		20		15		12		10		ns
14	RESET Hold Time		1		1		1		1		ns
15	RESET Active Pulse Width		16		16		16		16		CLKs
16	$\overline{INIT}$ Setup Time	Note 2	45		45		45		45		ns
17	$\overline{INIT}$ Hold Time	Note 2	1		1		1		1		ns
18	$\overline{INIT}$ Active Pulse Width		$3(t_1)+(t_{14})$		$3(t_1)+(t_{14})$		$3(t_1)+(t_{14})$		$3(t_1)+(t_{14})$		ns
19	$\overline{BUSY}$, $\overline{BPRN}$, $\overline{CBRQ}$, CBQCLK/$\overline{ALWAYS}$ Hold Time to $\overline{BCLK}$ or (RESET)		20		18		15		12		ns
20	$\overline{BUSY}$, $\overline{BPRN}$, $\overline{CBRQ}$, CBQCLK/$\overline{ALWAYS}$ Hold Time to $\overline{BCLK}$ or (RESET)		1		1		1		1		ns
21	$\overline{BCLK}$ to $\overline{BREQ}$ Delay	C = 60 pF		30		30		25		20	ns
22	$\overline{BCLK}$ to $\overline{BPRO}$ Delay	C = 60 pF		35		35		28		22	ns
23	$\overline{BPRN}$ to $\overline{BPRO}$ Delay	C = 60 pF		25		25		20		16	ns
24	$\overline{BCLK}$ to $\overline{BUSY}$ Active Delay	C = 300 pF	1	60	1	60	1	50	1	38	ns
25	$\overline{BCLK}$ to $\overline{BUSY}$ Float Delay	Note 1		35		35		28		22	ns
26	$\overline{BCLK}$ to $\overline{CBRQ}$ Active Delay	C = 300 pF		55		55		45		35	ns
27	$\overline{BCLK}$ to $\overline{CBRQ}$ Float Delay	Note 1		35		35		28		22	ns
28	$\overline{BCLK}$ to $\overline{AEN}$ Active Delay	C = 150 pF	1	25		25	1	20		16	ns
29	CLK to $\overline{AEN}$ Inactive Delay	C = 150 pF	3	25		25	3	20		16	ns
30	CLK to $\overline{LLOCK}$ Delay	C = 50pF		20		20		16		13	ns
31	RESET to $\overline{LLOCK}$ Delay	Note 2		35		35		28		22	ns
32	CLK to $\overline{BCLK}$ Setup Time	Note 3	38		38		35		25		ns

T_A = 0°C to 70°C
T_{CASE} = 0°C to 85°C
V_{CC} = 5V ± 5%

Notes: AC timing is referenced to 0.8V and 2.0V points.
1. When $I_O < I_{LO}$, float condition occurs.
2. CLK and $\overline{BCLK}$ are asynchronous to each other in actual use. But, this specification is required for component testing.
3. $\overline{INIT}$ is asynchronous to CLK and to $\overline{BCLK}$ during actual use. But, this specification is required for component testing.

Figure 13. AC Drive and Measurement Points CLK Input (BCLK Input)

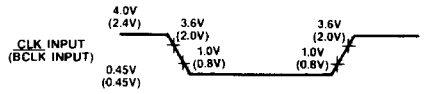


Figure 14. AC Setup, Hold and Delay Time Measurement

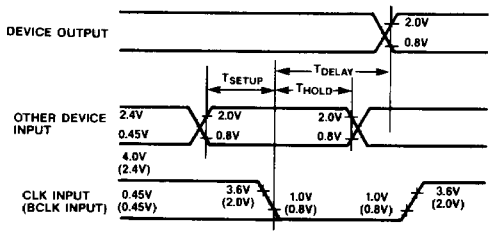
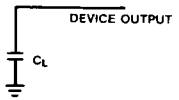


Figure 15. AC Test Loading on Outputs



WAVEFORMS

The following waveforms, Figures 16 through 24, contain examples of general cases of the timing relationships of the inputs and the outputs. These figures do not represent all the possible input and output transitions of all signals in all modes.

Refer to the identified special cases or a timing specification for the same or related function in another mode for examples of specific transitions.

The bus arbiter serves as an interface between the iAPX subsystem and MULTIBUS. The iAPX 286 subsystem operates synchronously to the $\overline{CLK}$ signal. The MULTIBUS operates synchronous to the $\overline{BCLK}$ signal.

$\overline{CLK}$ and $\overline{BCLK}$ operate asynchronously to each other and at different frequencies. The relative phase and frequency of $\overline{CLK}$ and $\overline{BCLK}$ at the time the input is sensed effects the exact clock period where a synchronous input to one clock will cause a synchronous response in the other clock.

The $\overline{CLK}$ period cannot be too long, relative to the $\overline{BCLK}$ period, t_1 greater than t_5 + 50ns, in order to maintain proper MULTIBUS arbitration. If the $\overline{CLK}$ period is too long relative to the $\overline{BCLK}$ period, another arbiter could gain control of the system bus before the current arbiter releases $\overline{AEN}$ synchronous to its $\overline{CLK}$.

The $\overline{AEN}$ release is synchronous to the fall of the $\overline{CLK}$ edge after the processor cycle ends. The $\overline{BREQ}$ and $\overline{BUSY}$ releases are synchronous to the fall of the $\overline{BCLK}$ after the processor cycle ends.

However, all 286 speed selections are MULTIBUS compatible because any $\overline{CLK}$ frequency greater than 6.66 MHz, processor speeds greater than 3.33 MHz, avoids conflict with 10 MHz $\overline{BCLK}$s.

Figure 16. MULTIBUS® Acquisition and Always-Release Operation

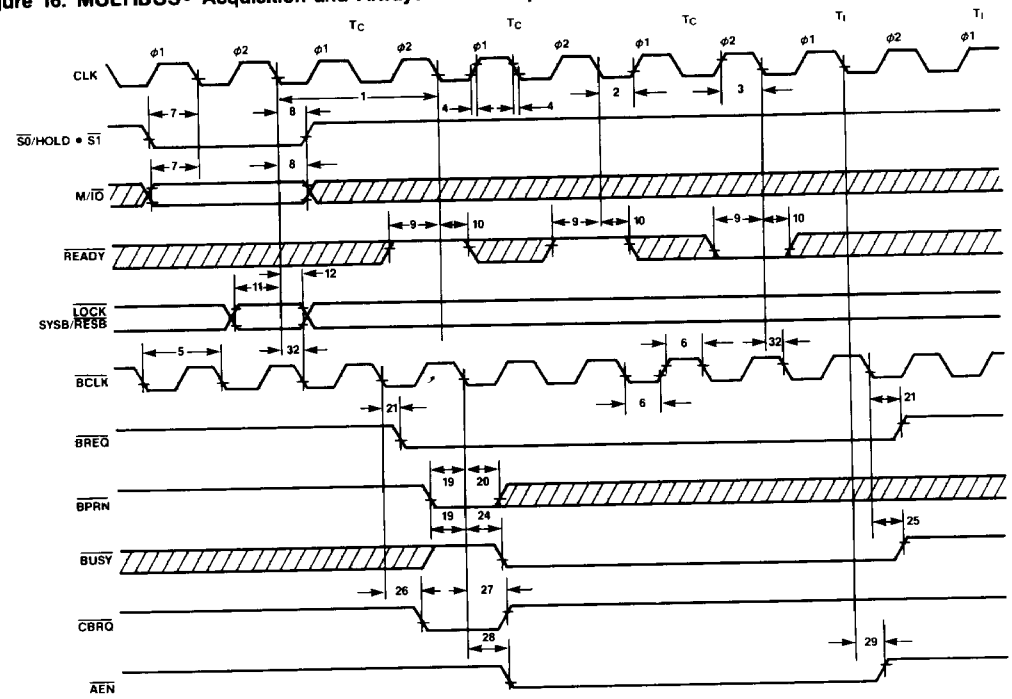


Figure 17. MULTIBUS® Release due to $\overline{\text{BPRN}}$ Inactive

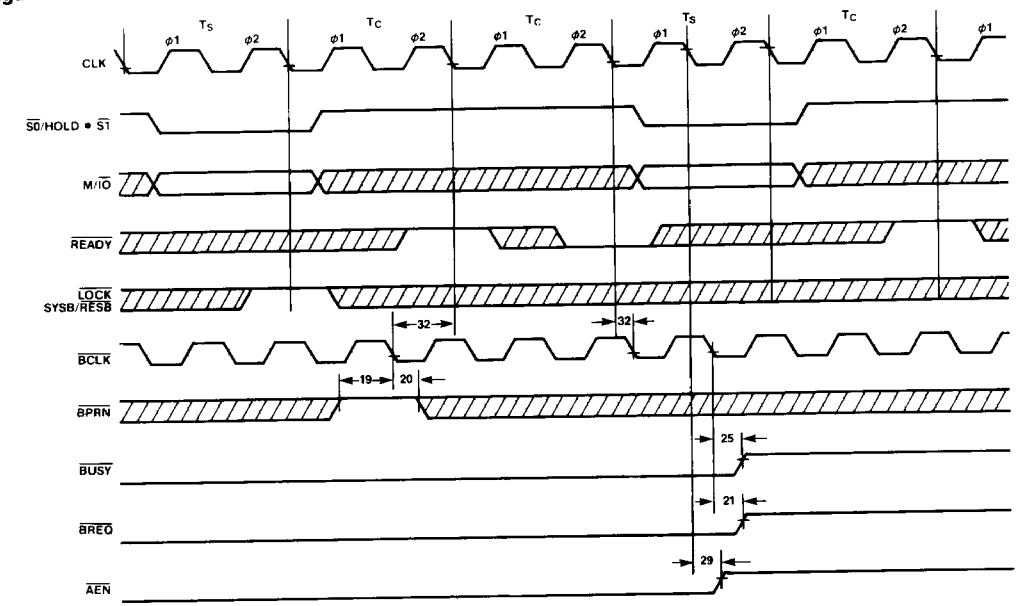


Figure 18. MULTIBUS® Release due to $\overline{CBRQ}$ Active

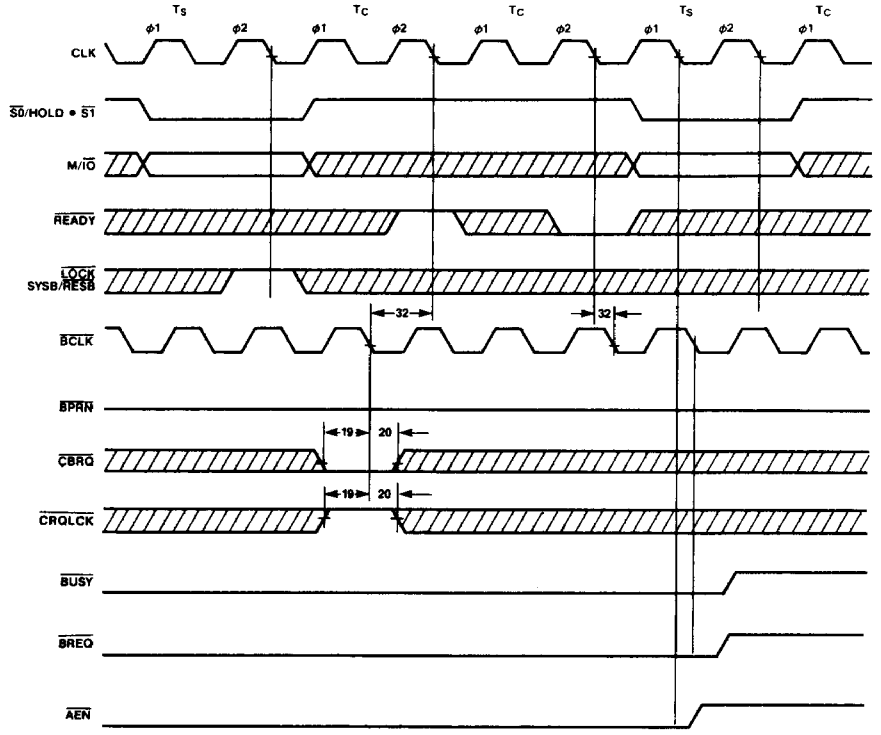


Figure 19. MULTIBUS® Acquisition During 80286 INTA Cycles

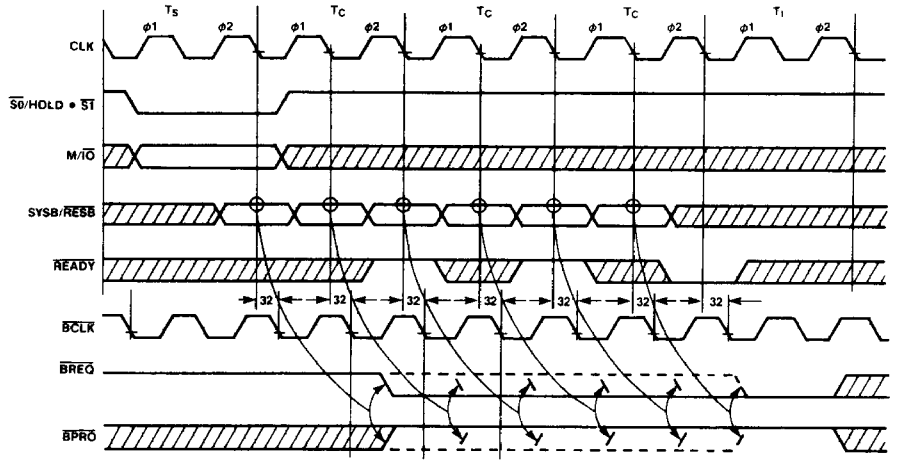


Figure 20. $\overline{BPRN}$ to $\overline{BPRO}$ Timing Relationship

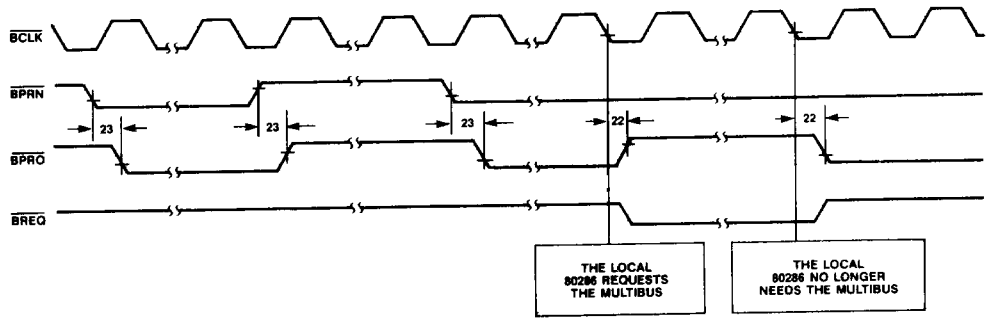


Figure 21. 80286 $\overline{LOCK}$ and 82289 $\overline{LLOCK}$ Relationship

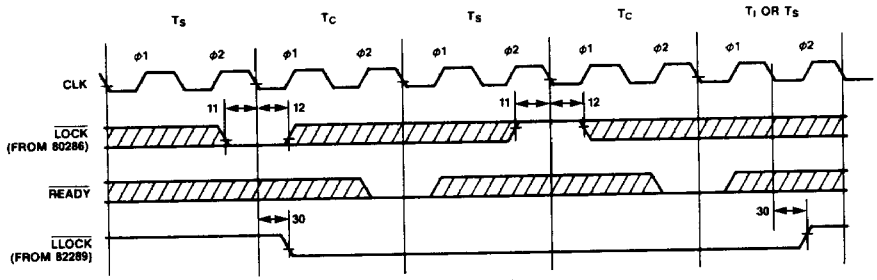


Figure 22. RESET Active Pulse

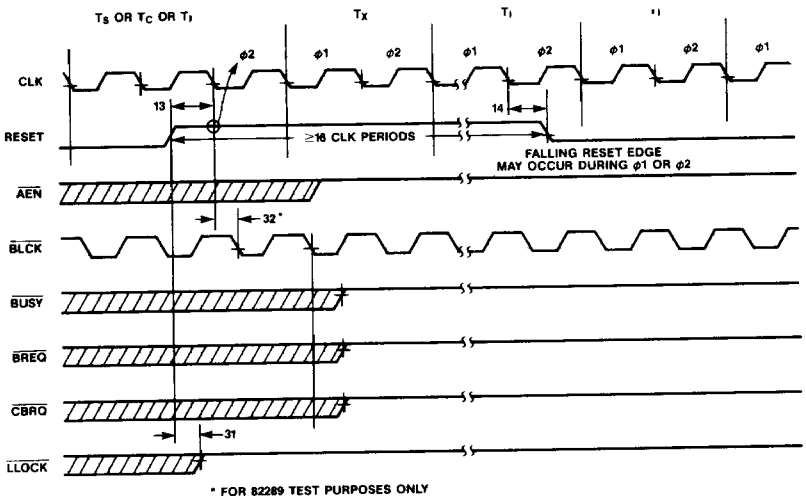


Figure 23. $\overline{\text{INIT}}$ Active Pulse

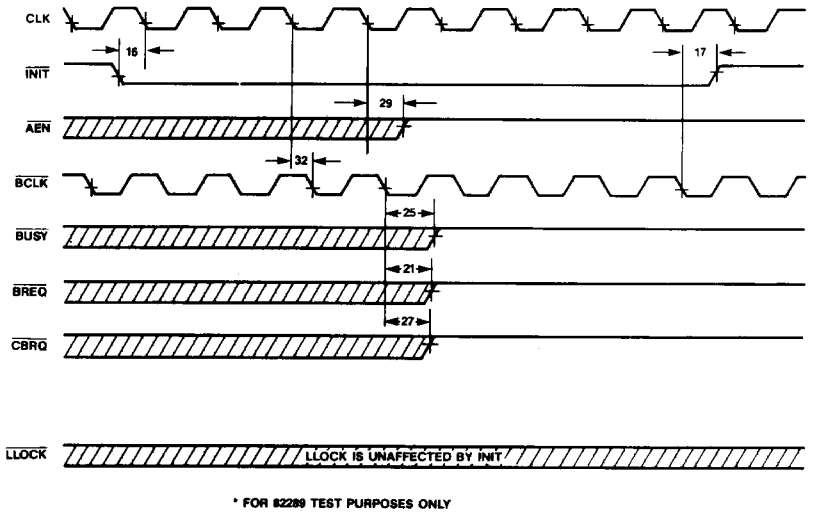
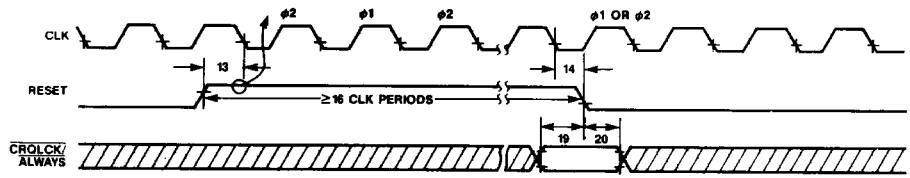
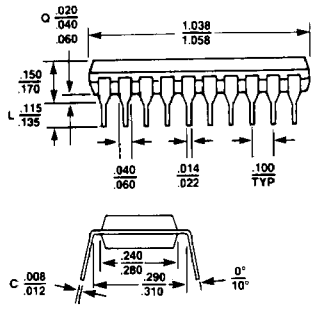


Figure 24. Programming the Always-Release/Common-Bus-Request-Release Option

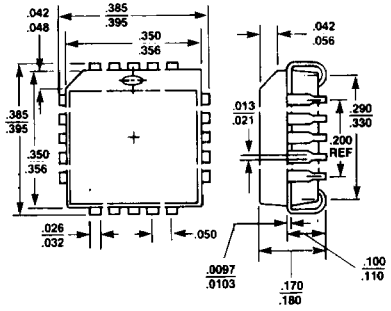


PACKAGE DIMENSIONS

20-pin DIP



20-pin PLCC



ALL DIMENSIONS IN INCHES

ORDERING INFORMATION AND PRODUCT CODE

KS 82C289 X X X

SAMSUNG SEMICONDUCTOR

Part Number

Speed

-8.0	8.0MHz
-10.0	10.0MHz
-12.5	12.5MHz
-16.0	16.0MHz

Packaging
P — Plastic DIP
L — Plastic Leaded Chip Carrier (PLCC)

Temperature
C — Commercial (0°C to +70°C)
I — Industrial (-40°C to +85°C)

SAMSUNG products are designated by a Product Code. When ordering, please refer to products by their full code. For unusual, and/or specific packaging or processing requirements not covered by the standard product line, please contact the SAMSUNG MOS Product Group.