

Am2914

Vectored Priority Interrupt Controller

DISTINCTIVE CHARACTERISTICS

- Accepts 8 interrupt inputs
Interrupts may be pulses or levels and are stored internally
- Built-in mask register
Six different operations can be performed on mask register
- Built-in status register
Status register holds code for lowest allowed interrupt
- Vectored output
Output is binary code for highest priority un-masked interrupt
- Expandable
Any number of Am2914's may be stacked for large interrupt systems
- Microprogrammable
Executes 16 different microinstructions
Instruction enable pin aids in vertical microprogramming
- High-speed operation
Delay from an interrupt clocked into the interrupt register to interrupt request output is typically 60 ns

TABLE OF CONTENTS

Block Diagram	2-159
Connection Diagrams	2-160
Instructions	2-160
Ordering Information	2-161
DC Characteristics	2-162
AC Characteristics	2-164
Burn-in Circuit	2-165
Detailed Logic Description	2-177

For applications information,
see Chapter VI of "Build a Microcomputer".

FUNCTIONAL DESCRIPTION

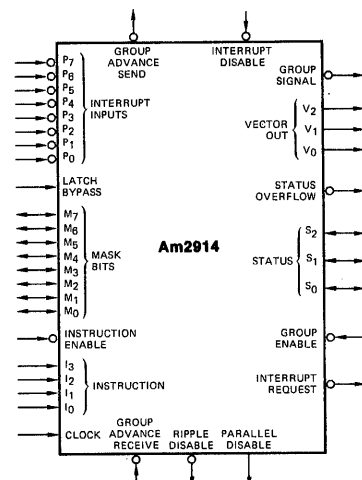
The Am2914 is a high-speed, eight-bit priority interrupt unit that is cascadable to handle any number of priority interrupt request levels. The high-speed of the Am2914 makes it ideal for use in Am2900 family microcomputer designs, but it can also be used with the Am9080A MOS microprocessor.

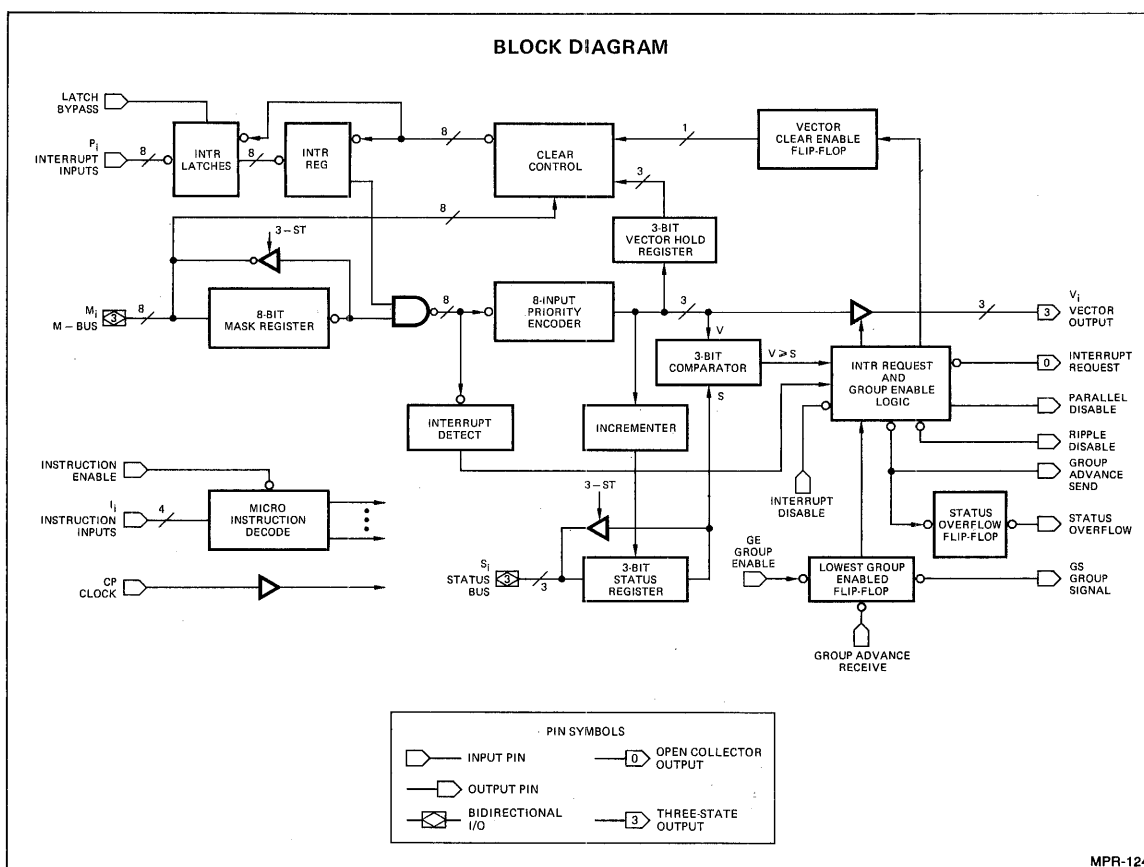
The Am2914 receives interrupt requests on 8 interrupt input lines (P₀-P₇). A LOW level is a request. An internal latch may be used to catch pulses on these lines, or the latch may be bypassed so the request lines drive the edge-triggered interrupt register directly. An 8-bit mask register is used to mask individual interrupts. Considerable flexibility is provided for controlling the mask register. Requests in the interrupt register are ANDed with the corresponding bits in the mask register and the results are sent to an 8-input priority encoder, which produces a three bit encoded vector representing the highest numbered input which is not masked.

An internal status register is used to point to the lowest priority at which an interrupt will be accepted. The contents of the status register are compared with the output of the priority encoder, and an interrupt request output will occur if the vector is greater than or equal to status. Whenever a vector is read from the Am2914 the status register is automatically updated to point to one level higher than the vector read. (The status register can be loaded externally or read out at any time using the S pins.) Signals are provided for moving the status upward across devices (Group Advance Send and Group Advance Receive) and for inhibiting lower priorities from higher order devices (Ripple Disable, Parallel Disable, and Interrupt Disable). A status overflow output indicates that an interrupt has been read at the highest priority.

The Am2914 is controlled by a 4-bit instruction field I₀-I₃. The command on the instruction lines is executed if IE is LOW and is ignored if IE is HIGH, allowing the 4 I bits to be shared with other devices.

LOGIC SYMBOL





MPR-124

BLOCK DIAGRAM DESCRIPTION

The Microinstruction Decode circuitry decodes the Interrupt Microinstructions and generates required control signals for the chip.

The Interrupt Register holds the Interrupt Inputs and is an eight-bit, edge-triggered register which is set on the rising edge of the CP Clock signal.

The Interrupt latches are set/reset-type latches. When the Latch Bypass signal is LOW, the latches are enabled and act as negative pulse catchers on the inputs to the Interrupt Register. When the Latch Bypass signal is HIGH, the Interrupt latches are transparent.

The Mask Register holds the eight mask bits associated with the eight interrupt levels. The register may be loaded from or read to the M Bus. Also, the entire register or individual mask bits may be set or cleared.

The Interrupt Detect circuitry detects the presence of any unmasked Interrupt Input. The eight-input Priority Encoder determines the highest priority, non-masked Interrupt Input and forms a binary coded interrupt vector. Following a Vector Read, the three-bit Vector Hold Register holds the binary coded interrupt vector. This stored vector is used for clearing interrupts.

The three-bit Status Register holds the status bits and may be loaded from or read to the S Bus. During a Vector Read, the Incrementer increments the interrupt vector by one, and the result is clocked into the Status Register. Thus the Status

Register always points to the lowest level at which an interrupt will be accepted.

The three-bit Comparator compares the Interrupt Vector with the contents of the Status Register and indicates if the Interrupt Vector is greater than or equal to the contents of the Status Register.

The Lowest Group Enabled Flip-Flop is used when a number of 2914's are cascaded. In a cascaded system, only one Lowest Group Enabled Flip-Flop is LOW at a time. It indicates the eight interrupt group, which contains the lowest priority interrupt level which will be accepted and is used to form the higher order status bits.

The Interrupt Request and Group Enable logic contain various gating to generate the Interrupt Request, Parallel Disable, Ripple Disable, and Group Advance Send signals.

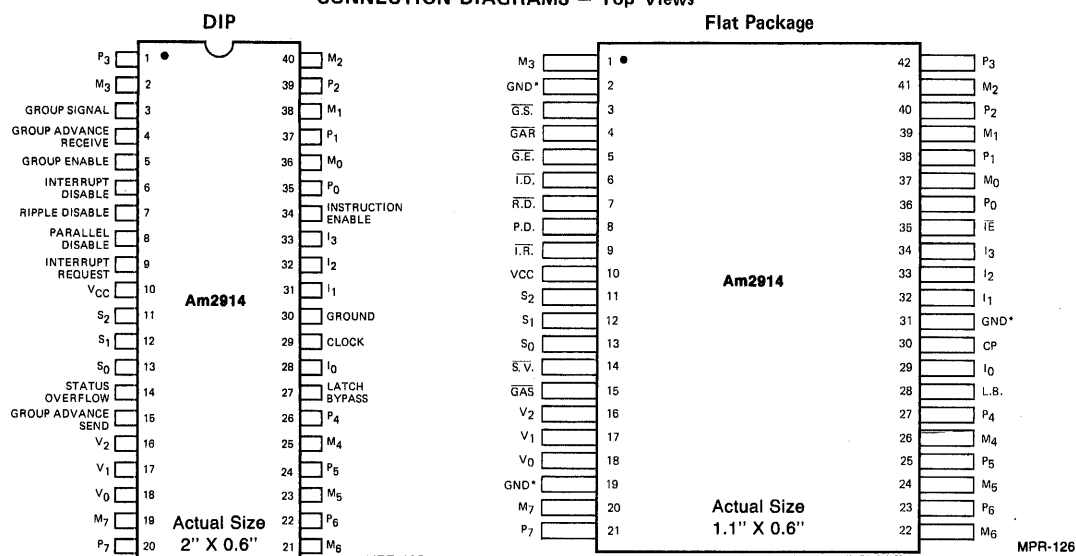
The Status Overflow signal is used to disable all interrupts. It indicates the highest priority interrupt vector has been read and the Status Register has overflowed.

The Clear Control logic generates the eight individual clear signals for the bits in the Interrupt Latches and Register. The Vector Clear Enable Flip-Flop indicates if the last vector read was from this group. When it is set, it enables the Clear Control Logic.

The CP clock signal is used to clock the Interrupt Register, Mask Register, Status Register, Vector Hold Register, and the Lowest Group Enabled, Vector Clear Enable and Status Overflow Flip-Flops, all on the clock LOW-to-HIGH transition.

Am2914

CONNECTION DIAGRAMS — Top Views



Metallization and Pad Layout

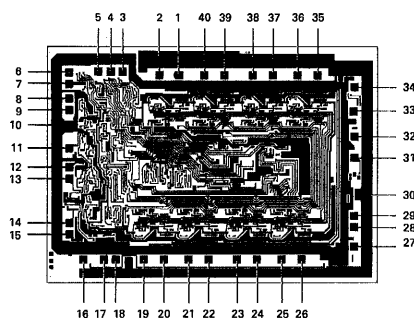


TABLE I
MICROINSTRUCTION SET FOR Am2914 PRIORITY INTERRUPT CIRCUIT

Decimal $I_3 I_2 I_1 I_0$	Mnemonic	Instruction	Decimal $I_3 I_2 I_1 I_0$	Mnemonic	Instruction
Mask Register Functions					
14	LDM	Load mask register from M bus	5	RDVC	Vectored Output Read vector output to V outputs, load V+1 into status register, load V into vector hold register and set vector clear enable flip-flop.
7	RDM	Read mask register to M bus	Priority Interrupt Register Clear		
12	CLRM	Clear mask register (enables all priorities)	1	CLRIN	Clear all interrupts
8	SETM	Set mask register (inhibits all interrupts)	3	CLMRM	Clear interrupts from mask register data (uses the M bus)
10	BCLRM	Bit clear mask register from M bus	2	CLRMB	Clear interrupts from M bus data
11	BSETM	Bit set mask register from M bus	4	CLRVC	Clear the individual interrupt associated with the last vector read
Status Register Functions					
9	LDSTA	Load status register from S bus and LGE flip-flop from GE input	Master Clear		
6	RDTA	Read status register to S bus	0	MCLR	Clear all interrupts, clear mask register, clear status register, clear LGE flip-flop, enable interrupt request.
Interrupt Request Control					
15	ENIN	Enable interrupt request			
13	DISIN	Disable interrupt request			

STANDARD SCREENING

(Conforms to MIL-STD-883 for Class C Parts)

Step	MIL-STD-883 Method	Conditions	Level	
			Am2914PC, DC	Am2914DM, FM
Pre-Seal Visual Inspection	2010	B	100%	100%
Stabilization Bake	1008	C 24-hour 150°C	100%	100%
Temperature Cycle	1010	C -65°C to +150°C 10 cycles	100%	100%
Centrifuge	2001	B 10,000 G	100% *	100%
Fine Leak	1014	A 5×10^{-8} atm-cc/sec	100% *	100%
Gross Leak	1014	C2 Fluorocarbon	100% *	100%
Electrical Test Subgroups 1 and 7	5004	See below for definitions of subgroups	100%	100%
Insert Additional Screening here for Class B Parts				
Group A Sample Tests				
Subgroup 1			LTPD = 5	LTPD = 5
Subgroup 2			LTPD = 7	LTPD = 7
Subgroup 3			LTPD = 7	LTPD = 7
Subgroup 7			LTPD = 7	LTPD = 7
Subgroup 8			LTPD = 7	LTPD = 7
Subgroup 9			LTPD = 7	LTPD = 7
	5005	See below for definitions of subgroups Maximum accept number is 3		

*Not applicable for Am2914PC.

ORDERING INFORMATION

Order the part number according to the table below to obtain the desired package, temperature range, and screening level.

Order Number	Package Type (Note 1)	Operating Range (Note 2)	Screening Level (Note 3)
AM2914PC	P-40	C	C-1
AM2914DC	D-40	C	C-1
AM2914DC-B	D-40	C	B-2 (Note 4)
AM2914DM	D-40	M	C-3
AM2914DM-B	D-40	M	B-3
AM2914FM	F-42	M	C-3
AM2914FM-B	F-42	M	B-3
AM2914XC	Dice	C	Visual inspection to MIL-STD-883 Method 2010B.
AM2914XM	Dice	M	

- Notes: 1. P = Molded DIP, D = Hermetic DIP, F = Flat Pak. Number following letter is number of leads. See Appendix B for detailed outline. Where Appendix B contains several dash numbers, any of the variations of the package may be used unless otherwise specified.
2. C = 0°C to +70°C, V_{CC} = 4.75V to 5.25V, M = -55°C to +125°C, V_{CC} = 4.50V to 5.50V.
3. See Appendix A for details of screening. Levels C-1 and C-3 conform to MIL-STD-883, Class C. Level B-3 conforms to MIL-STD-883, Class B.
4. 96 hour burn-in.

Am2914

MAXIMUM RATINGS (Above which the useful life may be impaired)

Storage Temperature	−65°C to +150°C
Temperature (Ambient) Under Bias	−55°C to +125°C
Supply Voltage to Ground Potential	−0.5V to +7.0V
DC Voltage Applied to Outputs for High Output State	+0.5V to +V _{CC} max.
DC Input Voltage	−0.5V to 5.5V
DC Output Current, Into Outputs	30mA
DC Input Current	−30mA to +5.0mA

OPERATING RANGE

P/N	Temperature	V _{CC}
Am2914PC, DC	0°C to +70°C	4.75V to 5.25V
Am2914DM, FM	−55°C to +125°C	4.50V to 5.50V

ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

(Group A, Subgroups 1, 2, and 3)

Am2914XC	T _A = 0°C to +70°C	V _{CC} = 5.0V ± 5% (COM'L)	MIN. = 4.75V	MAX. = 5.25V
Am2914XM	T _C = −55°C to +125°C	V _{CC} = 5.0V ± 10% (MIL)	MIN. = 4.50V	MAX. = 5.50V

Parameters	Description	Test Conditions (Note 1)	Min.	Typ. (Note 2)	Max.	Units
V _{OH}	Output HIGH Voltage	V _{CC} = MIN., V _{IN} = V _{IH} or V _{IL}	MIL, I _{OH} = −1.0mA COM'L, I _{OH} = −2.6mA	2.4		Volts
I _{CEX}	Output Leakage Current for IR Output	V _{CC} = MIN., V _O = 5.5V			250	μA
V _{OL}	Output LOW Voltage	V _{CC} = MIN., V _{IN} = V _{IH} or V _{IL}	I _{OL} = 4.0mA I _{OL} = 8.0mA I _{OL} = 12mA		0.4 0.45 0.5	Volts
V _{IH}	Input HIGH Level	Guaranteed input logical HIGH voltage for all inputs	2.0			Volts
V _{IL}	Input LOW Level	Guaranteed input logical LOW voltage for all inputs			0.8	Volts
V _I	Input Clamp Voltage	V _{CC} = MIN., I _{IN} = −18mA			−1.5	Volts
I _{IL}	Input LOW Current	V _{CC} = MAX., V _{IN} = 0.4V	M0-7 S0-2 L. B. I. D. All Others		−0.15 −0.1 −0.4 −2.0 −0.8	mA
I _{IH}	Input HIGH Current	V _{CC} = MAX., V _{IN} = 2.7V	M0-7 S0-2 IE, GE, GAR I. D. All Others		150 100 40 60 20	μA
I _I	Input HIGH Current	V _{CC} = MAX., V _{IN} = 5.5V			1.0	mA
I _O	Off-State Output Current	V _{CC} = MAX., V _{OUT} = 0.5V V _{OUT} = 2.4V	M0-7 S0-2 V0-2 M0-7 S0-2 V0-2		−150 −100 −50 150 100 50	μA
I _{CC}	Power Supply Current	V _{CC} = 5.0V, 25°C V _{CC} = MAX.	COM'L MIL	0°C 70°C −55°C 125°C	170 305 250 310 200	mA
I _{SC}	Output Short Circuit Current (Note 3)	V _{CC} = MAX.		−30	−85	mA

- Notes: 1. For conditions shown as MIN. or MAX., use the appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.

SWITCHING CHARACTERISTICS AT 25°C AND 5.0 VOLTS

Note: Guaranteed limits at 25°C and 5.0V are group A, subgroup 9 tests
All outputs fully loaded. $C_L = 50\text{pF}$. Measurements made at 1.5V with
input levels of 0V and 3.0V. All numbers are in ns.

For interrupt request output, $R_L = 470\Omega$

TABLE I. CLOCK AND INTERRUPT INPUT PULSE WIDTHS (ns)

Time	GUARANTEED
Minimum Clock LOW Time	30
Minimum Clock HIGH Time	30
Minimum Interrupt Input (P_0 - P_7) LOW Time for Guaranteed Acceptance (Pulse Mode)	25
Maximum Interrupt Input (P_0 - P_7) LOW Time for Guaranteed Rejection (Pulse Mode)	10

2

TABLE II. COMBINATIONAL PROPAGATION DELAYS (ns)

To Output From Input	TYPICAL						GUARANTEED					
	M Bus	S Bus	V012	Irpt Req	Ripple Disable	Group Advance Send	M Bus	S Bus	V012	Irpt Req	Ripple Disable	Group Advance Send
\overline{IE}	36	40	40	—	—	30	48	55	55	—	—	47
I0123	36	40	40	—	—	30	48	55	55	—	—	47
Irpt. Disable	—	—	25	35	8	19	—	—	37	42	18	25

TABLE III. DELAYS FROM CLOCK TO OUTPUTS (ns)

Clock Path	TYPICAL							GUARANTEED						
	To V012	To Irpt Req	To PD	To \overline{RD}	To \overline{GAS}	To Status O'flow	To \overline{GS}	To V012	To Irpt Req	To PD	To \overline{RD}	To \overline{GAS}	To Status O'flow	To \overline{GS}
Irpt Latches and Register	55	65	37	39	47	—	—	67	82	57	57	66	—	—
Mask Register	55	65	37	39	47	—	—	67	82	57	57	66	—	—
Status Register	45	55	28	31	37	—	—	59	74	57	57	58	—	—
Lowest Group Enabled Flip-Flop	—	—	22	25	—	—	17	—	—	42	45	—	—	32
Irpt Request Enable Flip-Flop	—	40	—	—	—	—	—	—	56	—	—	—	—	—
Status Overflow Flip-Flop	—	—	—	—	—	17	—	—	—	—	—	—	30	—

TABLE IV. SET-UP AND HOLD TIME REQUIREMENTS (ns)
(All relative to clock LOW-to-HIGH transition)

From Input	GUARANTEED	
	Set-up Time	Hold Time
S-Bus	11	8
M-Bus	11	8
$\overline{P_0}$ - $\overline{P_7}$	11	6
Latch Bypass	16	0
\overline{IE}	46	0
I0123 (See Note)	$t_{pWL} + 29$	0
\overline{GE}	11	11
\overline{GAR}	11	11
Irpt Disable	35	0
P_0 - P_7 Hold Time Relative to LB	—	21

Note: t_{pWL} is the Clock LOW Time. Both Set-up times must be met.

Am2914

SWITCHING CHARACTERISTICS OVER OPERATING VOLTAGE AND TEMPERATURE RANGE

(Group A, subgroup 10 and 11 tests and limits)

All outputs fully loaded, $C_L = 50\text{pF}$. Measurements made at 1.5V with input levels of 0V and 3.0V. For Interrupt Request Output, $R_L = 470\Omega$.

TABLE V. CLOCK AND INTERRUPT INPUT PULSE WIDTHS (ns)

Time	Am2914PC, DC, XC $T_A = 0^\circ\text{C to } +70^\circ\text{C}, 5\text{V} \pm 5\%$	Am2914DM, FM, XM $T_C = -55^\circ\text{C to } +125^\circ\text{C}, 5\text{V} \pm 10\%$
Minimum Clock LOW Time	30	30
Minimum Clock HIGH Time	30	30
Minimum Interrupt Input (P_0 - P_7) LOW Time for Guaranteed Acceptance (Pulse Mode)	40	40
Maximum Interrupt Input (P_0 - P_7) LOW Time for Guaranteed Rejection (Pulse Mode)	8	8
Minimum Clock Period, $\overline{IE} = H$ on current cycle and previous cycle	50	55
Minimum Clock Period, $\overline{IE} = L$ on current cycle or previous cycle	100	110

TABLE VI. MAXIMUM COMBINATIONAL PROPAGATION DELAYS (ns)

To Output From Input	Am2914PC, DC, XC $T_A = 0^\circ\text{C to } +70^\circ\text{C}, 5\text{V} \pm 5\%$						Am2914DM, FM, XM $T_C = -55^\circ\text{C to } +125^\circ\text{C}, 5\text{V} \pm 10\%$					
	M Bus	S Bus	V_{012}	Irpt Req	Ripple Disable	Group Advance Send	M Bus	S Bus	V_{012}	Irpt Req	Ripple Disable	Group Advance Send
\overline{IE}	52	60	65	—	—	56	60	68	70	—	—	62
I_{0123}	52	60	65	—	—	56	60	68	70	—	—	62
Irpt. Disable	—	—	45	52	20	30	—	—	48	60	22	33

TABLE VII. MAXIMUM DELAYS FROM CLOCK TO OUTPUTS (ns)

Clock Path	Am2914PC, DC, XC $T_A = 0^\circ\text{C to } +70^\circ\text{C}, 5\text{V} \pm 5\%$							Am2914DM, FM, XM $T_C = -55^\circ\text{C to } +125^\circ\text{C}, 5\text{V} \pm 10\%$						
	To V_{012}	To Irpt Req	To PD	To \overline{RD}	To \overline{GAS}	To Status O'flow	To \overline{GS}	To V_{012}	To Irpt Req	To PD	To \overline{RD}	To \overline{GAS}	To Status O'flow	To \overline{GS}
Irpt Latches and Register	76	97	67	67	80	—	—	82	105	75	75	85	—	—
Mask Register	76	97	67	67	80	—	—	82	105	75	75	85	—	—
Status Register	67	88	63	63	70	—	—	73	96	66	66	76	—	—
Lowest Group Enabled Flip-Flop	—	—	48	52	—	—	38	—	—	54	58	—	—	45
Irpt Request Enable Flip-Flop	—	62	—	—	—	—	—	—	66	—	—	—	—	—
Status Overflow Flip-Flop	—	—	—	—	—	35	—	—	—	—	—	—	40	—

TABLE VIII. SET-UP AND HOLD TIME REQUIREMENTS (ns)

(All relative to clock LOW-to-HIGH transition)

From Input	Am2914PC, DC, XC $T_A = 0^\circ\text{C to } +70^\circ\text{C}, 5\text{V} \pm 5\%$		Am2914DM, FM, XM $T_C = -55^\circ\text{C to } +125^\circ\text{C}, 5\text{V} \pm 10\%$	
	Set-Up Time	Hold Time	Set-Up Time	Hold Time
S-Bus	15	10	15	10
M-Bus	15	10	15	10
$\overline{P_0}$ - $\overline{P_7}$	15	8	15	8
Latch Bypass	20	0	20	0
\overline{IE}	55	0	55	0
I_{0123} (See Note)	$t_{pwL} + 33$	0	$t_{pwL} + 40$	0
\overline{GE}	15	13	15	13
\overline{GAR}	15	13	15	13
Irpt. Disable	42	0	42	0
P_0 - P_7 Hold Time Relative to LB	—	25	—	25

Note: t_{pwL} is the Clock LOW Time. Both Set-up times must be met.

A MICROPROGRAMMABLE, BIPOLAR, LSI INTERRUPT STRUCTURE USING THE Am2914

INTRODUCTION

Advanced Micro Devices' introduction of the Am2914 Vectored Priority Interrupt Controller now makes possible the structuring of a microprogrammable bipolar LSI interrupt system. The design engineer may use the Am2914 to simplify his design process, dramatically reduce the system cost, size and package count, and increase the speed, capability and reliability of his interrupt system.

The Am2914 is a modular, low cost, standard LSI component that may be microprogrammed to meet the requirements of specific applications. Today's engineer may utilize the Am2914 microprogrammability to provide functional flexibility and ease of engineering change, while taking advantage of its modularity to provide hardware regularity and future expansion capability.

THE INTERRUPT CONCEPT

In any state machine, a requirement exists for the efficient synchronization and response to asynchronous events such as power failure, machine malfunctions, control panel service requests, external timer signals, supervisory calls, program errors, and input/output device service requests. The merit of such an "asynchronous event handler" may be measured in terms of response time, system throughput, real time overhead, hardware cost and memory space required.

The simplest approach to asynchronous event handling is the poll approach. A status indicator is associated with each possible asynchronous event. The processor tests each indicator in sequence and, in effect, "asks" if service is required. This program-driven method is inefficient for a number of reasons. Much time is consumed polling when no service is required; programs must have frequent test points to poll indicators, and since indicators are polled in sequence, considerable time may elapse before the processor responds to an event. Thus, system throughput is low; real time overhead and response time are high, and a large memory space is required.

The interrupt method is a much more efficient way of servicing asynchronous requests. An asynchronous event requiring service generates an interrupt request signal to the processor. When the processor receives the interrupt request, it may suspend the program it is currently executing, execute an interrupt service routine which services the asynchronous request, then resume the execution of the suspended program. In this system, the execution of the service routine is initiated by an interrupt request; thus, the system is interrupt driven and service routines are executed only when service is requested. Although hardware cost may be higher in this type of system, it is more efficient since system throughput is higher, response time is faster, real time overhead is lower and less memory space is required.

INTERRUPT SYSTEM FUNCTIONAL DEFINITION

A complete and clear functional definition is key to the design of a good interrupt system. The following features are useful.

Multiple Interrupt Request Handling: Since interrupt requests are generated from a number of different sources, the interrupt system's ability to handle interrupt requests from several sources is important.

Interrupt Request Prioritization: Since the processor can service only one interrupt request at a time, it is important that the interrupt system has the ability to prioritize the requests and determine which has the highest priority.

Interrupt Service Routine "Nesting": This feature allows an interrupt service routine for a given priority request to be interrupted in turn, but only by a higher priority interrupt request. The service routine for the higher priority request is executed, then the execution of the interrupted service routine is resumed. If there are "n" interrupt requests, an "n" deep "nest" is possible.

Dynamic Interrupt Enabling/Disabling: The ability to enable/disable all interrupts "on the fly" under microprogram control can be used to prevent interruption of certain processes.

Dynamic Interrupt Request Masking: The ability to selectively inhibit or "mask" individual interrupt requests under microprogram control is useful.

Interrupt Request Vectoring: Many times, a particular interrupt request requires the execution of a unique interrupt service routine. For this reason, the generation of a unique binary coded vector for each interrupt request is very helpful. This vector can be used as a pointer to the start of a unique service routine.

Interrupt Request Priority Threshold: The ability to establish a priority threshold is valuable. In this type of operation, only those interrupt requests which have higher priority than a specified threshold priority are accepted. The threshold priority can be defined by microprogram or can be automatically established by hardware at the interrupt currently being serviced plus one. This automatic threshold prevents multiple interrupts from the same source. Also useful is the ability to read the threshold priority under microprogram control. Thus, the interrupt request being serviced may be determined by the microprogram.

Interrupt Request Clearing Flexibility: Flexibility in the method of clearing interrupt requests allows different modes of interrupt system operation. Of particular value are the abilities to clear the interrupt currently being serviced, clear all interrupts, or clear interrupts via a programmable mask register or bus.

Microprogrammability: Microprogrammability permits the construction of a general purpose or "universal" interrupt structure which can be microprogrammed to meet a specific application's requirements. The universality of the structure allows standardization of the hardware and amortization of the hardware development costs across a much broader user base. The end result is a flexible, low cost interrupt structure.

Hardware Modularity: Modular interrupt system hardware is beneficial in two ways. First, hardware modularity provides expansion capability. Additional modules may be added as the need to service additional requests arises. Secondly, hardware modularity provides a structural regularity which simplifies the system structure and also reduces the number of hardware part numbers.

Fast Interrupt System Response Time: Quick interrupt system response provides more efficient system operation. Fast response reduces real time overhead and increases overall system throughput.

INTERRUPT SYSTEM IMPLEMENTATION USING THE Am2914

The Am2914 provides all of the foregoing features on a single LSI chip. The Am2914 is a high-speed, eight-bit priority interrupt unit that is cascadable to handle any number of priority interrupt request levels. The Am2914's high speed is ideal for use in Am2900 Family microcomputer designs, but it can also be used with the Am9080A MOS microprocessor.

The Am2914 receives interrupt requests on eight Interrupt Input lines (P_0 - P_7). A LOW level is a request. An internal latch may be used to catch pulses (HIGH-LOW-HIGH) on these lines, or the latch may be bypassed so that the request lines drive the D-inputs to the edge-triggered Interrupt Register directly. An eight-bit Mask Register is used to mask individual interrupts. Considerable flexibility is provided for controlling the Mask Register. Requests in the Interrupt Register (P_0 - P_7) are ANDed with the corresponding bits in the mask register (M_0 - M_7) and the results are sent to an eight-input priority encoder, which produces a three-bit encoded vector representing the highest priority input which is not masked.

An internal Status Register is used to point to the lowest priority at which an interrupt will be accepted. The contents of the Status Register are compared with the output of the

priority encoder, and an Interrupt Request output will occur if the vector is greater than or equal to the contents of the Status Register. Whenever a vector is read from the Am2914, the Status Register is automatically updated to point to one level higher than the vector read. (The Status Register can be loaded externally or read out at any time using the S-Bus.) Signals are provided for moving the status upward across devices (Group Advance Send and Group Advance Receive) and for inhibiting lower priorities from higher order devices (Ripple Disable, Parallel Disable, and Interrupt Disable). A Status Overflow output indicates that an interrupt has been read at the highest priority.

The Am2914 is controlled by a four-bit microinstruction field I_0 - I_3 . The microinstruction is executed if \overline{IE} (Instruction Enable) is LOW and is ignored if \overline{IE} is HIGH, allowing the four I bits to be shared with other functions. Sixteen different microinstructions are executed. Figure 2 shows the microinstructions and the microinstruction codes.

MICROINSTRUCTION DESCRIPTION	MICROINSTRUCTION CODE $I_3 I_2 I_1 I_0$
MASTER CLEAR	0000
CLEAR ALL INTERRUPTS	0001
CLEAR INTERRUPTS FROM M-BUS	0010
CLEAR INTERRUPTS FROM MASK REGISTER	0011
CLEAR INTERRUPT, LAST VECTOR READ	0100
READ VECTOR	0101
READ STATUS REGISTER	0110
READ MASK REGISTER	0111
SET MASK REGISTER	1000
LOAD STATUS REGISTER	1001
BIT CLEAR MASK REGISTER	1010
BIT SET MASK REGISTER	1011
CLEAR MASK REGISTER	1100
DISABLE INTERRUPT REQUEST	1101
LOAD MASK REGISTER	1110
ENABLE INTERRUPT REQUEST	1111

Figure 2. Am2914 Microinstruction Set.

In this microinstruction set, the *Master Clear* microinstruction is selected as binary zero so that during a power up sequence, the microinstruction register in the microprogram control unit of the central processor can be cleared to all zeros. Thus, on the next clock cycle, the Am2914 will execute the *Master Clear* function. This includes clearing the Interrupt Latches and Register as well as the Mask Register and Status Register. The LGE flip-flop of the least significant group is set LOW because the Group Advance Receive input is tied LOW. All other Group Advance Receive inputs are tied to Group Advance Send outputs and these are forced HIGH during this instruction. This clear instruction also sets the Interrupt Request Enable flip-flop so that a fully interrupt driven system can be easily initiated from any interrupt.

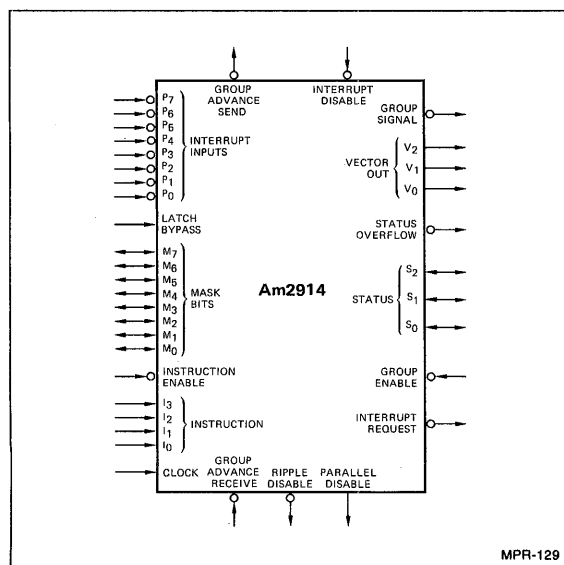


Figure 1. Am2914 Logic Symbol.

Am2914

The *Clear All Interrupts* microinstruction clears the Interrupt Latches and Register.

The *Clear Interrupts from Mask Register* microinstruction clears those Interrupt Latches and Register bits which have corresponding Mask Register bits set equal to one. The M-Bus is used by the Am2914 during the execution of this microinstruction and must be floating.

The *Clear Interrupts from M-Bus* microinstruction clears those Interrupt Latches and Register bits which have corresponding M-Bus bits set equal to one.

The *Clear Interrupt, Last Vector Read* microinstruction clears the Interrupt Latch and Register bit associated with the last vector read.

The *Read Vector* microinstruction is used to read the vector value of the highest priority request causing the interrupt. The vector outputs are three-state drivers that are enabled onto the $V_0V_1V_2$ bus during this instruction. This microinstruction also automatically loads the value "vector plus one" into the Status Register. In addition, this instruction sets the Vector Clear Enable flip-flop and loads the current vector value into the Vector Hold Register so that this value can be used by the *Clear Interrupt, Last Vector Read* microinstruction. This allows the user to read the vector associated with the interrupt, and at some later time clear the Interrupt Latch and Register bit associated with the vector read.

The *Load Status Register* microinstruction loads S-Bus data into the Status Register and also loads the LGE flip-flop from the Group Enable input.

During the *Read Status Register* microinstruction, the Status Register outputs are enabled onto the Status Bus ($S_0S_1S_2$). The Status Bus is a three-bit, bi-directional, three-state bus.

The *Load Mask Register* microinstruction loads data from the three-state, bi-directional M-Bus into the Mask Register.

The *Read Mask Register* microinstruction enables the Mask Register outputs onto the bi-directional, three-state M-Bus.

The *Set Mask Register* microinstruction sets all the bits in the Mask Register to one. This results in all interrupts being inhibited.

The entire Mask Register is cleared by the *Clear Mask Register* microinstruction. This enables all interrupts subject to the Interrupt Enable flip-flop and the Status Register.

The *Bit Clear Mask Register* microinstruction may be used to selectively clear individual Mask Register bits. This microinstruction clears those Mask Register bits which have corresponding M-Bus bits equal to one. Mask Register bits with corresponding M-Bus bits equal to zero are not affected.

The *Bit Set Mask Register* microinstruction sets those Mask Register bits which have corresponding M-Bus bits equal to one. Other Mask Register bits are not affected.

All Interrupt Requests may be disabled by execution of the *Disable Interrupt Request* microinstruction. This microinstruction resets an Interrupt Request Enable flip-flop on the chip.

The *Enable Interrupt Request* microinstruction sets the Interrupt Enable flip-flop. Thus, Interrupt Requests are enabled subject to the contents of the Mask and Status Registers.

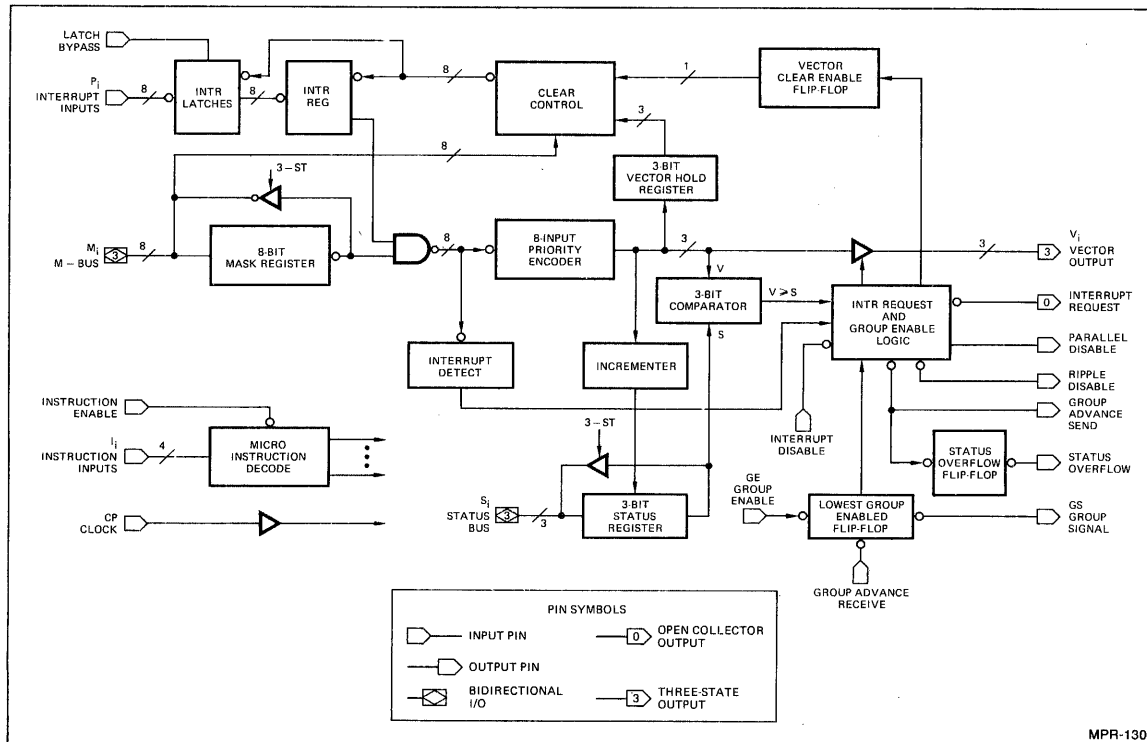


Figure 3. Am2914 Block Diagram.

Am2914 BLOCK DIAGRAM DESCRIPTION

The Am2914 block diagram is shown in Figure 3. The Microinstruction Decode circuitry decodes the Interrupt Microinstructions and generates required control signals for the chip.

The Interrupt Register holds the Interrupt Inputs and is an eight-bit, edge-triggered register which is set on the rising edge of the CP Clock signal if the Interrupt Input is LOW.

The Interrupt latches are set/reset latches. When the Latch Bypass signal is LOW, the latches are enabled and act as negative pulse catchers on the inputs to the Interrupt Register. When the Latch Bypass signal is HIGH, the Interrupt latches are transparent.

The Mask Register holds the eight mask bits associated with the eight interrupt levels. The register may be loaded from or read to the M-Bus. Also, the entire register or individual mask bits may be set or cleared.

The Interrupt Detect circuitry detects the presence of any unmasked Interrupt Input. The eight-input Priority Encoder determines the highest priority, non-masked Interrupt Input and forms a binary coded interrupt vector. Following a Vector Read, the three-bit Vector Hold Register holds the binary coded interrupt vector. This stored vector can be used later for clearing interrupts.

The three-bit Status Register holds the status bits and may be loaded from or read to the S-Bus. During a Vector Read, the Incrementer increments the interrupt vector by one, and the result is clocked into the Status Register. Thus, the Status Register points to a level one greater than the vector just read.

The three-bit Comparator compares the Interrupt Vector with the contents of the Status Register and indicates if the Interrupt Vector is greater than or equal to the contents of the Status Register.

The Lowest Group Enabled Flip-Flop is used when a number of Am2914's are cascaded. In a cascaded system, only one Lowest Group Enabled Flip-Flop is LOW at a time. It indicates the eight interrupt group, which contains the lowest priority interrupt level which will be accepted and is used to form the higher order status bits.

The Interrupt Request and Group Enable logic contain various gating to generate the Interrupt Request, Parallel Disable, Ripple Disable, and Group Advance Send signals.

The Status Overflow signal is used to disable all interrupts. It indicates the highest priority interrupt vector has been read and the Status Register has overflowed.

The Clear Control logic generates the eight individual clear signals for the bits in the Interrupt Latches and Register. The Vector Clear Enable Flip-Flop indicates if the last vector read was from this chip. When it is set it enables the Clear Control Logic.

The CP clock signal is used to clock the Interrupt Register, Mask Register, Status Register, Vector Hold Register, and the Lowest Group Enabled, Vector Clear Enable and Status Overflow Flip-Flops, all on the clock LOW-to-HIGH transition.

The Am2914 can be microprogrammed in many different ways. Figure 4 shows an example interrupt sequence. The *Read Vector* microinstruction is necessary in order to read the interrupt priority level. Since vector plus one is automatically loaded into the Status Register when a *Read Vector* microinstruction is executed, the Status Register possibly will overflow and disable all interrupts. For this reason, the Status

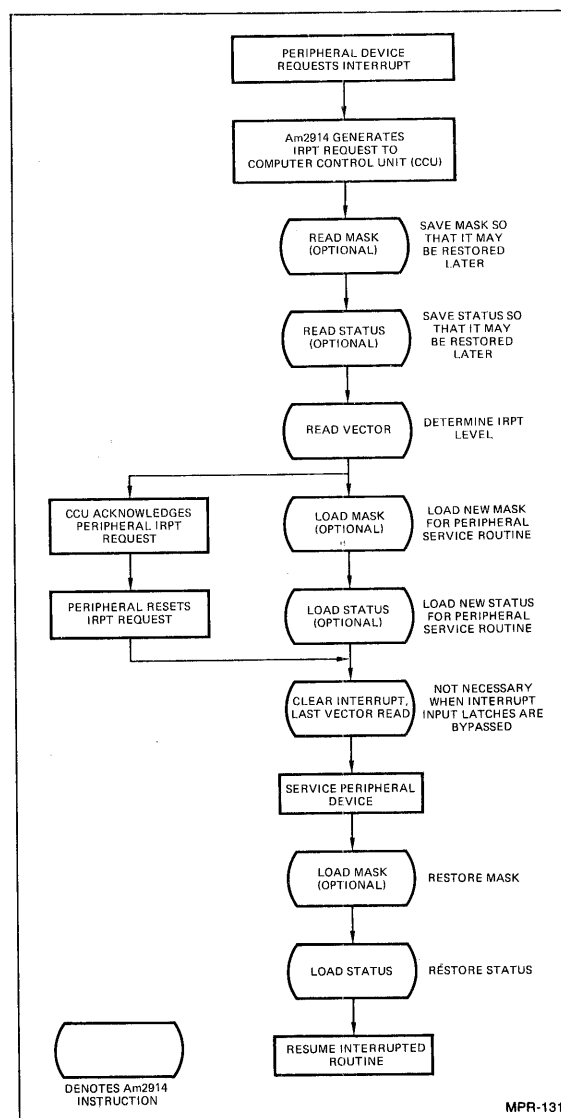


Figure 4. Example Interrupt Sequence.

Register must be reloaded periodically. The other Am2914 microinstructions are optional.

CASCADING THE Am2914

A number of input/output signals are provided for cascading the Am2914 Vectored Priority Interrupt Encoder. A definition of these I/O signals and their required connections follows:

Group Signal (\overline{GS}) – This signal is the output of the Lowest Group Enabled flip-flop and during a Read Status microinstruction is used to generate the high order bits of the Status word.

Group Enable (\overline{GE}) – This signal is one of the inputs to the Lowest Group Enable flip-flop and is used to load the flip-flop during the Load Status microinstruction.

Am2914

Group Advance Send (\overline{GAS}) – During a Read Vector microinstruction, this output signal is LOW when the highest priority vector (vector seven) of the group is being read. In a cascaded system Group Advance Send must be tied to the Group Advance Receive input of the next higher group in order to transfer status information.

Group Advance Receive (\overline{GAR}) – During a Master Clear or Read Vector microinstruction, this input signal is used with other internal signals to load the Lowest Group Enabled flip-flop. The Group Advance Receive input of the lowest priority group must be tied to ground.

Status Overflow (\overline{SV}) – This output signal becomes LOW after the highest priority vector (vector seven) of the group has been read and indicates the Status Register has overflowed. It stays LOW until a Master Clear or Load Status microinstruction is executed. The Status Overflow output of the highest priority group should be connected to the Interrupt Disable input of the same group and serves to disable all interrupts until new status is loaded or the system is master cleared. The Status Overflow outputs of lower priority groups should be left open (see Figure 7).

Interrupt Disable (\overline{ID}) – When LOW, this input signal inhibits the Interrupt Request output from the chip and also generates a Ripple Disable output.

Ripple Disable (\overline{RD}) – This output signal is used only in the Ripple Cascade Mode (see below). The Ripple Disable output is LOW when the Interrupt Disable input is LOW, the Lowest Group Enabled flip-flop is LOW, or an Interrupt Request is generated in the group. In the ripple cascade mode, the

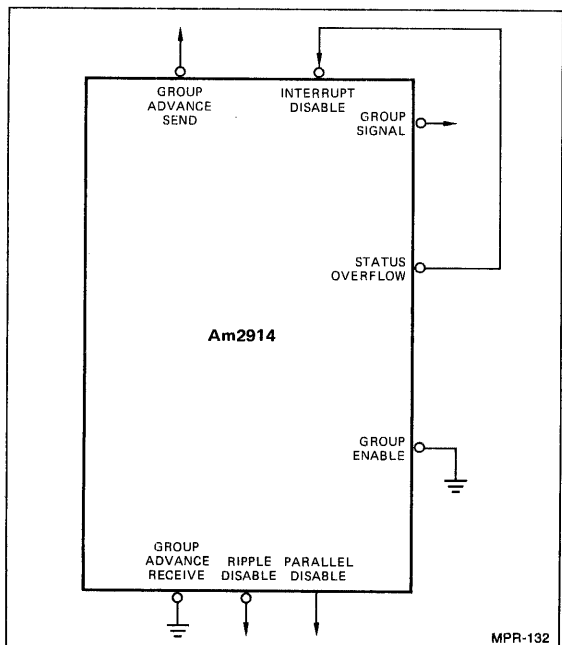


Figure 5. Cascade Lines Connection for Single Chip System.

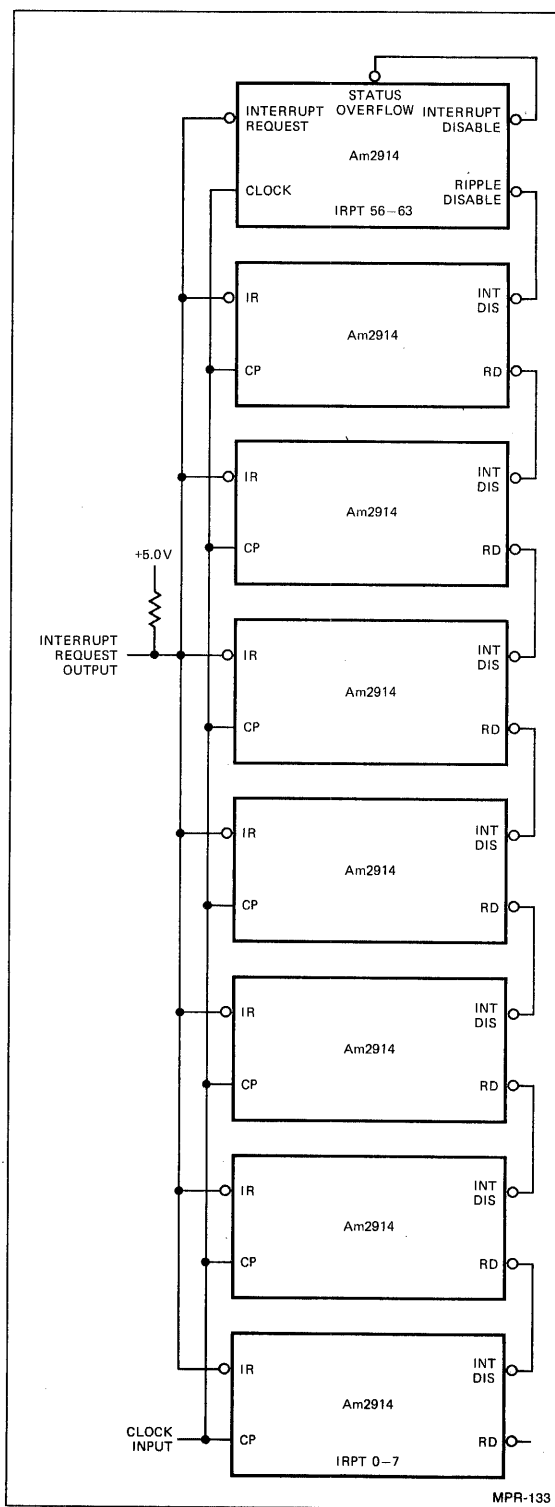


Figure 6. Interrupt Disable Connections for Ripple Cascade Mode.



Am2914

Ripple Disable output is tied to the Interrupt Disable input of the next lower priority group (see Figure 6).

Parallel Disable (PD) — This output is used only in the parallel cascade mode (see below). It is HIGH when the Lowest Group Enabled flip-flop is LOW or an Interrupt Request is generated in the group. It is not affected by the Interrupt Disable input.

A single Am2914 chip may be used to prioritize and encode up to eight interrupt inputs. Figure 5 shows how the above cascade lines should be connected in such a single chip system.

The Group Advance Receive and Group Enable inputs should be connected to ground so that the Lowest Group Enabled flip-flop is forced LOW during a *Master Clear* or *Load Status* microinstruction. Status Overflow should be connected to Interrupt Disable in order to disable interrupts when vector seven is read. The Group Advance Send, Ripple Disable, Group Signal and Parallel Disable pins should be left open.

The Am2914 may be cascaded in either a Ripple Cascade Mode or a Parallel Cascade Mode. In the Ripple Cascade Mode, the Interrupt Disable signal, which disables lower priority interrupts, is allowed to ripple through lower priority groups. Figures 6, 9 and 11 show the cascade connections required for a ripple cascade 64 input interrupt system.

In the parallel cascade mode, a parallel lookahead scheme is employed using the high-speed Am2902 Lookahead Carry Generator. Figures 7, 9 and 10 show the cascade connections required for a parallel cascade 64-input interrupt system. For this application, the Am2902 is used as a lookahead interrupt disable generator. A Parallel Disable output from any group results in the disabling of all lower priority groups in parallel. Figure 8 shows the Am2902 logic diagram and equations.

In Figures 9 and 10, the Am2913 Priority Interrupt Expander is shown forming the high order bits of the vector and status, respectively. The Am2913 is an eight-line to three-line priority encoder with three-state outputs which are enabled by the five output control signals G_1 , G_2 , G_3 , G_4 , and G_5 . In Figure 9, the Am2913 is connected so that its outputs are enabled during a Read Vector instruction, and in Figure 10 the Am2913 is connected so that its outputs are enabled during a Read Status instruction. The Am2913 logic diagram and truth table are shown in Figure 11.

The Am25LS138 three-line to eight-line Decoder also is shown in Figure 10. It is used to decode the three high order status bits during a Load Status instruction. The Am25LS138 logic diagram and truth table are shown in Figure 12.

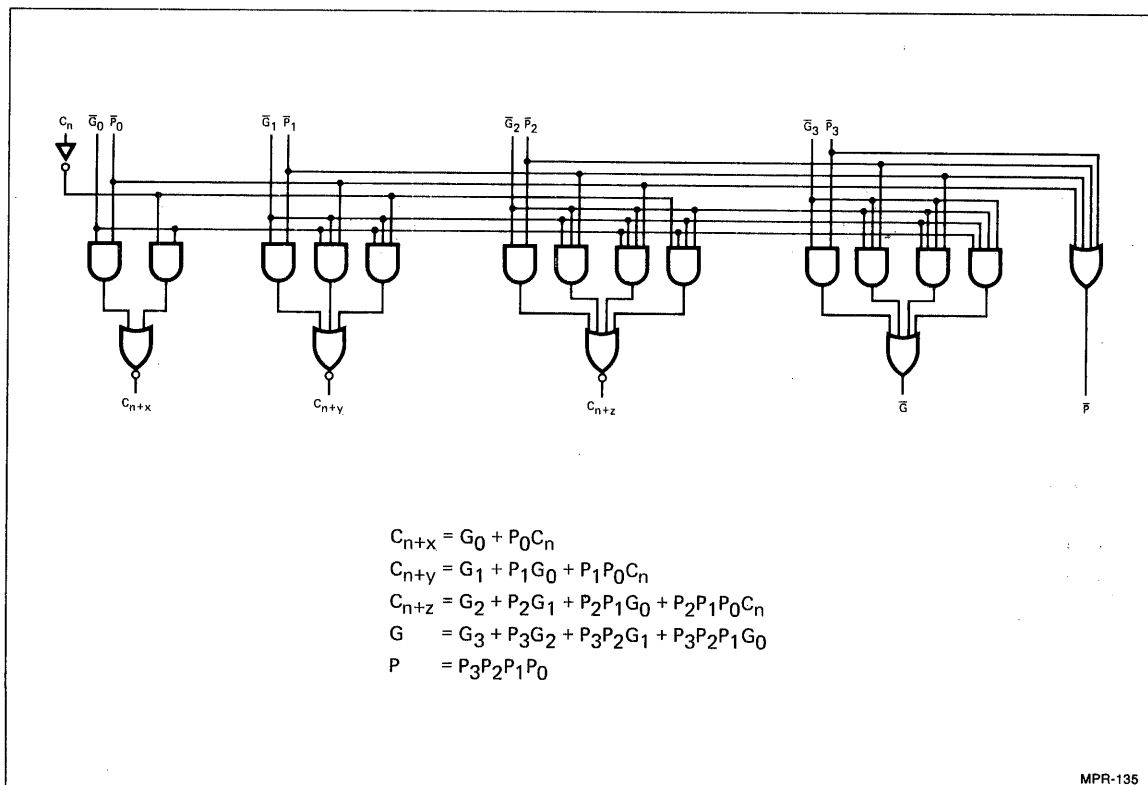


Figure 8. Am2902 Carry Look-Ahead Generator Logic Diagram and Equations.

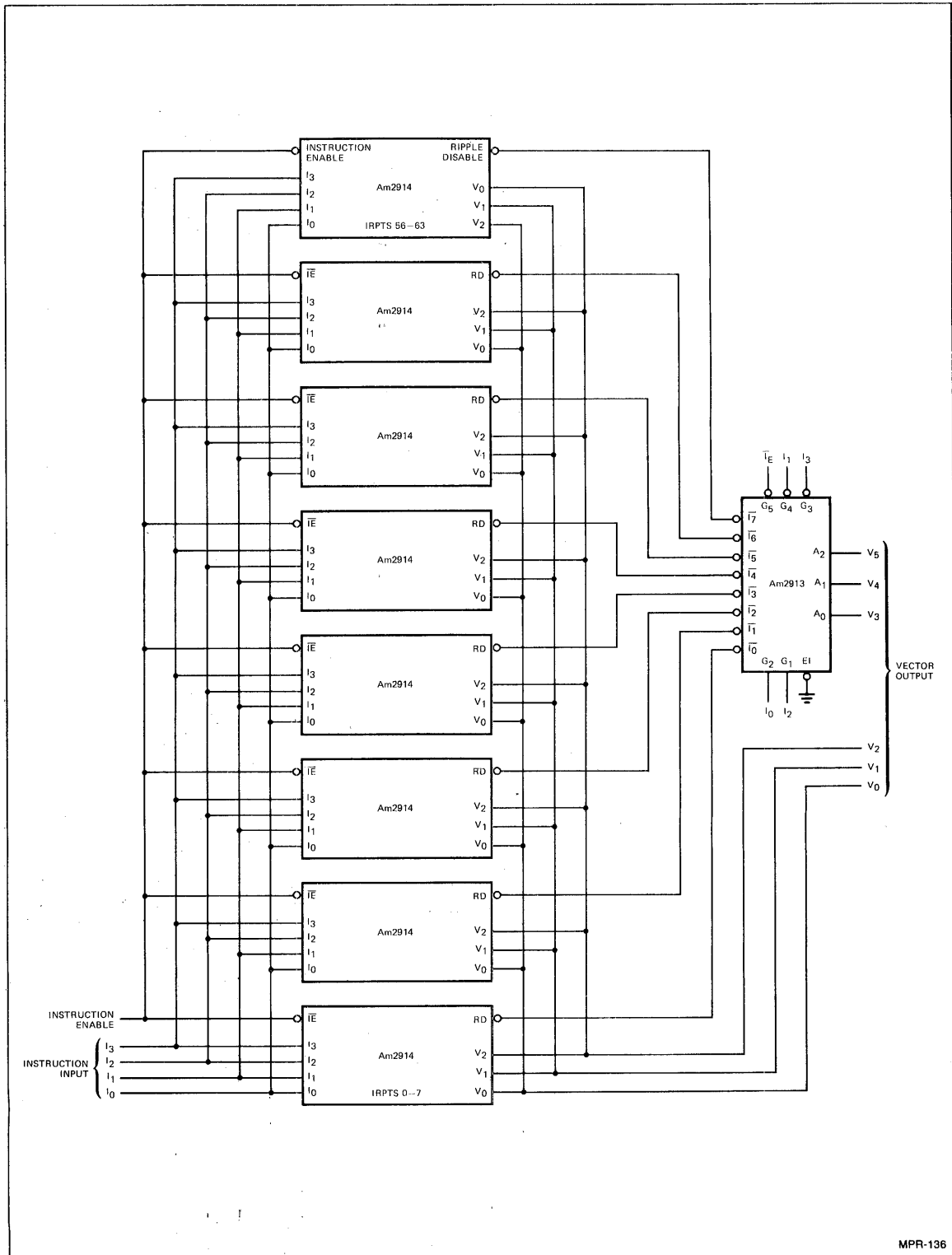


Figure 9. Vector Connections for both the Parallel and Ripple Cascade Modes.

2-174

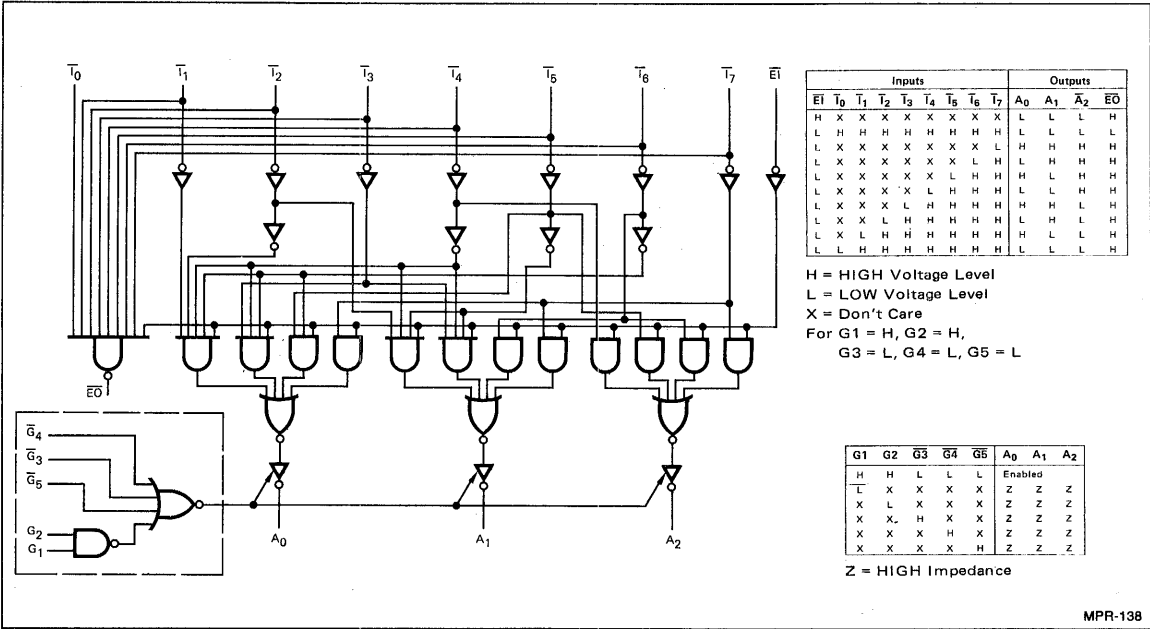


Figure 11. Am2913 Priority Interrupt Expander Logic Diagram and Truth Table.

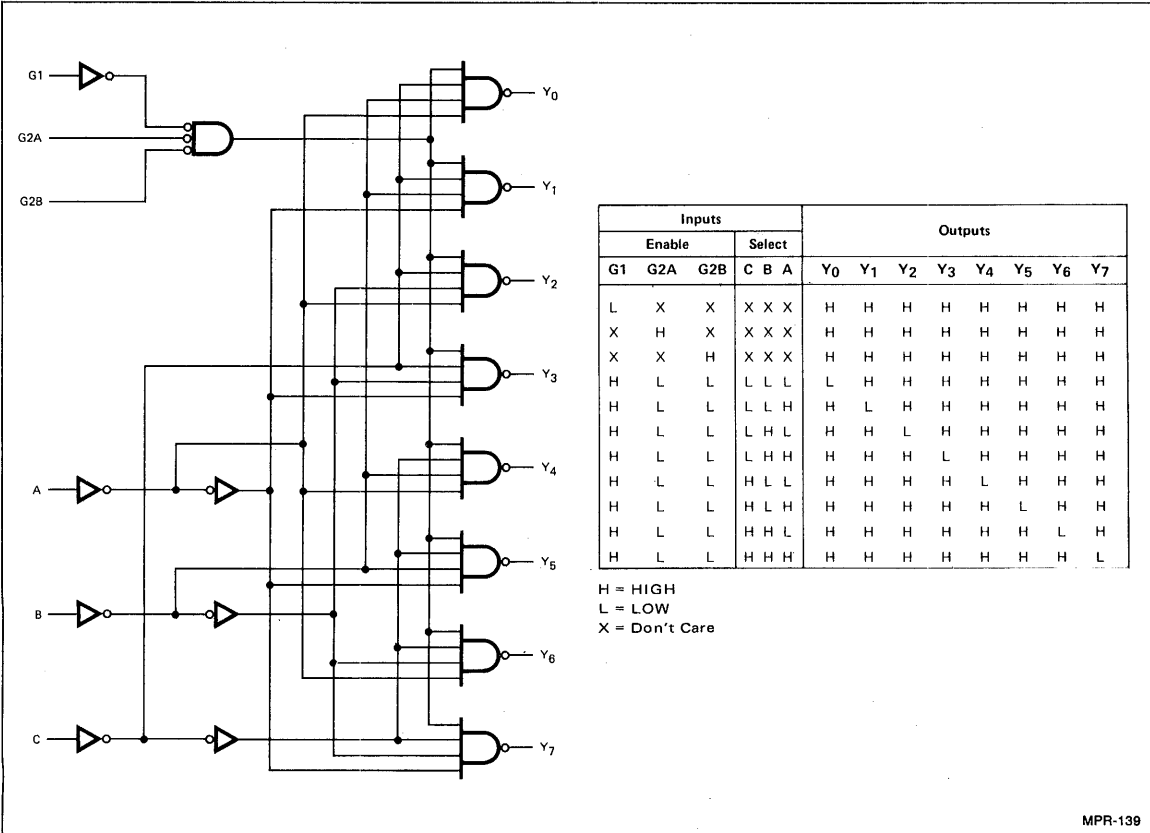


Figure 12. Am25LS138 3 to 8 Line Decoder Logic Diagram and Truth Table.

EXAMPLE INTERRUPT SYSTEMS DESIGNS FOR AN Am2900 SYSTEM

A classical computer architecture is shown in Figure 13. The Computer Control Unit controls the internal busses and subsystems of the processor, synchronizes internal and external events and grants or denies permission to external systems. The data bus is commonly used by all of the subsystems in the computer. Information, instructions, address operands, data and sometimes control signals are transmitted down the data bus under control of a microprogram. The microprogram selects the source of the data as well as the destination(s) of the data. The Address Bus is typically used to select a word in memory for an internal computer function or to select an input/output port for an external subsystem or peripheral function. The source of the data for the address bus, also selected by microprogram commands, may be the program counter, the memory address register, a direct memory address controller, an interface controller, etc.

The arithmetic/logic unit (ALU) is that portion of the processor that computes. Under control of the microprogram, the ALU performs a number of different arithmetic and logic functions on data in the working registers or from the data bus. The ALU also provides a set of condition codes as a result of the current arithmetic or logic operation. The condition codes, along with other computer status information, are

stored in a register for later use by the programmer or computer control unit.

The program counter and the memory address register are the two main sources of memory word and I/O address select data on the address bus. The program counter contains the address of the next instruction or instruction operand that is to be fetched from main memory, and the memory address register contains instruction address operands which are necessary to fetch the data required for the execution of the current instruction.

A subroutine address stack is provided to allow the return address linkage to be handled easily when exiting a subroutine. The address stack is a last-in, first-out stack that is controlled by a jump-to-subroutine, PUSH, or a return-from-subroutine, POP, instruction from the CCU microinstruction word.

The next microprogram address control (NMAC) circuitry controls the generation of microinstruction addresses. Based on microprogram control, interrupt requests, test conditions and commands from a control panel or other processor, the NMAC determines the address of the next microinstruction to be executed.

For a more detailed description of the above portions of the computer, refer to Advanced Micro Devices' Application Note *A Microprogrammed 16 Bit Computer* by James R.W. Clymer.

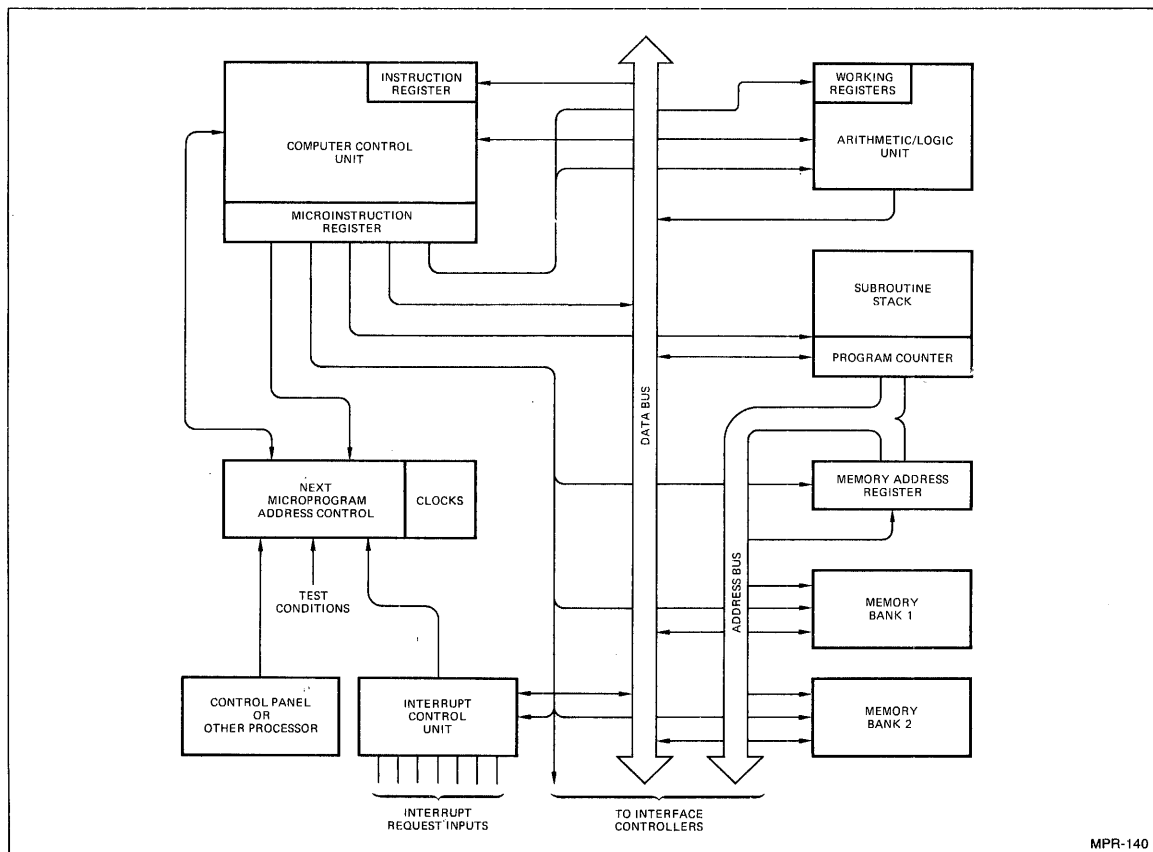


Figure 13. Generalized Computer Architecture.

Am2914 PRIORITY INTERRUPT ENCODER DETAILED LOGIC DESCRIPTION

INTRODUCTION

A clear understanding of the Am2914 Priority Interrupt controller's operation facilitates its efficient use. With that idea in mind, a detailed logic description of the Am2914 is presented here. A detailed logic diagram and control signal truth table are shown, and significant aspects of the Am2914 design are described verbally.

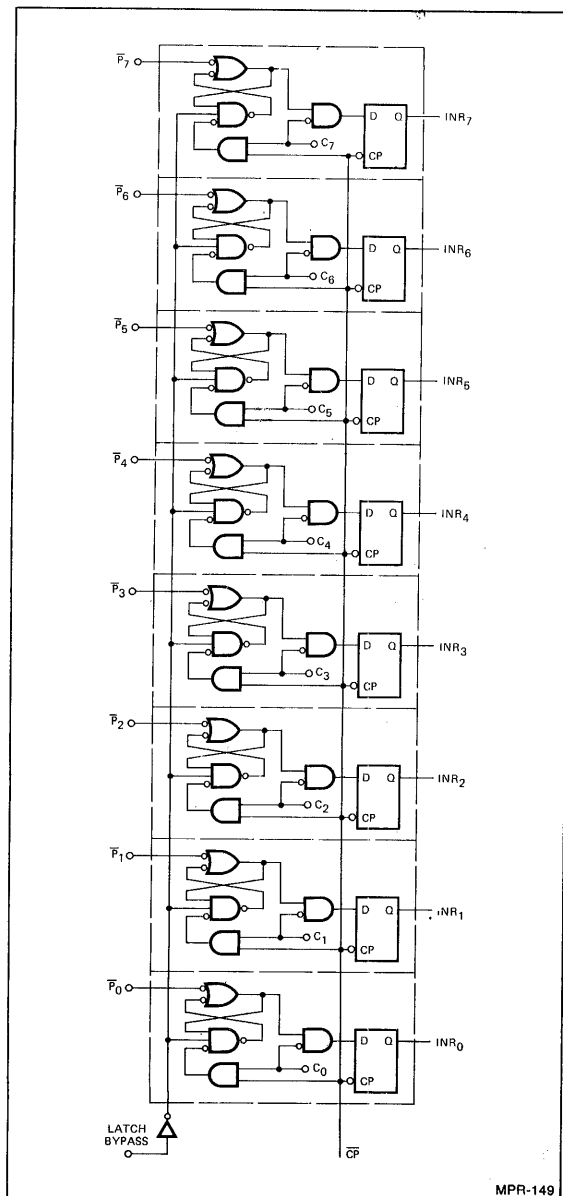


Figure 1. Interrupt Latches and Register.

LOGIC DIAGRAM DESCRIPTION

The Interrupt Latches and Register are shown in Figure 1. The Interrupt latches are set/reset-type latches. When the Latch Bypass signal is LOW, the latches are enabled and act as negative pulse catchers on the inputs to the Interrupt Register. When the Latch Bypass signal is HIGH, the Interrupt latches are transparent. The Interrupt Register holds the Interrupt Inputs and is an eight-bit, edge-triggered register. It is updated on the LOW-to-HIGH transition of the clock pulse (HIGH-to-LOW transition of the CP signal) as are all of the flip-flops on the chip.

2

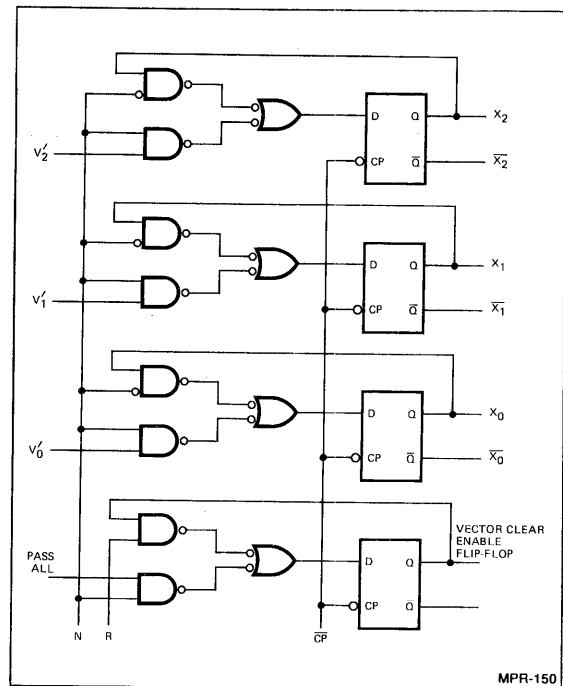


Figure 2. Vector Hold Register

When a Read Vector instruction is executed, the binary coded vector is loaded into the Vector Hold Register of Figure 2. This stored vector can be used later for clearing the interrupt associated with the last vector that was read. The Vector Clear Enable Flip-Flop of Figure 2 is set when a Read Vector instruction is executed and the PASS ALL signal is HIGH. A HIGH PASS ALL signal level indicates that this group is enabled and that an interrupt request in this group was detected and passed priority. The Vector Hold Register and the Vector Clear Enable Flip-Flop are cleared when a Master Clear, Clear All Interrupts, or Clear Interrupt Last Vector Read is executed. Table 1 shows the generation of the "N and R" control signals for each of these operations.

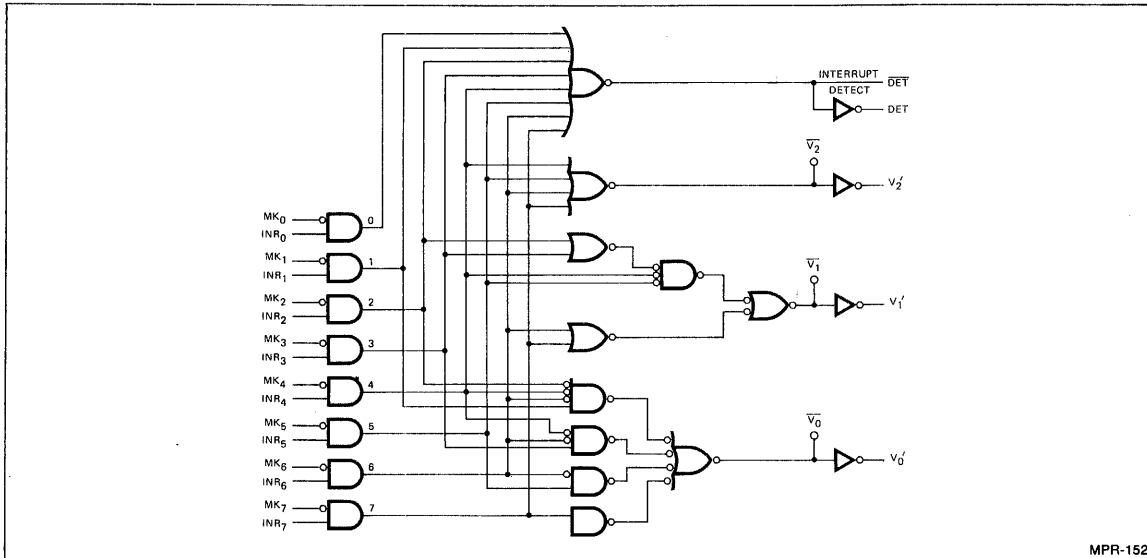


Figure 4. Interrupt Request Detect and Priority Decoder.

The Interrupt Request Detect and Priority Encode circuitry are shown in Figure 4. The Interrupt Detect circuitry detects the presence of any unmasked Interrupt Input. The

eight-input Priority Encoder determines the highest priority, non-masked Interrupt Input and forms a binary coded interrupt vector, V_0 - V_2 .

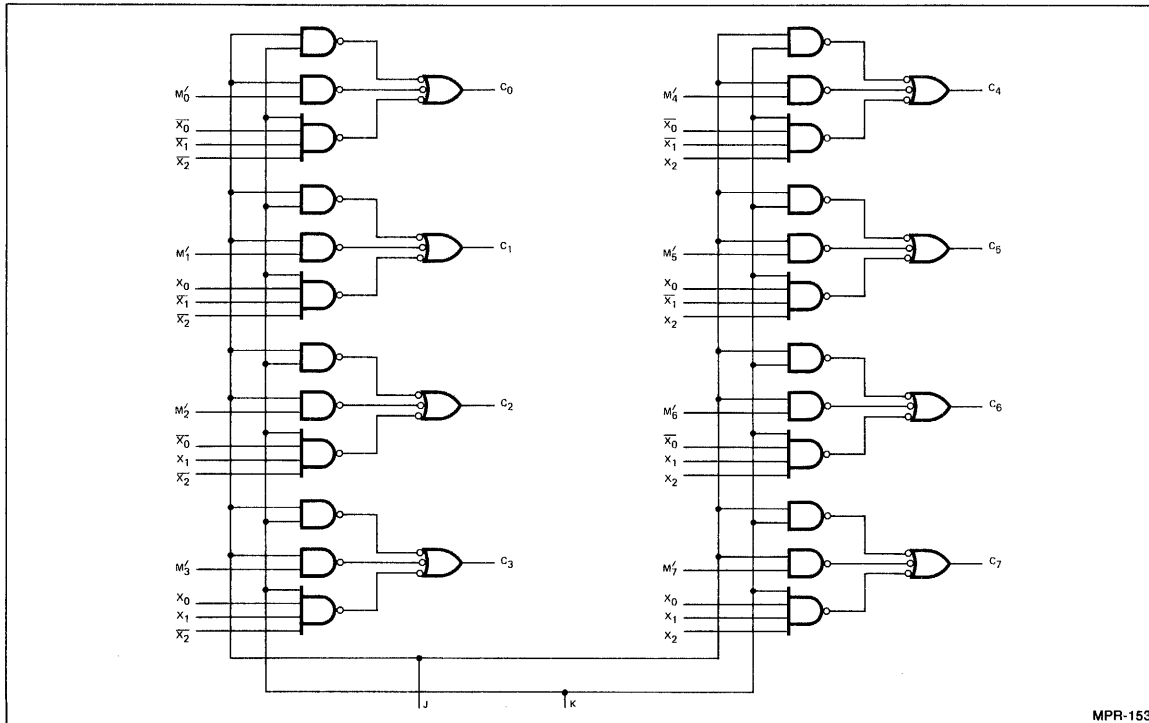


Figure 5. Clear Control.

The Clear Control logic of Figure 5 generates the eight individual clear signals for the eight Interrupt Register bits. Under microinstruction control, all interrupts, interrupts with corresponding mask register bits set, interrupts with

corresponding mask bus bits equal to one, or the interrupt associated with the last vector read may be cleared. Table 1 shows the generation of the "J" and "K" control signals for each of these operations.

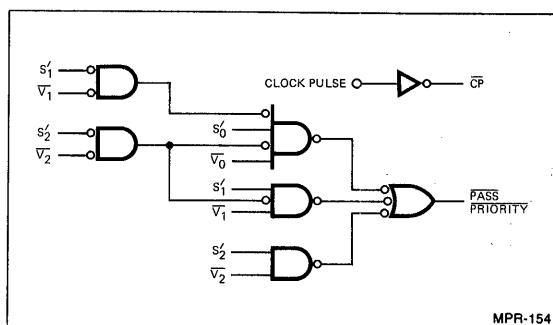


Figure 6. Three-Bit comparator.

The three-bit Comparator of Figure 6 compares the interrupt vector with the contents of the Status Register. A LOW signal level at the PASS PRIORITY output indicates that the interrupt vector is greater than or equal to the contents of the Status Register.

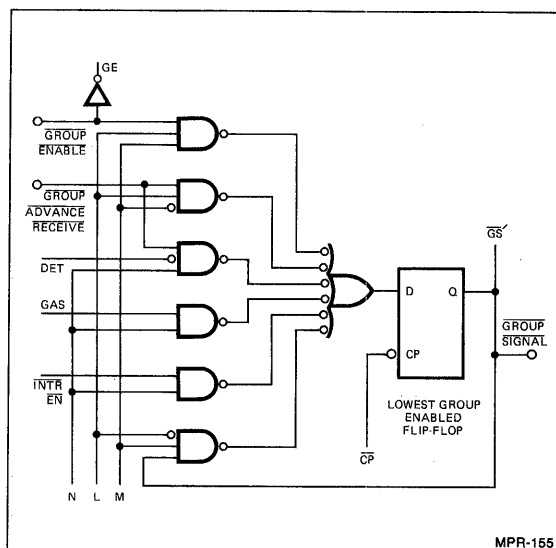


Figure 7. Group Enable Logic.

The Lowest Group Enabled Flip-Flop, Figure 7, is used when a number of Am2914's are cascaded. In a cascaded system, only one Lowest Group Enabled Flip-Flop is LOW at a time. It indicates the group which contains the lowest priority interrupt which will be accepted and is used to form the high order status bits. When a Load Status instruction is executed, the flip-flop is loaded from the GROUP ENABLE input. When a

Master Clear instruction is executed, it is loaded from the GROUP ADVANCE RECEIVE input. The flip-flop is set HIGH when a Read Vector instruction is executed if a Group Advance is not received and no interrupt in this group is detected, if a Group Advance is sent from this group, or if interrupts from this group are disabled. For all other instructions, the flip-flop remains the same. Table 1 shows the generation of the "N", "L" and "M" control signals for these operations.

The Status Register holds the status bits and may be loaded from or read to the "S" bus as shown in Figure 8. Note that when a Load Status instruction is executed, status from the "S" bus is loaded into the Status Register only if the GROUP ENABLE input is LOW; if the GROUP ENABLE input is HIGH, the Status Register is cleared. Also note that during a Read Status instruction, the Status Register outputs are enabled onto the "S" bus only if the Lowest Group Enabled Flip-Flop of this group is LOW. When a Read Vector instruction is executed, the incrementer increases the vector by one and the result is loaded into the Status Register. Thus, the Status Register always points to the lowest level at which an interrupt will be accepted. Table 1 shows the generation of the "F", "G" and "OE-S" control signals for Status Register operations.

The Interrupt Request Logic, shown in Figure 9, generates the RIPPLE DISABLE, PARALLEL DISABLE, INTERRUPT REQUEST, GROUP ADVANCE SEND, and STATUS OVERFLOW output signals. The PARALLEL DISABLE signal is generated when the Lowest Group Enabled signal is LOW or an interrupt request in this group is detected and passes priority. The RIPPLE DISABLE signal is generated when the PARALLEL DISABLE signal is generated and also when the INTERRUPT DISABLE input signal is LOW. The INTERRUPT REQUEST output signal is generated when interrupt requests in this group are enabled and a request is detected and passes priority. The GROUP ADVANCE SEND output signal is generated when a vector of value seven is being read. The Status Overflow Flip-Flop is set LOW when a vector of value seven is read and indicates the Status Register has overflowed. The Interrupt Request Enable Flip-Flop is either set or reset by the Enable Request or Disable Request microinstructions respectively, and is used to enable or disable the INTERRUPT REQUEST output. Table 1 shows the generation of control signals "D", "E", "S" and "H".

Note that the vector outputs are enabled only when a Read Vector is being executed. Also note that when a Read Vector instruction is executed, the vector outputs will be disabled after the execution of the instruction since the Status Register is loaded with V+1, and the INTERRUPT REQUEST will no longer be generated.

The Microinstruction Decode circuitry, Figure 10, decodes the Am2914 microinstructions and generates the required internal control signals. Table 1 shows the truth table for these functions and Figure 11 shows the function tables.

Table 1. Am2914 Control Signal Truth Table.
0 = LOW, 1 = HIGH

Microinstruction						Function	Mask Register				Status Register			Group Enable		Clear Control		Irpt Request Enable		Vector Hold Register		Other		
Decimal	I _E	I ₃	I ₂	I ₁	I ₀	Description	A	B	C	OE-M	F	G	OE-S	L	M	J	K	D	E	N	R	S	H	
0	0	0	0	0	0	Master Clear	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	0	1	1
1	0	0	0	0	1	Clear All Interrupts	1	0	1	0	0	1	1	0	1	1	1	1	1	X	0	0	1	0
2	0	0	0	1	0	Clear Intr Via M Bus	1	0	1	0	0	1	1	0	1	1	1	0	1	X	0	1	1	0
3	0	0	0	1	1	Clear Intr Via M Reg	1	0	1	1	0	1	1	0	1	1	0	1	1	X	0	1	1	0
4	0	0	1	0	0	Clear Intr, Last Vector	1	0	1	0	0	1	1	0	1	0	1/0	1	X	0	0	1	0	0
5	0	0	1	0	1	Read Vector	1	0	1	0	0/1	0	1	0	0	0	0	1	X	1	0	0	1	0
6	0	0	1	1	0	Read Status Reg	1	0	1	0	0	1	0	0	1	0	0	1	X	0	1	1	0	0
7	0	0	1	1	1	Read Mask Reg	1	0	1	1	0	1	1	0	1	0	0	1	X	0	1	1	0	0
8	0	1	0	0	0	Set Mask Reg	0	0	0	0	0	1	1	0	1	0	0	1	X	0	1	1	0	0
9	0	1	0	0	1	Load Status Reg	1	0	1	0	1	1	1	1	1	0	0	1	X	0	1	1	1	0
10	0	1	0	1	0	Bit Clear Mask Reg	0	1	0	0	0	1	1	0	1	0	0	1	X	0	1	1	0	0
11	0	1	0	1	1	Bit Set Mask Reg	1	1	1	0	0	1	1	0	1	0	0	1	X	0	1	1	0	0
12	0	1	1	0	0	Clear Mask Reg	0	0	1	0	0	1	1	0	1	0	0	1	X	0	1	1	0	0
13	0	1	1	0	1	Disable Request	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	1	1	0
14	0	1	1	1	0	Load Mask Reg	0	1	1	0	0	1	1	0	1	0	0	1	X	0	1	1	0	0
15	0	1	1	1	1	Enable Request	1	0	1	0	0	1	1	0	1	0	0	0	1	0	1	1	0	0
X	1	X	X	X	X	Instruction Disable	1	0	1	0	0	1	1	0	1	0	0	1	X	0	1	1	0	0

Notes: 1. Control line "F" during "READ VECTOR" instruction is 0 when "PASS ALL" is LOW and 1 when "PASS ALL" is HIGH.
2. Control line "K" during "Clear Intr, Last Vector" instruction is 0 when "Vector Clear Enable" is LOW and 1 when "Vector Clear Enable" is HIGH.

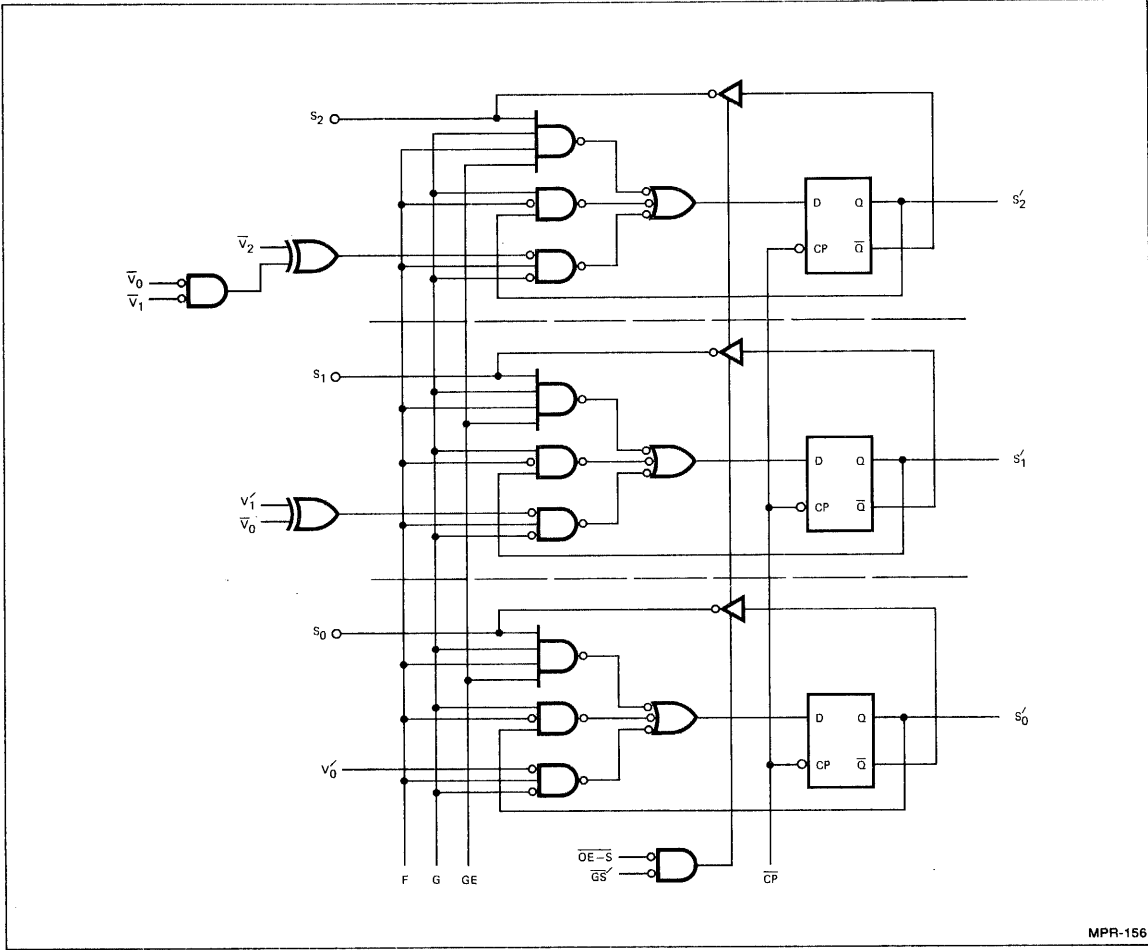


Figure 8. Incrementer and Status Register.

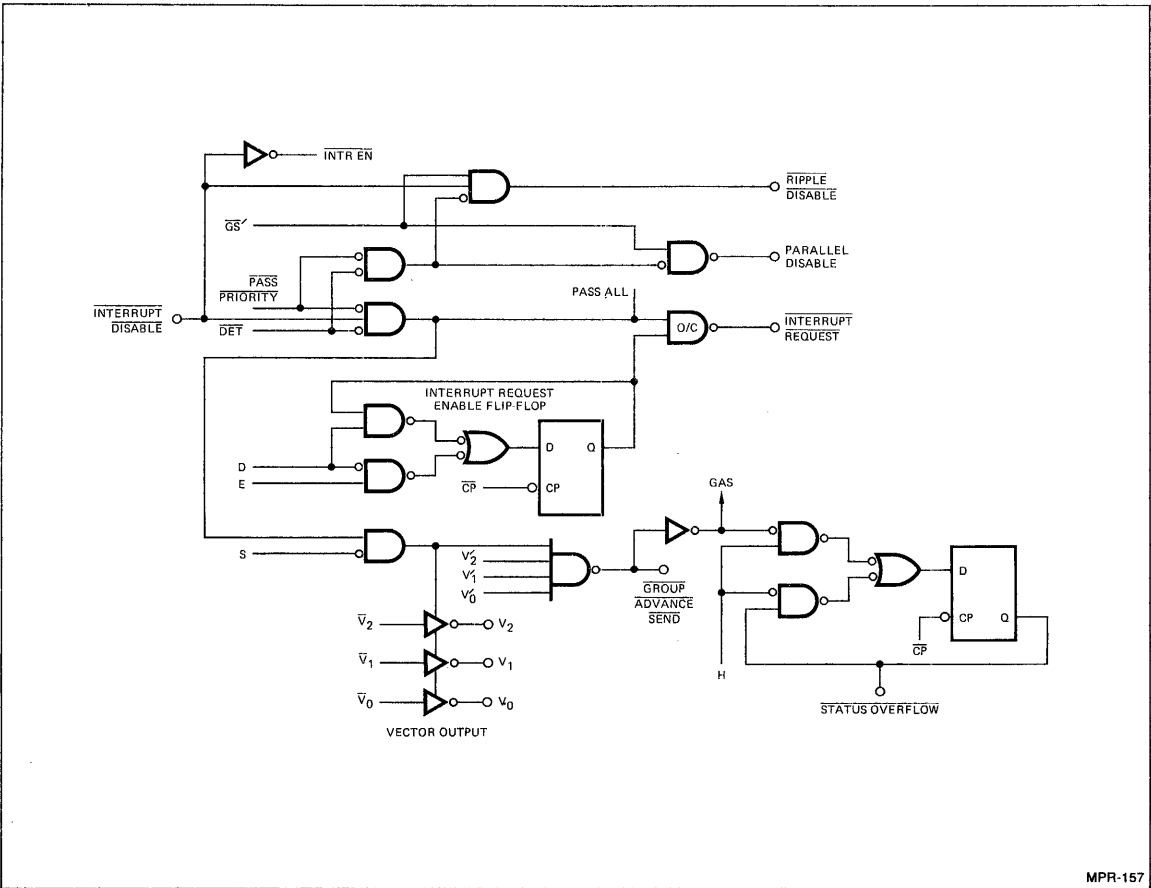


Figure 9. Interrupt Request Logic.

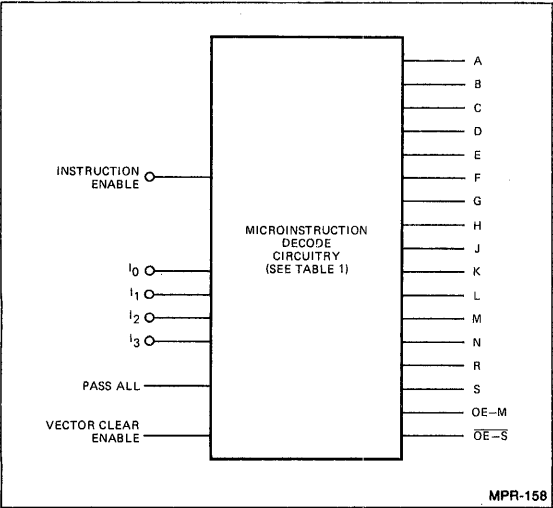


Figure 10.

MASK REGISTER			STATUS REGISTER		
A	B	C	F	G	FUNCTION
0	0	0	0	0	SET
0	0	1	0	1	CLEAR
0	1	0	0	1	HOLD
0	1	1	1	0	LOAD VECTOR + 1
1	0	1	1	1	LOAD VIA "S" BUS
1	1	0			
1	1	1			

CLEAR CONTROL			INTERRUPT REQUEST ENABLE FLIP-FLOP		
J	K	FUNCTION	D	E	FUNCTION
0	0	NO CLEAR	0	0	DISABLE IRPTS
0	1	CLEAR IRPT, VECTOR	0	1	ENABLE IRPTS
1	0	CLEAR IRPTS VIA M	1	X	HOLD
1	1	CLEAR ALL IRPTS			

VECTOR HOLD REGISTER			VECTOR CLEAR ENABLE FLIP-FLOP		
N		FUNCTION	N	R	FUNCTION
0		HOLD	0	0	CLEAR
1		LOAD	0	1	HOLD
			1	0	LOAD

LOWEST GROUP ENABLED FLIP-FLOP			STATUS OVERFLOW FLIP-FLOP		
L	M	FUNCTION	H		FUNCTION
0	0	UPDATE	0		HOLD
0	1	HOLD	1		LOAD
1	0	LOAD VIA GROUP ADVANCE RECEIVE			
1	1	LOAD VIA GROUP ENABLE			

Figure 11. Control Function Tables.

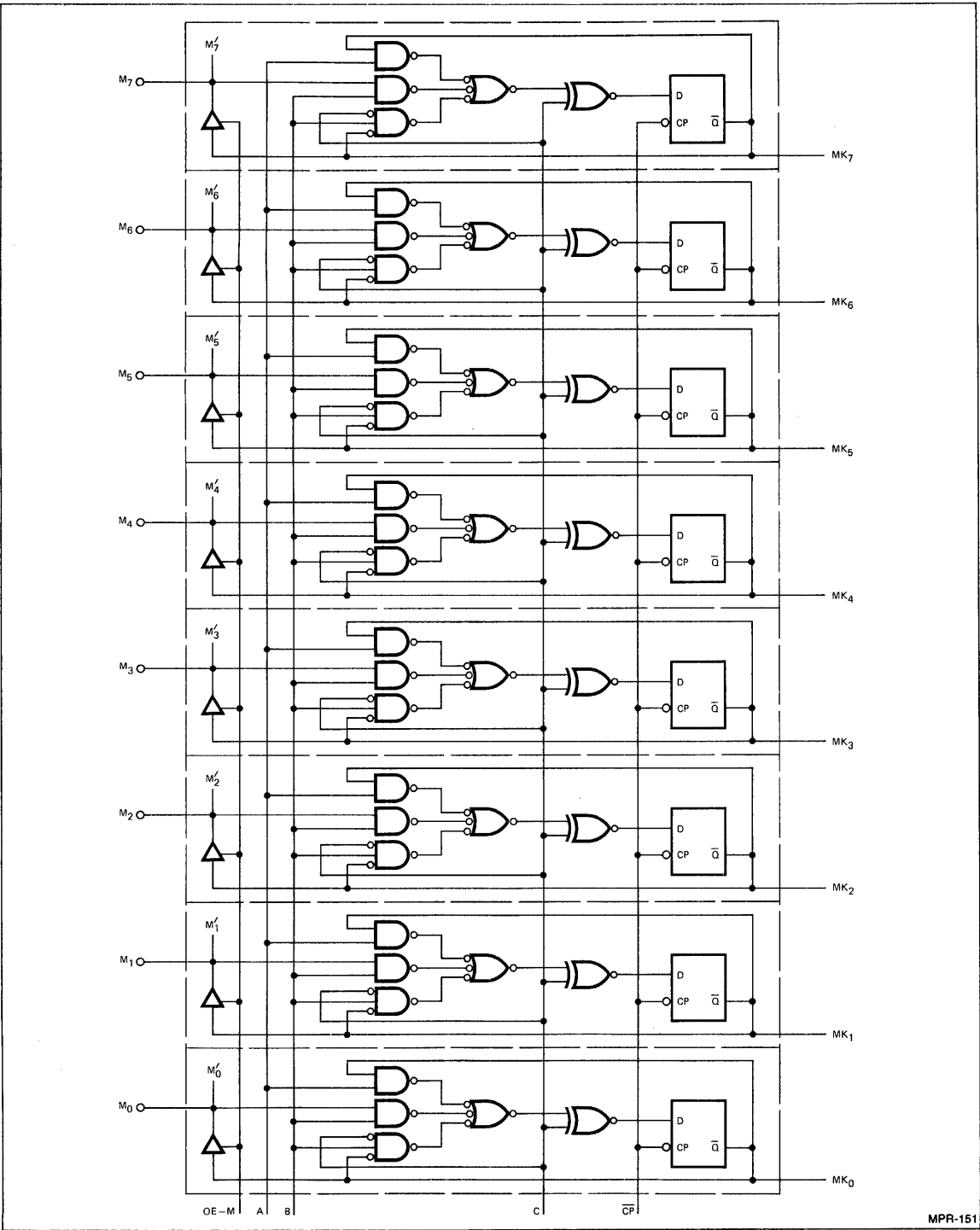


Figure 3. Mask Register.

The Mask Register shown in Figure 3 holds the eight mask bits associated with the eight interrupt levels. The register may be set or cleared, bit set or bit cleared from the "M"

bus, or loaded or read to the "M" bus. Table 1 shows the generation of the "A", "B", "C" and "OE-M" control signals for each of these operations.