82C89

CMOS Bus Arbiter

Features

- Pin Compatible with Bipolar 8289
- Scaled SAJI IV CMOS Process
- Low Power Operation
  - ICCSB 10μA
  - ICCOP 1mA/MHz
- Compatible with 5MHz and 8MHz 80C86 and 80C88
- Provides Multi-Master System Bus Control and Arbitration
- Provides Simple Interface With 82C88/8288 Bus Controller
- Synchronizes 80C86/8086, 80C88/8088 Processors with Multi-Master Bus
- Bipolar Drive Capability, Fully TTL Compatible
- Four Operating Modes for Flexible System Configuration
- Wide Operating Temperature Ranges:
  - C82C89 .... 0°C to +70°C
  - I82C89 .... -40°C to +85°C
  - M82C89 .... -55°C to +125°C

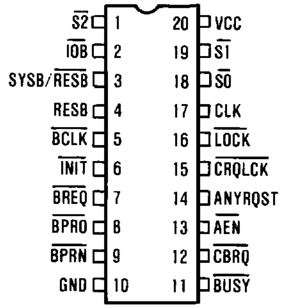
Description

The Harris 82C89 Bus Arbiter is manufactured using a self-aligned silicon gate CMOS process (Scaled SAJI IV). This circuit along with the 82C88 bus controller, provides full bus arbitration and control for multi-processor systems. The 82C89 is typically used in medium to large 80C86 or 80C88 systems where access to the bus by several processors must be coordinated. The 82C89 also provides high output current and capacitive drive to eliminate the need for additional bus buffering.

Static CMOS circuit design insures low operating power. The advanced Harris SAJI CMOS process results in performance equal to or greater than existing equivalent products at a significant power savings.

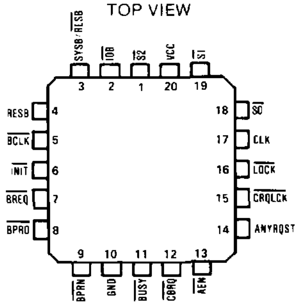
Pinouts

TOP VIEW

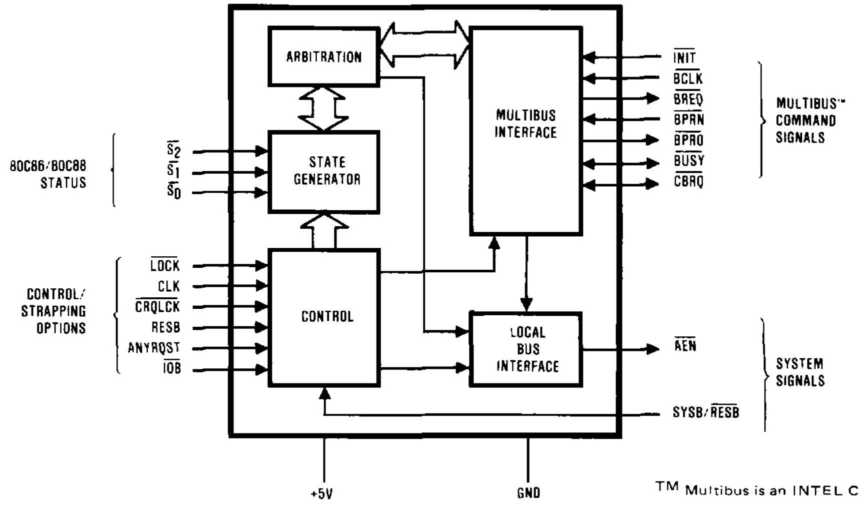


LCC/PLCC

TOP VIEW



Functional Diagram



TM Multibus is an INTEL Corp trademark

CAUTION. These devices are sensitive to electrostatic discharge Proper I C handling procedures should be followed

Pin Description

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
VCC	20		VCC: The +5V Power supply pin. A 0.1μF capacitor between pins 10 and 20 is recommended for decoupling.
GND	10		GROUND.
$\overline{\text{S0}}$, $\overline{\text{S1}}$, $\overline{\text{S2}}$	1, 18-19	I	STATUS INPUT PINS: The status input pins from an 80C86, 80C88 or 8089 processor. The 82C89 decodes these pins to initiate bus request and surrender actions. (See Table 1)
CLK	17	I	CLOCK: From the 82C84A or 82C85 clock chip and serves to establish when bus arbiter actions are initiated.
$\overline{\text{LOCK}}$	16	I	LOCK: A processor generated signal which when activated (low) prevents the arbiter from surrendering the multi-master system bus to any other bus arbiter, regardless of its priority.
$\overline{\text{CRQLCK}}$	15	I	COMMON REQUEST LOCK: An active low signal which prevents the arbiter from surrendering the multi-master system bus to any other bus arbiter requesting the bus through the $\overline{\text{CBRQ}}$ input pin.
RESB	4	I	RESIDENT BUS: A strapping option to configure the arbiter to operate in systems having both a multi-master system bus and a Resident Bus. Strapped high, the multi-master system bus is requested or surrendered as a function of the SYSB/$\overline{\text{RESB}}$ input pin. Strapped low, the SYSB/$\overline{\text{RESB}}$ input is ignored.
ANYRQST	14	I	ANY REQUEST: A strapping option which permits the multi-master system bus to be surrendered to a lower priority arbiter as if it were an arbiter of higher priority (i.e., when a lower priority arbiter requests the use of the multi-master system bus, the bus is surrendered as soon as it is possible). When ANYRQST is strapped low, the bus is surrendered according to Table 1. If ANYRQST is strapped high and $\overline{\text{CBRQ}}$ is activated, the bus is surrendered at the end of the present bus cycle. Strapping $\overline{\text{CBRQ}}$ low and ANYRQST high forces the 82C89 arbiter to surrender the multi-master system bus after each transfer cycle. Note that when surrender occurs $\overline{\text{BREQ}}$ is driven false (high).
$\overline{\text{IOB}}$	2	I	IO BUS: A strapping option which configures the 82C89 Arbiter to operate in systems having both an IO Bus (Peripheral Bus) and a multi-master system bus. The arbiter requests and surrenders the use of the multi-master system bus as a function of the status line, $\overline{\text{S2}}$. The multi-master system bus is permitted to be surrendered while the processor is performing IO commands and is requested whenever the processor performs a memory command. Interrupt cycles are assumed as coming from the peripheral bus and are treated as an IO command
$\overline{\text{AEN}}$	13	O	ADDRESS ENABLE: The output of the 82C89 Arbiter to the processor's address latches, to the 82C88 Bus Controller and 82C84A or 82C85 Clock Generator. $\overline{\text{AEN}}$ serves to instruct the Bus Controller and address latches when to three-state their output drivers.
$\overline{\text{INIT}}$	6	I	INITIALIZE. An active low multi-master system bus input signal used to reset all the bus arbiters on the multi-master system bus. After initialization, no arbiters have the use of the multi-master system bus.
SYSB/$\overline{\text{RESB}}$	3	I	SYSTEM BUS/RESIDENT BUS: An input signal when the arbiter is configured in the System/Resident Mode (RESB is strapped high) which determines when the multi-master system bus is requested and multi-master system bus surrendering is permitted. The signal is intended to originate from a form of address-mapping circuitry, such as a decoder or PROM attached to the resident address bus. Signal transitions and glitches are permitted on this pin from ϕ1 of T4 to ϕ1 of T2 of the processor cycle. During the period from ϕ1 of T2 to ϕ1 of T4, only clean transitions are permitted on this pin (no glitches). If a glitch occurs, the arbiter may capture or miss it, and the multi-master system bus may be requested or surrendered, depending upon the state of the glitch. The arbiter requests the multi-master system bus in the System/Resident Mode when the state of the SYSB/$\overline{\text{RESB}}$ pin is high and permits the bus to be surrendered when this pin is low.

Pin Description

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
$\overline{\text{CBRQ}}$	12	I/O	COMMON BUS REQUEST: An input signal which instructs the arbiter if there are any other arbiters of lower priority requesting the use of the multi-master system bus. The $\overline{\text{CBRQ}}$ pins (open-drain output) of all the 82C89 Bus Arbiters which surrender to the multi-master system bus upon request are connected together. The Bus Arbiter running the current transfer cycle will not itself pull the $\overline{\text{CBRQ}}$ line low. Any other arbiter connected to the $\overline{\text{CBRQ}}$ line can request the multi-master system bus. The arbiter presently running the current transfer cycle drops its $\overline{\text{BREQ}}$ signal and surrenders the bus whenever the proper surrender conditions exist. Strapping $\overline{\text{CBRQ}}$ low and ANYRQST high allows the multi-master system bus to be surrendered after each transfer cycle. See the pin definition of ANYRQST.
$\overline{\text{BCLK}}$	5	I	BUS CLOCK: The multi-master system bus clock to which all multi-master system bus interface signals are synchronized.
$\overline{\text{BREQ}}$	7	O	BUS REQUEST: An active low output signal in the Parallel Priority Resolving Scheme which the arbiter activates to request the use of the multi-master system bus.
$\overline{\text{BPRN}}$	9	I	BUS PRIORITY IN: The active low signal returned to the arbiter to instruct it that it may acquire the multi-master system bus on the next falling edge of $\overline{\text{BCLK}}$ $\overline{\text{BPRN}}$ active indicates to the arbiter that it is the highest priority requesting arbiter presently on the bus. The loss of $\overline{\text{BPRN}}$ instructs the arbiter that it has lost priority to a higher priority arbiter
$\overline{\text{BPRO}}$	8	O	BUS PRIORITY OUT: An active low output signal used in the serial priority resolving scheme where $\overline{\text{BPRO}}$ is daisy-chained to $\overline{\text{BPRN}}$ of the next lower priority arbiter.
$\overline{\text{BUSY}}$	11	I/O	BUSY: An active low open-drain multi-master system bus interface signal used to instruct all the arbiters on the bus when the multi-master system bus is available. When the multi-master system bus is available the highest requesting arbiter (determined by $\overline{\text{BPRN}}$) seizes the bus and pulls $\overline{\text{BUSY}}$ low to keep other arbiters off of the bus. When the arbiter is done with the bus, it releases the $\overline{\text{BUSY}}$ signal, permitting it to go high and thereby allowing another arbiter to acquire the multi-master system bus

Functional Description

The 82C89 Bus Arbiter operates in conjunction with the 82C88 Bus Controller to interface 80C86, 80C88 processors to a multi-master system bus (both the 80C86 and 80C88 are configured in their max mode). The processor is unaware of the arbiter's existence and issues commands as though it has exclusive use of the system bus. If the processor does not have the use of the multi-master system bus, the arbiter prevents the Bus Controller (82C88), the data transceivers and the address latches from accessing the system bus (e.g. all bus driver outputs are forced into the high impedance state). Since the command sequence was not issued by the 82C88, the system bus will appear as "Not Ready" and the processor will enter wait states. The processor will remain in Wait until the Bus Arbiter acquires the use of the multi-master system bus whereupon the arbiter will allow the bus controller, the data transceivers, and the address latches to access the system. Typically, once the command has been issued and a data transfer has taken place, a transfer acknowledge (XACK) is returned to the processor to indicate "READY" from the accessed slave device. The processor then completes its transfer cycle. Thus the arbiter serves to multiplex a processor (or bus master) onto a multi-master system bus and avoid contention problems between bus masters.

Arbitration Between Bus Masters

In general, higher priority masters obtain the bus when a lower priority master completes its present transfer cycle. Lower priority bus masters obtain the bus when a higher priority master is not accessing the system bus. A strapping option (ANYRQST) is provided to allow the arbiter to surrender the bus to a lower priority master as though it were a master of higher priority. If there are no other bus masters requesting the bus, the arbiter maintains the bus so long as its processor has not entered the HALT State. The arbiter will not voluntarily surrender the system bus and has to be forced off by another master's bus request, the HALT State being the only exception. Additional strapping options permit other modes of operation wherein the multi-master system bus is surrendered or requested under different sets of conditions.

Priority Resolving Techniques

Since there can be many bus masters on a multi-master system bus, some means of resolving priority between bus masters simultaneously requesting the bus must be provided. The 82C89 Bus Arbiter provides several resolving techniques. All the techniques are based on a priority concept that at a given time one bus master will have priority above all the rest. There are provisions for using parallel priority resolving techniques, serial priority resolving techniques, and rotating priority techniques.

Parallel Priority Resolving

The parallel priority resolving technique uses a separate bus request line $\overline{\text{BREQ}}$ for each arbiter on the multi-master system bus, see Figure 1. Each $\overline{\text{BREQ}}$ line enters into a priority encoder which generates the binary address of the highest priority $\overline{\text{BREQ}}$ line which is active. The binary address is decoded by a decoder to select the corresponding $\overline{\text{BPRN}}$ (Bus Priority In) line to be returned to the highest priority requesting arbiter. The arbiter receiving priority ($\overline{\text{BPRN}}$ true) then allows its associated bus master onto the multi-master system bus as soon as it becomes available (i.e., the bus is no longer busy). When one bus arbiter gains priority over another arbiter it cannot immediately seize the bus, it must wait until the present bus transaction is complete. Upon completing its transaction the present bus occupant recognizes that it no longer has priority and surrenders the bus by releasing $\overline{\text{BUSY}}$. $\overline{\text{BUSY}}$ is an active low "OR" tied signal line which goes to every bus arbiter on the system bus. When $\overline{\text{BUSY}}$ goes inactive (high), the arbiter which presently has bus priority ($\overline{\text{BPRN}}$ true) then seizes the bus and pulls $\overline{\text{BUSY}}$ low to keep other arbiters off of the bus. See waveform timing diagram, Figure 2. Note that all multi-master system bus transactions are synchronized to the bus clock ($\overline{\text{BCLK}}$). This allows the parallel priority resolving circuitry or any other priority resolving scheme employed to settle.

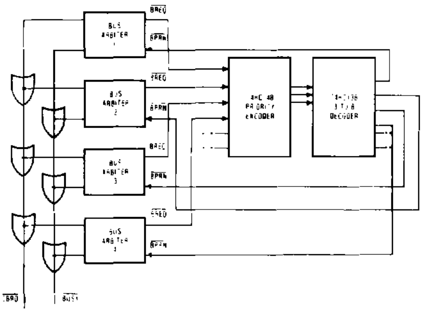
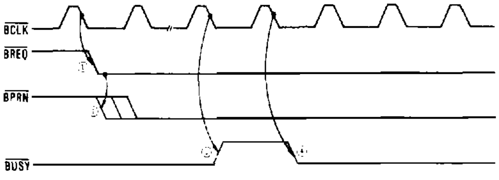


FIGURE 1. PARALLEL PRIORITY RESOLVING TECHNIQUE



NOTES

① Higher priority bus arbiter requests the Multi-Master system bus

② Attains priority.

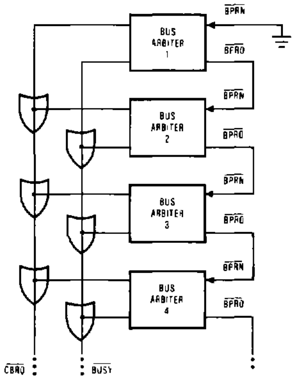
③ Lower priority bus arbiter releases $\overline{\text{BUSY}}$

④ Higher priority bus arbiter then acquires the bus and pulls $\overline{\text{BUSY}}$ down

FIGURE 2. HIGHER PRIORITY ARBITER OBTAINING THE BUS FROM A LOWER PRIORITY ARBITER

Serial Priority Resolving

The serial priority resolving technique eliminates the need for the priority encoder-decoder arrangement by daisy-chaining the bus arbiters together, connecting the higher priority bus arbiter's $\overline{\text{BPRO}}$ (Bus Priority Out) output to the $\overline{\text{BPRN}}$ of the next lower priority. See Figure 3.



NOTE

The number of arbiters that may be daisy-chained together in the serial priority resolving scheme is a function of $\overline{\text{BCLK}}$ and the propagation delay from arbiter to arbiter. Normally, at 10MHz only 3 arbiters may be daisy-chained.

FIGURE 3. SERIAL PRIORITY RESOLVING

Rotating Priority Resolving

The rotating priority resolving technique is similar to that of the parallel priority resolving technique except that priority is dynamically re-assigned. The priority encoder is replaced by a more complex circuit which rotates priority between requesting arbiters thus allowing each arbiter an equal chance to use the multi-master system bus, over time

Which Priority Resolving Technique To Use

There are advantages and disadvantages for each of the techniques described above. The rotating priority resolving technique requires substantial external logic to implement while the serial technique uses no external logic but can accommodate only a limited number of bus arbiters before the daisy-chain propagation delay exceeds the multi-master's system bus clock ($\overline{\text{BCLK}}$). The parallel priority resolving technique is in general a good compromise between the other two techniques. It allows for many arbiters to be present on the bus while not requiring too much logic to implement.

82C89 Modes Of Operation

There are two types of processors for which the 82C89 will provide support: An Input/Output processor (i.e. an NMOS 8089 IOP) and the 80C86, 80C88. Consequently, there are two basic operating modes in the 82C89 bus arbiter. One, the IOB (I/O Peripheral Bus) mode, permits the processor access to both an I/O Peripheral Bus and a multi-master system bus. The second, the RESB (Resident Bus mode), permits the processor to communicate over both a Resident Bus and a multi-master system bus. An I/O Peripheral Bus is a bus where all devices on that bus, including memory, are treated as I/O devices and are addressed by I/O commands. All memory commands are directed to another bus, the multi-master system bus. A Resident Bus can issue both memory and I/O commands, but it is a distinct and separate bus from the multi-master system bus. The distinction is that the Resident Bus has only one master, providing full availability and being dedicated to that one master.

The $\overline{\text{IOB}}$ strapping option configures the 82C89 Bus Arbiter into the $\overline{\text{IOB}}$ mode and the strapping option RESB configures it into the RESB mode. It might be noted at this point that if both strapping options are strapped false, the arbiter interfaces the processor to a multi-master system bus only (see Figure 4). With both options strapped true, the arbiter interfaces the processor to a multi-master system bus, a Resident Bus, and an I/O Bus.

In the $\overline{\text{IOB}}$ mode, the processor communicates and controls a host of peripherals over the Peripheral Bus. When the I/O Processor needs to communicate with system memory, it does so over the system memory bus. Figure 5 shows a possible I/O Processor system configuration.

The 80C86 and 80C88 processors can communicate with a Resident Bus and a multi-master system bus. Two bus controllers and only one Bus Arbiter would be needed in such a configuration as shown in Figure 6. In such a system configuration the processor would have access to

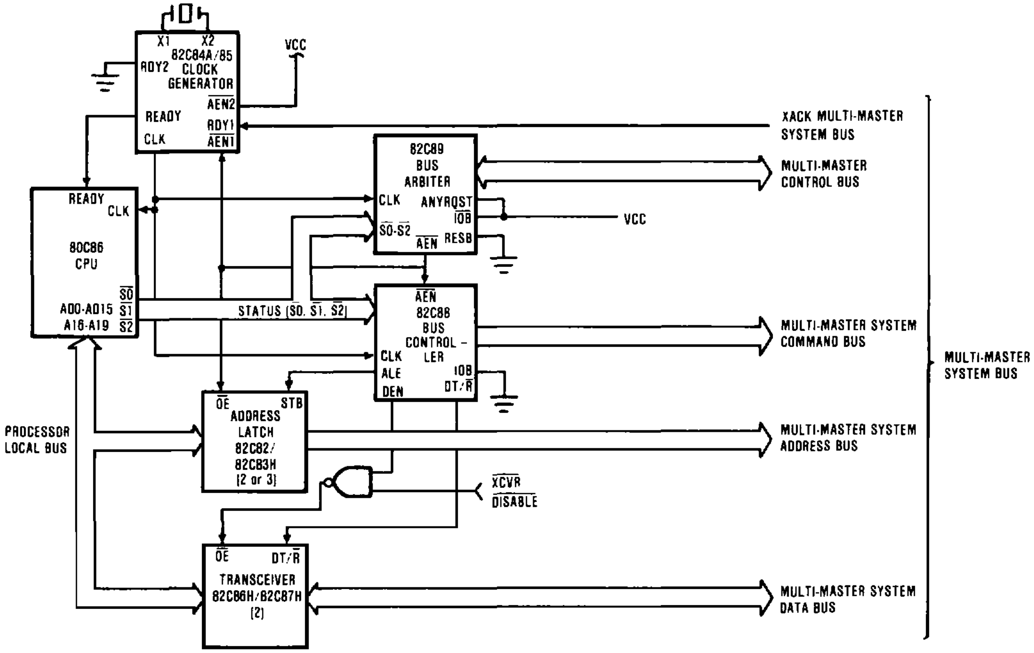


FIGURE 4. TYPICAL MEDIUM COMPLEXITY CPU SYSTEM

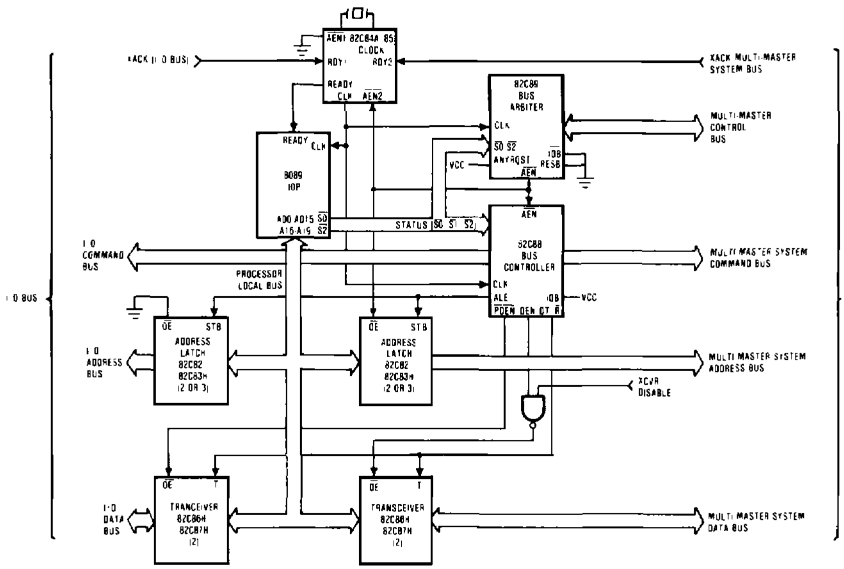


FIGURE 5. TYPICAL MEDIUM COMPLEXITY IOB SYSTEM

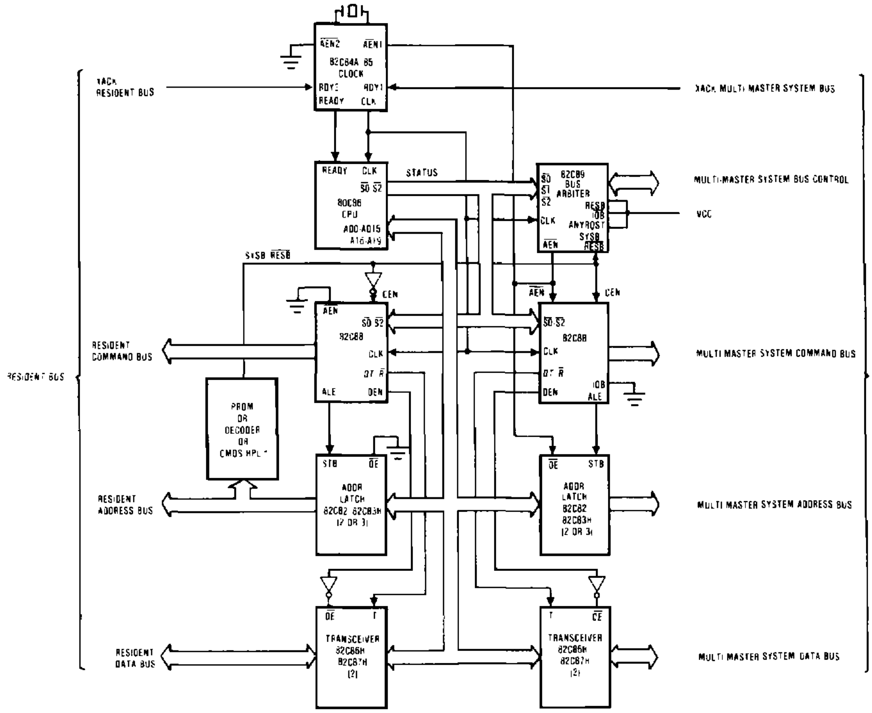


FIGURE 6. 82C89 BUS ARBITER SHOWN IN SYSTEM- RESIDENT BUS CONFIGURATION

* By adding another 82C89 arbiter and connecting its AEN to the 82C88 whose AEN is presently grounded The processor could have access to two multi-master buses

memory and peripherals of both busses. Memory mapping techniques are applied to select which bus is to be accessed. The SYSB/$\overline{\text{RESB}}$ input on the arbiter serves to instruct the arbiter as to whether or not the system bus is to be accessed. The signal connected to SYSB/$\overline{\text{RESB}}$ also enables or disables commands from one of the bus controllers.

A summary of the modes that the 82C89 has, along with its response to its status lines inputs, is shown in Table 1.

TABLE 1. SUMMARY OF 82C89 MODES, REQUESTING AND RELINQUISHING THE MULTI-MASTER SYSTEM BUS

STATUS LINES FROM 80C86 OR 80C88 OR 8088				IOB MODE ONLY $\overline{\text{IOB}}$ = LOW RESB = LOW	RESB MODE ONLY $\overline{\text{IOB}}$ = HIGH RESB = HIGH		IOB MODE RESB MODE $\overline{\text{IOB}}$ = LOW RESB = HIGH		SINGLE BUS MODE $\overline{\text{IOB}}$ = HIGH RESB = LOW
	$\overline{\text{S2}}$	$\overline{\text{S1}}$	$\overline{\text{S0}}$		SYSB/$\overline{\text{RESB}}$=High	SYSB/$\overline{\text{RESB}}$=Low	SYSB/$\overline{\text{RESB}}$=High	SYSB/$\overline{\text{RESB}}$=Low	
I/O COMMANDS	0	0	0	X	✓	X	X	X	✓
	0	0	1	X	✓	X	X	X	✓
	0	1	0	X	✓	X	X	X	✓
HALT	0	1	1	X	X	X	X	X	X
MEM COMMANDS	1	0	0	✓	✓	X	✓	X	✓
	1	0	1	✓	✓	X	✓	X	✓
	1	1	0	✓	✓	X	✓	X	✓
IDLE	1	1	1	X	X	X	X	X	X

NOTES

1 X = Multi-Master System Bus is allowed to be Surrendered.

2. ✓ = Multi-Master System Bus is Requested

MODE	PIN STRAPPING	MULTI-MASTER SYSTEM BUS	
		REQUESTED**	SURRENDERED*
Single Bus Multi-Master Mode	$\overline{\text{IOB}}$ = High RESB = Low	Whenever the processor's status lines go active	HLT + TI • $\overline{\text{CBRQ}}$ + HPBRQ †
RESB Mode Only	$\overline{\text{IOB}}$ = High RESB = High	SYSB/$\overline{\text{RESB}}$ = High • ACTIVE STATUS	(SYSB $\overline{\text{RESB}}$ = Low + TI) • $\overline{\text{CBRQ}}$ + HLT + HPBRQ
IOB Mode Only	$\overline{\text{IOB}}$ = Low RESB = Low	Memory Commands	(I/O Status + TI) • $\overline{\text{CBRQ}}$ + HLT + HPBRQ
IOB Mode RESB Mode	$\overline{\text{IOB}}$ = Low RESB = High	(Memory Command) • (SYSB/$\overline{\text{RESB}}$ = High)	((I O Status Commands) + SYSB/$\overline{\text{RESB}}$ = LOW) • $\overline{\text{CBRQ}}$ + HPBRQ † + HLT

NOTES:

* $\overline{\text{LOCK}}$ prevents surrender of Bus to any other arbiter $\overline{\text{CRQLCK}}$ prevents surrender of Bus to any lower priority arbiter

** Except for HALT and Passive or IDLE Status.

† HPBRQ. Higher priority Bus request or $\overline{\text{BPRN}}$ = 1.

1 $\overline{\text{IOB}}$ Active Low.

2. RESB Active High

3. + is read as "OR" and • as "AND"

4. TI = Processor Idle Status $\overline{\text{S2}}$, $\overline{\text{S1}}$, $\overline{\text{S0}}$ = 111

5. HLT = Processor Halt Status $\overline{\text{S2}}$, $\overline{\text{S1}}$, $\overline{\text{S0}}$ = 011

Absolute Maximum Ratings

Supply Voltage	+8.0 Volts
Input, Output or I/O Voltage Applied	GND -0.5V to VCC +0.5V
Maximum Package Power Dissipation	1 Watt
Storage Temperature Range	-65°C to +150°C
θ_{jc}	26°C/W (CERDIP package), 31°C/W (LCC package)
θ_{ja}	76°C/W (CERDIP package), 81°C/W (LCC package)
Gate Count	200 Gates
Junction Temperature	+150°C
Lead Temperature (Soldering, Ten Seconds)	+275°C

CAUTION: Stresses above those listed in the "ABSOLUTE MAXIMUM RATINGS" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied

Operating Conditions

Operating Voltage Range	+4.5V to +5.5V
Operating Temperature Range	
C82C89	0°C to +70°C
I82C89	-40°C to +85°C
M82C89	-55°C to +125°C

D.C. Electrical Specifications

VCC = 5.0V ± 10%; T_A = 0°C to +70°C (C82C89);
T_A = -40°C to +85°C (I82C89);
T_A = -55°C to +125°C (M82C89)

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
VIH	Logical One Input Voltage	2.0 2.2		V V	C82C89, I82C89 M82C89
VIL	Logical Zero Input Voltage		0.8	V	
VIHC	CLK Logical One Input Voltage	0.7 VCC		V	
VILC	CLK Logical Zero Input Voltage		0.2 VCC	V	
V_{OL}	Output Low Voltage $\overline{\text{BUSY}}$, $\overline{\text{CBRQ}}$ $\overline{\text{AEN}}$ $\overline{\text{BPRO}}$, $\overline{\text{BREQ}}$		 0.45 0.45 0.45	 V V V	 I_{OL} = 20mA I_{OL} = 16mA I_{OL} = 10mA
V_{OH}	Output High Voltage $\overline{\text{BUSY}}$, $\overline{\text{CBRQ}}$	Open-Drain			
	All other Outputs	3.0 VCC -0.4		V V	I_{OH} = -2.5mA I_{OH} = -100µA
II	Input Leakage Current	-1.0	1.0	µA	VIN = GND or VCC DIP Pins 1-6, 9, 14-19
IO	I/O Leakage	-10.0	10.0	µA	VO = GND orVCC DIP Pins 11-12
ICCSB	Standby Power Supply Current		10	µA	VCC = 5.5V VIN = VCC or GND Outputs Open
ICCOP	Operating Power Supply Current		1	mA/MHz	VCC = 5.5V Outputs Open See Note 1

NOTE 1 Maximum current defined by CLK or BCLK, whichever has the highest operating frequency

Capacitance T_A = 25°C

SYMBOL	PARAMETER	TYPICAL	UNITS	TEST CONDITIONS
C_{IN}	Input Capacitance	5	pF	FREQ = 1MHz, all measurements are referenced to device GND
C_{OUT}	Output Capacitance	15	pF	

A.C. Electrical Specifications

VCC = +5V ±10%, GND = 0V: T_A = 0°C to 70°C (C82C89)
T_A = -40°C to +85°C (I82C89)
T_A = -55°C to +125°C (M82C89)

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
(1) TCLCL	CLK Cycle Period	125		ns	Note 3
(2) TCLCH	CLK Low Time	55		ns	
(3) TCHCL	CLK High Time	35		ns	
(4) TSVCH	Status Active Setup	65	TCLCL-10	ns	
(5) TSHCL	Status Inactive Setup	50	TCLCL-10	ns	
(6) THVCH	Status Inactive Hold	10		ns	
(7) THVCL	Status Active Hold	10		ns	
(8) TBYSBL	$\overline{\text{BUSY}}$↑↓ Setup to $\overline{\text{BCLK}}$↓	20		ns	
(9) TCBSBL	$\overline{\text{CBRQ}}$↑↓ Setup to $\overline{\text{BCLK}}$↓	20		ns	
(10) TBLBL	$\overline{\text{BCLK}}$ Cycle Time	100		ns	
(11) TBHCL	$\overline{\text{BCLK}}$ High Time	30	0.65(TBLBL)	ns	
(12) TCLLL1	$\overline{\text{LOCK}}$ Inactive Hold	10		ns	
(13) TCLLL2	$\overline{\text{LOCK}}$ Active Setup	40		ns	
(14) TPNBL	$\overline{\text{BPRN}}$↑↓ to $\overline{\text{BCLK}}$ Setup Time	20		ns	
(15) TCLSR1	SYSB/$\overline{\text{RESB}}$ Setup	0		ns	
(16) TCLSR2	SYSB/$\overline{\text{RESB}}$ Hold	30		ns	
(17) TIVIH	Initialization Pulse Width	3 TBLBL+ 3 TCLCL		ns	
(18) TBLBRL	$\overline{\text{BCLK}}$ to $\overline{\text{BREQ}}$ Delay↑↓		35	ns	
(19) TBLPOH	$\overline{\text{BCLK}}$ to $\overline{\text{BPRO}}$↑↓		35	ns	Note 1 and 3
(20) TPNPO	$\overline{\text{BPRN}}$↓↑ to $\overline{\text{BPRO}}$↓↑ Delay		22	ns	Note 1 and 3
(21) TBLBYL	$\overline{\text{BCLK}}$ to $\overline{\text{BUSY}}$ Low		60	ns	Note 3
(22) TBLBYH	$\overline{\text{BCLK}}$ to $\overline{\text{BUSY}}$ Float		35	ns	Note 2 and 3
(23) TCLAEH	CLK to $\overline{\text{AEN}}$ High		65	ns	Note 3
(24) TBLAEL	$\overline{\text{BCLK}}$ to $\overline{\text{AEN}}$ Low		40	ns	Note 3
(25) TBLCBL	$\overline{\text{BCLK}}$ to $\overline{\text{CBRQ}}$ Low		60	ns	Note 3
(26) TBLCBH	$\overline{\text{BCLK}}$ to $\overline{\text{CBRQ}}$ Float		35	ns	Note 2 and 3
(27) TOLOH	Output Rise Time		20	ns	From 0.8V to 2 0V Note 4
(28) TOHOL	Output Fall Time		12	ns	From 2.0V to 0.8V Note 4
(29) TILIH	Input Rise Time		20	ns	From 0.8V to 2.0V
(30) TIHIL	Input Fall Time		20	ns	From 2 0V to 0.8V

NOTES:
1. $\overline{\text{BCLK}}$ generates the first $\overline{\text{BPRO}}$ wherein subsequent $\overline{\text{BPRO}}$ changes lower in the chain are generated through BPRON
2. Measured at 0.5V above GND.
3. All A C parameters tested as per test circuits in Figures 7—9 Input rise and fall times are driven at 1ns/V
4. Except $\overline{\text{BUSY}}$ and $\overline{\text{CBRQ}}$.

A.C. Test Circuits

*Includes stray and jig capacitance

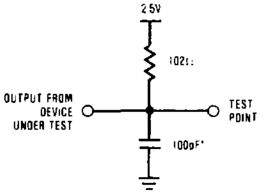


FIGURE 7.
BUSY, CBRQ LOAD CIRCUIT

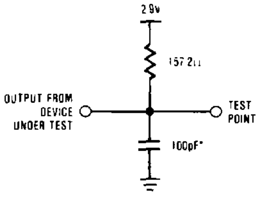


FIGURE 8.
AEN LOAD CIRCUIT

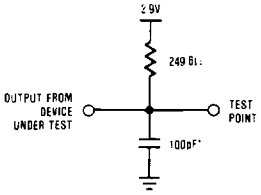
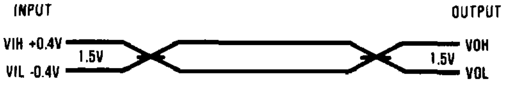


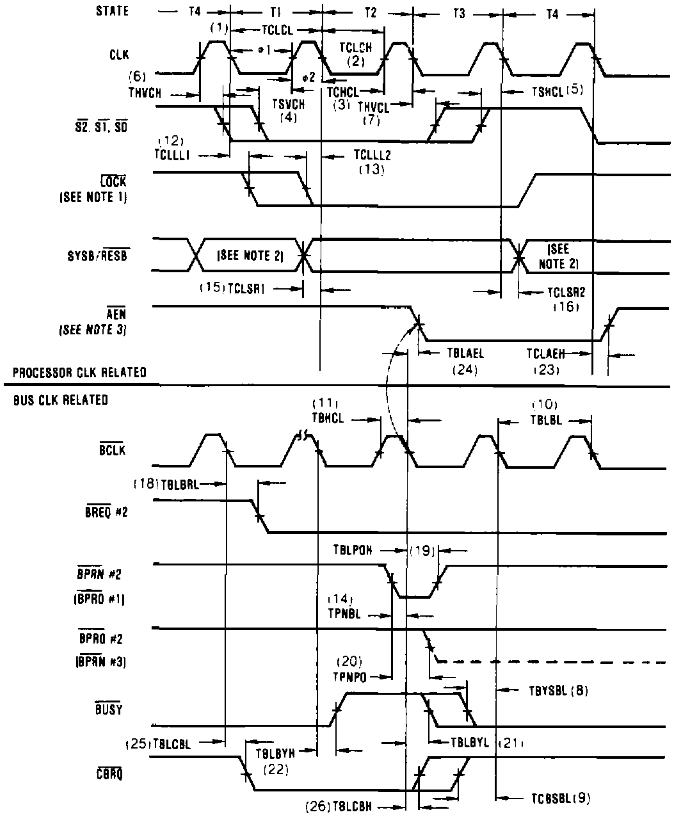
FIGURE 9.
BPRO, BREQ LOAD CIRCUIT

A.C. Testing Input, Output Waveforms



A C Testing. Inputs are driven at VIH +0.4V for a logic "1" and VIL -0.4V for a logic "0" The clock is driven at 4.1V and 0.4V. Timing measurements are made at 1.5V for both a logic "1" and "0"

Waveforms



NOTES:

1 $\overline{\text{Lock}}$ active can occur during any state as long as the relationships shown above with respect to the CLK are maintained. $\overline{\text{LOCK}}$ inactive has no critical time and can be asynchronous $\overline{\text{CRQLCK}}$ has no critical timing and is considered an asynchronous input signal.

2. Glitching of SYSB $\overline{\text{RESB}}$ is permitted during this time After $\phi2$ of T1, and before $\phi1$ of T4, SYSB $\overline{\text{RESB}}$ should be stable to maintain system efficiency

3 $\overline{\text{AEN}}$ leading edge is related to $\overline{\text{BCLK}}$ trailing edge to CLK The trailing edge of $\overline{\text{AEN}}$ occurs after bus priority is lost

ADDITIONAL NOTES

The signals related to CLK are typical processor signals and do not relate to the depicted sequence of events of the signals referenced to $\overline{\text{BCLK}}$ The signals shown related to the $\overline{\text{BCLK}}$ represent a hypothetical sequence of events for illustration Assume 3 bus arbiters of priorities 1,2 and 3 configured in serial priority resolving scheme (as shown in Figure 3) Assume arbiter 1 has the bus and is holding $\overline{\text{BUSY}}$ low Arbiter #2 detects its processor wants the bus and pulls low $\overline{\text{BREQ}}$ #2 If $\overline{\text{BPRN}}$ #2 is high (as shown) arbiter #2 will pull low $\overline{\text{CPRQ}}$ line $\overline{\text{CBRQ}}$ signals to the higher priority arbiter #1 that a lower priority arbiter wants the bus [A higher priority arbiter would be granted BPRN when it makes the bus request rather than having to wait for another arbiter to release the bus through $\overline{\text{CBRQ}}$] *Arbiter #1 will relinquish the multi-master system bus when it enters a state not requiring it (see Table 1), by lowering its $\overline{\text{BPRO}}$ #1 (tied to $\overline{\text{BPRN}}$ #2) and releasing $\overline{\text{BUSY}}$ Arbiter #2 now sees that is has priority from $\overline{\text{BPRN}}$ #2 being low and releases $\overline{\text{CBRQ}}$ As soon as BUSY signifies the bus is available (high) arbiter #2 pulls $\overline{\text{BUSY}}$ low on next falling edge of $\overline{\text{BCLK}}$ Note that if arbiter #2 didn't want the bus at the time it received priority it would pass priority to the next lower priority arbiter by lowering its $\overline{\text{BPRO}}$ #2 [TPNPO]

*Note that even a higher priority arbiter which is acquiring the bus through $\overline{\text{BPRN}}$ will momentarily drop $\overline{\text{CBRQ}}$ until it has acquired the bus