HONEYWELL INFORMATION SYSTEMS ITALIA		SPEC. NO.	SHEET	REVISION
		A78xxxxxx	1/44	DRAFT1
Prepared by Date A. GRASSI 85.11.18	Designation PRODUCT DESIGN DESCRIPTION S. G. M. 2 S T A T I O N P R O C E S S O R 0			
Approved by Date				

REVISION RECORD

REV	AUTHORITY	DATE	APPROVED BY	SHEETS AFFECTED
				- ALL -

shared memory = 32K bytes
local memory = 64K bytes

```
****     ****     ***     *****    *****
*   *    *   *    *   *   *        *
*   *    *   *    *   *   *        *
*   *    ****     *****   ****     *
*   *    *  *     *   *   *        *
*   *    *   *    *   *   *        *
****     *   *    *   *   *        *
```

INDEX

1. GENERAL DESCRIPTION

The Station Processor 0 is an intelligent communication processor with SGM2-VME bus interface, which supports up to 8 full duplex channels for serial communications and one Centronics/IBM parallel printer interface.

The eight serial channels allow RS-232C or RS-422A asynchronous local connections via 9 pins connectors. One of these RS-422A interfaces is also configured for local asynchronous communications at high speed.

In VME enviroment the Station Processor 0 is a slave controller fully compatible which responds to 32-bit addressing and 8 or 16 bit data transfers. Communications between the system CPUs and the SP can take place in three ways:

-- from host to SP or vice versa by message interchange via a shared-RAM allocated on the SP;

-- from host to SP by writing an 1-bit attention register to interrupt the SP;

-- from SP to host by an interrupter that generates interrupts to the VME bus on any of the seven levels and supplies an 8-bit vector during interrupt acknowledge cycle. The request level and the vector are programmable by the local processor.

The memory of the SP consists of a 32Kbytes EPROM area and of a static RAM area subdivided in 64kbytes of local-memory and 32kbytes of shared-memory.

1.1 HARDWARE DESCRIPTION

The Major Block Diagram of the Station Processor 0 board is described in details in Fig. 1.1 where it is possible to individuate the following functional blocks:

- MICROPROCESSOR

  the microprocessor used is the 16 bit Motorola MC68000. It operates at 12.5 Mhz clock and it has an addressing capability up to 16 Mbytes (see para. 1.1.1);

- INT & INTA LOGIC

  this logic permits to handle the interrupt lines (see para. 1.1.2.);

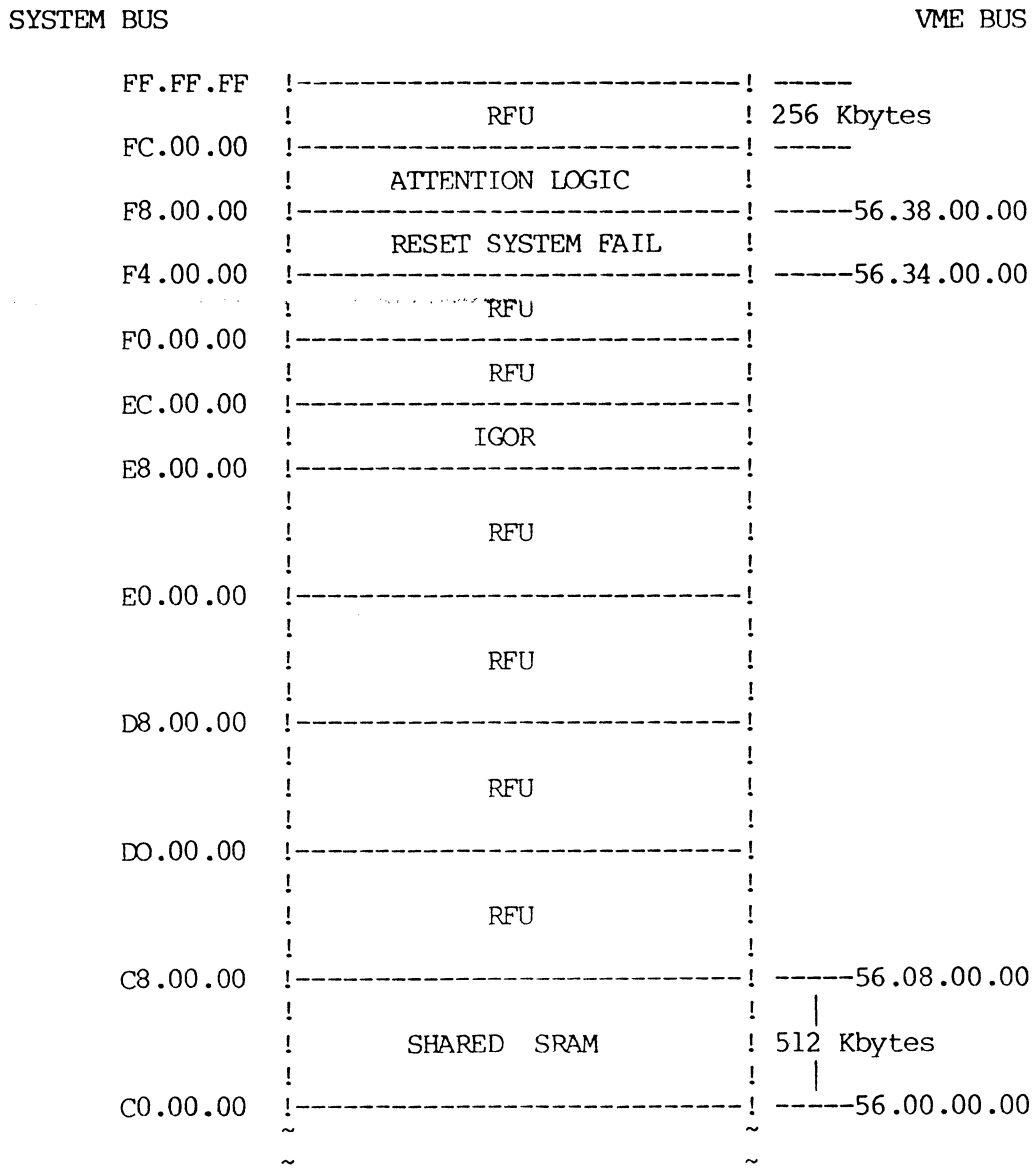
- CHIP SELECT AND CONTROL LOGIC FOR LOCAL/SHARED AREA

  the main functions carried out by this logic are the following:

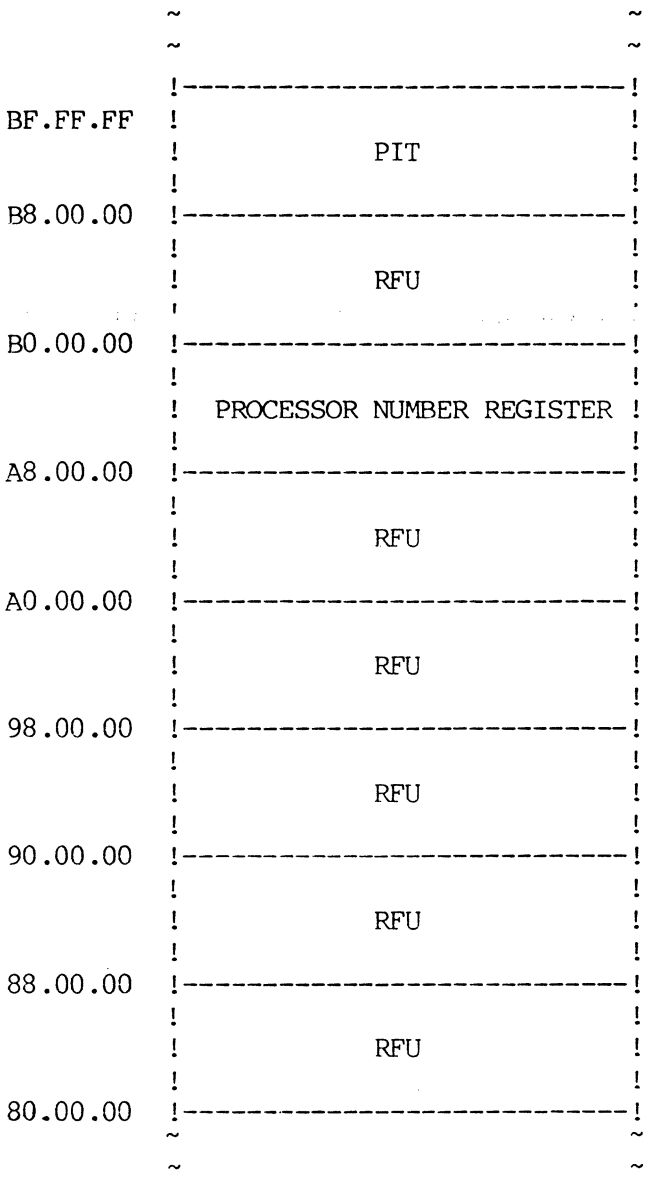
  - generation of the chip-selects;
  - generation of control signals (for example, the READ and WRITE clocks of the memory, etc);
  - generation of DATA TRANSFER ACKNOWLEDGE signal towards MC68000;
  - generation of a time-out signal (BERR line) every 16us if on-board peripheral don't return DTACK signal within this time;

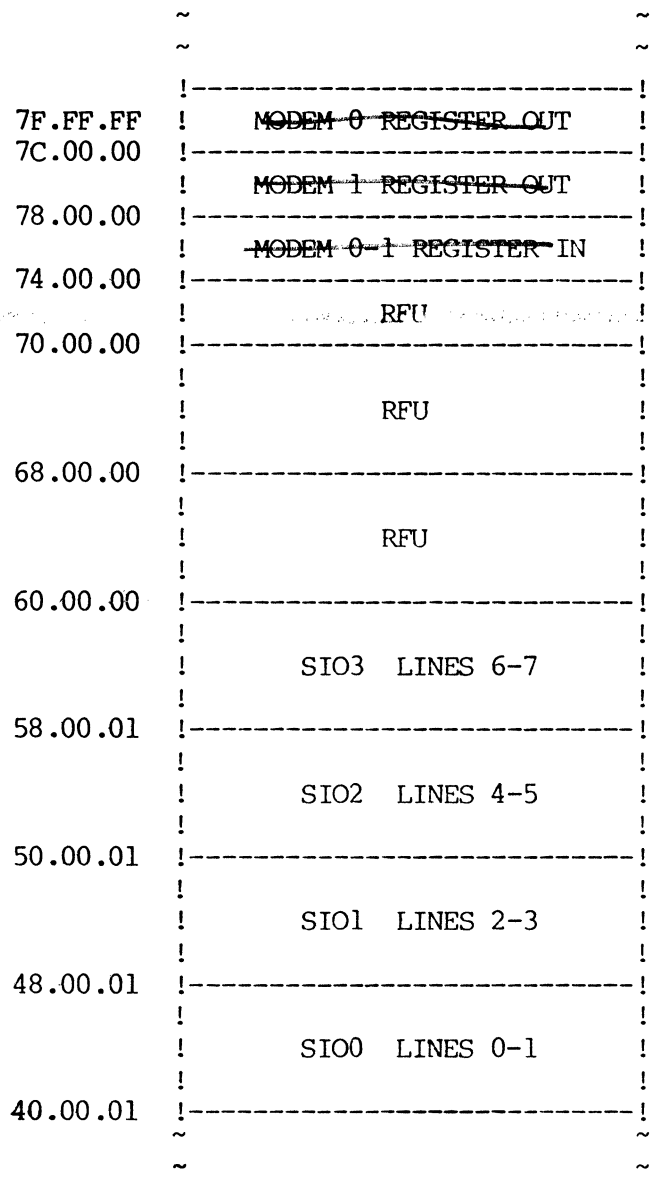
The 16 Mbyte addressing space is subdivided as shown below:

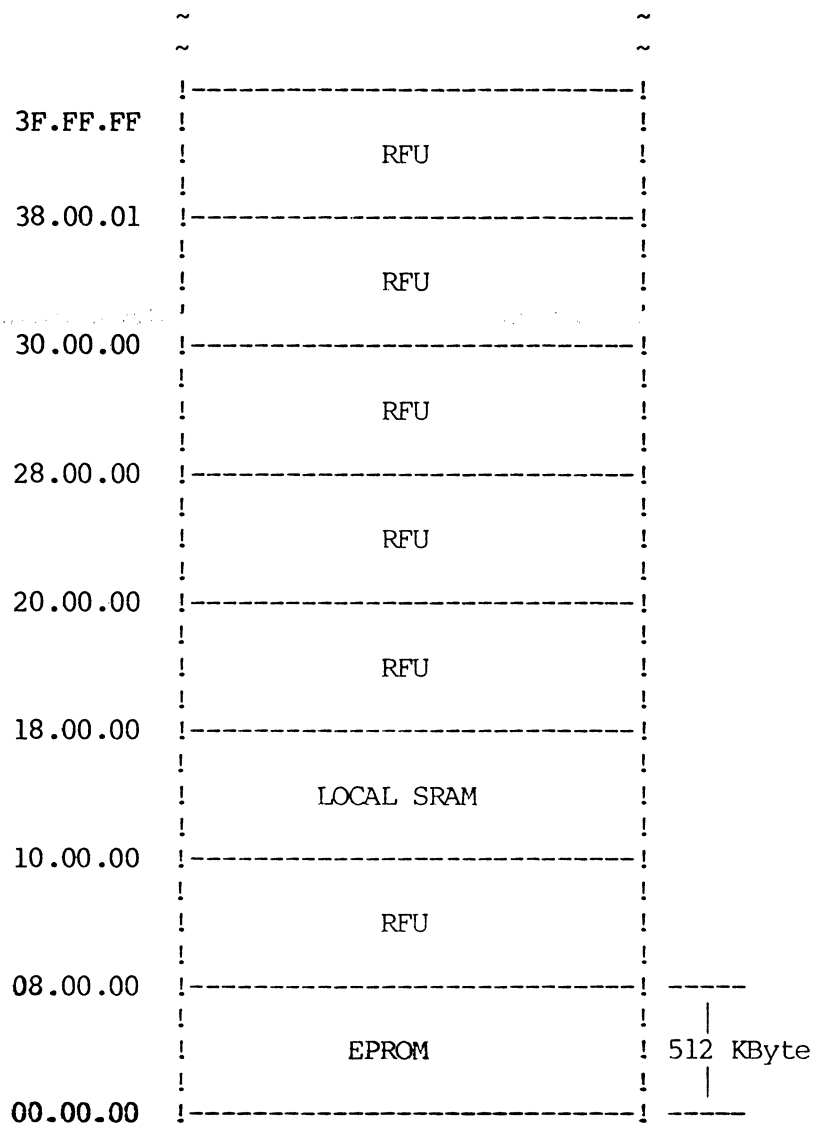
STATION PROCESSOR 0 - MASTER PROCESSOR MEMORY MAP



* Map for the first Station Processor with the Processor Number 0. For the others see para. 1.1.7 processor number and board type detection.


BF.FF.FF
PIT
B8.00.00
RFU
B0.00.00
PROCESSOR NUMBER REGISTER
A8.00.00
RFU
A0.00.00
RFU
98.00.00
RFU
90.00.00
RFU
88.00.00
RFU
80.00.00


7F.FF.FF
7C.00.00
78.00.00
74.00.00
70.00.00
68.00.00
60.00.00
58.00.01
50.00.01
48.00.01
40.00.01
MODEM 0 REGISTER OUT
MODEM 1 REGISTER OUT
MODEM 0-1 REGISTER IN
RFU
RFU
RFU
SIO3 LINES 6-7
SIO2 LINES 4-5
SIO1 LINES 2-3
SIO0 LINES 0-1


3F.FF.FF
RFU
38.00.01
RFU
30.00.00
RFU
28.00.00
RFU
20.00.00
RFU
18.00.00
LOCAL SRAM
10.00.00
RFU
08.00.00
EPROM
512 KByte
00.00.00

- ONBOARD/VME DECODE

this logic carries out the shared-bus requests to the BUSCON when either the local processor or the system CPUs want to transfer data into shared memory, or attention register, or BIL;

- EPROM

two 16 Kbytes packages (Eprom 27128) have been utilized. These two packages are connected in such a way as to constitute one 16 Kwords bank; the required access time is 200 ns which allows the dialogues to be carried out with one wait cycle. The 16Kwords EPROM bank is mapped in the following address range:

- 16 Kwords bank: 00.00.00 Hex --> 00.7F.FF Hex.

The Eprom code carries out the following functions:

- Exception Vector Table;
- Resident Diagnostic Routines;
- H/W initialization of the whole SP board;

- LOCAL MEMORY

the local memory consists of 8X8 Kbytes SRAM packages plus two 64K x1 SRAM packages utilized like check bit. The 8 Kbytes packages are connected in such a way to constitute one 32 Kwords bank. This memory can be accessed at byte length (right byte and left byte) and at word length without wait cycles.
The LOCAL MEMORY bank is mapped in the following address range:

10.00.00 Hex --> 10.FF.FF Hex;

When an error occurs in the Local Memory, the check logic sets the 68000 in the HALT state and asserts the VME SYSFAIL signal. SYSFAIL condition can be removed under Operating System control writing or reading (byte or word length) the RESET SYSTEM FAIL flip/flop located at the following address:

F4.00.00 Hex

The diagnostics can set the memory check bit to 0 for testing the error detection circuit. For this feature the PC4 Port C bit of the Parallel Interface Timer has been utilized.
To write always a 0 into the check memory, diagnostics must set the PC4 bit to 0. For The Port C Addressing space, refer to figure 1.1.4.3. (MC68230 Register Model).

- SERIAL INPUT OUTPUT

four USART (SIO) have been used. Each of these controllers can handle two Serial Ports (see para 1.1.3);

- PARALLEL INTERFACE TIMER (PIT)

it is a logic which permits to send single or periodic programmable interrupts to the MC68000 microprocessor and to connect the printers having an IBM and CENTRONICS type parallel interface (see para. 1.1.4). PIT is also utilized for diagnostics purpouses;

- VME BUS CONTROLLER (BUSCON)

it is an interface device that assures VMEbus compatibility, allowing either the SP0 microprocessor and the system CPUs to dialogue with the shared devices;

- SHARED MEMORY

the shared memory consists of four 8 Kbytes SRAM packages plus two 64Kx1 SRAM packages utilized like check bit. The 8 Kbytes packages are connected in such a way to constitute one 16 Kwords bank. This memory can be accessed at byte length (right byte and left byte) and at word length with at least four wait cycles.
The SHARED MEMORY bank is mapped in the following address range:

C0.00.00 Hex --> C0.7F.FF Hex;

When an error occurs in the Shared Memory, the check logig asserts the BUS ERROR signal to the processor that is reading the memory.

The diagnostics can set the memory check bit to 0 for testing the error detection circuit. For this feature the PC4 Port C bit of the Parallel Interface Timer has been utilized.
To write always a 0 into the check memory, diagnostics must set the PC4 bit to 0. For The Port C Addressing space, refer to figure 1.1.4.3. (MC68230 Register Model).

- VME BUS INTERRUPTER LOGIC
  this logic acts as an interrupt requester on VMEbus (see para. 1.1.5);

- ATTENTION LOGIC

  this logic permits the system CPUs to interrupt the SP (see para. 1.1.6);

- PROCESSOR NUMBER AND BOARD TYPE DETECTION

  this logic allows to recognize in which slot of the VME bus the board was inserted to configure the lines.
  It is also possible to know which kind of communicatio board was inserted (SP0 or LP0). (see para. 1.1.7).

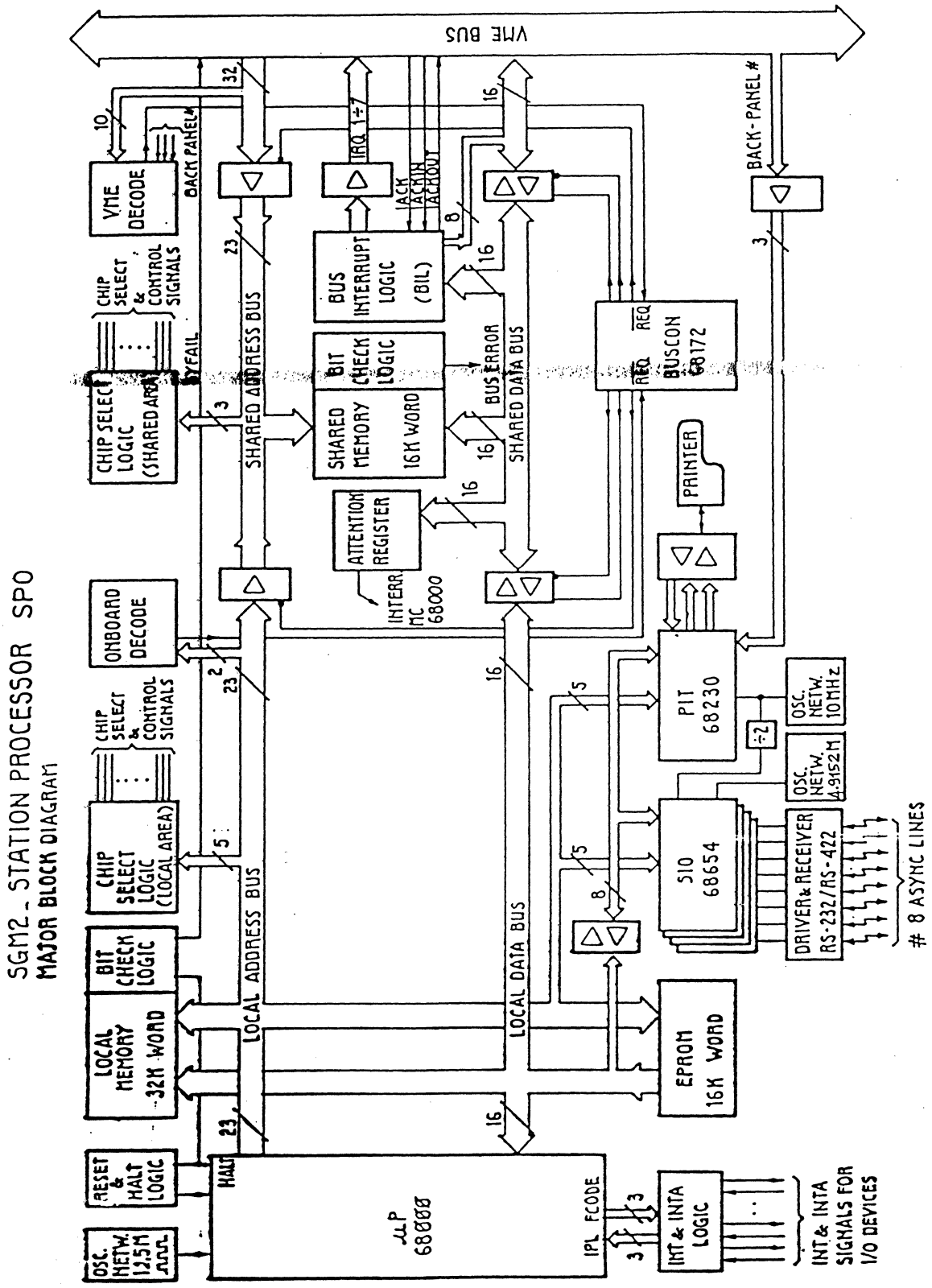


Fig. 1.1 - SP0 Major Block Diagram

1.1.1 I/O MICROPROCESSOR

The SP0 uses the Motorola "MC68000 16-BIT MICROPROCESSOR" with a 12.5 Mhz frequency clock.

The main characteristics of this microprocessor are the following:

- 64 pins Microprocessor;
- 16 bits Data Bus;
- 24 bits Address Bus;
- 16 Megabytes which are directly addressable;
- I/O mapped in memory;
- 17 internal 32 bits registers;
- 32 bits Program Counter;
- 16 bits Status Register;
- 56 different types of instructions;
- operation at BITs, DIGITs, BYTEs (8 bytes) WORDs (16 bits) LONG WORDs (32 bits) length;
- two priviliged statuses: SUPERVISOR status and USER status.

For further details refer to "16 BIT MICROPROCESSOR DATA MANUAL" Revision June 1983-B012B of MOTOROLA INC..

1.1.2 INT & INTA LOGIC

The INT & INTA LOGIC carries out the following functions:

- it receives the interrupt signals;
- it handles the interrupt priorities;
- it notifies to the MC68000 the interrupts with the highest priority by coding the three lines, IPL0, IPL1, IPL2 (Interrupt Control);
- it acnowledges and handles the Interrupt Acknowledge cycle using FC0, FC1, FC2 lines (Processor Status) and the A1, A2, A3 address signals.

The microprocessor of the SP board handles the following interrupts:

Level	Interrupt			
				- with the highest priority
7 (NMI)	NOT USED			
			/	(Special RX Cond.
			!	(RX Data Request
			! LINE#0	< TX Data Request
			!	(STS CHG Request
		SIO0	!	
			!	(Special RX Cond.
			! LINE#1	< RX Data Request
			!	(TX Data Request
			!	(STS CHG Request
6	SIOs		<	
			!	(Special RX Cond.
			!	(RX Data Request
			! LINE#2	< TX Data Request
			!	(STS CHG Request
		SIO1	!	
			!	(Special RX Cond.
			! LINE#3	< RX Data Request
			!	(TX Data Request
			\	(STS CHG Request
			/	(Special RX Cond.
			!	(RX Data Request
			! LINE#4	< TX Data Request
			!	(STS CHG Request
		SIO2	!	
			!	(Special RX Cond.
			! LINE#5	< RX Data Request
			!	(TX Data Request
			!	(STS CHG Request
5	SIOs		<	
			!	(Special RX Cond.
			!	(RX Data Request
			! LINE#6	< TX Data Request
			!	(STS CHG Request
		SIO3	!	
			!	(Special RX Cond.
			! LINE#7	< RX Data Request
			!	(TX Data Request
			\	(STS CHG Request
4	NOT USED			
3	TIMER			
2	ATTENTION REGISTER			
1	PARALLEL PRINTER			
0	NO INTERRUPTS			
				- with the lowest priority

All the interrupts are vectored

When an interrupt is acknowledged by the MC68000 microprocessor an "INTERRUPT ACKNOWLEDGE" cycle is performed and the INT & INTA LOGIC activates the interrupting device which responds by sending one VECTOR (byte) on the Lower Data Bus.

This vector is then latched and used by the MC68000 to select one of the 256 possible pointers of the Exception Vector Table located in Eprom.

The MC68000 Microprocessor can be set at an "Interrupt Priority Level" so that the interrupts having a lower or equal priority will not serviced.

1.1.3 SERIAL INPUT OUTPUT

The SP0 serial ports allow Local Asynchronous Communications by means of RS-232C (V24/V28) and RS-422A (V11) electrical interface.

The choice between the electrical interfaces RS-232C and RS-422A is made using different cables.

The handling of the serial ports are made by the Serial Input Output (SIO) 68564 which are LSI chips with 48 pins dual in line packages operating at 5 Mhz clock.

Their main characteristics are the following:

- compatible with MC68000;
- two independent full-duplex channels;
- directly addressable registers (all control register are read/write);
- receive data registers are quadruply buffered, transmit registers are doubly buffered;;
- Self-test capability;
- daisy chain priority interrupt logic provides automatic interrupt vectoring without external logic;
- Asynchronous features:
  * 5,6,7 or 8 bits/character
  * 1,1/2 or 2 stop bits
  * even, odd or no parity
  * x1, x16, x32 and x64 clock modes
  * break generation and detection
  * parity, overrun and framing error detection

1.1.3.1 CONFIGURATION

The configuration consists of 8 Serial Ports . These ports or lines are indicated as follows:

```
          /                          / RS-232C (up to 38.400 Bps)
          ! LINE#0 > ASYNCHRONOUS
          !                          \ RS-422A (up to 19.200 Bps)
SIO0      <
          .                          / RS-232C (up to 38.400 Bps)
          ! LINE#1 > ASYNCHRONOUS
          \                          \ RS-422A (up to 76.800 Bps)

          /                          / RS-232C (up to 19.200 Bps)
          ! LINE#2 > ASYNCHRONOUS
          !                          \ RS-422A (up to 19.200 Bps)
SIO1      <
          !                          / RS-232C (up to 19.200 Bps)
          ! LINE#3 > ASYNCHRONOUS
          \                          \ RS-422A (up to 19.200 Bps)

          /                          / RS-232C (up to 19.200 Bps)
          ! LINE#4 > ASYNCHRONOUS
          !                          \ RS-422A (up to 19.200 Bps)
SIO2      <
          !                          / RS-232C (up to 19.200 Bps)
          ! LINE#5 > ASYNCHRONOUS
          \                          \ RS-422A (up to 19.200 Bps)

          /                          / RS-232C (up to 19.200 Bps)
          ! LINE#6 > ASYNCHRONOUS
          !                          \ RS-422A (up to 19.200 Bps)
SIO3      <
          !                          / RS-232C (up to 19.200 Bps)
          ! LINE#7 > ASYNCHRONOUS
          \                          \ RS-422A (up to 19.200 Bps)
```

To the LINE#0 is assigned the role of CONSOLE.

The maximum distance reachable with the RS-232C interface is 15 meters.
According to the DSA-46 that defines the standard for DTE to DCE **direct** connection via balanced voltage digital interface circuits, the **maximum** distance reachable with the RS-422A is:

BIT RATE	DISTANCE
up to 20KBps.	1200 meters
76.8 KBps.	320 meters

The SIOs Address Summary Table is given below (all the SIOs accesses must be performed on the LOWER DATA BUS):

ADDRESS (Hex)	ABBREVIATION	# LINE	REGISTER NAME	ACCESS READ/ WRITE	ACCESS READ ONLY
40.00.01	CMDREG	0	Command Register	X	
40.00.03	MODECTL	0	Mode Control Register	X	
40.00.05	INTCTL	0	Interr. Control Reg.	X	
40.00.07	SYNC 1	0	Sync Word Register 1	X	
40.00.09	SYNC 2	0	Sync Word Register 2	X	
40.00.0B	RCVCTL	0	Receiver Control Reg.	X	
40.00.0D	XMTCTL	0	Transmitter Contr.Reg	X	
40.00.0F	STAT 0	0	Status Register 0		X
40.00.11	STAT 1	0	Status Register 1		X
40.00.13	DATARG	0	Data Register	X	
40.00.15	TCREG	0	Time Constant Reg. .	X	
40.00.17	BRGCTL	0	Baud Rate Gen.Cnt.Reg	X	
40.00.19	VECTRG	0/1	Int.Vect.Reg.(Note 2)	X	
40.00.1B		0	(Note 1) .	X	
40.00.1D		0	(Note 1)	X	
40.00.1F		0	(Note 1)	X	
40.00.21	CMDREG	1	Command Register	X	
40.00.23	MODECTL	1	Mode Control Register	X	
40.00.25	INTCTL	1	Interr. Control Reg.	X	
40.00.27	SYNC 1	1	Sync Word Register 1	X	
40.00.29	SYNC 2	1	Sync Word Register 2	X	
40.00.2B	RCVCTL	1	Receiver Control Reg.	X	
40.00.2D	XMTCTL	1	Transmitter Contr.Reg	X	
40.00.2F	STAT 0	1	Status Register 0		X
40.00.31	STAT 1	1	Status Register 1		X
40.00.33	DATARG	1	Data Register	X	
40.00.35	TCREG	1	Time Constant Reg. .	X	
40.00.37	BRGCTL	1	Baud Rate Gen.Cnt.Reg	X	
40.00.39	VECTRG	0/1	Int.Vect.Reg.(Note 2)	X	
40.00.3B		1	(Note 1) .	X	
40.00.3D		1	(Note 1)	X	
40 00 3F		1	(Note 1)	X	

Notes:
1 - Not used, read as "FFH".
2 - Only one Vector Register, accessible through either channel.

ADDRESS (Hex)	ABBREVIATION	# LINE	REGISTER NAME	ACCESS READ/ WRITE	ACCESS READ ONLY
48.00.01	CMDREG	2	Command Register	X	
48.00.03	MODECTL	2	Mode Control Register	X	
48.00.05	INTCTL	2	Interr. Control Reg.	X	
48.00.07	SYNC 1	2	Sync Word Register 1	X	
48.00.09	SYNC 2	2	Sync Word Register 2	X	
48.00.0B	RCVCTL	2	Receiver Control Reg.	X	
48.00.0D	XMTCTL	2	Transmitter Contr.Reg	X	
48.00.0F	STAT 0	2	Status Register 0		X
48.00.11	STAT 1	2	Status Register 1		X
48.00.13	DATARG	2	Data Register	X	
48.00.15	TCREG	2	Time Constant Reg. .	X	
48.00.17	BRGCTL	2	Baud Rate Gen.Cnt.Reg	X	
48.00.19	VECTRG	2/3	Int.Vect.Reg.(Note 2)	X	
48.00.1B		2	(Note 1) .	X	
48.00.1D		2	(Note 1)	X	
48.00.1F		2	(Note 1)	X	
48.00.21	CMDREG	3	Command Register	X	
48.00.23	MODECTL	3	Mode Control Register	X	
48.00.25	INTCTL	3	Interr. Control Reg.	X	
48.00.27	SYNC 1	3	Sync Word Register 1	X	
48.00.29	SYNC 2	3	Sync Word Register 2	X	
48.00.2B	RCVCTL	3	Receiver Control Reg.	X	
48.00.2D	XMTCTL	3	Transmitter Contr.Reg	X	
48.00.2F	STAT 0	3	Status Register 0		X
48.00.31	STAT 1	3	Status Register 1		X
48.00.33	DATARG	3	Data Register	X	
48.00.35	TCREG	3	Time Constant Reg. .	X	
48.00.37	BRGCTL	3	Baud Rate Gen.Cnt.Reg	X	
48.00.39	VECTRG	2/3	Int.Vect.Reg.(Note 2)	X	
48.00.3B		3	(Note 1) .	X	
48.00.3D		3	(Note 1)	X	
48 00 3F		3	(Note 1)	X	

Notes:
1 - Not used, read as "FFH".
2 - Only one Vector Register, accessible through either channel.

ADDRESS (Hex)	ABBREVIATION	# LINE	REGISTER NAME	ACCESS READ/WRITE	ACCESS READ ONLY
50.00.01	CMDREG	4	Command Register	X	
50.00.03	MODECTL	4	Mode Control Register	X	
50.00.05	INTCTL	4	Interr. Control Reg.	X	
50.00.07	SYNC 1	4	Sync Word Register 1	X	
50.00.09	SYNC 2	4	Sync Word Register 2	X	
50.00.0B	RCVCTL	4	Receiver Control Reg.	X	
50.00.0D	XMTCTL	4	Transmitter Contr.Reg	X	
50.00.0F	STAT 0	4	Status Register 0		X
50.00.11	STAT 1	4	Status Register 1		X
50.00.13	DATARG	4	Data Register	X	
50.00.15	TCREG	4	Time Constant Reg. .	X	
50.00.17	BRGCTL	4	Baud Rate Gen.Cnt.Reg	X	
50.00.19	VECTRG	4/5	Int.Vect.Reg.(Note 2)	X	
50.00.1B		4	(Note 1) .	X	
50.00.1D		4	(Note 1)	X	
50.00.1F		4	(Note 1)	X	
50.00.21	CMDREG	5	Command Register	X	
50.00.23	MODECTL	5	Mode Control Register	X	
50.00.25	INTCTL	5	Interr. Control Reg.	X	
50.00.27	SYNC 1	5	Sync Word Register 1	X	
50.00.29	SYNC 2	5	Sync Word Register 2	X	
50.00.2B	RCVCTL	5	Receiver Control Reg.	X	
50.00.2D	XMTCTL	5	Transmitter Contr.Reg	X	
50.00.2F	STAT 0	5	Status Register 0		X
50.00.31	STAT 1	5	Status Register 1		X
50.00.33	DATARG	5	Data Register	X	
50.00.35	TCREG	5	Time Constant Reg. .	X	
50.00.37	BRGCTL	5	Baud Rate Gen.Cnt.Reg	X	
50.00.39	VECTRG	4/5	Int.Vect.Reg.(Note 2)	X	
50.00.3B		5	(Note 1) .	X	
50.00.3D		5	(Note 1)	X	
50 00 3F		5	(Note 1)	X	

Notes:
1 - Not used, read as "FFH".
2 - Only one Vector Register, accessible through either channel.

ADDRESS (Hex)	ABBREVIATION	# LINE	REGISTER NAME	ACCESS READ/WRITE	ACCESS READ ONLY
58.00.01	CMDREG	6	Command Register	X	
58.00.03	MODECTL	6	Mode Control Register	X	
58.00.05	INTCTL	6	Interr. Control Reg.	X	
58.00.07	SYNC 1	6	Sync Word Register 1	X	
58.00.09	SYNC 2	6	Sync Word Register 2	X	
58.00.0B	RCVCTL	6	Receiver Control Reg.	X	
58.00.0D	XMTCTL	6	Transmitter Contr.Reg	X	
58.00.0F	STAT 0	6	Status Register 0		X
58.00.11	STAT 1	6	Status Register 1		X
58.00.13	DATARG	6	Data Register	X	
58.00.15	TCREG	6	Time Constant Reg. .	X	
58.00.17	BRGCTL	6	Baud Rate Gen.Cnt.Reg	X	
58.00.19	VECTRG	6/7	Int.Vect.Reg.(Note 2)	X	
58.00.1B		6	(Note 1) .	X	
58.00.1D		6	(Note 1)	X	
58.00.1F		6	(Note 1)	X	
58.00.21	CMDREG	7	Command Register	X	
58.00.23	MODECTL	7	Mode Control Register	X	
58.00.25	INTCTL	7	Interr. Control Reg.	X	
58.00.27	SYNC 1	7	Sync Word Register 1	X	
58.00.29	SYNC 2	7	Sync Word Register 2	X	
58.00.2B	RCVCTL	7	Receiver Control Reg.	X	
58.00.2D	XMTCTL	7	Transmitter Contr.Reg	X	
58.00.2F	STAT 0	7	Status Register 0		X
58.00.31	STAT 1	7	Status Register 1		X
58.00.33	DATARG	7	Data Register	X	
58.00.35	TCREG	7	Time Constant Reg. .	X	
58.00.37	BRGCTL	7	Baud Rate Gen.Cnt.Reg	X	
58.00.39	VECTRG	6/7	Int.Vect.Reg.(Note 2)	X	
58.00.3B		7	(Note 1) .	X	
58.00.3D		7	(Note 1)	X	
58 00 3F		7	(Note 1)	X	

Notes:
1 - Not used, read as "FFH".
2 - Only one Vector Register, accessible through either channel.

The line Bit Rate must be specified during the SIO initialization sequence by loading an 8 bit Time Constant for every Bit Rate Generator.

The following table supplies the Time Constant values for the most frequently line speeds:

		Time con Decimal (X1 CK)	Time con Hex. (X1 CK)	Divided by (X1 CK)	Time con Decimal (X16 CK)	Time con Hex. (X16 CK)	Divided by (X16 CK)
	76800	16	10	4	1	1	4
BIT	38400	32	20	4	2	2	4
	19200	64	40	4	4	4	4
	9600	128	80	4	8	8	4
RATE	4800	16	10	64	16	10	4
	2400	32	20	64	32	20	4
	1200	64	40	64	4	4	64
	600	128	80	64	8	8	64
	300	255	FF	64	16	10	64

Note: N.A. means Not Applicable.

The Time Constant value given above, have been calculated in the following mode:

$$\text{OUTPUT FREQUENCY (*)} = \frac{\text{INPUT FREQUENCY}}{\text{(divided by selected)x(Time Constant value in decimal)}}$$

(*) Output Frequency of the Bit Rate Generator. Pay attention to the clock rate!

The Input Frequency is 4.9152 and the Clock Rate (x1, x16, x32, X64), is settable in the bit 6 and 7 of the MODE CONTROL REGISTER.

For further details refer to 68564 data sheet.

1.1.3.2 ELECTRICAL INTERFACE

The Serial Ports use as interface connectors 9 path CANNONs with female pins.

The connector of each line includes both the RS-232C and RS-422A interface signals. The table given below shows the pin assignment:

Connec. Pin N.	Interface Circuit Name	Note
1	Cable Shield Ground	* $
2	Transmitted Data	*
3	Received Data	*
4	Receive Data A	$
5	Receive Data B	$
6	Not Used	*
7	Signal Ground	* $
8	Transmission Data A	$
9	Transmission Data B	$

* RS-232C Interface
$ RS-422A Interface

The interface signals are driven and received using the following standard packages:

RS-232C		RS-422A	
+ 1488	(Driver)	+ 3487	(Driver)
+ 75154	(Receiver)	+ 3486	(Receiver)

On the RS-232C and RS-422A interfaces there is a network which forces the SPACE condition on the Receive Data signal of the SIO when the cable is disconnected or the terminal is off.

1.1.4 TIMER and PARALLEL PRINTER INTERFACE

This hardware block consists essentially of Motorola MC68230 PI/T chip, which provides a programmable timer plus a versatile double buffered parallel interfaces.

a) TIMER - The PI/T timer contains a 24-bit synchronous down counter that is loaded from three 8-bit Counter Preload Registers. The 24-bit counter must be clocked by the output of a 5-bit (divided-by-32) prescaler to generate periodic interrupts, a square wave, a single interrupt after a programmed time period, or it can be used for elapsed time misurement.
Also, the end of count can be checked by software without interrupt use.

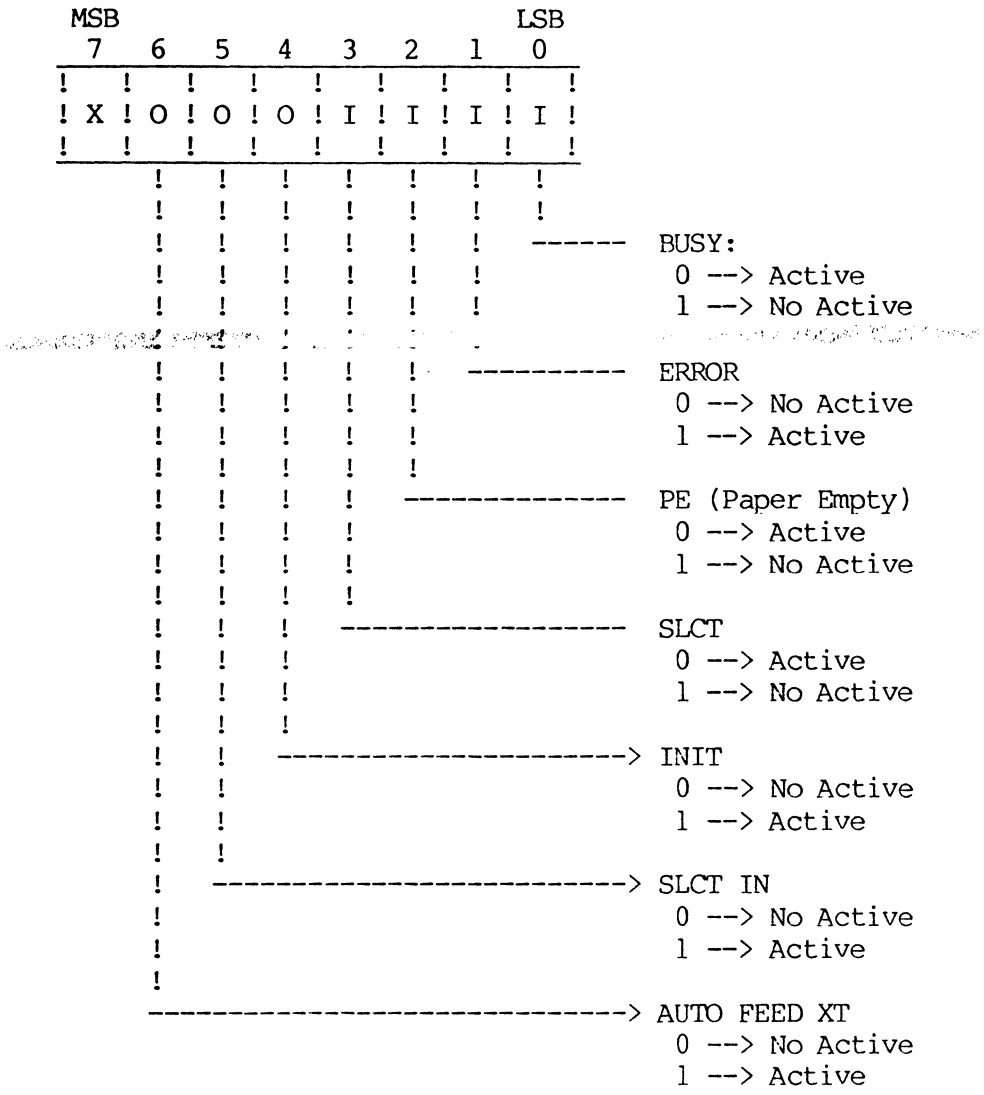
A register model that includes the corresponding Register Selects is shown in Fig. 1.1.4.3. For further information refer to "16-BIT MICROPROCESSOR DATA MANUAL - 1983 MOTOROLA" pag. 4-509 - 4-537.

b) PARALLEL PRINTER INTERFACE - The Parallel Printer Interface allows the connection to printers with electric parallel interface both of the IBM and CENTRONICS types. Fig. 1.1.4.1 shows the major block diagram.

This interface is implemented by mean a programmable parallel interface (MC68230) plus some DRIVERs and RECEIVERs of the Low Power Schottky type. The dialogue with the printer must be performed programming the MC68230 in the following mode:

- Port A must be set with Mode 0 and submode 01;
- Port B must be set with Mode 0 and submode 1X;
- all pins of the Port A must be programmed in output mode to drive the printer data;
- some pins of the Port B must be programmed in output mode to drive printer command and some ones in input mode to receive printer status, as shown below:

PI/T PORT B (bit Input/Output)



- handshake pin H4 must be used as an edge-sensitive status input pin to produce an interrupt to MC68000 when there is the trailing edge of the ACKNLG signal with the meaning of character request. The character should be sent only if the printer is ready (not busy) and no error condition is present.

N.B. - The write of the data register (Port A) involve a hardware generation of the data STROBE signal timing without software management necessity (see Fig. 1.1.4.2 which shows the timings relevant to CENTRONICS and IBM interfaces).

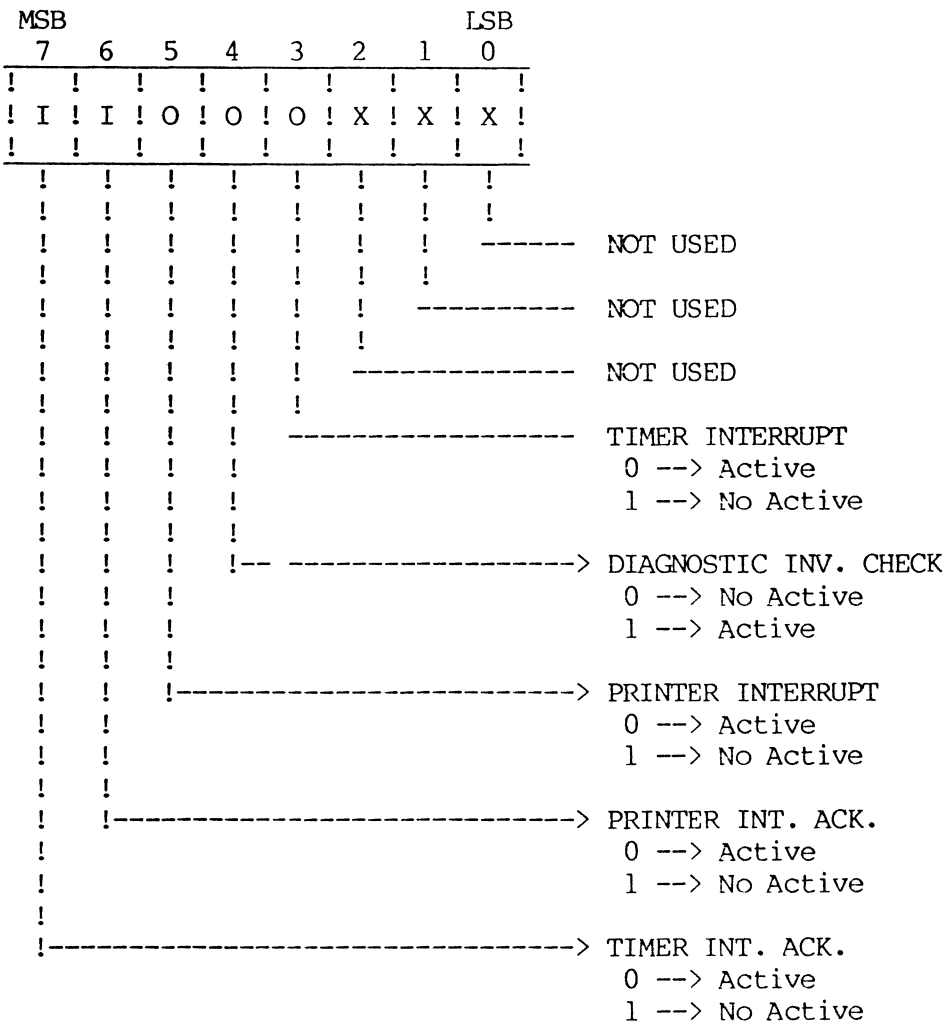
A register model that includes the corresponding Register Selects is shown in Fig. 1.1.4.3. For further information refer to "16-BIT MICROPROCESSOR DATA MANUAL - 1983 MOTOROLA" pag. 4-509 - 4-537.

The I/O signals are made available via a CANNON connector having 37 paths with female pins: the connector pin out and a detailed description of the interface signals are shown in table 1.1.4.1.

The SP board connects the printer unit via a signal cable (with twisted-pair and shield) 5 meters max in lenght with a 37 pin connector at the board unit end, and a 36 pin connector on the printer end.

The bits of the Port C are used in this way:

PI/T PORT C (bit Input/Output)



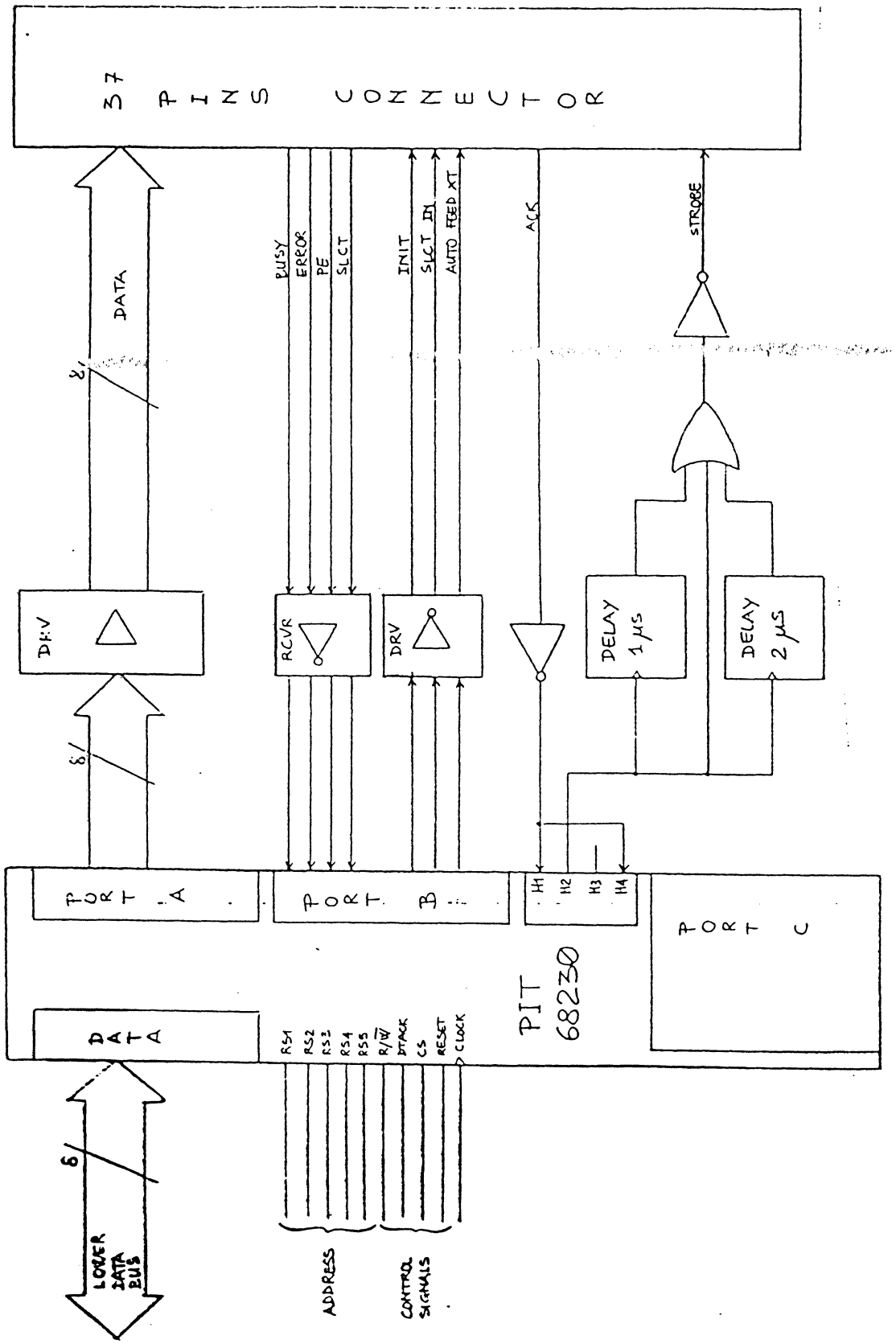
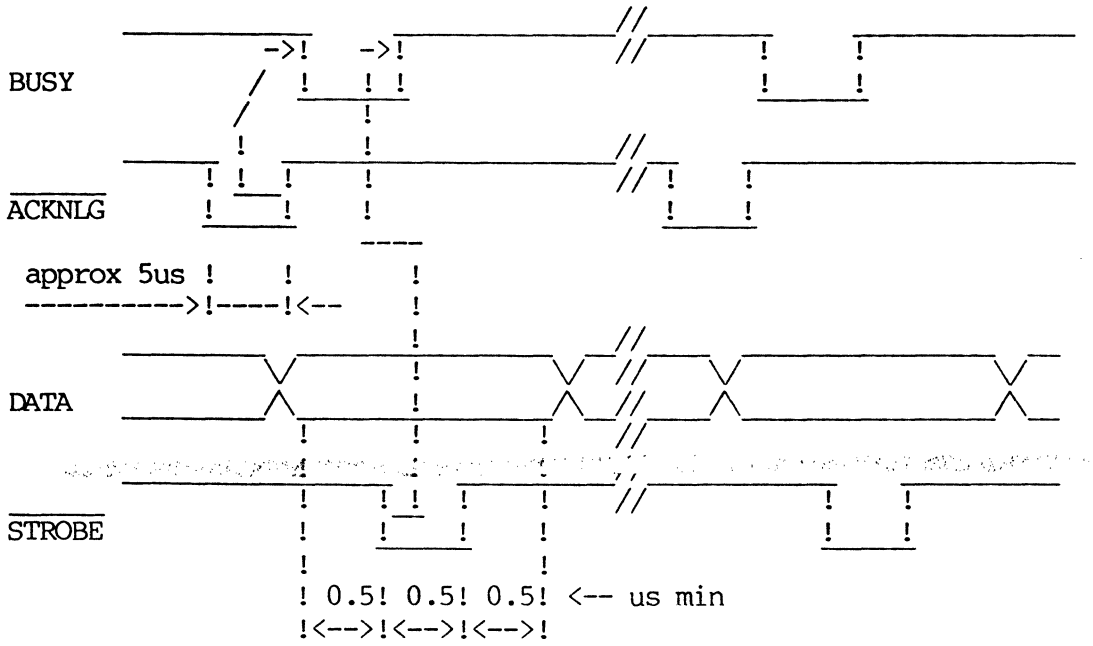
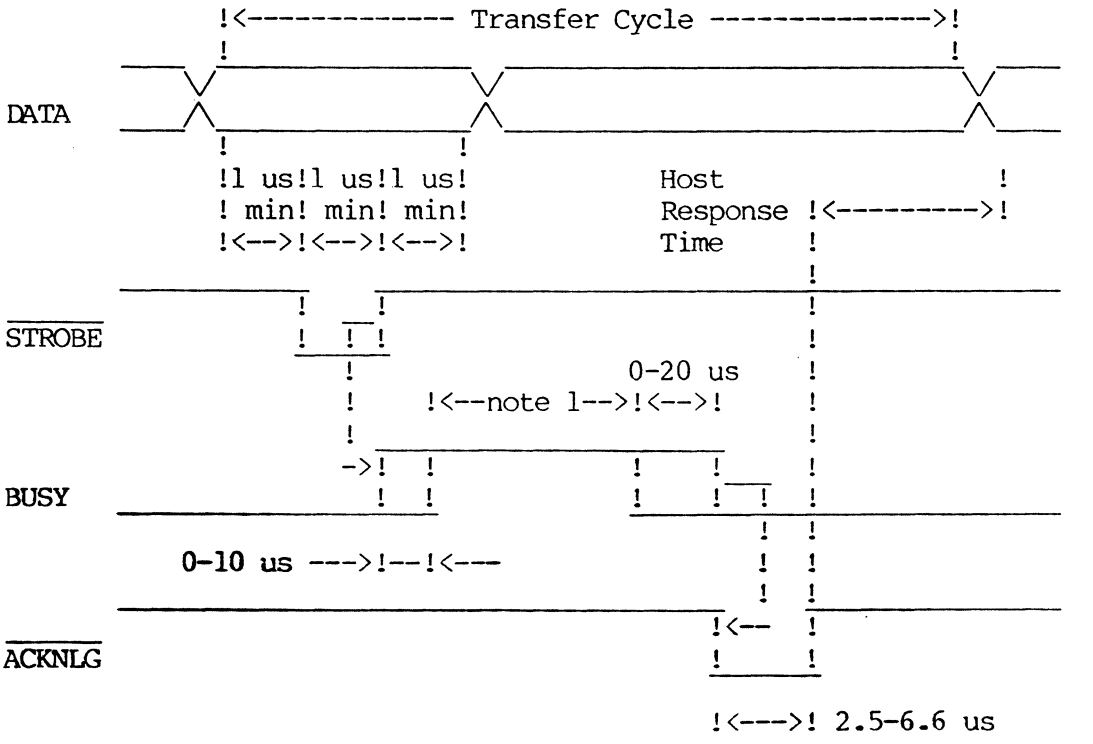


Fig. 1.1.4.1 Parallel Printer Interface Block Diagram



a) IBM Interface Timing



Note 1: Max duration is a function of the required operation.

b) CENTRONICS Interface Timing

Fig. 1.1.4.2 Parallel Printer Interface Timing Diagram

ADDRESS (Hex)	7	6	5	4	3	2	1	0	
B8.00.01	Port Mode Control			H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	Port General Control Register
B8.00.03	*	SVCRQ Select			Interrupt FFS		Port Interrupt Priority Control			Port Service Request Register
B8.00.05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port A Data Direction Register
B8.00.07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port B Data Direction Register
B8.00.09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port C Data Direction Register
B8.00.0B	Interrupt Vector Number						*	*		Port Interrupt Vector Register
B8.00.0D	Port A Submode		H2 Control			H2 Int Enable	H1 SVCRQ Enable	H1 Stat Ctrl		Port A Control Register
B8.00.0F	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRQ Enable	H3 Stat Ctrl		Port B Control Register
B8.00.11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port A Data Register
B8.00.13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port B Data Register
B8.00.15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port A Alternate Register
B8.00.17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port B Alternate Register
B8.00.19	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Port C Data Register
B8.00.1B	H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S		Port Status Register
B8.00.1D	*	*	*	*	*	*	*	*		(null)
B8.00.1F	*	*	*	*	*	*	*	*		(null)
B8.00.21	TOUT/$\overline{\text{TIACK}}$ Control			Z D Ctrl	*	Clock Control			Timer Enable	Timer Control Register
B8.00.23	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Timer Interrupt Vector Register
B8.00.25	*	*	*	*	*	*	*	*		(null)
B8.00.27	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16		Counter Preload Register (High)
B8.00.29	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8		(Mid)
B8.00.2B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		(Low)
B8.00.2D	*	*	*	*	*	*	*	*		(null)
B8.00.2F	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16		Count Register (High)
B8.00.31	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8		(Mid)
B8.00.33	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		(Low)
B8.00.35	*	*	*	*	*	*	*	ZDS		Timer Status Register
B8.00.37	*	*	*	*	*	*	*	*		(null)
B8.00.39	*	*	*	*	*	*	*	*		(null)
B8.00.3B	*	*	*	*	*	*	*	*		(null)
B8.00.3D	*	*	*	*	*	*	*	*		(null)
B8.00.3F	*	*	*	*	*	*	*	*		(null)

(*) - Unused, read as zero.

Fig. 1.1.4.3 MC68230 Register Model

Parallel Printer + Timer

Table 1.1.4.1 Parallel-Printer Connector Pin Assignment and Interface Signal Description

Signal Pin N.	Signal Name	Source	Description
1	$\overline{\text{STROBE}}$	HOST	Data sampling strobe: it clocks data lines into the printer interface logic. The signal level is normally HIGH; write-out of data is performed at the LOW level of this signal. STROBE pulse width requirements are shown in Fig. 1.9.5.2 for IBM and CENTRONICS interfaces.
2	DATA0	HOST	* Least Significant Bit (2^0)
3	DATA1	HOST	
4	DATA2	HOST	These signals represent the character to be printed or the control code to be executed by the printer: normally, these informations are given in ASCII code. Each signal is at HIGH level when data is logical "1" and LOW when logical "0". Data Set-up and Data Hold Times requirements are shown in Fig. 1.9.5.2 for IBM and CENTRONICS interfaces.
5	DATA3	HOST	
6	DATA4	HOST	
7	DATA5	HOST	
8	DATA6	HOST	
9	DATA7	HOST	* Most Significant Bit (2^7)
10	$\overline{\text{ACKNLG}}$	PRINTER	Active LOW pulse. It indicates that the data has been loaded into the buffer or the command has been executed and that the printer is ready to accept other data. In Fig. 1.9.5.2 are shown timing consideration for IBM and CENTRONICS interfaces.

Table 1.1.4.1 Parallel-Printer Connector Pin Assignment and Interface Signal Description (Cont'd.)

Signal Pin N.	Signal Name	Source	Description
11	BUSY	PRINTER	Active HIGH level. It indicates the printer is not ready to accept any data or control code. It is high in the following cases: - During data entry; - During printing operation; - When the printer is in OFF-LINE (or LOCAL) state; and, only for CENTRONICS interface, in these other cases: - As long as the $\overline{\text{INIT}}$ signal is LOW; - When the printer is in STAND-BY status; in this case it will anyway accept XON and DEL codes; and, only for IBM interface, in this other case: - During printer error status. Timing considerations about BUSY signal are shown in Fig. 1.9.5.2 for both interfaces.
12	PE	PRINTER	PE (Paper Empty) is active at HIGH level; it indicates that the printer is out of paper.
13	SLCT	PRINTER	SLCT (Select) signal is active at HIGH level; it indicates that the printer is in the selected state, i.e. is in READY state.

Table 1.1.4.1 Parallel-Printer Connector Pin Assignment and Interface Signal Description (Cont'd.)

Signal Pin N.	Signal Name	Source	Description
14	$\overline{\text{AUTO FEED XT}}$	HOST	This signal is applicable only for IBM interface. When this signal is driven at LOW level, the paper is automatically fed one line after printing.
32	$\overline{\text{INIT}}$	HOST	For the CENTRONICS interfaces this signal is named $\overline{\text{PRIME}}$ and a LOW level on this line causes the output signal $\overline{\text{BUSY}}$ to go high for as long as the $\overline{\text{INIT}}$ signal is low. For the IBM interface, when the level of this signal becomes LOW the printer is reset to its initial state and the printer buffer is cleared. This signal is normally at HIGH level, and its pulse width must be more than 50 us at the receiving terminal.
33	$\overline{\text{ERROR}}$	PRINTER	This line is named $\overline{\text{FAULT}}$ for the CENTRONICS interface. The level of this signal becomes LOW when the printer is in: - PAPER END state; - OFF LINE state; - Error state.
37	$\overline{\text{SLCT IN}}$	HOST	This signal is applicable only for IBM interface. Data entry to the printer is possible only when the level of this signal is LOW.

Table 1.1.4.1 Parallel-Printer Connector Pin Assignment and Interface Signal Description (Cont'd.)

Signal Pin N.	Signal Name	Source	Description
16-19 20-21 22-23 24-25 26-27 28-29 30-31 34	GND	---	Logic GND level.
15-18 35-36	R.F.U.	---	Pins not used.
17	ZVP00	---	Safety Ground.

TO INTERRUPT 68020

1.1.5 VME BUS INTERRUPTER LOGIC (BIL)

The VME Bus Interrupter Logic allows to generate interrupts to the VME bus on any of the seven levels and supplies an 8-bit vector during interrupt acknowledge cycle. Besides it handles the daisy-chain configuration.

The heart of this logic is the SIGNETICS 68154 Interrupt Generator (IGOR), that provides this interface between an interrupting device and the VME Bus.

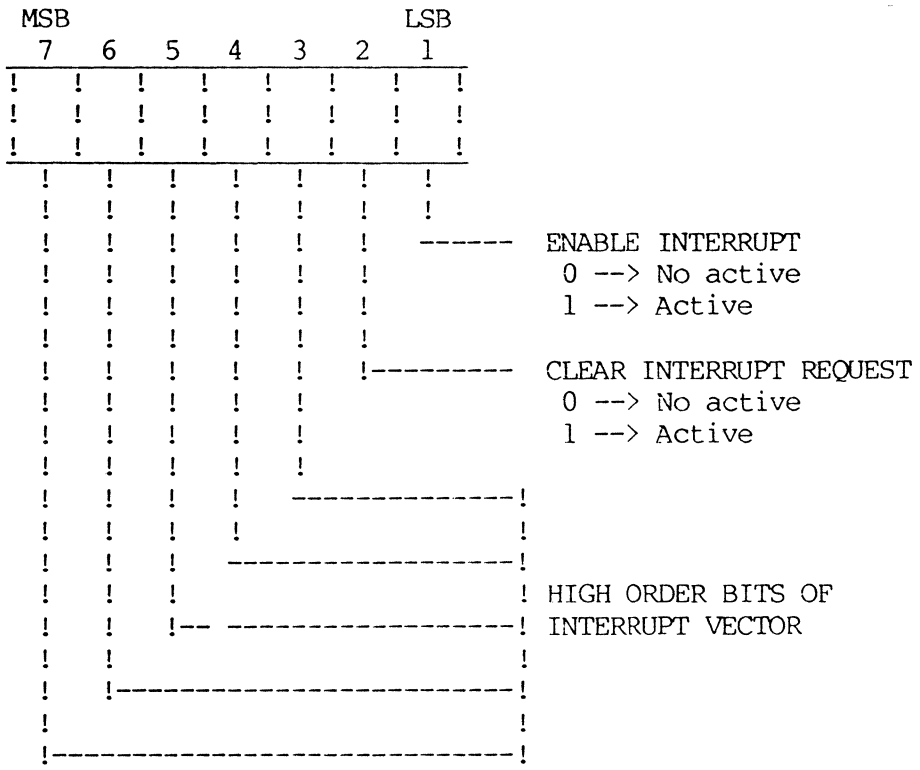
Inside it has two registers:

- the INTERRUPT VECTOR REGISTER R0 (located at Hex E8.00.01)

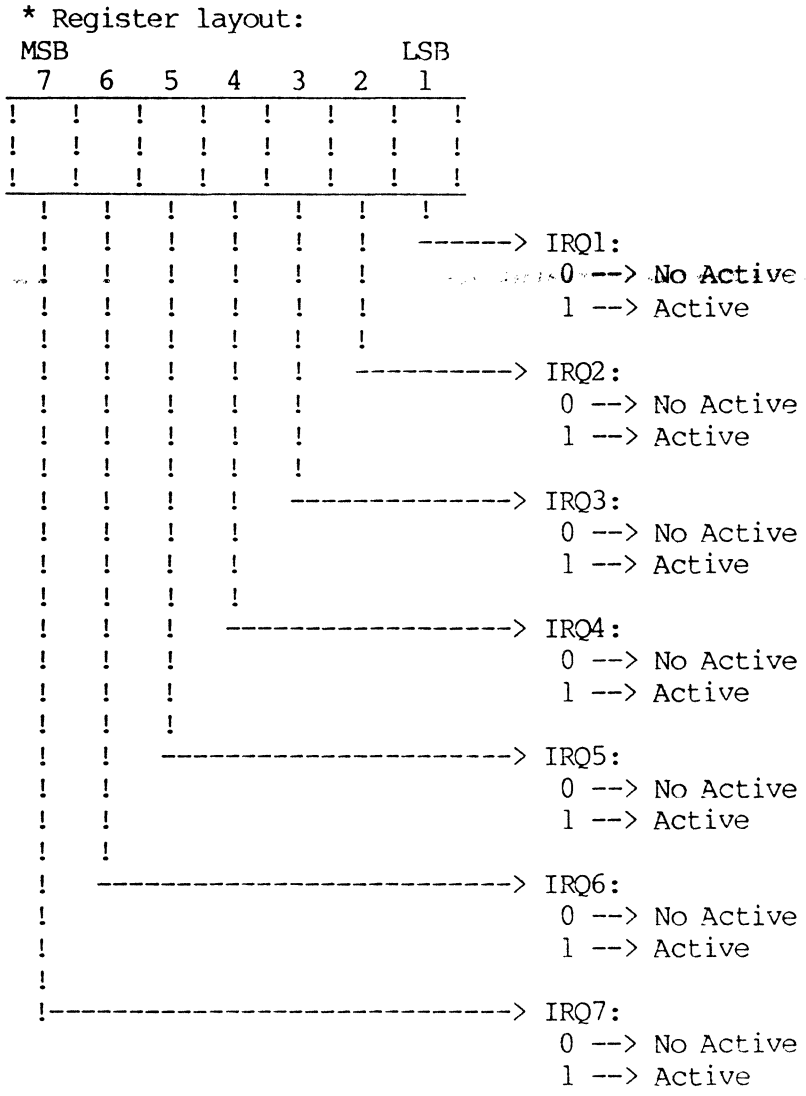
- the INTERRUPT REQUEST REGISTER R1 (located at Hex E8.00.05)

The local processor writes the interrupt vector register to generate an interrupt on any interrupt request level of the VME Bus.

The R0 register has this layout:



The R1 register has this layout:



The 7/6/5/4/3 bits of the interrupt vector register plus the A3 A2 A1 address bits of the VME Bus form the interrupt vector register passed during an interrupt acknowledge cycle.

Writing a 1 to bit 2 of the R0 register are reset all interrupt levels in the interrupt request register. Bit 2 will always be read as 0.

Setting bit 1 of R0 all interrupt levels are enabled.

Any Number (up to seven) of interrupt request can be generated in single access of R1 but they are not stackable on the same level. To generate another interrupt request on a level currently asserted, the user must wait until that level has been acknowledge. During an interrupt acknowledge cycle the corresponding bit of the interrupt level will automatically cleared by the device.

Refers to the Signetics SCB68154 data sheet on Signetics book (January 1986 pages 2-358 2-368).

1.1.6 ATTENTION LOGIC AND SYSTEM FAIL DETECTION

The Attention Logic allows to generate interrupts to the MC68000 from System's CPU and supplies an 8-bit vector, loaded by the interrupting device, during interrupt acknowledge cycle.
Below are shown the registers layouts and access mode of this logic:

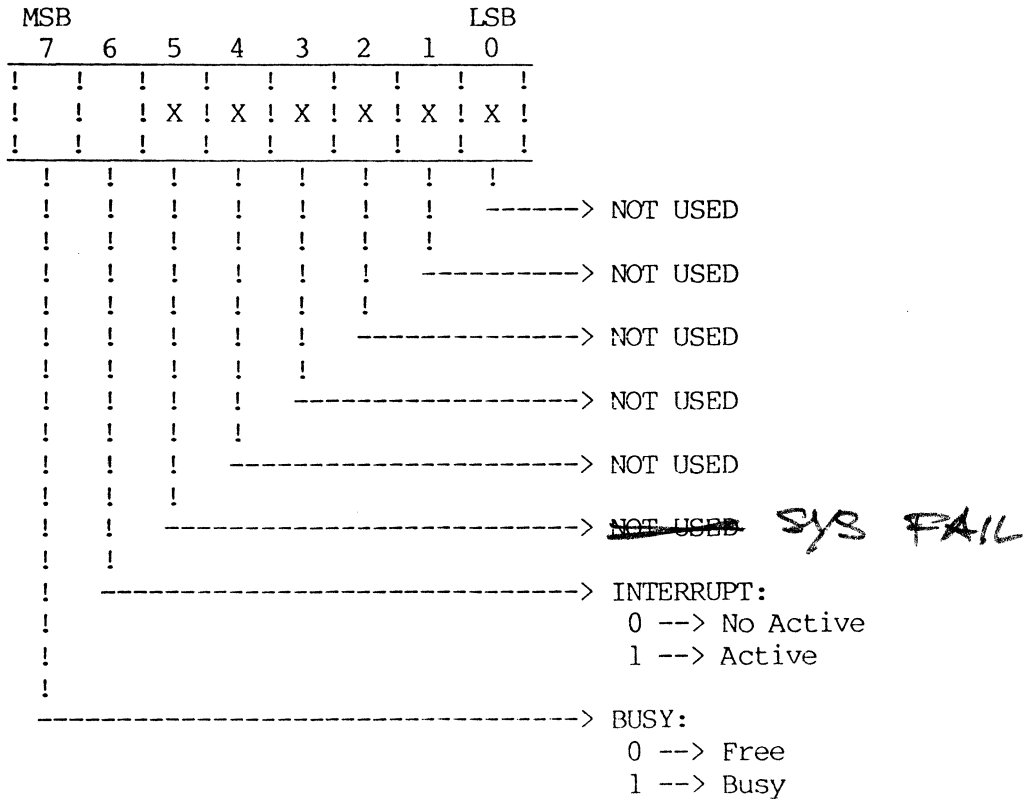
- ATTENTION INTERRUPT REGISTER:

  * Write/Read 8 bit register.

  * Access mode:

    Low ADDRESS: F8.00.00 Hex. WRITE ADDRESS 56.38.00.00
    Read/Write UPPER DATA BYTE.
    TAS.

  * Register layout:



- ATTENTION VECTOR REGISTER

  * Write/Read 8 bit register.

  * Access mode:

    ADDRESS: F8.00.01 Hex;
    Write/Read LOWER DATA BYTE.

  * Register layout:

MSB 7	6	5	4	3	2	1	LSB 0
!	!	!	!	!	!	!	!
!	!	!	!	!	!	!	!
!	!	!	!	!	!	!	!

The informations contained in these registers can be also supplied at word lenght.

It is possible to give only one interrupt at a time to the 68000 SP0 CPU, setting the bit SIX after having checked with the TAS instruction, the 7 bit (BUSY) of the Interrupt Register that no one else interrupt is active.

The reset of the Interrupt Register is made by software control accessing to the interrupt register.

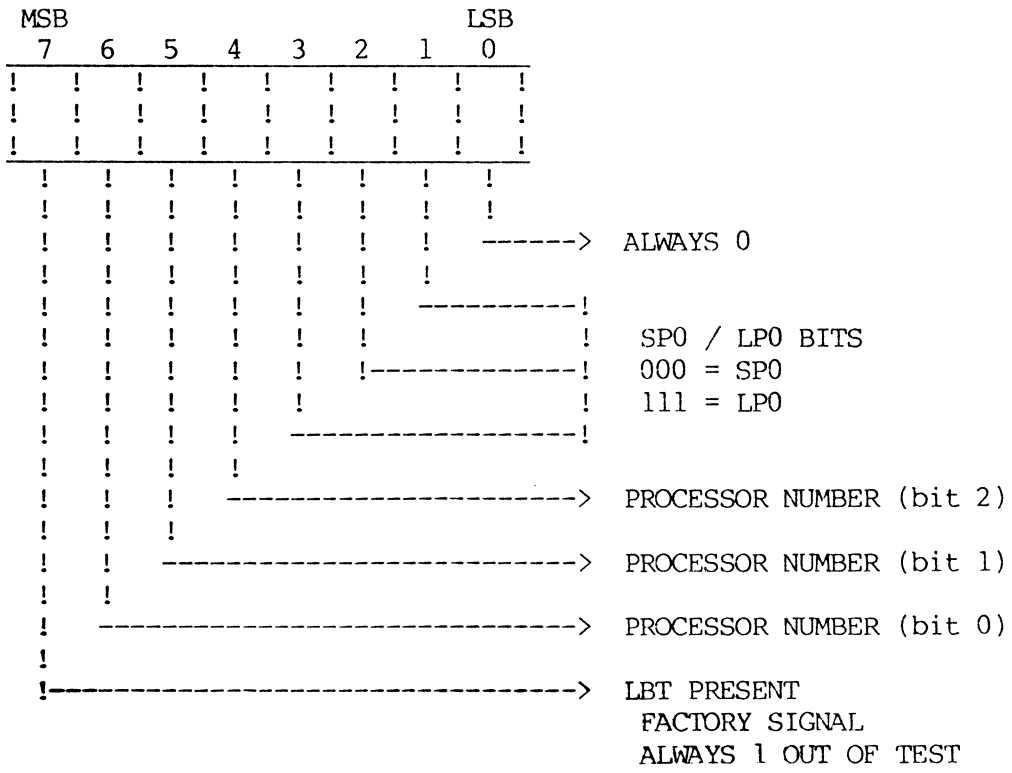
The system fail condition is detectable in the bit 5 of the ATTENTION INTERRUPT REGISTER . The software running in the SGM2 CPU can know which SP0 board in the VME Bus is in the system fail condition.

1.1.7 PROCESSOR NUMBER AND BOARD TYPE DETECTION

There is a register loacated at Hex A8.00.00 where is possible to know some information about the board:

- in which slot of the SGM2 VME Bus the SP0 board was inserted;
- the type of the board inserted (SP0 or LP0);
- detecting enviroment;

The layout of this register is:



The table below describes the VME bus addressing space of each board with the processor number bits equal to:

6	5	4	
0	0	0	PROCESSOR NUMBER 0 - SP0 located at Hex 56.00.00.00 (VME BUS ADDRESSING SPACE)
1	0	0	PROCESSOR NUMBER 1 - SP0 located at Hex 56.40.00.00 (VME BUS ADDRESSING SPACE)
0	1	0	PROCESSOR NUMBER 2 - SP0 located at Hex 56.80.00.00 (VME BUS ADDRESSING SPACE)
1	1	0	PROCESSOR NUMBER 3 - SP0 located at Hex 56.C0.00.00 (VME BUS ADDRESSING SPACE)
0	0	1	PROCESSOR NUMBER 4 - SP0 located at Hex 57.00.00.00 (VME BUS ADDRESSING SPACE)
1	0	1	PROCESSOR NUMBER 5 - SP0 located at Hex 57.40.00.00 (VME BUS ADDRESSING SPACE)
0	1	1	PROCESSOR NUMBER 6 - SP0 located at Hex 57.80.00.00 (VME BUS ADDRESSING SPACE)
1	1	1	PROCESSOR NUMBER 7 - SP0 located at Hex 57.C0.00.00 (VME BUS ADDRESSING SPACE)

2. VME BUS PIN ASSIGNMENT

(*) - Signal low level active

VMEbus J1/P1 PIN ASSIGNMEMENT (ROW A)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
A 1	VDAT00+00	D00
A 2	VDAT01+00	D01
A 3	VDAT02+00	D02
A 4	VDAT03+00	D03
A 5	VDAT04+00	D04
A 6	VDAT05+00	D05
A 7	VDAT06+00	D06
A 8	VDAT07+00	D07
A 9	ZGND	GND
A10	VSYCLK+00	SYSCLK
A11	ZGND	GND
A12	VDSTB1-00	DS1*
A13	VDSTB0-00	DS0*
A14	VWRITE-00	WRITE*
A15	ZGND	GND
A16	VDTACK-00	DTACK*
A17	ZGND	GND
A18	VADSTB-00	AS*
A19	ZGND	GND
A20	VINACK-00	IACK*
A21	VIACKI-00	IACKIN*
A22	VIACKO-00	IACKOUT*
A23	VADMD4+00	AM4
A24	VADD07+00	A07
A25	VADD06+00	A06
A26	VADD05+00	A05
A27	VADD04+00	A04
A28	VADD03+00	A03
A29	VADD02+00	A02
A30	VADD01+00	A01
A31	ZVN12	-12 V
A32	ZVP05	+5 V

CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES
AND STANDARD VME MNEMONIC (Cont.)

VMEbus J1/P1 PIN ASSIGNMEMENT (ROW B)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
B 1		BBSY*
B 2		BCLR*
B 3		ACFAIL*
B 4	SBG0CC-00	BG0IN*
B 5	SBG0CC-00	BG0OUT*
B 6	SBG1CC-00	BG1IN*
B 7	SBG1CC-00	BG1OUT*
B 8	SBG2CC-00	BG2IN*
B 9	SBG2CC-00	BG2OUT*
B10	SBG3CC-00	BG3IN*
B11	SBG3CC-00	BG3OUT*
B12		BR0*
B13		BR1*
B14		BR2*
B15		BR3*
B16	VADMD0+00	AM0
B17	VADMD1+00	AM1
B18		AM2
B19	VADMD3+00	AM3
B20	ZGND	GND
B21		SERCLK (1)
B22		SERDAT (1)
B23	ZGND	GND
B24	VINRQ7-00	IRQ7*
B25	VINRQ6-00	IRQ6*
B26	VINRQ5-00	IRQ5*
B27	VINRQ4-00	IRQ4*
B28	VINRQ3-00	IRQ3*
B29	VINRQ2-00	IRQ2*
B30	VINRQ1-00	IRQ1*
B31	ZVP5SB	+ 5 V STDBY
B32	ZVP05	+ 5 V

CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES AND STANDARD VME MNEMONIC (Cont.)

NOTE:

(1) SERCLK and SERDAT represent provision for a special serial communication bus protocol still being finalized

VMEbus J1/P1 PIN ASSIGNMEMENT (ROW C)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
C 1	VDAT08+00	D08
C 2	VDAT09+00	D09
C 3	VDAT10+00	D10
C 4	VDAT11+00	D11
C 5	VDAT12+00	D12
C 6	VDAT13+00	D13
C 7	VDAT14+00	D14
C 8	VDAT15+00	D15
C 9	ZGND	GND
C10	VSYFAL-00	SYSFAIL*
C11	VBUERR-00	BERR*
C12	VSYRES-00	SYSRESET*
C13	VLWORD-00	LWORD*
C14	VADMD5+00	AM5
C15	VADD23+00	A23
C16	VADD22+00	A22
C17	VADD21+00	A21
C18	VADD20+00	A20
C19	VADD19+00	A19
C20	VADD18+00	A18
C21	VADD17+00	A17
C22	VADD16+00	A16
C23	VADD15+00	A15
C24	VADD14+00	A14
C25	VADD13+00	A13
C26	VADD12+00	A12
C27	VADD11+00	A11
C28	VADD10+00	A10
C29	VADD09+00	A09
C30	VADD08+00	A08
C31	ZVP12	+ 12 V
C32	ZVP05	+ 5 V

CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES
AND STANDARD VME MNEMONIC (Cont.)

VMEbus J2/P2 PIN ASSIGNMEMENT (ROW A)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
A 1		USER I/O
A 2	ZGND	USER I/O (GND)
A 3		USER I/O
A 4		USER I/O
A 5		USER I/O
A 6		USER I/O
A 7		USER I/O (GND)
A 8		USER I/O
A 9		USER I/O
A10		USER I/O
A11		USER I/O
A12		USER I/O
A13	ZVP05	USER I/O (+5 V)
A14		USER I/O
A15		USER I/O
A16		USER I/O
A17		USER I/O
A18	VPNUM0+00	USER I/O (PROCESSOR NUMBER)
A19	VPNUM1+00	USER I/O (PROCESSOR NUMBER)
A20	VPNUM2+00	USER I/O (PROCESSOR NUMBER)
A21		USER I/O
A22	ZGND	USER I/O (GND)
A23		USER I/O
A24	ZGND	USER I/O (GND)
A25		USER I/O
A26		USER I/O
A27		USER I/O
A28		USER I/O
A29	ZVP05	USER I/O (+5 V)
A30	ZVP05	USER I/O (+5 V)
A31	ZGND	USER I/O (GND)
A32	ZGND	USER I/O (GND)

CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES
AND STANDARD VME MNEMONIC (Cont.)

VMEbus J2/P2 PIN ASSIGNMEMENT (ROW B)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
B 1	ZVP05	+ 5 V
B 2	ZGND	GND
B 3	VRESERVD+01	RESERVED
B 4	VADD24+00	A24
B 5	VADD25+00	A25
B 6	VADD26+00	A26
B 7	VADD27+00	A27
B 8	VADD28+00	A28
B 9	VADD29+00	A29
B10	VADD30+00	A30
B11	VADD31+00	A31
B12	ZGND	GND
B13	ZVP05	+5 V
B14		D16
B15		D17
B16		D18
B17		D19
B18		D20
B19		D21
B20		D22
B21		D23
B22	ZGND	GND
B23		D24
B24		D25
B25		D26
B26		D27
B27		D28
B28		D29
B29		D30
B30		D31
B31	ZGND	GND
B32	ZVP05	+5 V

CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES
AND STANDARD VME MNEMONIC (Cont.)

VMEbus J2/P2 PIN ASSIGNMEMENT (ROW C)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
C 1		USER I/O
C 2	ZGND	USER I/O (GND)
C 3		USER I/O
C 4		USER I/O
C 5		USER I/O
C 6		USER I/O
C 7		USER I/O
C 8		USER I/O
C 9		USER I/O
C10		USER I/O
C11		USER I/O
C12		USER I/O
C13	ZVP05	USER I/O (+5 V)
C14		USER I/O
C15		USER I/O
C16		USER I/O
C17		USER I/O
C18		USER I/O
C19		USER I/O
C20		USER I/O
C21		USER I/O
C22	ZGND	USER I/O (GND)
C23		USER I/O
C24	ZGND	USER I/O (GND)
C25		USER I/O
C26		USER I/O
C27		USER I/O
C28		USER I/O
C29	ZVP05	USER I/O (+5 V)
C30	ZVP05	USER I/O (+5 V)
C31	ZGND	USER I/O (GND)
C32	ZGND	USER I/O (GND)

CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES
AND STANDARD VME MNEMONIC

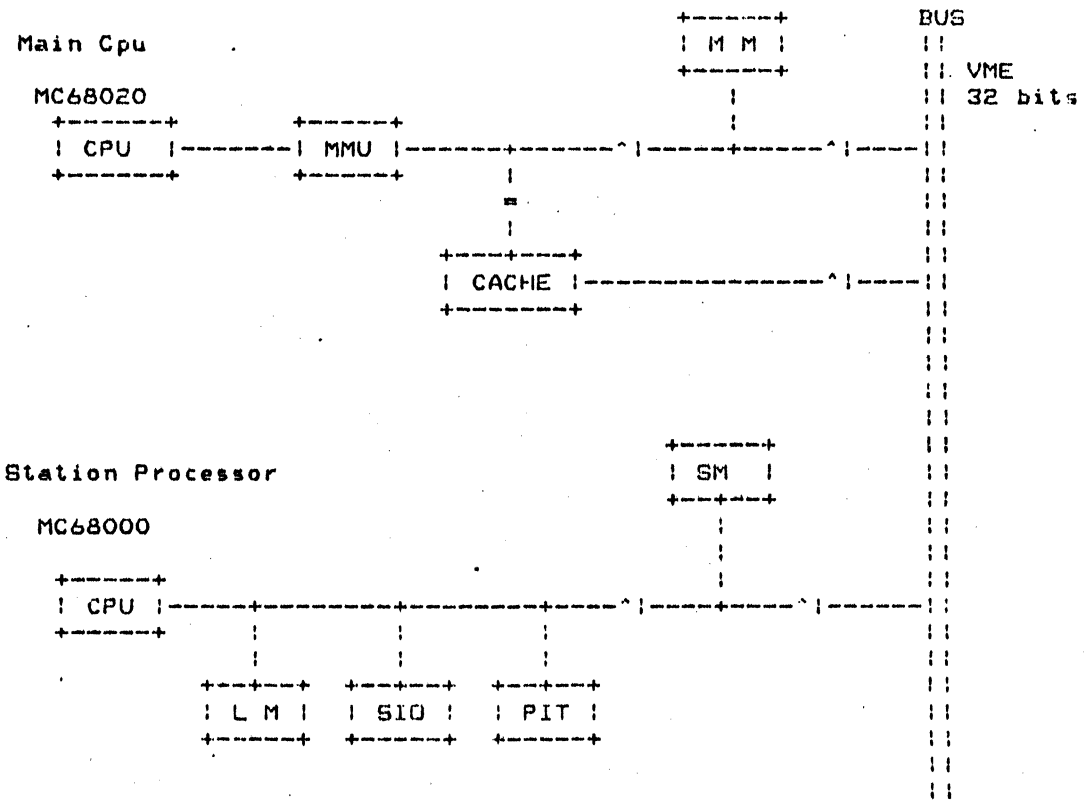
```
*$DOC xxxxxxxx,' SGM2 STATION* PROCESSOR      %UPD 85/11/30 '
*$BEGIN DOC
          EJECT
#$PAR 1,'                  Introduction'
          SPACE    2
          Purpose of this document is to describe the architecture of
        Station Processor(S.P.) on the multiprocessor machine SGM2, and its
        interface with UNIX kernel running on main CPU, when connecting
      TTY terminals.
          SPACE    3
*$PAR 1.1,'              Product requirements'
          SPACE    2
          Purpose of the S.P. board is to connect  up to 8 terminals,
        two types of board are provided.

          SP is only slave mode.
          SPACE
        The first one(board M)has:
          - 8 connectors (9 pins) for local connections.
            .RS-232 async.interface (up to 19200 bps - 15m)
            .RS-422 async.interface (up to 19200 bps - 1200m)
                                    (up to 56400 bps - 443m)
          - 1 connector for printer

          - SW: only TTY like.
          SPACE
        The second one(board F)has:
          - 2 connectors (25 pins) for local or remote connections.
            .RS-232 sync./async. interface (up to 19200 bps).
          - 2 connectors (9 pins) for local connections.
            . as the first one.
          - 2 connectors (15 pins) for local connections.
            . RS-422 sync./async. interface (up to 56400 bps async.)
                                            (up to 100000 bps sync.)
          - 1 connector for printer

          - SW: BSC,SNA or TTY like
                DMA is supported(only 25/15 pins SIO to LM).
          SPACE    3
*$PAR 1.2,'              Reference document'
          SPACE    2
          . ALTAIR EPS Doc Numb. xxxxxxxx
          . SGM2 H/W PDD Doc Numb. xxxxxxx
          . Mostek Serial Input Output MK68564 Technical Manual
          . Parallel Interface/Timer MC68230
          . Motorola MDS-400 Technical Data
          . UNIX System Administrator's Manual(system V)
          . UNIX Device Drivers(PDP-11)
          . SGM-DS UNIX Internals
          . General multiriser system(SGM) PDD
          EJECT
*$PAR 2,'                  Competition analysis'
          EJECT
*$PAR 3,'                  Functional description'
          SPACE    3.
*$PAR 3.1,'                User visible funtionality'
          SPACE    2
          There is a general improvement of the response time at the
        terminals and an improvement of main CPU power due to minor
        processing executed for the terminal handling.
          SPACE    3
*$PAR 3.2,'                User interface'
```

```
          SPACE    2
          No modification are required to any user application to
        support this new architecture.
          The standard Unix system V 2.2 is  implemented(termio(7),
        printer).
          SPACE    3
*$PAR 3.3,'             Security requirements'
          SPACE    3
*$PAR 3.4,'             Availability Reliability requirements'
          SPACE    3
*$PAR 3.5,'             Separately priced software'
          SPACE    3
*$PAR 3.6,'             Functionality not use visible'
          SPACE    2
          A HDS-400 microprocessor H/W S/W development station is
        available for debugging, testing and performance measurements.
          EJECT
*$PAR 4,'               H/W Supported configuration'
          SPACE    2
          The following schema shows the connection of 1 SP, up to 4
        boards can be connected to the same bus on mono CPU configuration
          SPACE    5
```




```
          Main CPU : Motorola 68020 32 bit microprocessor
          S.P. CPU :    "     68000 16  "        "
          Serial input/output (SIO) : Motorola 68564 chip
          Up to 4 SIO can be connected to one board.
          SM  : 32kX16 shared  memory : it is shared between the main
                                        CPU  and SP CPU.
          LM  : 32kX16 local memory
          PIT : Motorola 68230 chip, Printer and Timer.
          EJECT
```

```
*$PAR 5,'               Performance requirements'
        SPACE    3
*$PAR 5.1,'             Performance objectives'
        SPACE    2
        The maximum H/W troughput for a single board is computed as:
      max # of teminals per 19200 bps.
        This means that, for 8 terminals,   153600    bps    may be
      processed simoultaneously. Therefore a character must be
      processed in 65 micro sec.

        Nevertheless S/W throughput is computable as about 60000 bps.
        SPACE    3
*$PAR 5.2,'             Performance measurements specification'
        SPACE    2
        The most significant test to verify the performance objectives
      is to execute file transfer with other computers systems in TTY
      mode. In fact the processing for TTY terminals is greater then
      other type of connections.
        SPACE    3
*$PAR 5.3,'             Measurement facilities'
        SPACE    2
        The measurement will be performed using computers as DPS4
      to execute CP to CP connection and using HDS-400 for timing
    evaluation.
        EJECT
*$PAR 6,'               Transferability'
        SPACE    2
        No modification are required to user application to support
      the new architecture of the SGM2 machine.
        EJECT
*$PAR 7,'               Architecture definition'
        SPACE    2
*$PAR 7.1,'             Interface UNIX(CPU) and S.P.'
        SPACE    2
        The interface used to exchange data and commands between
      the CPU and the S.P. is based on circular buffers with the mecha-
      nism of the producer and consumer pointers. These buffers are in
      the shared memory in the S.P. board.
        EJECT
        The following interface is foreseen:
        SPACE    2
```

UNIX (CPU) USER	DRIVER (CPU)	STATION PROCESSOR
Open	compile TTY struct	.open device .return status to drv
Read	polls buffers/wait	:performs input proc. .sends interrupt to driver when line or crt is present.
Write	fills the buffers	.performs output .sends interrupt to driver for L.W. mark

Close	compile TTY struct	.disable terminal .return status to drv
Ioctl	compile TTY struct	.performs parameter changements .sends interr.when function is executed

EJECT
*$PAR 7.1.1,' Interrupt to Main CPU'
SPACE 2

Only one Vector number associated with the SP is provided from SP to Main CPU for all types of interrupt. Each Vector number is unique in system.

There is only one interrupt circular queue per one Main CPU for every interrupt types.

The interrupt is issued when the interrupts command must be queued and this queue is empty. If it's not empty, the command is queued only.

note: SP may be to have two interrupt for Main CPU:M.CPU 1/2.

The following type of interrupt needs to syncronise with Main CPU:

- QUIT
- INTR
- BREAK
- SWITCH

In this case,SP suspends its processing and waits the acknowledge of the interrupt from the Main CPU. Data after the receipt of QUIT are remained in Raw buffer. Generally these interrupts cause the "SIGNAL" execution to user processing. However the received data from terminal are moved into the Raw buffer until it becomes full.
At this time, if ack is not received yet, these data are lost.

The types of interrupt(from SP to Main CPU) are:
SPACE

(1) device SIO tty

- Input: data present : when the Main interface Input buffer is empty and the input data has been received from a terminal :
  - . Line mode
    received 1st delimiter(CR,LF..)
  - . Character mode
    one character receipt.
    (see:7.2.1.2.1.1 (2))
- Output: Low water mark is reached in the Main

Interface Output buffer or print buffer
after High water mark event.

- Asynchronous event :

The following signals are received from a terminal :

. INTR
. BREAK
. QUIT
. SWITCH
.

- acknowledge :
. OPEN
. CLOSE
. IOCTL

- errore :
.
.

(2) device printer
SPACE

- Output : tbd

- Acknowledge : OPEN
CLOSE

- Error : open error ,printer busy.....

SPACE
An interrupt element(4 bytes) is define below:
SPACE 2

D	T	I	F	E	F U

From SP to Main CPU

SPACE 1
D = device type (4 bits)
1: printer
2: SIO tty
3:
.

T = termimal # (4 bits) if D=2
I = interrupt type (4 bits)
1: input
2: output
3: command (ack)
4: asycronous
5: error
F = flag (4 bits)

if D=2 & I=3 -> 1: OPEN ok
2: CLOSE ok
3: IOCTL ok

if D=2 & I=4 -> 1: QUIT
2: INTR
3: BREAK
4: SWITCH
5: hungup

E = Error code (8 bits)

if D=1
1: open error
2: printer busy

if D=2
1: tbd
2:

FU = future use (8 bits)

Note: If the interrupt queue becomes full ,system will be aborted!!!!!!!!!

EJECT

#$PAR 7.1.2,' Command from Main CPU to SP'

SPACE 2

There is a command circular queue of which an entry is compiled by Main CPU and this queue is polled by SP(in Command Event).

The following commands are forseen:

- INPUT(SIO)
- OPEN (SIO or printer)
- CLOSE(SIO or printer)
- IOCTL(SIO)
- Acknowledge(SIO)

SPACE 2

When the Ioctl command is put in this queue, the ioctl area in shared memory has been already compiled by Main CPU. The ioctl area in shared memory is copied into Local memory after the execution of command.

A command queue element(4 bytes) is defined below:

D	T	C	F	A	FU	FU

D = device type (4 bits)
1: printer
2: SIO tty
.
.

T = terminal # (4 bits)

C = command type (4 bits)
1: INPUT
2: OPEN
3: CLOSE
4: IOCTL
5: Acknowledge

F = flag (4 bits)

if C=3 ->
1: TCGETA
2: TCSETA
3: TCSETAW
4: TCSEAF
5: TCSBRK
6: TCXONC
7: TCFLSH

if C=4 ->
1: QUIT
2: INTR
3: BREAK
4: SWITCH

```
        A = argument      (4 bits)
                           if C=3 & F=5 0: send break

                                        F=6 0: suspend output
                                            1: restart output

                                        F=7 0: flush input Q
                                            1: flush output Q
                                            2: flush input/output Q
        , FU = future use:
*$END DOC
 ************************:   another idea   may be deleted:*************
     There are 2 interrupt queues used by SP and main CPU in
     flip-flop mode. When the CPU works on the first one, the SP
     enqueue interrupt element on the other one until the first
     one is empty.
     When an interrupt must be issued to CPU and one queue is not empty
     the element is queued into the other queue. When the first queue is
     empty an interrupt is issued on the second queue. In this way no
     syncronization mechanism is necessary between SP and main CPU.
     The interrupt queues structures are in the shared memory:
         SPACE    2
                  +----------+       +----------+      +----------+
         Queue 1  | Pointer |------>| Elem. 1 |------| Elem. n |
                  +----------+       +----------+      +----------+
         SPACE    2
                  +----------+       +----------+      +----------+
         Queue 2  | Pointer |------>| Elem. 1 |------| Elem. n |
                  +----------+       +----------+      +----------+
         Space    2
*$BEGIN DOC
         EJECT
*$PAR 7.2,'           Station Processor architecture'
         SPACE    3
         The main function of S.P. are to handle:

           - interface with Main host
           - SIO ( terminale)
           - printer
           - timer
         SPACE    5
*$PAR 7.2.1,'         Main functions'
         SPACE    3
         There are two main  modules to execute the main functions:
         SPACE
           - interrupt handler module
           - Main Loop module

         EJECT
*$PAR 7.2.1.1,'       Interrupt(internal)'
         SPACE    3
*$PAR 7.2.1.1.1,'     Type of interrupt'
         SPACE    3
         There are following type of interrupts:

         (1) between CPU(68000) and SIO

              * from SIO to CPU :
                                  - Rx data FIFO contains a character
                                  (input)
```

- one character is transmitted via Txd and Tx buffer(shift register) is empty (output)
- External status(break,...)
- Special receive condition(parity error).

(2) between CPU and Timer :

* from Timer to CPU :
  - time out from Printer Processing (no receive ack)
  - time out from Output " (timer delay)
  - time out from Input " (par "TIME" or scan rtn)

(3) between CPU and Printer:

* from Printer to CPU :
  - receipt of ACK

(4) between Main CPU and SP CPU

* from Main CPU to SP CPU(attention)
  - receipt of command

The SP interrupt level is defined below:

```
level 7   N.U          higest priority
      6   SIO 0,1
      5   SIO 2,3
      4   N.U
      3   timer
      2   Attention(Main CPU to SP)
      1   printer
```

EJECT

*$PAR 7.2.1.1.2,' Interrupt handler'

SPACE 3

The interrupt handler only sets the processing request flag in the appropriate flag field depending on the type of interrupt (vector number).

Before returning normal mode ,if any other interrupt pending are on, the corresponding process will be treated continuosly under mask mode.

The following switches are set:

- Return switch ,which indicates that the control is returned back to the first entry of Main loop .
- Reset Pointer switch ,which indicates that the corresponding processing request flag pointer is reset.

The processing request flag fields are :

- Input processing request flags field n bits
- Timer processing request flags field m bits
- Printer processing request flag field 1 bit
- Output processing 2 request flags field n bits

n = number of supported terminals
max 8 at first step
m = number of requested timer ...TBD

(1) Input processing request flags field (IPRF)

It consists of n bits and each bit corresponds to one terminale device. When the interrupt has been received from a terminal ,the corresponding flag is set.

(2) Timer processing request flags field (TPRF)

It consists of m bits and each bit corresponds to the requested functionfrom the output or printer processing. When the requested time is elapsed, the corresponding flag is set.

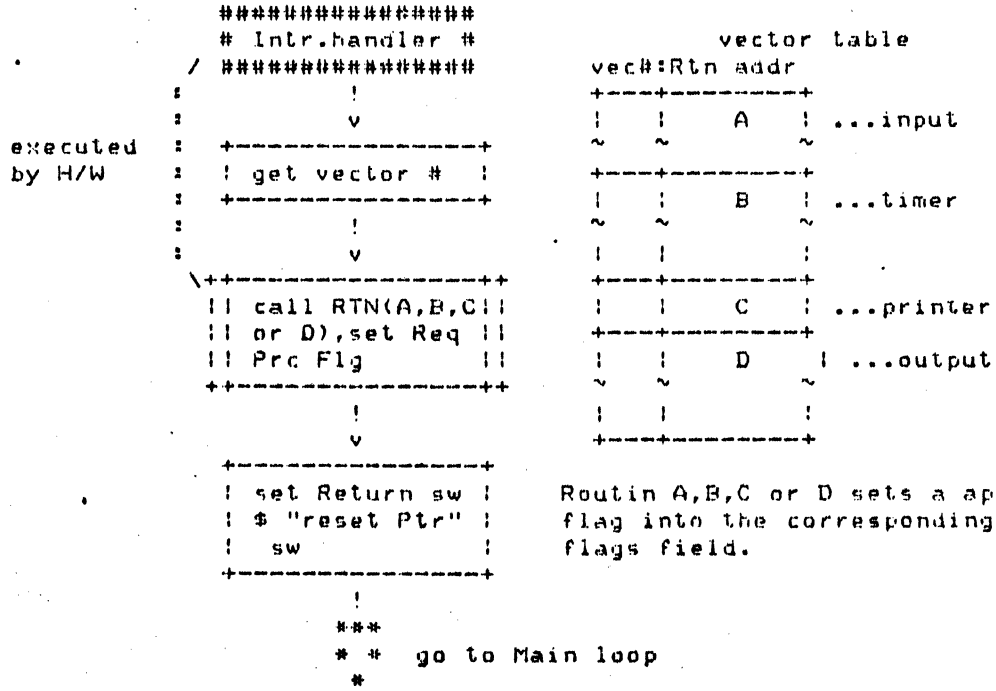
(3) Printer processing request flag field (PPRF)

It consist of only 1 bit . when the ACK sign has been received from the printer device this flag is set.

(4) Output processing 2 request flags field (OPRF)

It consists of n bits and each bit corresponds to a terminal. when Tx register in SIO becomes empty , the corresponding flag is set.

In the case of Input processig, The receipt data will be moved into the Raw buffer immediately.
EJECT
The general flow is shown below:



Note: SIO hardware error will be distiguished in routine A

and the signe is sent to Main CPu via interrupt command queue.

The following errors are forseen:
-
-

EJECT
*$PAR 7.2.1.2,' Main Loop module'
SPACE 3
The Main Loop module searches and performs the processing to execute depending on the given Event below:

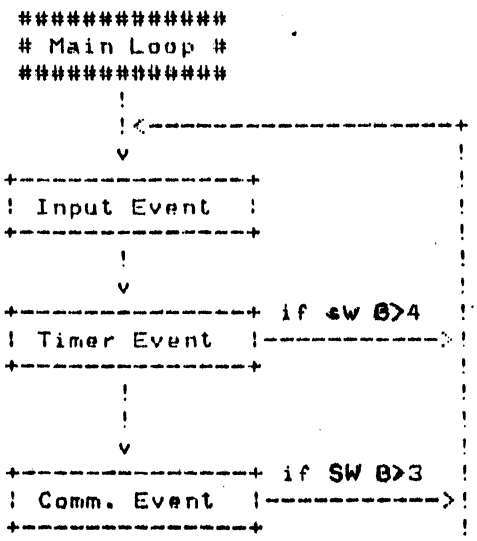
- Input Event 1st priority
- Timer Event 2nd priority
- Command Event 3rd priority
- Printer Event 4th priority
- Output Event 5th priority

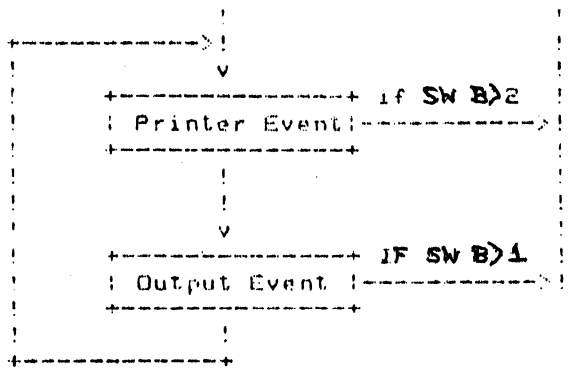
Each Input processing having the request flag ON will be executed and scan is continued until the Input processing request flags have been completely consumed(ie flag all OFF).

When these Input processing request flags become all Off, the next low priority Event(Timer) will be scanned.
When the control has been arrived to the last priority Event (Output Event), the Output Event scanes and executes the output processing having the Output P. request flag or having the data in the associated output buffer.

However each Event may be suspended by any interrupt and will be restarted from the suspended point to complete the processing. The control will be returned to the first Event of the Main Loop to continue from the higher priority Event.

The scan algorithm is round robin way:
The scan will be always restarted from the next processing request position which was finished on the previous scan.
EJECT
The general flow is shown below:
SPACE 2





SW B is set by interrupt handler having
the following value:

- input 5
- timer 4
- attention 3
- printer 2
- output . 1

EJECT

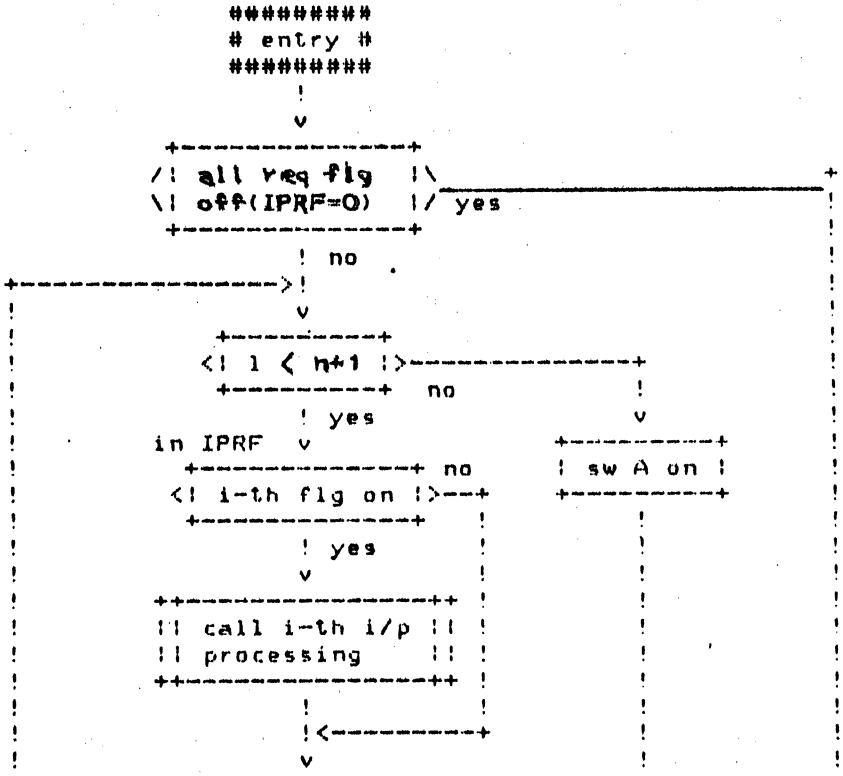
*$PAR 7.2.1.2.1,' Input Event'

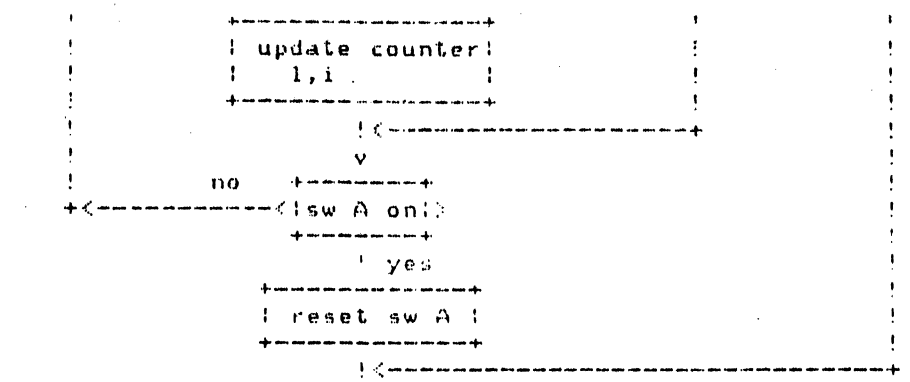
SPACE 2

Each Input processing associated with the terminals will be called when the Input P. Request flag(IPRF),which is set by interrupt handler, is On.

The general flow is shown below:

EJECT

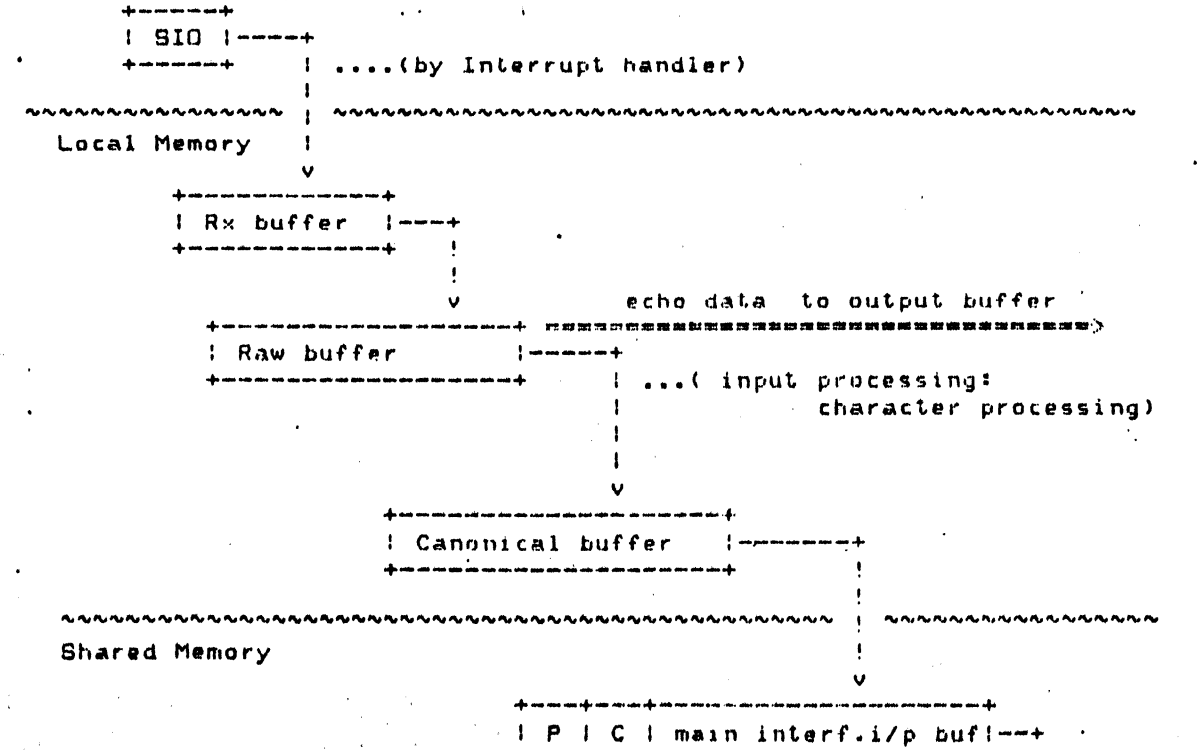


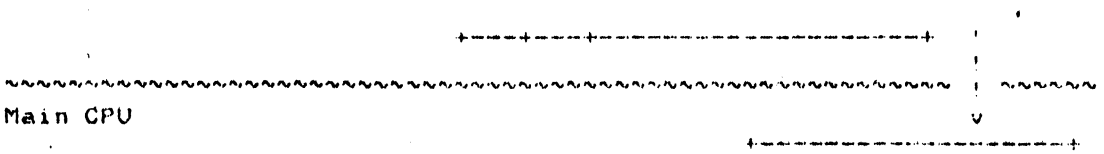


sw A: reset IPRF pointer switch
l: dummy loop pointer =< n
i: corresponding terminal number
IPRF: Input Processing Request Flg.
n: number of supported terminales

EJECT
*$PAR 7.2.1.2.1.1,'. Input processing'
SPACE 2
Purpose of this processing is to execute the character processing according to the user specification,taking characters from Raw buffer and putting them into Input buffer.

The data flow is shown below:
SPACE 2





SPACE 3
The input processing is executed based on the following rule:

(1) Line(canonical) mode: ..(ICANON on)

The data is moved from Raw buffer into Input buffer one character at time. The pointer in Input buffer is updated only when the line is completed(ie: the delimiter character has been received).

If a whole line cannot be put completely in the Input buffer the remained data are stored into the Raw buffer until it becomes full.

At this time ,the data in the Raw buffer and the data already moved in the Input buffer are lost.

If Main interface i/p buffer free space is not sufficent, this procedre will be waited until the command (SIO INPUT) is received from Main CPU via command queue.

(2) CHARACTER(RAW) MODE:

The data is moved from Raw into Input buffer one character at time. When the Input buffer becomes full, the data are stored into the Raw buffer until it becomes full. At this point all data in the Raw buffer are lost. The interrupt to Main CPU is issued when the Input buffer is empty and:

- every a cetain number of character(specified in "MIN" par.) has been reached.
- when the time out is elapsed even if received character number is less than the value of par."MIN"(specified in par. "TIME").

When the echo function is specified , the data are moved from Raw buffer into the Output buffer.

When the flush request is issued by Input processing, the following buffers associated the terminal are cleared:

- Output buffer (LM)
- Transmission buffer (LM)
- canonical buffer (LM)
- Raw buffer (LM)
- Received buffer (LM)
- Main interface o/p buffer (SM)

When the stop command(control+q) has been received, the Output processing associated with the terminal is suspended immediately.

However the received data after the STOP command from the line is stored in Raw buffer and in Transmission buffer if echo function is defined. If Raw buffer becomes full, it is flushed.

For the echo data, the echo data space is contained sufficiently in the Transmission buffer(about 80 characters). The Input processing will be restarted when START command(control+s) has been received.

For the receipt of INTR,QUIT,BREAK,SWITCH, the Input/Output processings associated with the terminal are forced to terminate and buffers are flushed .
In the case of SWITCH, only Input buffer is flushed.
In the case of BREAK , the action is done based on the basic terminal input control in "termio"(IGNBRK,BRKINT,IGNPAR:c-iflag).
These suspended processing will be restarted when the Ack is received from Main CPU via command queue.

```
        EJECT
*$PAR 7.2.1.2.2,'     Timer Event'
        SPACE    2


        EJECT
*$PAR 7.2.1.2.3,'     Command Event'
        SPACE    2
```

One entry is fetched from command queue and the following operation is done based on the command type:

- OPEN Initialises device parametrs associated with the device.

- CLOSE Turns off the indicated device terminal and flushes the associated buffers.

- IOCTL the following commands are forseen:
  - TCGETA
  - TCSETA
  - TCSETAW
  - TCSEAF
  - TCSBRK
  - TCXONC
  - TCFLSH

  When one of the above command has been received, the corresponding ioctl states in SM, which is already compiled by Main CPU, are moved into the relative ioctl save area in LM immediately or after end of drawing output, based on the command type.

- Acknowledge

  It's means that Main intf.i/p buffer had been flushed by Main CPU. And so if data is present in Raw buffer, the input processing will be called.

- INPUT

  The corresponding suspended input processing owing the lack of Main intf.i/p buffer space will be restarted.

```
        EJECT
*$PAR 7.2.1.2.4,'     Printer Event'
        SPACE    3
        EJECT
*$PAR 7.2.1.2.5,'     Output Event'
        SPACE    3
```

The Output processing is divided into two parts:

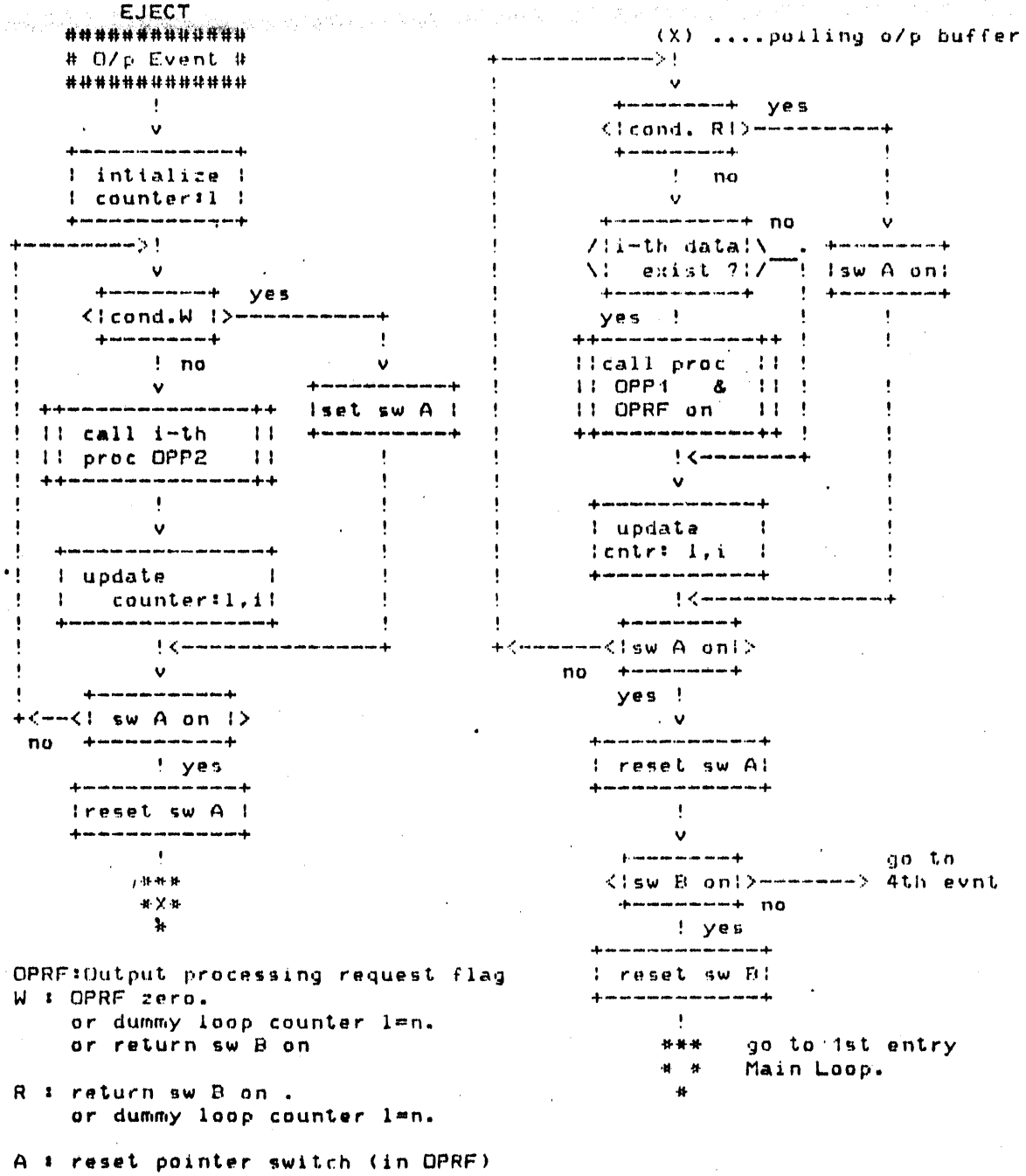
-Output Processing 1: (OPP1) the output buffer having the data is scanned and if it's found, the Opp1 is executed to move the data into the transmitter buffer via o/p character

processing. Then Output processing 2 request flag is set.

-Output processing 2: (OPP2) this processing having the Opp2 request flag will be executed. The data in transmitter buffer will be moved into the transmitter(Tx) register in SIO.

note: OPP2 request flag is set by OPP1 or by interrupt handler after the receipt of an interrupt from SIO indicating the Tx register empty.

The general flow is shown below:



OPRF:Output processing request flag

W : OPRF zero.
or dummy loop counter 1=n.
or return sw B on

R : return sw B on .
or dummy loop counter 1=n.

A : reset pointer switch (in OPRF)

B : return switch (return to 1st entry of Main loop)

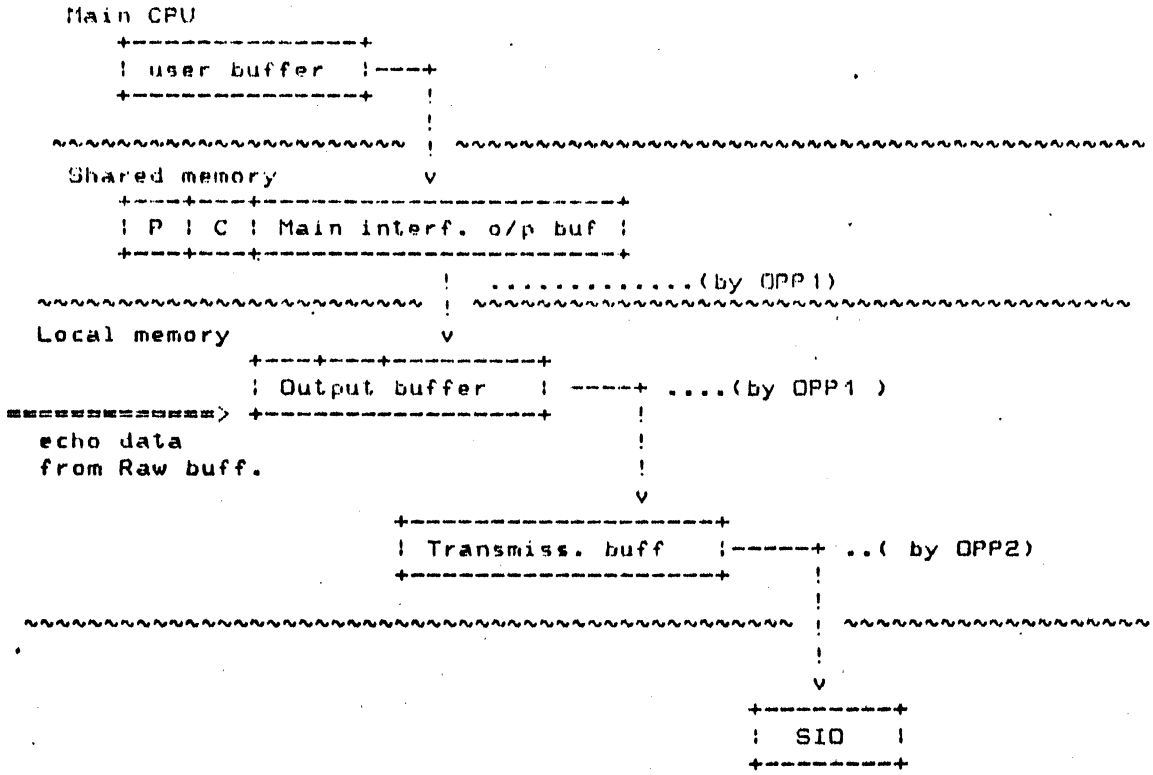
EJECT

*$PAR 7.2.1.2.5.1,' Output processing'

SPACE 2

Purpose of this processing is to execute the data transmission from the Output buffer in the SM, of which data is compiled by M CPU, to the destination terminal via Transmission buffer in the LM, adding delay and handling tabs,CR/NL and echo characters.

The data flow is shown below:



(1) OPP1 This routine copies the character from the Output buffer to the transmission buffer for the device (asociated terminal),adding delays and expanding tabs. until no characters left on the Output buffer. If the number of characters in the Output buffer is below the "Low water mark" ,the sign(via interrupt queue) is sent to Main CPU. And Tx buffer data ready(OPRF) switch to call OPP2 processing is set on.

(2) OPP2 This routine is called whenever a transmit interrupt is generated by SIO, based on the OPRF switch on. A character is placed in the Tranmit register in SIO from Tx buffer, being terminal ready state. If the state of the terminal is TTXON, input processing is started and CSTART(control+q) is sent to the line. Otherwise ,if the state of the terminal is TTXOFF, CSTOP(control+s) is sent to the line. this state is set to TTXON(or TTXOFF) by Main CPU(or OPP1) when the number of characters goes below (or above)the

"High water mark" on the Output buffer.

```
        EJECT
*$PAR 7.2.1.3,'       Data structure '
        SPACE   3
```

The data and working area are divided into two parts:

- shared memory area(SM) (32kx16)
- local memory area(LM) (32kx16)

The informations between Main cpu and SP are in the shared memory area. The other informations (ie: internal use only) and the program executed in SP cpu are im the local memory area.

The following structure have a circular mechanism:

- Main interface input buffer (SM)
- main interface output buffer (SM)
- Interrupt queue (SM)
- Command queue (SM)

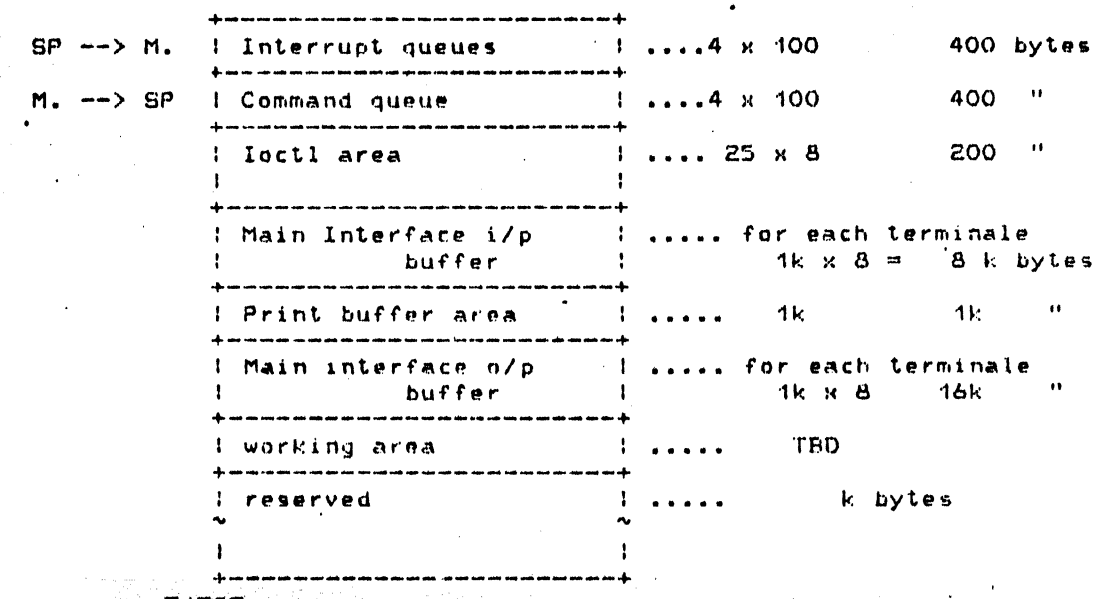
- Output buffer for printer (SM)

```
        EJECT
*$PAR 7.2.1.3.1,'     Shared memory area'
        SPACE   3
```

The shared memory area is used to exchange the data and command between Main cpu and SP via VME bus.

The same field will not be accessed at the same time(Lock byte mechanism).

The shared memory map is shown below:




```
        EJECT
*$PAR 7.2.1.3.2,'     Local memory area'
        SPACE   3
```

Local memory consists of the following area:

- program
- buffer . Raw input buffer \
  . canonical buffer | control block

```
                    . output bufffer            !       mechanism
                    . transmitter buffer        /       (clist)
                    . transmitter buffer for printer (Tx-pr)

          - table   . processing table  :flag pointer status ...etc
                    . SP system table
                    . Ioctl save area
                    . tty structure area
                    . device regisres area

        The informations for internal use will be alocated into the
      local memory because the access time in local memory is faster
      than that in shared memory.
        EJECT
*$PAR 8,'               Packaging,delivery,installation req.'
        SPACE   2
        Not applicable
        SPACE   5
*$PAR 9,'               Maintainability'
        SPACE   2
        TBD
        SPACE   5
*$PAR 10,'              Testability'
        SPACE   2
        Not applicable
        SPACE   5
*$PAR 11,'              Implementation strategy and subsetting'
        SPACE   2
        Not applicable
        SPACE   5
*$PAR 12,'              External documentation requirements'
        SPACE   2
        TBD
        SPACE   5
*$PAR 13,'              Standard compliance'
        SPACE   2
        TBD
        SPACE   5
*$END DOC
        END
```