MC68488

General Purpose Interface Adapter

The MC68488 GPIA provides the means to interface between the IEEE-488 standard instrument bus and the M6800 MPU Family. The GPIB instrument bus provides a means of controlling and moving data between instruments connected to it.

The MC68488 will automatically handle all handshake protocol needed on the instrument bus.

- Single- or Dual-Primary Address Recognition
- Secondary Address Capability (Talker or Listener)
- Complete Source and Acceptor Handshakes
- Programmable Interrupts
- RFD Holdoff to Prevent Data Overrun
- Operates with DMA Controller
- Serial- and Parallel-Polling Capability
- Talk-Only or Listen-Only Capability
- Selectable Automatic Features to Minimize Software
- Synchronization Trigger Output
- M6800 Bus Compatible

3

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

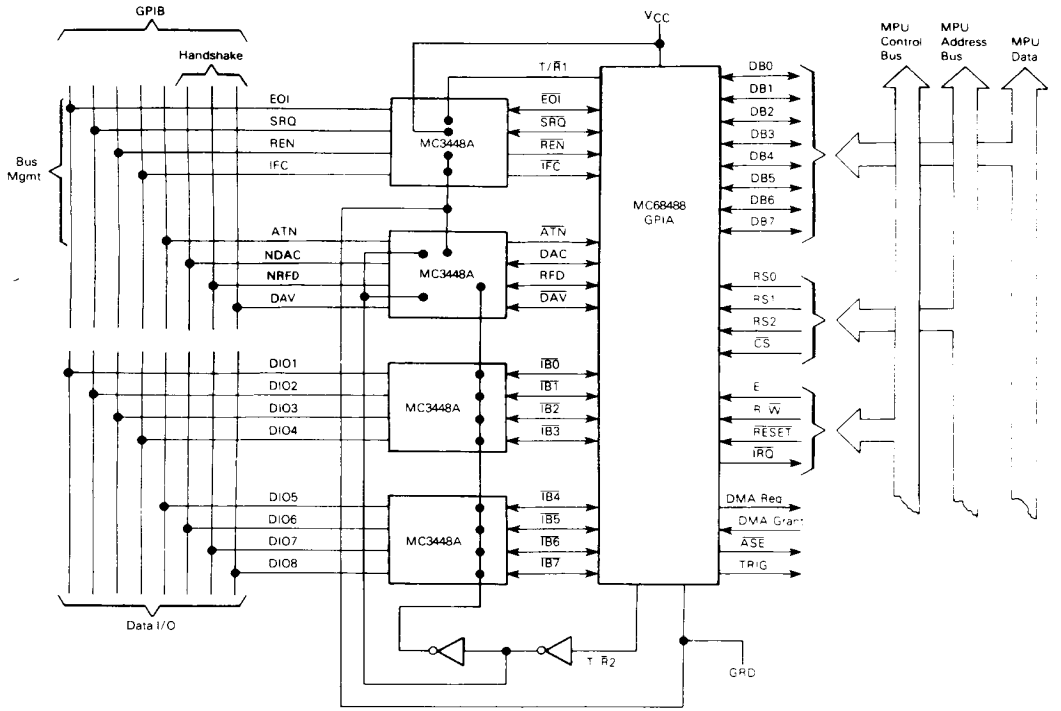
This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance	θ_{JA}		°C/W
Cerdip		60	
Plastic		100	

This document contains information on a new product. Specifications and information herein are subject to change without notice.

FIGURE 1 — GPIB INTERFACE

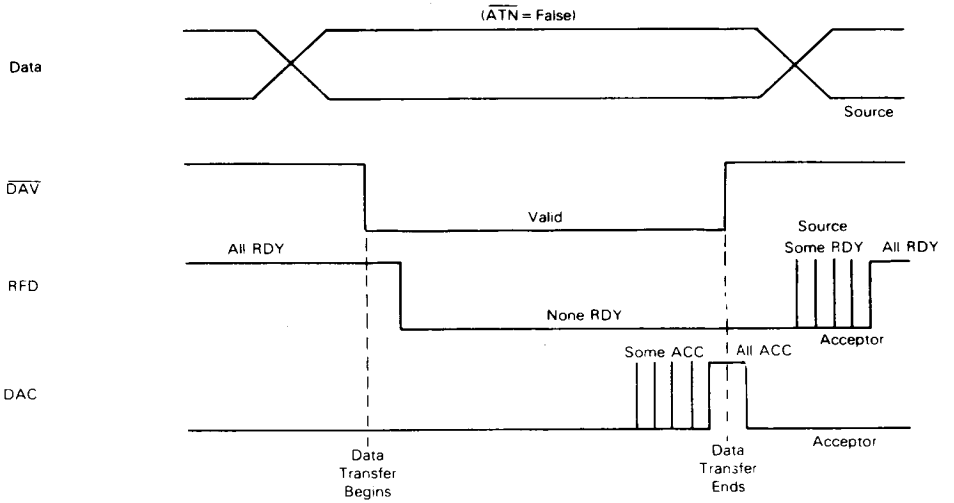


Note: The four MC3448A quad bus transceivers can be replaced by two MC3447 octal bus transceivers

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage		V_{IL}	V_{SS} – 0.3	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μA
Three State (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
DC Output High Voltage (I_{load} = – 205 μA)	D0-D7	V_{OH}	V_{SS} + 2.4	–	–	V
DC Output Low Voltage (I_{Load} = 1.6 mA) (I_{Load} = 3.2 mA)	D0-D7 SRQ, IRQ	V_{OL}	– –	– –	V_{SS} + 0.4 V_{SS} + 0.4	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	SRQ, IRQ	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation		P_{INT}	–	600	750	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7 All Others	C_{in}	– –	– –	12.5 7.5	pF

FIGURE 2 — SOURCE AND ACCEPTOR HANDSHAKE



This diagram displays logical voltage levels on the MC68488 pins. The MC68488 pins are labeled as the complement of the specified 488 bus callout; i.e., $\overline{DAV}$ rather than DAV. RFD rather than NRFD and DAC rather than NDAC. This was done to stay with standard positive logic format, which is used with all M6800 family devices.

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

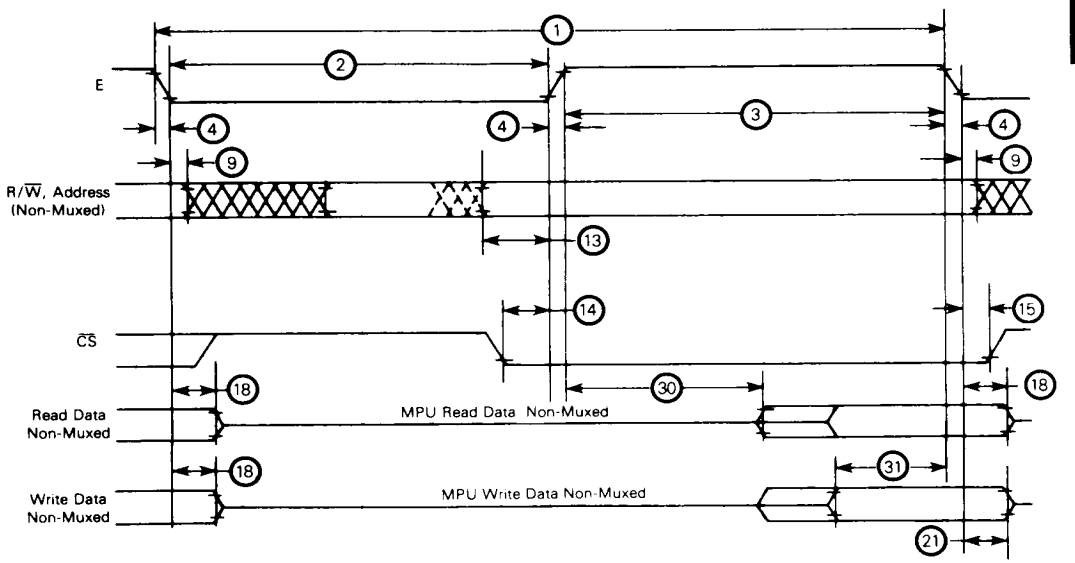
BUS TIMING (See Notes 1, 2, and 3)

Ident. Number	Characteristics	Symbol	MC68488		MC68A488*		MC68B488*		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50**	20	50**	20	50**	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*See Table 1 for GPIB transceiver considerations when using MC68A488 or MC68B488.

**The data bus output buffers are no longer sourcing or sinking current by t_{DHR} maximum (high-impedance)

FIGURE 3 — BUS TIMING



NOTES:
1. Not all signals are applicable to every part.
2. Voltage levels shown are $V_L \leq 0.8$ V, $V_H \geq 2.4$ V, unless otherwise specified.
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FIGURE 4 — OUTPUT BUS TIMING

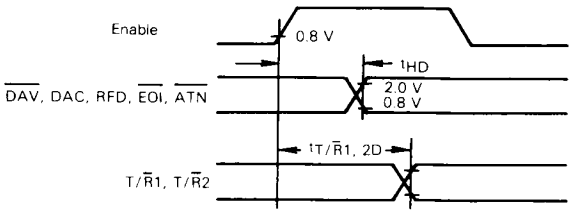


TABLE 1 — AC TIME VALUES

Characteristics		Symbol*	Typ	Unit
Settling Time for Multiple Message	SH	T_1	≥ 2	μs**
Response to ATN	SH, AH, T, L	t_3	≤ 200	ns
Interface Message Accept Time‡	AH	T_3	> 0	§
Response to IFC or REN False	T, TE, L, LE	t_4	< 100	μs
Response to ATN•EOI	PP	t_5	≤ 200	ns

* Time values specified by a lower case t indicate the maximum time allowed to make a state transition. Time values specified by an upper case T indicate the minimum time that a function must remain in a state before exiting.

** If three-state drivers are used on the DIO-DAV and EOI lines, T_1 may be:
(1) ≥ 1100 ns
(2) Or ≥ 700 ns if it is known that within the controller ATN is driven by a three-state driver.
(3) Or ≥ 500 ns for all subsequent bytes following the first sent after each false transition of ATN (the first byte must be sent in accordance with (1) or (2)).

‡ Time required for interface functions to accept, not necessarily respond to interface messages.

§ Implementation dependent.

When using an E clock of 1.5 MHz on the MC68A488, the GPIB data lines, DAV, and EOI lines must have three-state drivers – See Note **
When using an E clock of 2.0 MHz on the MC68B488 the GPIB data lines, DAV, EOI, and ATN lines must have three-state drivers – See Note **.

GENERAL DESCRIPTION

The IEEE-488 instrument bus standard is a bit-parallel, byte-serial bus structure designed for communication to and from intelligent instruments. Using this standard, many instruments may be interconnected, remotely and automatically controlled, or programmed. Data may be taken from, sent to, or transferred between instruments. A bus controller dictates the role of each device by making the attention line true and sending talk or listen addresses on the instrument bus data lines; those devices which have matching addresses are activated. Device addresses are set into each GPIA from switches or jumpers on a PC board by a microprocessor as a part of the initialization sequence.

When the controller makes the attention line true, instrument bus commands may also be sent to single or multiple GPIAs.

Information is transmitted on the instrument bus data lines under sequential control of the three handshake lines. No step in the sequence can be initiated until the previous step is completed. Information transfer can proceed as fast as the devices can respond, but no faster than the slowest device presently addressed as active. This permits several devices of different speeds to receive the same data concurrently.

The GPIA is designed to work with standard 488-bus driver ICs (MC3447As or MC3448As) to meet the complete electrical specifications of the IEEE-488 bus. Additionally, a powered-off instrument may be powered-on without disturbing the 488 bus. With some additional logic, the GPIA could be used with other microprocessors.

The MC68488 GPIA has been designed to interface between the M6800 family microprocessor and the complex protocol of the IEEE-488 instrument bus. Many instrument bus protocol functions are handled automatically by the GPIA and require no additional MPU action. Other functions require minimum MPU response due to a large number of internal registers conveying information on the state of the GPIA and the instrument bus.

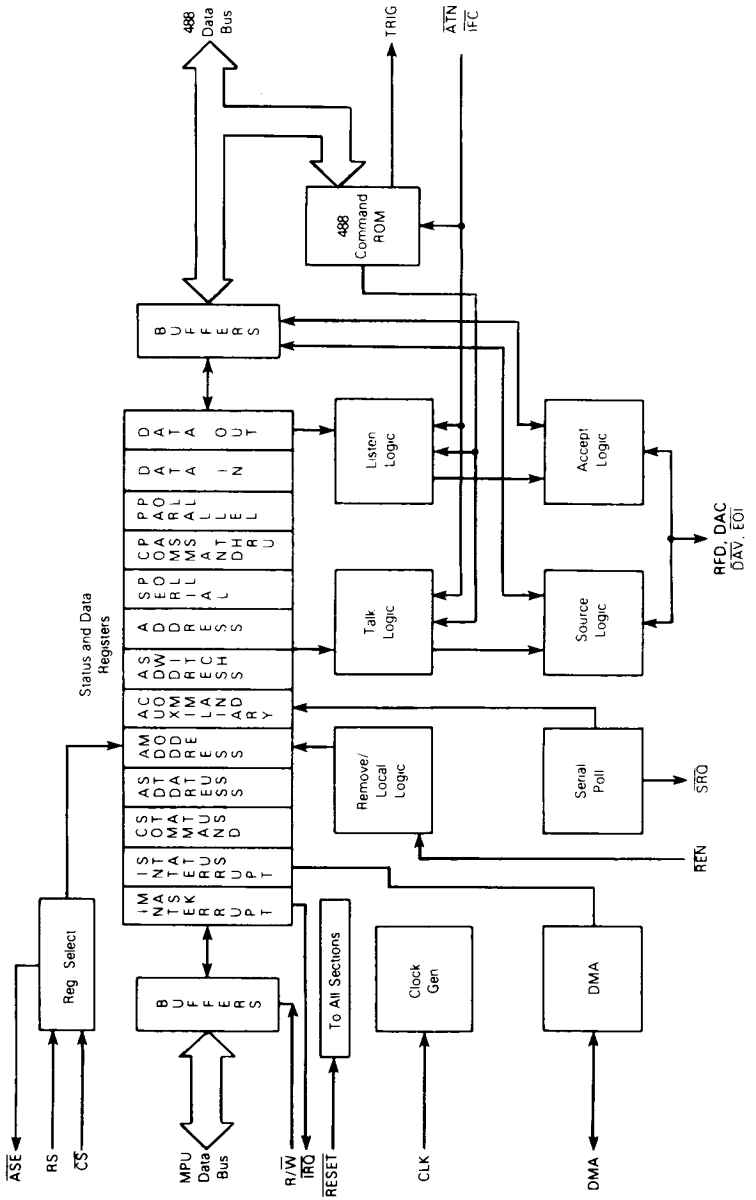
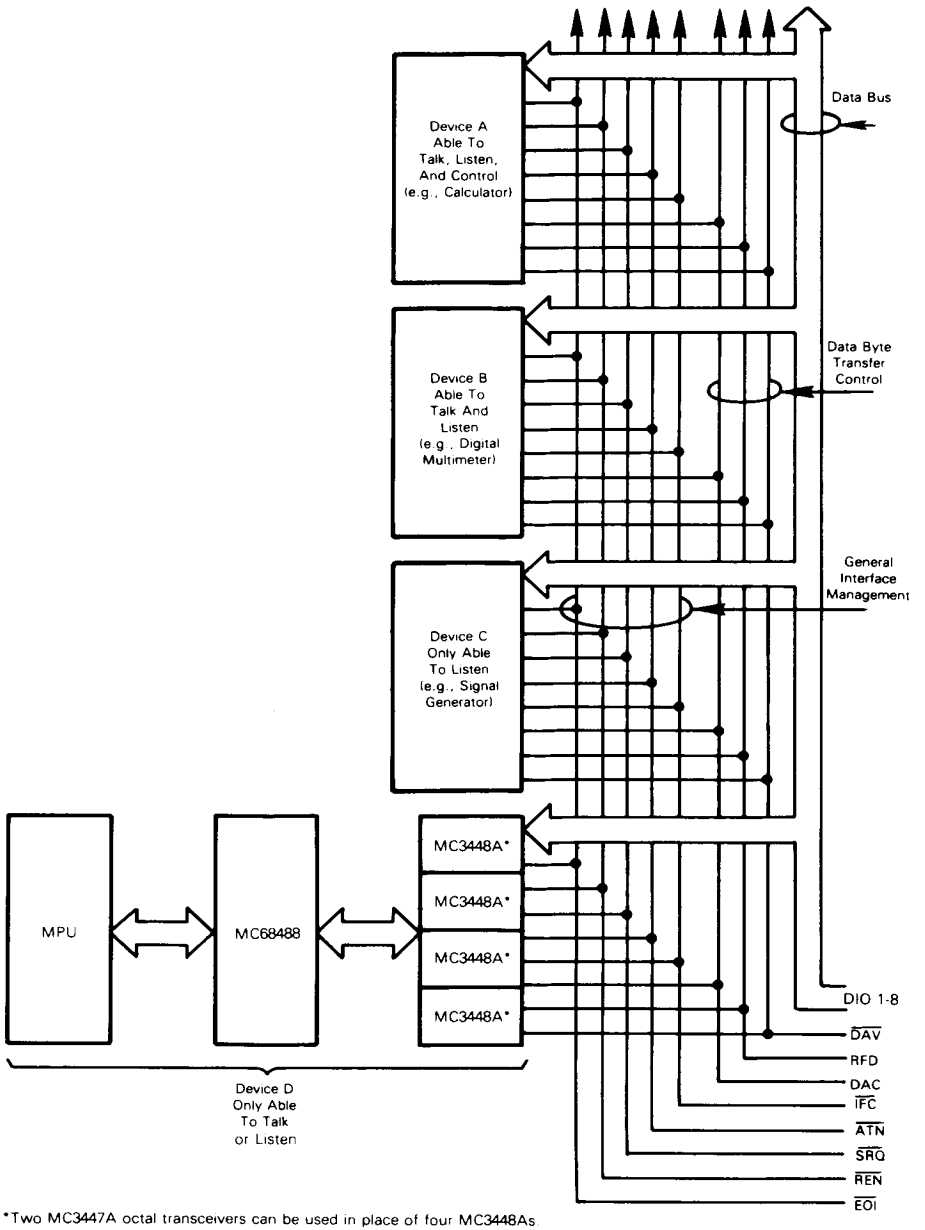


FIGURE 5 — GPIA BLOCK DIAGRAM

3

FIGURE 6 — GPIB SYSTEM



*Two MC3447A octal transceivers can be used in place of four MC3448As

PIN DESCRIPTION

All inputs to the GPIA are high impedance and TTL compatible. All outputs from the GPIA are compatible with standard TTL. $\overline{IRQ}$ (Interrupt Request) and $\overline{SRQ}$, however, are open-drain outputs (no internal pullup).

INTERFACE WITH MPU

BIDIRECTIONAL DATA (D0-D7) — The bidirectional data lines allow the transfer of data between the MPU and GPIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a GPIA read operation or the DMA controller performs a memory write operation. The Read/Write line is high when the GPIA is selected for a read operation.

CHIP SELECT ($\overline{CS}$) — This input signal is used to select the GPIA. $\overline{CS}$ must be low for selection of the device. Chip Select decoding is normally accomplished with external logic.

READ/WRITE INPUT (R/$\overline{W}$) — This signal is generated by the MPU or DMA controller to control register access and direction of data transfer on the data bus. A low state on the GPIA Read/Write and DMA Grant lines allows for the selection of one of seven write-only reigsters when used in conjunction with register select lines RS0, RS1, and RS2. A high state on the GPIA Read/Write and low state on the DMA Grant line allows for the selection of one of eight read-only registers when used in conjunction with register select lines RS0, RS1, and RS2.

REGISTER SELECT (RS0, RS1, RS2) — The three register select inputs are used to select the various registers inside the GPIA. These three lines are used in conjunction with the Read/Write line to select a particular register that is to be written or read. Table 2 shows the register select coding.

INTERRUPT REQUEST ($\overline{IRQ}$) — The $\overline{IRQ}$ output goes to the common interrupt bus line for the MPU. This is an open-drain output which is wire-ORed to the $\overline{IRQ}$ bus line. The $\overline{IRQ}$ is asserted low when an enable interrupt occurs and stays low until the MPU reads the interrupt status register. Reading R0R will reset $\overline{IRQ}$ to the high state.

TABLE 2 — REGISTER ACCESS

RS2	RS1	RS0	R/$\overline{W}$	Register Title	Register Symbol
0	0	0	1	Interrupt Status	R0R
0	0	0	0	Interrupt Mask	R0W
0	0	1	1	Command Status	R1R
0	0	1	0	Unused	–
0	1	0	1	Address Status	R2R
0	1	0	0	Address Mode	R2W
0	1	1	1	Auxiliary Command	R3R
0	1	1	0	Auxiliary Command	R3W
1	0	0	1	Address Switch*	R4R
1	0	0	0	Address	R4W
1	0	1	1	Serial Poll	R5R
1	0	1	0	Serial Poll	R5W
1	1	0	1	Command Pass-Through	R6R
1	1	0	0	Parallel Poll	R6W
1	1	1	1	Data In	R7R
1	1	1	0	Data Out	R7W

*External to MC68488

$\overline{RESET}$ — The $\overline{RESET}$ input provides a means of resetting the GPIA from a hardware source. In the low state, the $\overline{RESET}$ input causes the following:

- The Interrupt "Mask" register is reset;
- All status conditions are reset;
- The GPIA is placed in the Untalk/Unlisten state;
- The Parallel Poll, Serial Poll, Data In, and Data Out registers are reset;
- The Address register and Address mode register are cleared;
- All stored conditions in the Auxiliary Command register except bit 7 are reset — (bit 7 is set);
- T/R1, 2 will go to the low state.

When $\overline{RESET}$ returns high (the inactive state) the GPIA will remain in the reset condition until the MPU writes bit 7 of the Auxiliary Command register (R3W) low. Prior to the release of the software reset bit, the only register that can be accessed is the Address register. The conditions affected by the $\overline{RESET}$ pin cannot be changed while this pin is low.

E (ENABLE CLOCK) — E activates the address inputs (CS, RS0, RS1, and RS2) and R/$\overline{W}$ input and enables data transfer on the MPU data bus. It is also used internally as a state counter allowing the device to change interface states. The E input should be connected to a free-running clock source such as the MC6800 ϕ2 or the Enable Signal of other M6800 family MPUs.

GPIA/GPIB INTERFACE BUS SIGNALS

The GPIA provides a set of eighteen interface signal lines between the M6800 and the IEEE-488 Standard bus.

NOTE

The IEEE-488 Standard defines these signals as negative logic. In this document all MPU and MC68488 signals are defined as positive logic.

SIGNAL LINES ($\overline{IB0}$-$\overline{IB7}$) — These bidirectional lines allow for the flow of 7-bit ASCII interface messages and device dependent messages. Data appears on these lines in a bit-parallel byte-serial form. These lines are buffered by transceivers and applied to the IEEE-488 Standard bus (DIO1-DIO8).

BYTE TRANSFER LINES (DAC, RFD, $\overline{DAV}$) — These lines allow for proper transfer of each data byte on the bus between sources and acceptors. RFD goes passively high indicating "Ready For Data". A source will indicate the "data is valid" by pulling $\overline{DAV}$ low. Upon the reception of valid data, DAC will go passively high indicating that the "data has been accepted" by all acceptors. The handshake lines have internal pullup resistors.

BUS MANAGEMENT LINES ($\overline{ATN}$, $\overline{IFC}$, $\overline{SRQ}$, $\overline{EOI}$, $\overline{REN}$) — These lines are used to manage an orderly flow of information across the interface lines.

ATTENTION ($\overline{ATN}$) — is continuously monitored by the GPIA. The device responds to any changes on this line in less than 200 ns by activating the transmit/receive control signals. If the $\overline{EOI}$ line and $\overline{ATN}$ are low at the same time, GPIA will place the contents of a parallel poll register on the IEEE-488 Standard bus.

INTERFACE CLEAR ($\overline{IFC}$) – is used by a system controller to put the GPIA in a known quiescent state. The occurrence of $\overline{IFC}$ will place the GPIA in the Listener/Talker idle state (LIDS or TIDS). If the MC68488 is in a Listener Active state with a byte of data in the Data-In register (BI bit set) an $\overline{IFC}$ will place the part in LIDS but will not destroy the received byte nor the status indication (BI). Any interface function that requires the device to be in either the Listener or Talker Active state (e.g., a Serial Poll enable command) will be reset if an $\overline{IFC}$ occurs. A command that originates from the MPU (e.g., to, lo, fget, hida) will only be affected during the occurrence of an $\overline{IFC}$ (when $\overline{IFC}$ is low) and will return to its programmed state when $\overline{IFC}$ returns high; i.e., $\overline{IFC}$ will not affect local messages. For example: if the GPIA is in TACS (Talker Active State) and has placed a byte in the Data-Out register it has made a new byte available (nba). If $\overline{IFC}$ occurs while the source handshake is in SDYS, the talker function will be returned to its idle state but nba (a local message) will not be destroyed. When the GPIA is again made a talker, the byte in the Data-Out register (placed there before $\overline{IFC}$) will be placed onto the GPIB. The address register is not affected by an $\overline{IFC}$.

SERVICE REQUEST ($\overline{SRQ}$) – is used to indicate a need for attention in addition to requesting an interruption in the current sequence of events. This indicates to the controller that a device on the bus is in need of service. This output becomes active low by setting the rsv bit (bit 6) of R5W. This line is an open drain and an external pullup resistor (nominal 3.3k ohm) must be used.

REMOTE ENABLE (REN) – is used to select one of two alternate sources of device programming data – local and remote control. When this input is low the GPIA is enabled to move to the REMS state. Note that REN being low is a necessary but not a sufficient condition for moving to REMS.

END OF IDENTIFY ($\overline{EOI}$) – Serves a dual purpose. When the GPIA is in TACS and the MPU writes bit 5 or R3W (feoi) this pin becomes an output and signals the end of a multibyte transfer. If the system controller makes the $\overline{EOI}$ line true in conjunction with $\overline{ATN}$, the contents of the Parallel Poll register will be placed on the IEEE-488 Standard bus.

TRANSMIT/RECEIVE CONTROL SIGNALS (T/$\overline{R}$1, T/$\overline{R}$2) – These two signals are used to control the quad or octal transceivers which drive the interface bus. It is assumed that transceivers equivalent to the MC3447 or MC3448A will be used where each transceiver has a separate Transmit/Receive control pin. These pins can support one TTL load each. The outputs can then be grouped and the control for $\overline{SRQ}$ hardwired high to transmit. The Transmit/Receive inputs of $\overline{REN}$, $\overline{IFC}$, and $\overline{ATN}$ are hardwired low to receive. $\overline{EOI}$ is controlled by T/$\overline{R}$1 through the MC3447/MC3448A (or equivalents) allowing it to transmit or receive. T/$\overline{R}$1 operates exactly as T/$\overline{R}$2 except during the parallel polling sequence. During parallel poll, $\overline{EOI}$ will be made an input by T/$\overline{R}$1 while $\overline{DAV}$ and IB0/IB7 lines are outputs.

SPECIAL CONTROL SIGNALS

DMA CONTROL LINES (DMA GRANT, DMA REQUEST – The DMA request line is used to signal a DMA controller that a data transfer is pending. The DMA request line is set high if either the BI or BO status bits are set in the Interrupt Status Register (R0R). The DMA request line is cleared when the DMA Grant is true. The DMA Grant line is used to signal the GPIA that the DMA has control of the MPU data and address lines. The DMA Grant, when set high, selects register 7. It also inhibits the RS0, RS1, and RS2 lines. During this time the $\overline{CS}$ input must be high. The DMA Grant also inverts the function of the R/$\overline{W}$ line making it $\overline{R}$/W. Thus, if the DMAC supplies a write function to a memory location, this same line will perform a read of the GPIA (R7R) and vice versa.

NOTE

DMA GRANT MUST BE GROUNDED WHEN NOT IN USE.

TRIGGER OUTPUT (TRIG) – The TRIG pin provides an output corresponding to the GET and fget commands. A hardware or software reset places this output at a low level. The trigger output can be programmed high by either of two methods:

1. Setting fget (bit 0 of R3W) by the MPU causes the trigger output to be set. It remains set until the fget bit is programmed low or until a reset occurs.
2. The Trigger Output is set upon reception of a GET command from the controller. It is reset when the GPIA moves out of DTAS (Device Trigger Active State); i.e., when GET, LADS, or ACDS occur.

ADDRESS SWITCH ENABLE ($\overline{ASE}$) – The $\overline{ASE}$ output is used to enable three-state buffers that connect instrument address switches to the MPU data bus. This output pin is pulsed low when the Address Switch Register of the GPIA is read (R4R), i.e., a read of R4R will drive the $\overline{ASE}$ line low for the E clock that is used to read R4R.

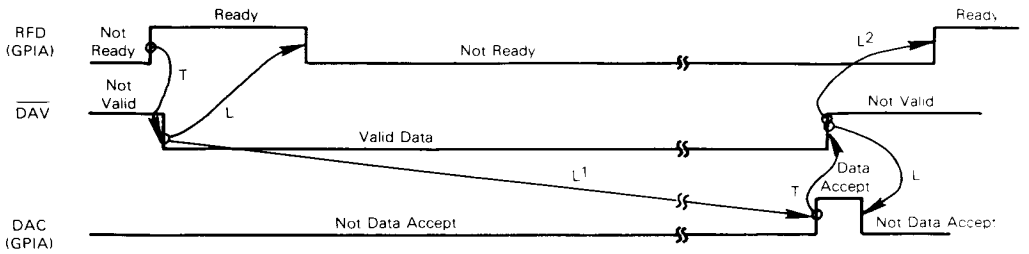
GPIB HANDSHAKE SEQUENCE

The GPIB handshake line transitions are debounced inside the GPIA with the E-clock to provide a high degree of noise immunity. Due to the asynchronous nature between the GPIB handshake line transitions and the internal debounce circuit sampling, the time required for handshake completion can vary by 1 E clock cycle.

LISTENER MODE – The handshake sequence begins when the GPIA makes RFD true (Figure 8). A second byte cannot be transferred on the GPIB until the GPIA again makes RFD true for the next handshake. The total time required by the GPIA to debounce all of the handshake lines in the appropriate time sequence is 7-8 E clock cycles. The 1 cycle variation is due to the asynchronous nature of the GPIB with respect to the GPIA debounce circuitry. To determine the maximum throughput rate add this number to the number of instructions or DMA cycles used to service each transfer.

TALKER MODE — The handshake sequence begins when the listener(s) on the GPIB make the RFD line true (Figure 9). When this occurs and the MPU has written a byte to R7W the GPIA will make the $\overline{DAV}$ line low indicating to the listeners that valid data is on the GPIB. When this byte is accepted and RFD is again made true the next transfer can begin. The GPIA debounce circuitry requires 6-7 E clock cycles to complete a handshake sequence. As with the listener there is a 1 E clock fluctuation due to the asynchronous nature between the GPIB handshake and the GPIA debounce circuitry. To determine the maximum throughput rate add this number to the number of instructions or DMA cycles used to service each transfer.

FIGURE 7 — GPIA IN LISTEN MODE*



* The GPIA in the listener mode controls the DAC and RFD lines. The $\overline{DAV}$ line is controlled by the Talker on the GPIB. Note that the RFD and DAC lines are wire ANDed on the GPIB; thus, these lines returning to the high state are dependent on all devices programmed as listeners releasing the lines.

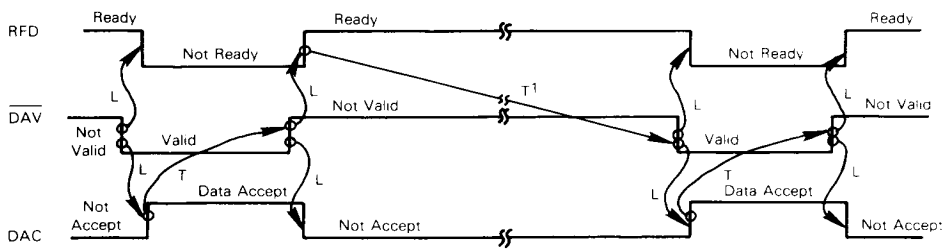
L The listener(s) on the GPIB causes this action.

T The talker on the GPIB causes this action.

1 The release of DAC may require action by the MPU (reading R7R for data byte transfers or writing dacr (R3W) high for certain commands). For some commands DAC is automatically released by the GPIA. Consult the MC68488 user's manual for details.

2 The RFD line is normally automatically released by the GPIA. Certain conditions, however, require MPU intervention to provide this release. Consult the MC68488 user's manual for details.

FIGURE 8 — GPIA IN TALKER MODE*



* The GPIA in the talker mode controls the $\overline{DAV}$ line. The RFD and DAC lines are controlled by the listener(s) on the GPIB.

L The listener(s) on the GPIB causes this action.

T The talker on the GPIB causes this action.

1 Two conditions must occur before the $\overline{DAV}$ line goes to the valid state. The RFD line must be high and a data byte must be placed in the data out register (nba must be true).

GPIA INTERNAL CONTROLS AND REGISTERS*

There are fifteen locations accessible to the MPU data bus which are used for transferring data to control the various functions on the chip and provide current chip status. Seven of these registers are write only and eight registers are read only. The various registers are accessed according to the three least-significant bits of the MPU address bus and the status of the Read/Write line. One of the fifteen registers is external to the IC but an address switch register is provided for reading the address switches. Table 2 shows actual bit contents of each of the registers.

DATA-IN REGISTER R7R — The data-in register is an actual 8-bit storage register used to move data from the interface bus when the chip is a listener. Reading the register does not destroy information in the data-out register. DAC (data accepted) will remain low until the MPU removes the bytes from the data-in register. The chip will automatically finish the handshake by allowing DAC to go high. In RFD (ready for data) holdoff mode, a new handshake is not initiated until a command is sent allowing the chip to release holdoff. This will delay a talker until the available information has been processed.

Data-In Register
(Read Only)

D17	D16	D15	D14	D13	D12	D11	D10

D10-D17 — Correspond to DIO1-DIO8 of the 488-1975 Standard and $\overline{IB0}$-$\overline{IB7}$ of the MC68488.

***NOTE**: Upper and lower case type designations will be used with the register bits to indicate remote or local messages respectively.

DATA-OUT REGISTER R7W — The data-out register is an actual 8-bit storage register used to move data out of the chip onto the interface bus. Reading from the data-in register has no effect on the information in the data-out register. Writing to the data-out register has no effect on the information in the data-in register.

Data-Out Register
(Write Only)

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

DO0-DO7 — Correspond to DIO1-DIO8 of the 488-1978 Standard and $\overline{IB0}$-$\overline{IB7}$ of the MC68488.

INTERRUPT MASK REGISTER R0W — The Interrupt Mask Register is a 7-bit storage register used to select the particular events that will cause an interrupt to be sent to the MPU. The seven control bits may be set independently of each other. If dsel (bit 7 of the Address Mode Register) is set high CMD bit 2 will interrupt on SPAS or RLC. If dsel is set low CMD will interrupt on UACG, UUCG, and DCAS in addition to RLC and SPAS. The Command Status Register R1R may then be used to determine which command caused the interrupt. Setting GET bit 5 allows an interrupt to occur on Group Execute Trigger Command. END bit 1 allows an interrupt to occur if $\overline{EOI}$ is true (low) and $\overline{ATN}$ is false (high). APT bit 3 allows an interrupt to occur indicating that a secondary address is available to be examined by the MPU if apte (bit 0 of Address Mode Register) is enabled and listener or talker primary address is received and a Secondary Command Group is received. A typical response for a valid secondary address would be to set msa (bit 3 of Auxiliary Command Register) true and dacr (bit 4 Auxiliary Command Register) true, releasing the DAC handshake. BI indicates that a data

TABLE 3 — REGISTER CONTENTS

	7	6	5	4	3	2	1	0	
R0W	IRQ	BO	GET		APT	CMD	END	BI	Interrupt "Mask Register"
R0R	INT	BO	GET		APT	CMD	END	BI	Interrupt Status Register
R1R	UACG	REM	LOK		RLC	SPAS	DCAS	UUCG	Command Status Register
R1W									Unused
R2R	ma	to	lo	ATN	TACS	LACS	LPAS	TPAS	Address Status Register
R2W	dsel	to	lo		hlde	hlda		apte	Address Mode Register
R3R	RESET	DAC	DAV	RFD	msa	rtl	ulpa	fget	Auxiliary Command Register
R3W		rfdr	feoi	dacr			dacd		Auxiliary Command Register
R4R	UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1	Address Switch Register
R4W	lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1	Address Register
R5R	S7	SRQS	S5	S4	S3	S2	S1	S0	Serial Poll Register
R5W		rsv							
R6R	B7	B6	B5	B4	B3	B2	B1	B0	Command Pass-Through Register
R6W	PPR8	PPR7	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	Parallel Poll Register
R7R	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Data In Register
R7W	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	Data Out Register

Notes:
1. Upper case letters indicate a message resulting from the IEEE-488 Standard bus.
2. Lower case letters indicate a message resulting from the MPU data bus.
3. The bit terminology of the Data In and Data registers represent the numbering of the IEEE-488 Standard bus and not the 6800 MPU bus — see Section 3.1.2.

byte is waiting in the data-in register. BI is set high when data-in register is full. BO indicates that a byte from the data-out register has been accepted. BO is set when the data-out register is empty. IRQ enabled high allows any interrupt to be passed to the MPU.

Interrupt Mask Register
(Write Only)

IRQ	BO	GET	X	APT	CMD	END	BI

IRQ – Mask bit for IRQ pin
BO – Interrupt on byte output
GET – Interrupt on Group Execute Trigger
APT – Interrupt on Secondary Address Pass-Through
CMD – Interrupt on SPAS + RLC + $\overline{dsel}$ (DCAS + UUCG + UACG)
END – Interrupt on EOI and $\overline{ATN}$
BI – Interrupt on byte input

THE INTERRUPT STATUS REGISTER R0R – The Interrupt Status Register is a 7-bit storage register which corresponds to the interrupt mask register with an additional bit INT bit 7. Except for the INT bit the other bits in the status register are set regardless of the state of the interrupt mask register when the corresponding event occurs. The $\overline{IRQ}$ (MPU interrupt) is cleared when the MPU reads from the register. INT bit 7 is the logical OR of the other six bits ANDed with the respective bit of R0W.

Interrupt Status Register
(Read Only)

INT	BO	GET	X	APT	CMD	END	BI

INT – Logical OR of all other bits in this register ANDed with the respective bits in the interrupt mask register.
BO – A byte of data has been output
GET – A Group Execute Trigger has occurred
APT – An Address Pass-Through has occurred
CMD – SPAS + RLC + $\overline{dsel}$ (DCAS + UUCG + UACG) has occurred
END – An EOI has occurred with $\overline{ATN} = 0$
BI – A byte has been received

SERIAL POLL REGISTER R4R/W – The Serial Poll Register is an 8-bit storage register which can be both written into and read by the MPU. It is used for establishing the status byte that the chip sends out when it is serial poll enabled. Status may be placed in bits 0 through 5 and bit 7. Bit 6 rsv (request for service) is used to drive the logic which controls the $\overline{SRQ}$ line on the bus telling the controller that service is needed. This same logic generated the signal SRQS which is substituted in bit 6 position when the status byte is read by the MPU $\overline{IB0}$-$\overline{IB7}$. In order to initiate a rsv (request for service), the MPU sets bit 6 true (generating rsv signal) and this in turn causes the chip to pull down the $\overline{SRQ}$ line. SRQS is the same as rsv when SPAS is false. Bit 6 as read by the MPU will be the SRQS (Service Request State).

Serial Poll Register
(Read)

S8	SRQS	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits.
SRQS – Bus in Service Request State

Serial Poll Register
(Write)

S8	rsv	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits
rsv – generate a service request

PARALLEL POLL REGISTER R6W – This register will be loaded by the MPU and the bits in this register will be delivered to the instrument bus IB0-IB7 during PPAS (Parallel Poll Active State). This register powers up in the PP0 (Parallel Poll No Capability) state. The reset bit (Auxiliary Command Register bit 7) will clear this register to the PP0 state.

The parallel poll interface function is executed by this chip using the PP2 subset (Omit Controller Configuration Capability). The controller cannot directly configure the parallel poll output of this chip. This must be done by the MPU. The controller will be able to indirectly configure the parallel poll by issuing an addressed command which has been defined in the MPU software.

Parallel Poll Register
(Write Only)

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1

Bits delivered to bus during Parallel-Poll Active State (PPAS)
Register powers up in the PP0 state
Parallel Poll is executed using the PP2 subset

ADDRESS MODE REGISTER R2W – The address mode register is a storage register with six bits for control: to, lo, hlde, hlda, dsel, and apte. The to bit 6 selects the talker/listener and addresses the chip to talk only. The lo bit 5 selects the talker/listener and sets the chip to listen only. The apte bit 0 is used to enable the extended addressing mode. If apte is set low the device goes from the TPAS (Talker Primary Address State) directly to the TADS (Talker Addressed State). The hlda bit 2 holds off RFD (Ready for Data) on ALL DATA until rfdr is set true. The hlde bit 3 holds off RFD on EOI enabled (low) and ATN not enabled (high). This allows the last byte in a block of data to be continually read as needed. Writing rfdr true (high) will allow the next handshake to proceed.

Address Mode Register
(Write Only)

dsel	to	lo	X	hdle	hdla	X	apte

dsel – configure for automatic completion of handshake sequence on occurrence of GET, UACG, UUCG, SDC, or DCL commands
to – set to talk-only mode
lo – set to listen-only mode
hdle – Hold-off RFD on end (END = EOI∧$\overline{ATN}$)
hdla – Hold-off RFD on all data
apte – Enable the address pass-through feature

3

ADDRESS STATUS REGISTER R2R – The address status register is not a storage register but simply an 8-bit port used to couple internal signal modes to the MPU bus. The status flags represented here are stored internally in the logic of the chip. These status bits indicate the addressed state of the talker/listener as well as flags that specify whether the chip is in the talk only or listen only mode. The ATN, bit 4, contains the condition of the Attention Line. The ma signal is true when the chip is in:

TACS – Talker Active State
TADS – Talker Addressed State
LACS – Listener Active State
LADS – Listener Addressed State
SPAS – Serial Poll Active State

Address Status Register
(Read Only)

ma	to	lo	ATN	TACS	LACS	LPAS	TPAS

ma – my address has occurred
to – the talk-only mode is enabled
lo – the listen-only mode is enabled
ATN – the Attention command is asserted
TACS – GPIA is in the Talker Active State
LACS – GPIA is in the Listener Active State
LPAS – GPIA is in the Listener Primary Addressed State
TPAS – GPIA is in the Talker Primary Addressed State

ADDRESS SWITCH REGISTER R4R – The address switch register is external to the chip. There is an enable line ($\overline{ASE}$) to be used to enable three-state drivers connected between the address switches and the MPU. When the MPU addresses the address switch register the $\overline{ASE}$ line directs the switch information to be sent to the MPU. The five least-significant bits of the 8-bit register are used to specify the bus address of the device and the remaining three bits may be used at the discretion of the user. The most probable use of one or two of the bits is for controlling the listener only or talk only functions.

Address Switch Register
(Read Only)

UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1

AD1-AD5 – Device address
UD1-UD3 – User definable bits

When this "register" is addressed, the $\overline{ASE}$ pin is set which allows external address switch information from the bus device to be read.

ADDRESS REGISTER R4W – The Address Register is an 8-bit storage register. The purpose of this register is to carry the primary address of the device. The primary address is placed in the five least-significant bits of the register. If external switches are used for device addressing these are normally read from the Address Switch Register and then placed in the Address Register by the MPU.

AD1 through AD5 bits 0-5 are for the device's address. The lsbe bit 7 is set to enable the Dual Primary Addressing Mode. During this mode the device will respond to two consecutive addresses, one address with AD1 equal to 0 and the other address with AD1 equal to 1. For example, if the device's address is HEX 0F, the Dual Primary Addressing Mode would allow the device to be addressed at both HEX 0F and HEX 0E. The dal bit 6 is set to disable the listener and the dat bit 5 is set to disable the talker.

This register is cleared by the $\overline{RESET}$ input only (not by the reset bit of the Auxiliary Command Register bit 7).

When $\overline{ATN}$ is enabled and the primary address is received on the $\overline{IB0\text{-}7}$ lines, the MC68488 will set bit 7 of the address status register (ma). This places the MC68488 in the TPAS or LPAS.

When $\overline{ATN}$ is disabled the GPIA may go to one of three states: TACS, LACS, or SPAS.

Address Register
(Write Only)

lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1

lsbe – enable dual primary addressing mode
dal – disable the listener
dat – disable the talker
AD1-AD5 – Primary device address, usually read from address switch register
Register is cleared by the $\overline{RESET}$ input pin only.

AUXILIARY COMMAND REGISTER R3R/W – Bit 7, reset, initializes the chip to the following states: (reset is set true by external $\overline{RESET}$ input pin and by writing into the register from the MPU).

SIDS – Source Idle State
AIDS – Acceptor Idle State
TIDS – Talker Idle State
LIDS – Listener Idle State
LOCS – Local State
NPRS – Negative Poll Response State
PPIS – Parallel Poll Idle State
PUCS – Parallel Poll Unaddressed to Configure State
PPO – Parallel Poll No capability

rfdr (release RFD handshake) bit 6 allows for completion of the handshake that was stopped by RFD (Ready For Data) holdoff commands hlda and hlde.

fget (force group execute trigger) bit 0 has the same effect as the GET (Group Execute Trigger) command from the controller.

rtl (return to local) bit 2 allows the device to respond to local controls and the associated device functions are operative.

dacr (release DAC handshake) bit 4 is set high to allow DAC to go passively true. This bit is set to indicate that the MPU has examined a secondary address or an undefined command.

upla (upper/lower primary address) bit 1 will indicate the state of the LSB of the address received on the DIO1-8 bus lines at the time the last Primary Address was received. This bit can be read but not written by the MPU.

msa (valid secondary address) bit 3 is set true (high) when TPAS (Talker Primary Addressed State) or LPAS (Listener Primary Addressed State) is true. The chip will become addressed to listen or talk. The primary address must have been previously received.

RFD, $\overline{DAV}$, DAC – (Ready For Data, Data Valid, Data Accepted) bits assume the same state as the corresponding signal on the MC68488 package pins. The MPU may only read this bit. These signals are not synchronized with the MPU clock.

dacd (data accept disable) bit 1 set high by the MPU will prevent completion of the automatic handshake on Addresses or Commands. dacr is used to complete the handshake.

feoi (forced end or identify) bit 5 tells the chip to send $\overline{EOI}$ low. The $\overline{EOI}$ line is then returned high after the next byte is transmitted. NOTE: The following signals are not stored but revert to a false (low) level one clock cycle (MPUϕ2) after they are set true (high):

1. rfdr
2. feoi
3. dacr

These signals can be written but not read by the MPU.

Auxiliary Command Register

reset	rfdr	feoi	dacr	msa	rtl	dacd	fget	Write
	DAC	DAV	RFE			ulpa		Read

reset – initialize the chip to the following status:
(1) all interrupts cleared
(2) following bus states are in effect: SIDS, AIDS, TIDS, LIDS, LOCS, PPIS, PUCS, and PP0
(3) bit is set by $\overline{RESET}$ input pin
msa – if GPIA is in LPAS or TDAS, setting msa will force GPIA to LADS or TADS
rtl – return to local if local lockout is disabled
ulpa – state of LSB of bus at last-primary-address receive time
fget – force group execute trigger command from the MPU has occurred
rfdr – continue handshake stopped by RFD holdoff
feoi – set EOI true, clears after next byte transmitted
dacr – MPU has examined an undefined command or secondary address
dacd – prevents completion of automatic handshake on Addresses or Commands

COMMAND STATUS REGISTER R1R – The command status register flags command or state as they occur. These flags or states are simply coupled on the MPU bus. There are five major address commands. REM shows the remote/local state of the talker/listener. REM bit 6, set low, implies the local state. LOK bit 5 shows the local lockout status of the talker/listener. RLC bit 3 is set when a change of state of the remote/local flip-flop occurs and reset when the command status register is read. DCAS bit 1 indicates that either the device clear or selected device clear has been received activating the device clear function. SPAS bit 2 indicates that the SPE command has been received activating the device serial poll function. UACG bit 7 indicates that an undefined address command has been received and depending on programming the MPU decides whether to execute or ignore it. UUCG bit 0 indicates that an undefined universal command has been received.

Command Status Register (Read)

UACG	REM	LOK	X	RLC	SPAS	DCAS	UUCG

UACG – Undefined Addressed Command
REM – Remote Enabled
LOK – Local Lockout Enabled
RLC – Remote/Local State Changed
SPAS – Serial Poll Active State is in effect
DCAS – Device Clear Active State is in effect
UUCG – Undefined Universal Command

COMMAND PASS-THROUGH REGISTER R6R – The command pass through is an 8-bit port with no storage. When this port is addressed by MPU it connects the instrument data bus ($\overline{IB0}$-$\overline{IB7}$) to the MPU data bus D0-D7. This port can be used to pass commands and secondary addresses that aren't automatically interpreted through to the MPU for inspection.

Command Pass-Through Register (Read Only)

B7	B6	B5	B4	B3	B2	B1	B0

An 8-bit input port used to pass commands and secondary addresses to MPU which are not automatically interpreted by the GPIA.

PROGRAMMING CONSIDERATIONS

The following is a list of considerations when using the MH version of the MC68488:

1. Handshake Interruption

   Once a handshake sequence begins on the IEEE-488 bus it should be allowed to complete in a normal fashion, as described in the IEEE Standard. If this sequence is interrupted (e.g., the controller forces the DAV line to the not data valid state prematurely) the integrity of the data is lost and the interface devices can go to unintended states, as explained in the standard. NOTE: The MC68488 does not interrupt a handshake. It always allows the handshake to complete in the correct sequence; however, it is possible for a device, other than a MC68488, connected to the bus to interrupt the sequence. The controller can do this through an asynchronous Bus Take-over (asserting the ATN line during a handshake). If this occurs the controller should follow the asynchronous take-over with an IFC Uniline Command (ATN can be either asserted or not asserted at this time). It is also possible for some devices to interrupt the handshake by prematurely making the DAV line false (this type of interrupt should be avoided and it should be noted that the MC68488 does not interrupt the handshake sequence). If the DAV handshake line is made false (high) before DAC is made false (high) during a handshake sequence, the listener GPIA(s) will respond as follows:

   a) If IFC is sent by the controller with ATN false before another handshake sequence is initiated the MC68488 will reset back to an idle state. The GPIA at this point is ready to be reprogrammed.

The BI status bit may be set, depending on when the handshake was interrupted, but any byte in R7R cannot be considered valid. NOTE: If IFC is sent with ATN true, it must be sent again with ATN false.

b) If another handshake is initiated before IFC is sent with ATN false, the GPIA does not generate interrupts for subsequent data bytes received by the listener GPIA(s). The device responds to commands and moves into and out of TACS, LACS, etc., but no further BI interrupts are generated. The only solutions to this situation are to reset the MC68488 or have the MPU perform a read of R7R register in the GPIA.

2. Interrupt Structure

The status bits in R0R, when set, cause an interrupt (drives the $\overline{IRQ}$ line low) if the appropriate interrupt mask bits in R0W are set. The $\overline{IRQ}$ line is sensitive to a low-to-high transition produced by the logical OR of the appropriate bits in R0R. If, for example the BI status bit is set and causes an $\overline{IRQ}$ interrupt, the MPU reads R0R (this read will reset the $\overline{IRQ}$ line but not the status bit) and detect that the BI bit is set. The software should then direct the MPU to read the data byte from R7R, which in turn causes the BI bit to be reset. If after the status register (R0R) was read and before R7R is read, another interrupt status bit is set (e.g., the CMD bit) this second condition does not cause an interrupt. The BI bit being set at the time CMD occurred prevents the $\overline{IRQ}$ line from detecting the necessary low-to-high transition and an interrupt could be missed. To prevent this, the last set of instructions in the software interrupt handler should be a reset of the interrupt mask register, followed by programming this same register to its original state. This always produces the needed low-to-high transition, preventing missed interrupts.

3. The "nba" for TACS affects "nba" for SPAS

If nba for TACS is false (there is not a data byte pending in R7W) then the serial poll handshake sequence for the status byte to the controller occurs once. If nba for TACS is true (there is a data byte in R7R waiting for a handshake to listeners) then the status byte will be sent to the controller each time the controller completes a handshake and indicates that it is ready for more data.

4. The "nba" for SPAS affects "nba" for TACS

The controller places the GPIA into the Serial Poll Active State (SPAS) by sending serial Poll enable, sending the device talk address, and then releasing ATN. If the controller does this and never accepts the serial Poll Status byte (never makes the RFD handshake line true) but rather the controller asserts ATN and sends Serial Poll Disable (SPD), then the GPIA moves into and out of SPAS without completing the status byte handshake routine. In this state the "nba" for SPAS remains true and affects "nba" for TACS in the following way:

When the controller places the GPIA in TACS the part makes DAV true as soon as RFD is made true by the listeners in a normal sequence. However, the GPIA continues the handshake sequence, using the contents of R7W, over and over. (i.e., Each time the listeners accept the current data byte and makes RFD true, the GPIA makes DAV true automatically and begins another handshake sequence.) The B0 status bit is set, however, and if the MPU writes to R7W the new data byte is sent to the listeners over and over, using a handshake routine. This continual sending of data bytes occurs until the controller places the GPIA back in SPAS and completes the handshake routine for the Serial Poll Status byte making "nba" for SPAS false again.

This situation does not occur if the controller handshakes the status byte when it places the GPIA in SPAS.

5. Dual Addressing

Dual addressing implies the use of two adjacent primary addresses and, as such, care should be taken when selecting the primary addresses for this mode. Decimal address 30 (11110) should not be used because the dual address counterpart of decimal 30 is decimal 31 (11111). Since address 31 has the same bit code as that of either the Untalk or Unlisten Commands this value is an invalid primary address for the IEEE-488 system.

6. "Ghost Interrupts"

A "ghost interrupt" is an interrupt that occurs as a result of the MC68488, but when the status register is checked no status bits are set. There are two conditions that can legitimately cause a "ghost interrupt." They are:

a) SPAS status bit

If the controller conducts a serial poll by sending Serial Poll Enable (SPE) and then sends the GPIA talk address, the SPAS status bit is set and can cause an interrupt. After the controller receives the Serial Poll Status byte it will send Serial Poll Disable (SPD) which resets the SPAS status bit. If the controller can perform this sequence of events before the interrupt handler can check the SPAS status bit, the MPU will not find any status bits set ("ghost interrupt"). The possibilities are twofold:

1) If this device had actually requested the service, then the MPU, after receiving the interrupt ("ghost" or not), should check bit 6 of the Serial Poll register. If this bit is reset the MPU knows that a Serial Poll was conducted and can reset the rsv as per normal Serial Poll handling procedures.

2) If this device did not request the service request and SPAS is not set, the software should detect this as a "ghost interrupt," ignore it, and proceed with normal operations.

See "Serial Poll Procedure" (#11) for further Serial Poll operation.

b) B0 status bit

The B0 Status bit is set whenever the MC68488 is in the Talker Active State and the output register (R7W) is empty. After the listener(s) accept the current data byte on the IEEE bus, the B0 status bit will again be set and with the appropriate mask bits set, will cause an interrupt. When the talker sends the last byte of a string it is possible for the controller to detect this, synchronously take control of the bus, and untalk the talker; however, when the last byte is accepted the B0 status bit is again set and if so programmed, causes another interrupt. It is possible for the controller to untalk the device thereby resetting B0 before the MPU interrupt handler is able to check the status register. Under these conditions a "ghost interrupt" occurs. See "Send Last Byte Procedure" (#10) for further description and solution.

7. UACG Status Bit

The UACG status bit is set anytime the GPIA receives an Undefined Address Command Group (UACG) message from the controller. This bit is not qualified with the addressed state of the part. The MPU software, after detecting a UACG, must check the ma bit in the address status register to see if the device is addressed. If the UACG message is a selected command only pertinent to addressed listeners, the software, after receiving the command by reading R6R, should release the handshake (write dacr high in R3W). This allows the controller to make ATN false. If the device has been addressed to listen/talk and ATN is made false the LACS/TACS status bit in the address status register will be set. The MPU can then check these bits.

8. END Status Bit

The END status bit in R0R is used to indicate to addressed listeners that the next byte received by the addressed talker is the last byte of a string. This bit is not qualified with the handshake and thus occurs ahead of the reception of the last data byte. This alerts the MPU that the final byte will soon follow. Because of this, two interrupts, if so programmed, will occur. One for the END bit and one for the BI bit when the final byte is transferred with a handshake. For those situations where it is inconvenient to have two interrupts the END status bit can be masked, not allowing it to cause an interrupt.

9. feoi (force end or identify)

This control bit (bit 5, R3W) is used when the MC68488 is an Active Talker, to indicate to the listener(s) on the IEEE bus the end of a data string transfer. The MC68488 asserts the EOI management line when the feoi control bit is set and the device is in the Talker Active State (TACS). The feoi bit is set by the MPU writing this bit high and automatically resets one E clock cycle after it was set. The use of this function is as follows: When sending a string of data the feoi control bit should be set prior to sending the final data byte. This causes the EOI management line to be asserted (low). The final data byte can now be sent. The EOI line remains asserted until this byte is accepted, at which time it returns high.

Care must be used when setting the feoi control bit. Once feoi has been written high, the EOI line is asserted when the MC68488 is an Active Talker and remains asserted until the next data byte is sent and accepted. This is true even if feoi is written high while the device is not an Active Talker. In this case the EOI management line is asserted as soon as the MC68488 is again made an Active Talker. Once the feoi control bit is set, only a device reset prevents the END message from being sent when the MC68488 becomes an Active Talker.

10. Send Last Byte Procedure (Talker Mode)

The procedure used for sending the last byte is described below. When using the $\overline{EOI}$ management line, the MPU software must first set the feoi control bit (asserting $\overline{EOI}$), and then send the last byte. When the last byte is accepted by all listeners, the B0 status bit of the talker device is set. The B0 status bit is not qualified with the $\overline{EOI}$ line, but is set whenever the current data byte is accepted by all listeners and the device is in the Talker Active State (TACS). (Note that when the controller asserts $\overline{ATN}$ to send commands, the GPIA moves out of TACS causing B0 to reset and remains out of TACS as long as $\overline{ATN}$ is asserted.) After the data block transfer, the controller takes control of the bus (asserts $\overline{ATN}$) and reconfigures the GPIB system. In performing this task, the controller sends command(s) that untalk the device (MLA, OTA, UNT) or reassigns it as a Talker (MTA) asking for further data transfers. Since the GPIB operates asynchronously with respect to the device MPU bus, it is possible for the controller to take control of the GPIB and cause actions that change the state of the B0 status bit in the middle of the MPU interrupt routine. As a result, care needs to be exercised when responding to the B0 status bit interrupt occurring after transferring the last byte. Any of the following conditions can occur.

1) Device Untalked – If either My Listen Address, Other Talk Address, or the Untalk command is sent, the device is placed in the Talker Idle State (TIDS) – the device is Untalked. In this case the B0 status bit is set as soon as the last data byte is accepted, reset when the controller asserts $\overline{ATN}$, and B0 will remain reset after $\overline{ATN}$ is released.

   (a) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register before the controller asserts $\overline{ATN}$. This status indication, however, is misleading as another byte transfer is not intended. The device is soon to be Untalked.

   (b) The B0 status bit indicates a reset condition if the MPU reads the Interrupt Status Register after $\overline{ATN}$ has been asserted – a "ghost interrupt" is produced. This B0 status bit remains reset after $\overline{ATN}$ is made false (high).

2) Device Reassigned as a Talker – The controller reassigns the device to talk by sending My Talk Address. In this case the B0 status bit is set as

3

soon as the last data byte is accepted, reset when the controller asserts $\overline{ATN}$ to send MTA, and is again set when $\overline{ATN}$ is made false by the controller.

(a) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register before the controller asserts $\overline{ATN}$. This case is identical to part (a) for "Device Untalked" shown above.

(b) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register while $\overline{ATN}$ is asserted — a "ghost interrupt" is produced.

(c) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register after $\overline{ATN}$ is made false (high). This status indication is requesting a byte transfer and should be acted upon accordingly.

To alleviate the above ambiguity and "ghost interrupt" situation, the GPIB handshake must be synchronized with action by the device MPU. The following step-by-step procedure provides this needed synchronization and eliminates the ambiguity when servicing the B0 status bit after sending the last byte.

1) Before sending the last byte of a block transfer, the feoi bit (if used) should be set. In addition, the dacd bit in R3W should be set, holding off the handshake upon reception of any command (establishes the required synchronization between MPU and controller).
2) If operating under interrupts, the B0 interrupt mask should be reset. This prevents generation of a B0 status interrupt when the last byte is received.
3) Send the last data byte.
4) The MPU now monitors the ATN bit in the Address Status Register (R2R). When the ATN bit is set, the $\overline{ATN}$ line has been asserted and it will remain asserted until completion of the handshake. The procedure, described herein, assumes that $\overline{ATN}$ line is asserted between block transfers and at least one command sent. The fact that $\overline{ATN}$ is asserted indicates that the device is no longer in TACS and, thus, the B0 status bit is reset.
5) The dacd bit in R3W can now be written low, removing the manual handshake hold-off on subsequent commands. With the same write instruction, the dacr bit should be set, releasing the handshake on the current command (write a hex 10 to R3W).
6) The B0 interrupt mask bit can now be set, enabling interrupts for another block transfer.

After following this procedure, a B0 status condition will occur only if a second block of data is requested by the controller. In addition, the possibility of a B0 "ghost interrupt" is eliminated.

11. Serial Poll Procedure

The MPU initiates a service request by writing rsv (bit 6, R5W) high in the GPIA. At the same time, the appropriate code should be placed in the other 7 bits. Bit 6 being set causes the $\overline{SRQ}$ management line to go low. The GPIA enters the Serial Poll Active State (SPAS) when it receives SPE and is an active talker. When it enters SPAS, the following occurs: the SPAS status bit (bit 2, R1R) is set, the CMD status bit (bit 2, R0R) is set, the $\overline{SRQ}$ line is asserted passively false (high), the SRQ status bit (bit 6, R5R) is reset, and the contents of R5R is placed on the GPIB data bus.

When the GPIA enters SPAS, the SPAS status bit (R1R) is set. This, in turn, causes the CMD status bit in R0R to be set. In an interrupt driven system with the CMD and IRQ mask bits set, this causes an MPU interrupt. These status bits are not latched conditions and only monitor the current state of the GPIA. If the controller places the GPIA in SPAS (sends SPE and MTA), receives the Serial Poll status byte and removes the GPIA from SPAS (sends SPD) before the MPU reads the Interrupt Status register, the contents of this register shows hex 10. Since the MPU knows that this device issued the service request, it should check bit 6 of R5W if an MPU interrupt is generated but no status bit is set. If bit 6, R5R, is reset, the MPU will know the controller has performed a Serial Poll on it. However, the SRQ status bit being reset does not indicate that the status byte was accepted by the controller — that is, the handshake was completed. Rather, it indicates that the GPIA has been placed in SPAS and that the status byte has been placed on the GPIB. In systems with slow responding controllers, the SRQ bit in R5R can be reset while the SPAS status bit is still set. In this case to determine when the status byte was accepted, the MPU can monitor SPAS status bit. This bit is reset when the controller has removed the GPIA from the SPAS. Once in SPAS, the controller must accept the Serial Poll byte before removing the device from SPAS. The rsv bit cannot be written low until the status byte has been accepted, but should be written low as soon as the status byte has been accepted by the controller.

If this device has issued a service request to the controller, the following provides a procedure for handling a SPAS interrupt. The procedure only discusses Serial Poll (SPAS) interrupts. Interrupts resulting from other sources need to be incorporated as appropriate for the system application. In an interrupt driven system, the MPU normally reads the Interrupt Status Register to find the cause of the interrupt. The Interrupt Status Register must be read to release the $\overline{IRQ}$ line and, in most cases, it will be read to check if something other than SPAS caused the interrupt. However, since it is possible that the SPAS status can be set and then reset before the MPU reads the register, the following procedure should also be used (even though the SPAS status is reset).

1) The MPU should monitor the SRQ bit in the Serial Poll Register. This can occur as a result of either an interrupt or a polling routine.
2) When the SRQ bit returns to zero, it indicates that the MC68488 has been placed in the Serial Poll Active State (SPAS). This does not mean that the device is in SPAS, because the con-

troller could have placed the MC68488 in SPAS and then removed the device from SPAS before the MPU reads the Serial Poll Register (R5R).

3) After the SRQ bit in R5R returns to zero, the MPU should read the Command Status Register and Monitor the SPAS status bit. When this bit returns to 0, it indicates that the Serial Poll Status byte has been accepted by the controller and that the MC68488 has been removed from the Serial Poll Active State (SPAS).
4) After the SPAS status bit returns to 0, the rsv bit (in R5W) should be written low.

The GPIA uses the source handshake to send the Serial Poll status byte to the controller. The GPIA does this by placing the status byte on the GPIB, and when the controller makes RFD true, the GPIA makes $\overline{DAV}$ true (low), and the handshake takes place according to the IEEE-488 Standard handshake protocol. If nba for the GPIA TACS function is false at this time, the GPIA will send this byte only once; i.e., the GPIA does not make $\overline{DAV}$ true (low) a second time. If nba for the GPIA TACS function is true at this time, the GPIA sends this byte over and over, provided the controller continually makes RFD true at the end of the handshake without reconfiguring the device; i.e., the GPIA in this situation makes $\overline{DAV}$ true (low) each time it receives an RFD true from the controller. The only time nba can be true for TACS is if the device was an active talker prior to the Serial Poll sequence, and the GPIA MPU had loaded a byte in R7W. Now, if the controller synchronously takes over the bus before this byte is placed on the GPIB, the nba for TACS will be true.

NOTE

After a Serial Poll has been conducted on the GPIA, and the SRQ bit (bit 6, R5W = 0) is reset, the MPU must write the rsv (bit 6, R5W) low before another service request can be initiated.

APPENDIX

GPIA MASK SET DIFFERENCES

There have been two mask sets produced for the GPIA (MC68488). They are:

G6G MASK SET – sampled in the fall of '77 (first mask set). This mask set was produced through December of 1978 and can be identified by the letters "GG" preceding the date code on top of the package.

M2H MASK SET – parts available January '79 (final mask set). Any parts ordered after this date will be M2H parts. The M2H mask set replaces the G6G mask set and can be identified by the letters MH or M2H preceding the date code on top of the package. The mask set designation for later production runs is P9W. The P9W mask set is identical to the M2H in all aspects.

There are seven areas of differences between the G6G and M2H/P9W mask sets. They are:

1. RLC Status bit
   This bit is used to implement the Remote/Local interface function. In the GG mask set the Remote/Local option should not be used, because the RLC status bit in R1R will lock up in the zero state. In the MH mask version the RLC bit is completely functional and will report any change in the REM status bit.
2. Extended Addressing
   The GG mask version of the GPIA will not discontinue secondary addressing when the primary address of another GPIA is sent by the controller; i.e., after entering LPAS, the primary address of another device will not transfer the GPIA to LPIS. This transition from LPAS to LPIS was not fully implemented in the GG mask set. The MH mask set has fully implemented this interface function. With this version, if the GPIA is programmed for extended addressing and receives its primary address, it will move to LPAS. If at this point the primary address of another controller is sent, the GPIA will go to its idle state (LIDS/TIDS) as per IEEE-488 1978 standard requirements.
3. TPAS and LPAS Status Bits
   In the GG mask set the LPAS status bit will report either LPAS or LADS. Likewise, the TPAS status bit will report either TPAS or TADS. In the MH mask set these bits only report LPAS and TPAS respectively.
4. DAC Release
   When the GPIA (GG mask set) in the listener mode receives a byte of data from the IEEE bus the DAC handshake will be held off until the MPU reads this data byte. The E-pulse that reads this data from R7R also releases DAC, indicating to the talker that the byte has been accepted. In the GG mask set the DAC handshake line is released on the low-to-high transition (leading edge) of the E-pulse, but the data is actually read (accepted by the MPU) on the high-to-low transition (trailing edge). If there is a very fast talker or long E-pulse width, it is possible for the talker to receive a data accept (DAC) and place the next data byte in the Data-In Register before the current one has been read out by the MPU. This will overwrite the data on the MPU bus and result in missed data. For this to occur the talker must be able to detect the DAC line going high and place the next data byte on the bus (making DAV true) before the E-pulse goes low. For a 1 MHz E-pulse this is approximately 400 ns. The MH or M2H mask set corrects this by releasing DAC after the trailing edge of the E-pulse.
5. dsel (deselect)
   One of the functions of the dsel bit (bit 7 of R2W) is to deselect the Group Execute Trigger (GET) command from setting the GET status bit and causing an interrupt. The GG mask version of the GPIA, when dsel is set, prevents the GET status bit from being set, but it is still possible to get an $\overline{IRQ}$ ($\overline{IRQ}$ output goes low), if enabled, when the GET command is detected. Thus, dsel inhibits the GET status condition, but not the associated interrupt. The result is a "ghost interrupt" whenever the controller sends the GET command. The MH mask set, when in dsel mode, inhibits both GET status and its associated interrupt and eliminates this "ghost interrupt."

6. hold-on-all-data (hlda)
   When in the listener mode, the GPIA provides a means of holding off the handshake on reception of data until the MPU releases the handshake. This mode occurs if the hlda (hold-on-all-data) bit in R2W is set. The MPU releases the handshake by writing rfdr (ready-for-data-release) in R3W high. In the GG mask version, if while receiving data in the listener mode the controller takes over synchronously and makes the GPIA a talker and then changes the GPIA at a later time, back to a listener, the RFD handshake will be held off on the listener command rather than waiting for the first data byte. The MH mask only holds off the handshake on data and does not hold off the handshake on any command.
7. new-byte-available (nba) During a Serial Poll
   In the GG mask version, if the GPIA had been in the talker active state prior to the controller conducting a serial poll, it is possible for nba to be lost. The situation is as follows: if a byte of data has been written into R7W and the controller takes over the bus synchronously at SDYS (SH state diagram, Figure 3, page 20, IEEE-488 1978 Specification) to perform a serial poll, the GG mask version of the GPIA will respond in one of two ways:
   a) If the controller never requests the contents of the Serial Poll Register from the GPIA, and when the GPIA is returned as a talker, the data in R7W is available to the listeners on the bus. (Data byte is not destroyed).
   b) If the controller requests the contents of the Serial Poll Register from the active talker, when the controller returns the GPIA to the Active Talker State, the data which was in R7W will have been handshaked as though it had been accepted by the active listeners (data byte will be destroyed).

The original IEEE Standard had a discrepancy as to what happens to this data byte (nba) under these circumstances. This discrepancy has been alleviated. The MH mask conforms to the latest revision and does not destroy the data in R7W when a Serial Poll occurs; i.e., if in TACS with a nba pending when the controller releases the bus to the talker, the byte in R7W will be transferred, via handshake, to the listeners (data byte is not destroyed).

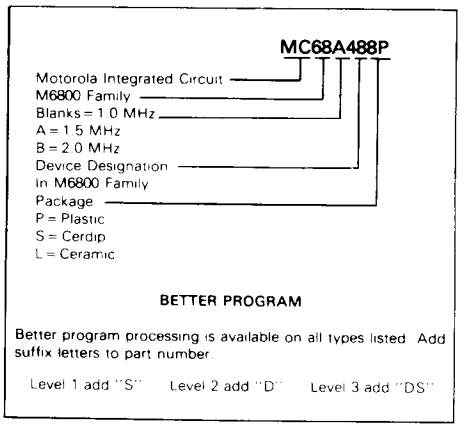
SOFTWARE DIFFERENCES BETWEEN MASK SETS

The seven changes mentioned in the previous sections are the only changes from the GG to the MH mask set. All of these changes except number 3 (TPAS and LPAS status bits) are transparent to the user software.

The change to TPAS and LPAS status bits is a functional change. In the GG mask, user software could monitor LPAS and TPAS for address recognition in the primary address mode because LPAS is set as soon as the GPIA receives its Listen Address (MLA) and TPAS is set as soon as the GPIA receives its Talker Address (MTA); i.e., the LPAS bit is set when the GPIA enters LADS and the TPAS bit is set when the GPIA enters TADS. In the MH mask set these bits do not report LADS/TADS and as such they can only be used in the extended address mode. In the primary address mode the software for the MH mask set should monitor LACS/TACS (bits 2 and 3 of the address status register) rather than LPAS/TPAS: TACS/LACS indicates when the device is in the Talker/Listener Active State.

MECHANICAL DATA

ORDERING INFORMATION



Level 1 "S" = 10 Temp Cycles – (−25 to 150°C), Hi Temp testing at T_A max
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2

Speed	Device	Temperature Range
1.0 MHz	MC68488P, S MC68488CP, CS	0°C to 70°C −40°C to +85°C
1.5 MHz	MC68A488P, S MC68A488CP, CS	0°C to 70°C −40°C to +85°C
2.0 MHz	MC68488P, S	0 to +70°C

PIN ASSIGNMENT