SGM2

SYSTEM

DOCUMENT ORDER A 78139983
REVISION AA
STATUS FINAL
DATE SEPTEMBER 24, 1986
DOCUMENT TREE

SGM2

PRODUCT DESIGN DESCRIPTION

PREPARED BY

SGM2 DEVELOPMENT TEAM

REVIEWED BY

M.NOBILE

APPROVED BY

E.MACCARIO

Honeywell

Honeywell Information Systems Italia

Honeywell Honeywell Information Systems Italia Loc. Pregnana Milanese Italia	SGM2 P D D	SPEC. NO. A 78139983	SHEET 1/340	REV AA

REVISION RECORD

REV.	AUTHORITY	DATE	SIGN.RE	SHEETS AFFECTED
AA	ZPGC860960	86.10.2.		1st issue

SGM2

PRODUCT DESIGN DESCRIPTION

PREPARED BY: SGM2 DEVELOPMENT TEAM

REVIEWED BY: M. NOBILE

INDEX

	PAGE
SECTION A: CENTRAL PROCESSOR UNIT (CP0) PDD	4
SECTION B: CACHE MEMORY (CH0) PDD	55
SECTION C: SYSTEM CONTROLLER (SC0) PDD	140
SECTION D: STATION PROCESSOR (SP0) PDD	152
SECTION E: LINE PROCESSOR (LP0) PDD	200
SECTION F: DISK CONTROLLER (DCS) PDD	272
SECTION G: MEMORY SUB-SYSTEM PDD	312
SECTION H: CUSTOM MEMORY MANAGEMENT UNIT (CM1) PDD	321

SECTION A

CENTRAL PROCESSOR UNIT (CP0)

PRODUCT DESIGN DESCRIPTION

PREPARED BY: A. DORIGHET

REVIEWED BY: A. LAZZARI

INDEX OF SECTION A

		PAGE
A.1	GENERAL DESCRIPTION	7
A.2	IDP MAJOR BLOCKS DESCRIPTION	8
A.2.1	MPU	10
A.2.2	CM1 INTERFACE	11
A.2.3	FPU	18
A.2.4	LOGICAL AND PHYSICAL SPACES & DECODE LOGIC	19
A.2.4.1	LOGICAL ADDRESSED SPACE	19
A.2.4.2	PHYSICAL ADDRESSED SPACE	20
A.2.4.3	VMEbus ADDRESSING SPACE	25
A.2.4.4	DATA TRANSFER PATHS AND DECODE LOGIC	28
A.2.5	SHARED & VMEbus CONTROL LOGIC (BUSCON)	31
A.2.5.1	GENERAL DESCRIPTION	31
A.2.5.2	ARBITER	33
A.2.6	VMEbus INTERFACE	35
A.2.6.1	ADDRESS MODIFIER CODE	35
A.2.6.2	UTILITY SIGNAL LINES	36
A.2.6.3	VME REQUESTER	36
A.2.6.4	VME INTERRUPT HANDLER	37
A.2.6.5	VME INTERRUPTER	37
A.2.6.6	SWAPPER	37
A.2.7	CACHE (*) INTERFACE	44
A.2.8	MEMORY SUB-SYSTEM INTERFACE	49
A.2.9	EPROM	51
A.2.10	PARALLEL I/O AND TIMER	51
A.2.11	OTHER I/O REGISTERS	53
A.2.12	INTERRUPT HANDLER LOGIC	54

(*) optional

LIST OF ILLUSTRATIONS

		PAGE
FIG. A.1	IDP BLOCK DIAGRAM	9
FIG. A.2	BUSCON ARBITER BLOCK DIAGRAM	34

LIST OF TABLES

		PAGE
TAB. A.1	CM1 CONNECTORS PIN ASSIGNMENT	12
TAB. A.2	MAP OF LOGICAL ADDRESSED SPACE	19
TAB A.3.1	PHYSICAL MAP (DETAIL OF INTERNAL SPACE)	20
TAB. A.3.2	PHYSICAL MAP (DETAIL OF CP02 EXTERNAL MEMORY SPACE)	21
TAB. A.3.3	PHYSYCAL MAP (DETAIL OF CP01 EXTERNAL MEMORY SPACE)	22
TAB. A.3.4	PHYSICAL MAP (RFU EXTERNAL MEMORY SPACE AND DETAIL OF PROPRIETARY AND STANDARD SLAVES SPACE)	23
TAB. A.3.5	PHYSICAL MAP (STANDARD-32 SLAVES SPACE)	24
TAB. A.4.1	VMEbus MAP (EXTENDED ADDRESSING)	25
TAB. A.4.2	VMEbus MAP (STANDARD ADDRESSING)	26
TAB. A.4.3	VMEbus MAP (SHORT ADDRESSING)	27
TAB. A.5	DECODE LOGIC SIGNALS DESCRIPTION	29
TAB. A.6	CP0-BUSCON INTERFACE	31
TAB. A.7	ADDRESS MODIFIER CODES	35
TAB. A.8	CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES AND STANDARD VME MNEMONIC	38
TAB. A.9	Y02 CACHE CONNECTOR PIN ASSIGNMENT	45
TAB. A.10	CP0-CACHE INTERFACE	48
TAB. A.11	CP0-MEMORY SUB-SYSTEM INTERFACE	49
TAB. A.12	MC68230 (PI/T) PIN ASSIGNMENT	52
TAB. A.13	INTERNAL RESOURCES ADDRESS MAP	53
TAB. A.14	INTERRUPT REQUEST LINES HANDLED BY CP0	54

A.1 GENERAL DESCRIPTION

The INTERIOR DECOR PROCESSOR (IDP) is the basic module of the SGM2 multi-IDP architecture (up to two IDPs are allowed in the maximum SGM2 system configuration).

Each IDP contains:

* one CENTRAL PROCESSOR UNIT (CP0 PWA)
* one CUSTOM MEMORY MANAGEMENT UNIT (CM1 PWA), connected as a mezzanine board to CP0 PWA
* one MEMORY SUB-SYSTEM (up to three SEi PWAs; 1, 2 or 4 Mbyte in any choice)
* optionally, one CACHE (CH0 PWA).

The CENTRAL PROCESSOR UNIT (CP0 PWA) is based on 68020 32-bit µP, running at 16.67 MHz, and features:

* interface to a demand-paged VIRTUAL MEMORY MANAGEMENT UNIT, residing on a piggy-back board and supporting multi-tasking, multi-user and multi-processor applications
* floating-point coprocessor (68881)
* interface to a 12 Mbyte max IDP MEMORY SUB-SYSTEM, residing on up to three boards (1M, 2M or 4M in any choice), with DYNAMIC RAM CONTROLLER and Error Detection And Correction functions distributed on each board
* interface to an optional 16 kbyte 8-sector associative CACHE
* full VMEbus master/slave interface
* two 16 kbyte EPROM, configured as a 16 kword bank, containing T&D, boot-strap routines and vector tables
* 24-bit timer with parallel I/O port.

A.2 IDP MAJOR BLOCKS DESCRIPTION

The block diagram of IDP module is shown in fig. A.1; the hashed blocks of this diagram:

* MEMORY SUB-SYSTEM
* CACHE
* CUSTOM MEMORY MANAGEMENT UNIT

reside on separate PWAs, while CP0 provides interfaces to them and contains all other IDP's blocks.

The following functional modules and interfaces reside on CP0:

* MPU 68020, the computing resource of the board, described in par. A.2.1
* CUSTOM MMU INTERFACE, described in par. A.2.2, providing all signals and controls to link a daughter board containing a CUSTOM MEMORY MANAGEMENT UNIT (CM1) based on a SUN-like (1) approach
* FPU 68881, suited for floating-point operations, described in par. A.2.3
* MEMORY SUB-SYSTEM INTERFACE, described in par. A.2.8, providing data, addressess and controls to link up to three SEi PWAs (12 Mbyte max) of DYNAMIC RAM MEMORY
* CACHE INTERFACE, described in par. A.2.7, providing all signals to link an optional CACHE MEMORY
* DECODE LOGIC, described in par. A.2.4.4
* EPROM
* TIMER & parallel port
* I/O registers
* SHARED & VMEbus CONTROL LOGIC (BUS CONTROLLER), described in par. A.2.5, wich provides controls for SHARED and VMEbus
* VMEbus interface, described in par. A.2.6.

CP0 CYCLE TIMING

16.67 MHz (60 ns clock period)

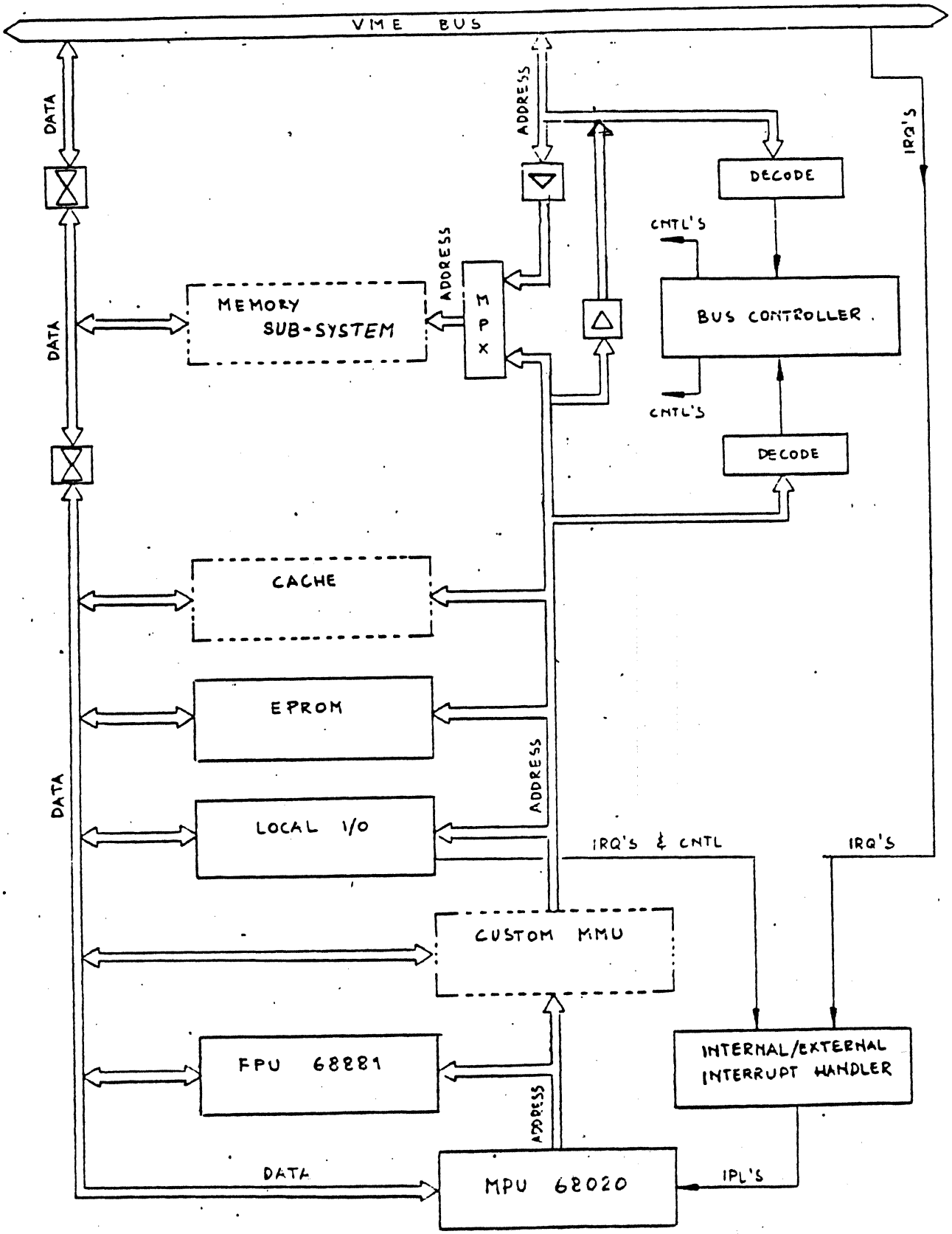
0 wait-state when operating with the cache
2 wait-state when operating with main memory (4-byte read/write)
4-8 wait-state to access controllers'/processors' resources connected to system bus

PROFILE TIMER

It provides a "window" on the system's activity, opened by interrupting MPU with an asynchronous (2) frequency, based on an astable multivibrator (100 Hz), thus allowing statistical evaluations.

(1) SUN: STANFORD UNIVERSITY NETWORKING

(2) with respect to clock-based interrupt, provided as operating system's programmable time reference


VME BUS
DATA
ADDRESS
IRQ'S
DECODE
CNTL'S
ADDRESS
MEMORY SUB-SYSTEM
M P X
BUS CONTROLLER
DATA
CNTL'S
DECODE
CACHE
EPROM
ADDRESS
DATA
LOCAL I/O
IRQ'S & CNTL
IRQ'S
CUSTOM MMU
FPU 68881
INTERNAL/EXTERNAL INTERRUPT HANDLER
ADDRESS
DATA
MPU 68020
IPL'S

A.2.1 MC68020 32-BIT VIRTUAL MEMORY MICROPROCESSOR

Using VLSI technology, the MC68020 is implemented with 32-bit registers and data paths, 32-bit addresses, a rich basic instruction set and versatile addressing modes. The resources available to the MC68020 user consist of the following:

* virtual memory/machine support
* sixteen 32-bit general-purpose data and address registers
* two 32-bit Supervisor stack pointers
* five special purpose control registers
* 4 Gbyte direct addressing range
* 18 addressing modes
* memory mapped I/O
* coprocessor interface
* high performance on-chip instruction cache (64-longword entries)
* operations on seven data types
* complete floating-point support via MC68881 coprocessor.

For further details refer to "MC68020 32-bit Microprocessor User's Manual" - Second Edition (MC68020UM/AD REV 1).

A.2.2 CUSTOM MEMORY MANAGEMENT UNIT (CM1) INTERFACE

The CP0 board provides a CM1 interface to connect a piggy-back board with virtual memory management. That board carries out a translation between logical and physical address, as per descriptor tables residing on the same board. The engineering development will be based on a SUN-like approach, with a page dimension of 4 kbyte.

Translation will be provided only for addresses pertaining to MAPPED MEMORY SPACE (see also par. A.2.4.1). All other addresses are not translated (they are transparent).

To perform this function, two 96-pin connectors (Y05 and Y06) provides data, addresses and control signals described in table A.1.

CM1 MAJOR CHARACTERISTICS

* two level translation - segment and page map
* number of contest directly supported: 64 Supervisor plus 64 User
* moves instruction fully supported
* 256 segments per context
* 16 pages per frame
* 4 kbyte page size
* 32 Mbyte virtual address space
* RAM memory parity checked

Translation of the address must be accomplished in 45 ns.

For more details see also SGM2 CUSTOM MEMORY MANAGEMENT UNIT (CM1) PDD (SECTION H).

CM1 Y05 CONNECTOR PIN ASSIGNMENT (ROW A)

PIN NAME	SIGNAL NAME	DIR (1)	DESCRIPTION
A 1	ZVP05	N.A.	
A 2	ZGND	N.A.	
A 3	CMISMU+00	I	CACHE MISS
A 4	CHPRES-00	I	CACHE PRESENT
A 5	CBEHMA-00	I	BUS ERROR & HALT MASK
A 6	ZGND	N.A.	
A 7	RESET-20	I	MPU RESET
A 8	RWMP+00	I	MPU READ/WRITE
A 9	ZGND	N.A.	
A10	HALT-MU	O	HALT (TO MPU)
A11	ZGND	N.A.	
A12	CPULDS-02	I	RFU
A13	ZGND	N.A.	
A14	PADD12+00	O	PHYSICAL ADDRESS
A15	PADD13+00	O	" "
A16	PADD14+00	O	" "
A17	PADD15+00	O	" "
A18	PADD16+00	O	" "
A19	PADD17+00	O	" "
A20	ZGND	N.A.	
A21	PADD18+00	O	PHYSICAL ADDRESS
A22	PADD19+00	O	" "
A23	PADD20+00	O	" "
A24	PADD21+00	O	" "
A25	PADD22+00	O	" "
A26	PADD23+00	O	" "
A27	PADD24+00	O	" "
A28	PADD25+00	O	" "
A29	ZGND	N.A.	
A30	PAS-MU	O	WIRED TO Y2 CACHE CONNECTOR
A31	ZGND	N.A.	
A32	ZVP05	N.A.	

TABLE A.1: CM1 CONNECTORS PIN ASSIGNMENT

(1) I: INPUT TO CM1
O: OUTPUT FROM CM1

CM1 Y05 CONNECTOR PIN ASSIGNMENT (ROW B)

PIN NAME	SIGNAL NAME	DIR (1)	DESCRIPTION
B 1	ZVP05	N.A.	
B 2	ZGND	N.A.	
B 3	DATA15+MU	I/O	FROM/TO MPU DATA BUS
B 4	DATA14+MU	I/O	" " " "
B 5	DATA13+MU	I/O	" " " "
B 6	DATA12+MU	I/O	" " " "
B 7	DATA11+MU	I/O	" " " "
B 8	DATA10+MU	I/O	" " " "
B 9	DATA09+MU	I/O	" " " "
B10	DATA08+MU	I/O	" " " "
B11	ZGND	N.A.	
B12	FCODE0+MU	I	MPU FUNCTION CODE 0
B13	FCODE1+MU	I	MPU FUNCTION CODE 1
B14	FCODE2+MU	I	MPU FUNCTION CODE 2
B15	ZGND	N.A.	
B16	BERR-MU	O	BUS ERROR TO MPU
B17	ZGND	N.A.	
B18	DSACK1-MU	O	\ DATA & SIZE TRANSFER
B19	DSACK0-MU	O	/ ACKNOWLEDGE (TO MPU)
B20	ZGND	N.A.	
B21	BCLK0+MU	I	MPU CLOCK (COHERENT)
B22	ZGND	N.A.	
B23	PADD08+00	I	ADDRESS FROM MPU
B24	PADD09+00	I	" " "
B25	PADD10+00	I	" " "
B26	PADD11+00	I	" " "
B27	LADD12+00	I	" " "
B28	LADD13+00	I	" " "
B29	LADD14+00	I	" " "
B30	LADD15+00	I	" " "
B31	ZGND	N.A.	
B32	ZVP05	N.A.	

TABLE A.1: CM1 CONNECTORS PIN ASSIGNMENT (CONT.)

(1) I: INPUT TO CM1
O: OUTPUT FROM CM1

CM1 Y05 CONNECTOR PIN ASSIGNMENT (ROW C)

PIN NAME	SIGNAL NAME	DIR (1)	DESCRIPTION
C 1	ZVP05	N.A.	
C 2	ZGND	N.A.	
C 3	DATA07+MU	I/O	FROM/TO MPU DATA BUS
C 4	DATA06+MU	I/O	" " " "
C 5	DATA05+MU	I/O	" " " "
C 6	DATA04+MU	I/O	" " " "
C 7	DATA03+MU	I/O	" " " "
C 8	DATA02+MU	I/O	" " " "
C 9	DATA01+MU	I/O	" " " "
C10	DATA00+MU	I/O	" " " "
C11	ZGND	N.A.	
C12	CPULAS-02	I	RFU
C13	ZGND	N.A.	
C14	LADD16+00	I	ADDRESS FROM MPU
C15	LADD17+00	I	" " "
C16	LADD18+00	I	" " "
C17	LADD19+00	I	" " "
C18	LADD20+00	I	" " "
C19	LADD21+00	I	" " "
C20	LADD22+00	I	" " "
C21	LADD23+00	I	" " "
C22	ZGND	N.A.	
C23	LADD24+00	I	ADDRESS FROM MPU
C24	LADD25+00	I	" " "
C25	LADD26+00	I	" " "
C26	LADD27+00	I	" " "
C27	LADD28+00	I	" " "
C28	LADD29+00	I	" " "
C29	LADD30+00	I	" " "
C30	LADD31+00	I	" " "
C31	ZGND	N.A.	
C32	ZVP05	N.A.	

TABLE A.1: CM1 CONNECTORS PIN ASSIGNMENT (CONT.)

(1) I: INPUT TO CM1
O: OUTPUT FROM CM1

CM1 Y06 CONNECTOR PIN ASSIGNMENT (ROW A)

PIN NAME	SIGNAL NAME	DIR (1)	DESCRIPTION
A 1	ZVP05	N.A.	
A 2	ZGND	N.A.	
A 3	N.C.	N.A.	
A 4	N.C.	N.A.	
A 5	CLKLAS-00	I	CLOCKED LAS FROM MPU
A 6	CLKLAS+00	I	" " " "
A 7	ZGND	N.A.	
A 8	N.C.	N.A.	
A 9	ZGND	N.A.	
A10	PADD25+CH	O	\
A11	PADD24+CH	O	|
A12	PADD23+CH	O	|
A13	PADD22+CH	O	> TO Z2 CACHE CONNECTOR
A14	PADD21+CH	O	> AND VME BUFFERS
A15	PADD20+CH	O	|
A16	PADD19+CH	O	|
A17	PADD18+CH	O	/
A18	ZGND	N.A.	
A19	PADD17+CH	O	\
A20	PADD16+CH	O	|
A21	PADD15+CH	O	> TO Z2 CACHE CONNECTOR
A22	PADD14+CH	O	> AND VME BUFFERS
A23	PADD13+CH	O	|
A24	PADD12+CH	O	/
A25	ZGND	N.A.	
A26	N.C.	N.A.	
A27	N.C.	N.A.	
A28	N.C.	N.A.	
A29	ZGND	N.A.	
A30	N.C.	N.A.	
A31	ZGND	N.A.	
A32	ZVP05	N.A.	

TABLE A.1: CM1 CONNECTORS PIN ASSIGNMENT (CONT.)

(1) I: INPUT TO CM1
O: OUTPUT FROM CM1

CM1 Y06 CONNECTOR PIN ASSIGNMENT (ROW B)

PIN NAME	SIGNAL NAME	DIR (1)	DESCRIPTION
B 1	ZVP05	N.A.	
B 2	ZGND	N.A.	
B 3	DATA31+MU	I/O	FROM/TO MPU DATA BUS
B 4	DATA30+MU	I/O	" " " "
B 5	DATA29+MU	I/O	" " " "
B 6	DATA28+MU	I/O	" " " "
B 7	DATA27+MU	I/O	" " " "
B 8	DATA26+MU	I/O	" " " "
B 9	DATA25+MU	I/O	" " " "
B10	DATA24+MU	I/O	" " " "
B11	ZGND	N.A.	
B12	MEMDSK+00	I	MEMORY DATA TRANSFER ACK
B13	ZGND	N.A.	
B14	PADD31+00	O	PHYSICAL ADDRESS
B15	PADD30+00	O	" "
B16	PADD29+00	O	" "
B17	PADD28+00	O	" "
B18	PADD27+00	O	" "
B19	PADD26+00	O	" "
B20	ZGND	N.A.	
B21	SIZE0+MU	I	MPU TRANSFER SIZE BIT 0
B22	SIZE1+MU	I	MPU TRANSFER SIZE BIT 1
B23	N.C.	N.A.	
B24	N.C.	N.A.	
B25	ZGND	N.A.	
B26	N.C.	N.A.	
B27	ZGND	N.A.	
B28	N.C.	N.A.	
B29	ALLCYC+00	I	ALLOWED SPACE (TAB. A.5)
B30	CPUSP-00	I	CPU SPACE (TAB. A.5)
B31	ZGND	N.A.	
B32	ZVP05	N.A.	

TABLE A.1: CM1 CONNECTORS PIN ASSIGNMENT (CONT.)

(1) I: INPUT TO CM1
O: OUTPUT FROM CM1

CM1 Y06 CONNECTOR PIN ASSIGNMENT (ROW C)

PIN NAME	SIGNAL NAME	DIR (1)	DESCRIPTION
C 1	ZVP05	N.A.	
C 2	ZGND	N.A.	
C 3	DATA23+MU	I/O	FROM/TO MPU DATA BUS
C 4	DATA22+MU	I/O	" " " "
C 5	DATA21+MU	I/O	" " " "
C 6	DATA20+MU	I/O	" " " "
C 7	DATA19+MU	I/O	" " " "
C 8	DATA18+MU	I/O	" " " "
C 9	DATA17+MU	I/O	" " " "
C10	DATA16+MU	I/O	" " " "
C11	ZGND	N.A.	
C12	PADD00+MU	I	ADDRESS FROM MPU
C13	PADD01+MU	I	" " "
C14	PADD02+MU	I	" " "
C15	PADD03+MU	I	" " "
C16	PADD04+MU	I	" " "
C17	PADD05+MU	I	" " "
C18	PADD06+MU	I	" " "
C19	PADD07+MU	I	" " "
C20	ZGND	N.A.	
C21	N.C.	N.A.	
C22	N.C.	N.A.	
C23	N.C.	N.A.	
C24	N.C.	N.A.	
C25	ZGND	N.A.	
C26	RMC-00	I	MPU READ-MODIFY-WRITE CYCLE
C27	ZGND	N.A.	
C28	MMUTP0+00	O	MMU TYPE BIT 0 (2)
C29	MMUTP1+00	O	MMU TYPE BIT 1 (2)
C30	N.C.	N.A.	
C31	ZGND	N.A.	
C32	ZVP05	N.A.	

TABLE A.1: CM1 CONNECTORS PIN ASSIGNMENT (CONT.)

(1) I: INPUT TO CM1
O: OUTPUT FROM CM1

(2) SEE TABLE A.12

A.2.3 MC68881 FLOATING-POINT COPROCESSOR

The MC68881 is a high performance floating-point unit designed to interface with the MC68020 as a coprocessor. It fully supports the MC68020's virtual machine architecture, and is implemented in HCMOS, Motorola's new low power, small geometry process. At 5 V the MC68881 consumes less than 1 W of power.

The MC68881 utilizes the M68000 family coprocessor interface to provide a logical extension of the MPU integer data processing capabilities. It does this by providing a very high performance floating-point arithmetic unit and a set of floating-point data registers that are utilized in a manner that it is analogous to the use of the integer data registers. The MC68881 instruction set is a natural extension of all earlier members of the M68000 Family, and supports all of the addressing modes of the host MPU, which access FPU in a mannner wich is transparent to the programmer. It can execute concurrently with the MC68020's processing to achieve a high throughput.

The MC68881 is internally divided into two processing elements, the Bus Interface Unit (BIU) and Execution Unit (EU). The EU executes all MC68881 instructions while the BIU communicates with the MC68020.

The major features of the MC68881 are:

* eight general purpose floating-point data registers, each supporting a full 80-bit extended precision real data format (a 64-bit mantissa plus a sign bit and a 15-bit signed exponent)
* a 67-bit arithmetic unit to allow very fast calculations, with intermediate precision greater than the extended precision format
* a 67-bit barrel shifter for high-speed shifting operations (for normalizing, etc.)
* forty-six instructions, including 35 arithmetic operations
* full conformation to the IEEE P754 standard, including all requirements and suggestions
* support of functions not defined by the IEEE standard, including a full set of trigonometric and trancendental functions
* seven data types: byte, word and long integers, single, double, and extended precision real numbers, and packed binary coded decimal string real numbers
* twenty-two constants available in the on-chip ROM, including π, e, and powers of 10
* virtual memory/machine operations
* efficient mechanism for procedure calls, context switches, and interrupt handling
* fully concurrent instruction execution with the main processor
* use with any host processor, on an 8-, 16- or 32-bit data bus

Fur further details, refer to "MC68881 Floating-Point Coprocessor User's Manual" (MC68881UM/AD).

A.2.4 LOGICAL AND PHYSICAL SPACES & DECODE LOGIC

A.2.4.1 LOGICAL ADDRESSED SPACE

The map of 4 Gbyte space addressed by 32-bit MPU logical address bus is showned in table A.2.

1 00 00 00 00	EXTERNAL UNMAPPED SPACE	4G - 256M
10 00 00 00		
0F XX XX XX	CP0 INTERNAL RESOURCES	
0F 00 00 00		
0E XX XX XX	RFU	
0E 00 00 00		
0D XX XX XX	CM1 REGISTERS	
0D 00 00 00		
0C XX XX XX	CM1 PAGE STATUS TABLE	
0C 00 00 00		
0B 00 00 00	CM1 PAGE TABLE	INTERNAL
0A 00 00 00		> UNMAPPED >
09 00 00 00	CM1 SEGMENT TABLE	SPACE
08 00 00 00		
07 XX XX XX	RFU	
07 00 00 00		
06 XX XX XX	RFU	
06 00 00 00		
05 XX XX XX	RFU	
05 00 00 00		
04 XX XX XX	RFU	
04 00 00 00		
03 XX XX XX	RFU	
03 00 00 00		
02 XX XX XX	RFU	INTERNAL
02 00 00 00		> MAPPED
01 XX XX XX	RFU	SPACE
01 00 00 00		
00 XX XX XX	MAPPED MEMORY SPACE(*)	16 MByte
00 00 00 00		

UNMAPPED SPACE

TABLE A.2: MAP OF LOGICAL ADDRESSED SPACE

(*) Only for this window the physical to logical relation is:

00 XX XX XX >> (MMU) N0 YY YY YY

where N=0 for direct access to MEMORY SUB-SYSTEM of each CP0
N=1 when CP02 access the MEMORY SUB-SYSTEM of CP01 (via VME)
N=2 when CP01 access the MEMORY SUB-SYSTEM of CP02 (via VME)
N=3 or 4 are reserved for future uses

Translation of logical address into physical address takes place only for logical addresses pertaining to MAPPED SPACE (the lower 64 Mbyte of logical addressed space) and is provided by a CUSTOM MEMORY MANAGEMENT UNIT (CM1); all remaining addresses, pertaining to UNMAPPED SPACE, are transparent for CM1.

A.2.4.2 PHYSICAL ADDRESSED SPACE

The 4 Gbyte space addressed by 32-bit physical address bus is divided as shown in table A.3.1 throught A.3.5, by means of a dedicated DECODE LOGIC.

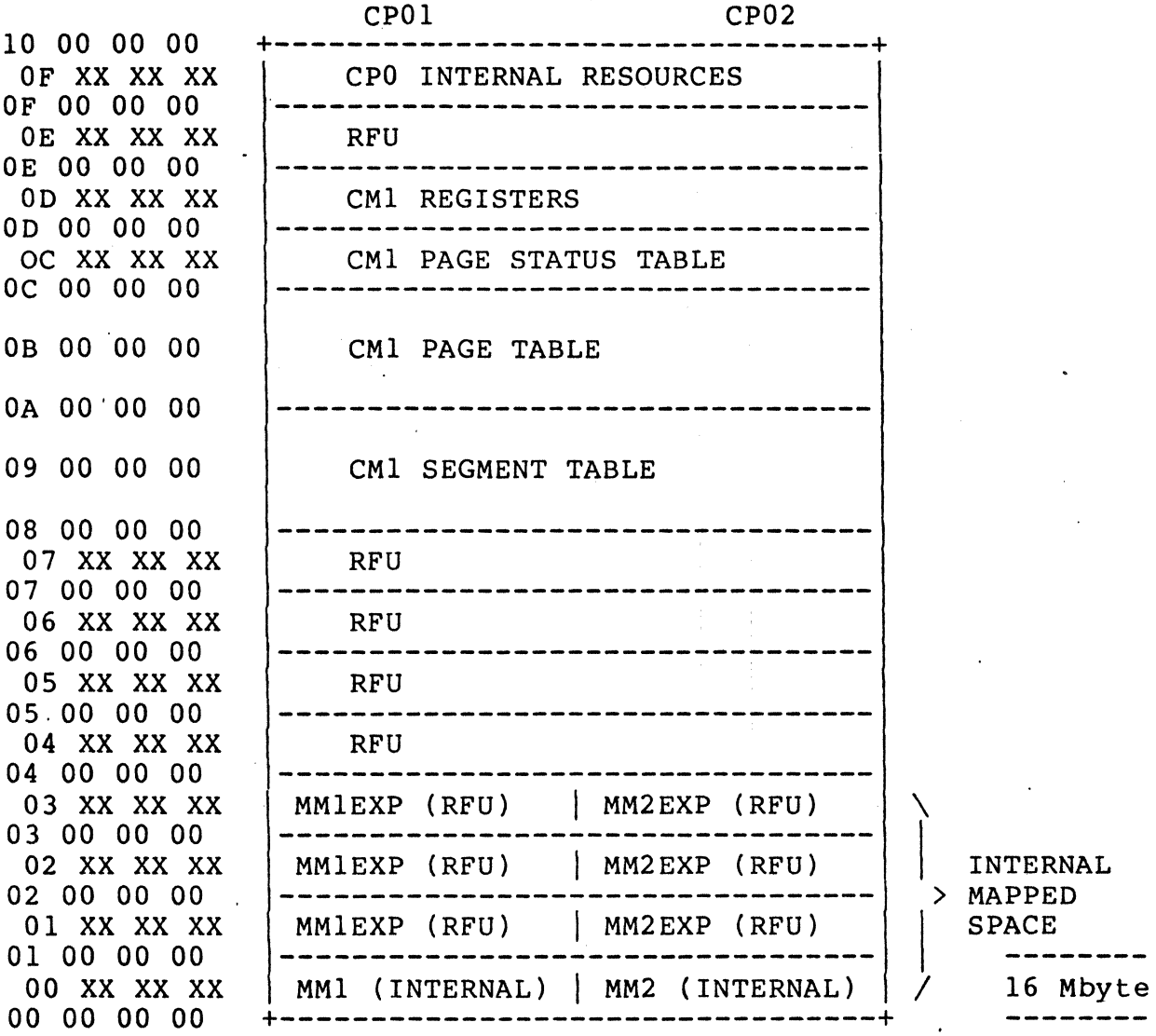


TABLE A.3.1: PHYSICAL MAP (DETAIL OF INTERNAL SPACE)

NOTE: MM1 is the MEMORY SUB-SYSTEM of CP01 (12 Mbyte max)
MM2 is the MEMORY SUB-SYSTEM of CP02 (12 Mbyte max)
MM1EXP are reserved for expansion of MM1
MM2EXP are reserved for expansion of MM2

	CP01	CP02	
20 00 00 00	RESERVED		
1F XX XX XX			
1F 00 00 00	RESERVED		
1E XX XX XX			
1E 00 00 00	RESERVED		
1D XX XX XX			
1D 00 00 00	RESERVED		
1C XX XX XX			
1C 00 00 00	RESERVED		
1B XX XX XX			
1B 00 00 00	RESERVED		
1A XX XX XX			
1A 00 00 00	RESERVED		
19 XX XX XX			
19 00 00 00	RESERVED		
18 XX XX XX			
18 00 00 00	RESERVED		
17 XX XX XX			
17 00 00 00	RESERVED		
16 XX XX XX			
16 00 00 00	RESERVED		
15 XX XX XX			
15 00 00 00	RESERVED		
14 XX XX XX			
14 00 00 00	RESERVED		
13 XX XX XX	RESERVED	MM1 (RFU)	
13 00 00 00	RESERVED	MM1 (RFU)	
12 XX XX XX			
12 00 00 00	RESERVED	MM1 (RFU)	
11 XX XX XX			
11 00 00 00	RESERVED	MM1 (RFU)	
10 XX XX XX	RESERVED	MM1*(EXTERNAL)	16 Mbyte
10 00 00 00			

TABLE A.3.2: PHYSICAL MAP (DETAIL OF CP02 EXTERNAL MEMORY SPACE)

NOTE: MM1* IS THE MEMORY SUB-SYSTEM OF CP01 WICH CAN BE ACCESSED BY CP02

	CP01	CP02	
30 00 00 00	RESERVED		
2F XX XX XX			
2F 00 00 00	RESERVED		
2E XX XX XX			
2E 00 00 00	RESERVED		
2D XX XX XX			
2D 00 00 00	RESERVED		
2C XX XX XX			
2C 00 00 00	RESERVED		
2B XX XX XX			
2B 00 00 00	RESERVED		
2A XX XX XX			
2A 00 00 00	RESERVED		
29 XX XX XX			
29 00 00 00	RESERVED		
28 XX XX XX			
28 00 00 00	RESERVED		
27 XX XX XX			
27 00 00 00	RESERVED		
26 XX XX XX			
26 00 00 00	RESERVED		
25 XX XX XX			
25 00 00 00	RESERVED		
24 XX XX XX			
24 00 00 00	RESERVED		
23 XX XX XX	MM2 (RFU)	RESERVED	
23 00 00 00			
22 XX XX XX	MM2 (RFU)	RESERVED	
22 00 00 00			
21 XX XX XX	MM2 (RFU)	RESERVED	
21 00 00 00			
20 XX XX XX	MM2*(EXTERNAL)	RESERVED	16 Mbyte
20 00 00 00			

TABLE A.3.3: PHYSICAL MAP (DETAIL OF CP01 EXTERNAL MEMORY SPACE)

NOTE: MM2* IS THE MEMORY SUB-SYSTEM OF CP02 WICH CAN BE ACCESSED BY CP01

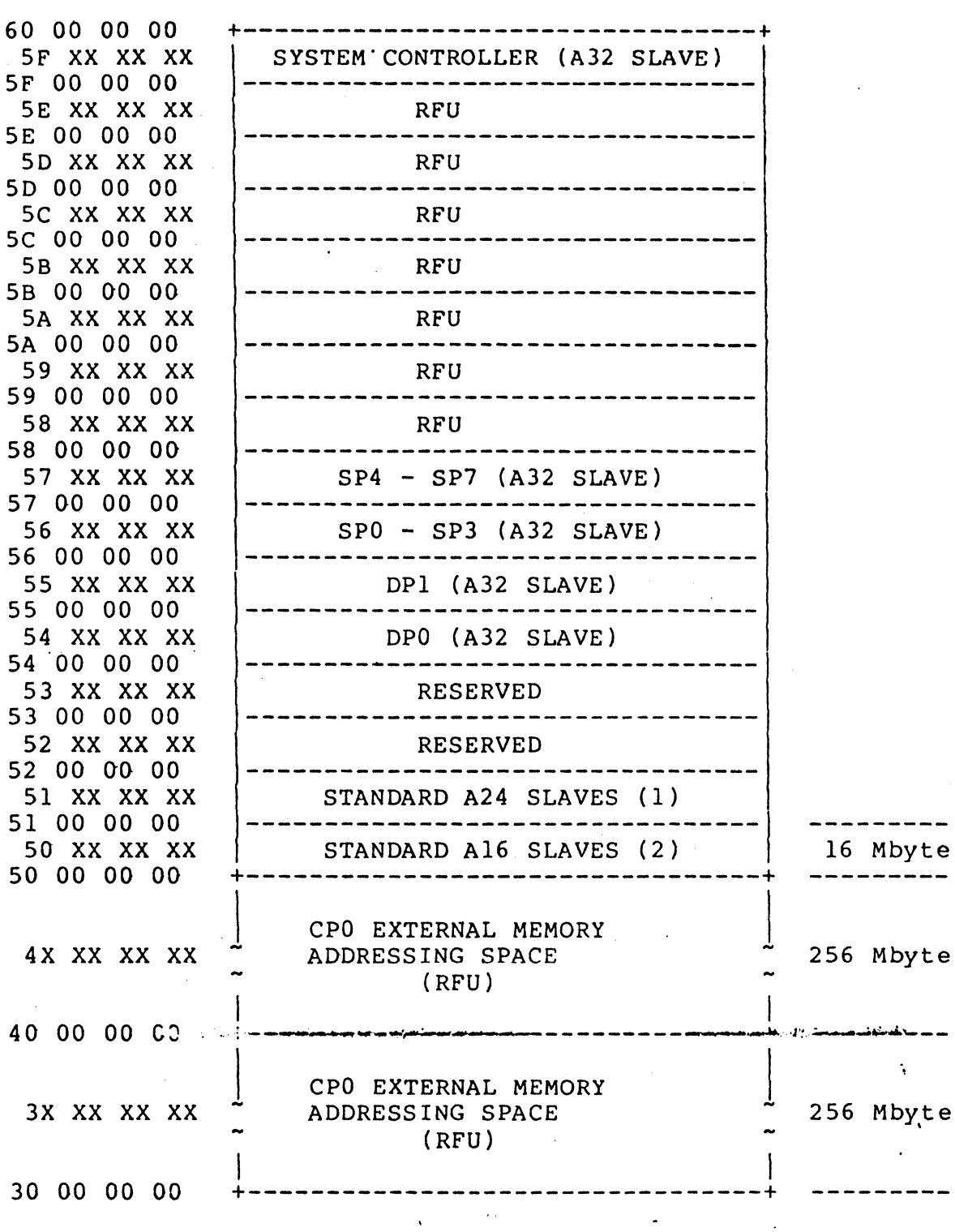


TABLE A.3.4: PHYSICAL MAP (RFU EXTERNAL MEMORY SPACE AND DETAIL OF PROPRIETARY AND STANDARD SLAVES SPACE)

(1) When physical address is in this range, CP0 acts as an A24 Busmaster and VMEbus map depicted in Table A.4.2 is applicable.

(2) When physical address is in this range, CP0 acts as an A16 Busmaster and VMEbus map depicted in Table A.4.3 is applicable.

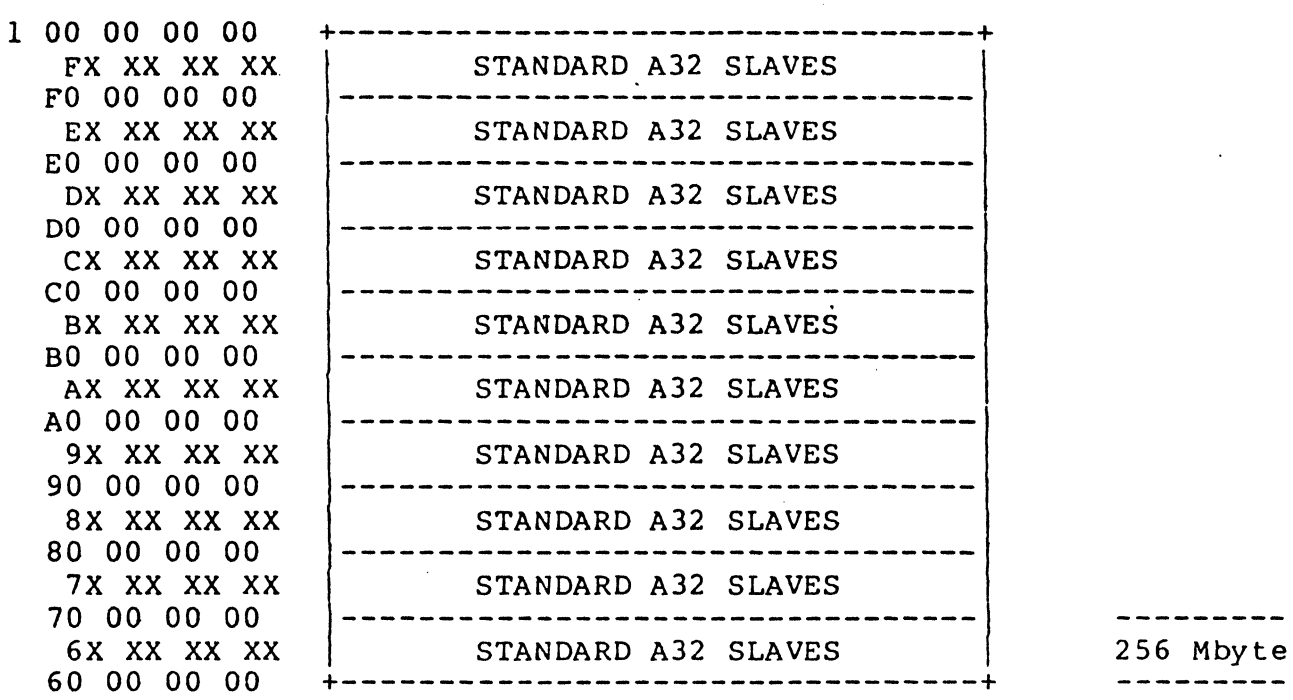


TABLE A.3.5: PHYSICAL MAP (STANDARD A32 SLAVES SPACE)

The least significant bit (VPNUM0+00) of Processor Identifier Number (PIN), wich is wired to both CP0 on three I/O user defined lines, allows a CP0 to know if it is CP01 (PIN=6, PB0=0) or CP02 (PIN=7, PB0=1).

A.2.4.3 VMEbus ADDRESSING SPACE

EXTENDED ADDRESSING

1 00 00 00 00		
FX XX XX XX	STANDARD A32 SLAVES	
F0 00 00 00		
EX XX XX XX	STANDARD A32 SLAVES	
E0 00 00 00		
DX XX XX XX	STANDARD A32 SLAVES	
D0 00 00 00		
CX XX XX XX	STANDARD A32 SLAVES	
C0 00 00 00		
BX XX XX XX	STANDARD A32 SLAVES	
B0 00 00 00		
AX XX XX XX	STANDARD A32 SLAVES	
A0 00 00 00		
9X XX XX XX	STANDARD A32 SLAVES	
90 00 00 00		
8X XX XX XX	STANDARD A32 SLAVES	
80 00 00 00		
7X XX XX XX	STANDARD A32 SLAVES	
70 00 00 00		
6X XX XX XX	STANDARD A32 SLAVES	
60 00 00 00		
5X XX XX XX	HISI A32 SLAVES	
50 00 00 00		
4X XX XX XX	MM (RFU)	
40 00 00 00		
3X XX XX XX	MM (RFU)	
30 00 00 00		
2X XX XX XX	MM2	
20 00 00 00		
1X XX XX XX	MM1	
10 00 00 00		
0X XX XX XX	RESERVED	256 Mbyte
00 00 00 00		

TABLE A.4.1: VMEbus MAP (EXTENDED ADDRESSING)

STANDARD ADDRESSING (1)

1 00 00 00		
FX XX XX	STANDARD A24 SLAVES	
F0 00 00		
EX XX XX	STANDARD A24 SLAVES	
E0 00 00		
DX XX XX	STANDARD A24 SLAVES	
D0 00 00		
CX XX XX	STANDARD A24 SLAVES	
C0 00 00		
BX XX XX	STANDARD A24 SLAVES	
B0 00 00		
AX XX XX	STANDARD A24 SLAVES	
A0 00 00		
9X XX XX	STANDARD A24 SLAVES	
90 00 00		
8X XX XX	STANDARD A24 SLAVES	
80 00 00		
7X XX XX	STANDARD A24 SLAVES	
70 00 00		
6X XX XX	STANDARS A24 SLAVES	
60 00 00		
5X XX XX	HISI A24 SLAVES	
50 00 00		
4X XX XX	MM (RFU)	
40 00 00		
3X XX XX	MM (RFU)	
30 00 00		
2X XX XX	MM2 (1 Mbyte WINDOW) (2)	
20 00 00		
1X XX XX	MM1 (1 Mbyte WINDOW) (2)	
10 00 00		-------
0X XX XX	RESERVED	1 Mbyte
00 00 00		-------

TABLE A.4.2: VMEbus MAP (STANDARD ADDRESSING)

(1) To address as an A24 Busmaster, CP0 issues an address in the range 51000000-51FFFFFF; the eight most significant bits are decoded to set a standard addressing on address modifiers lines, while bits A23-A01 are used to drive the corresponding significant VMEbus address lines.

(2) The lowest Mbyte of each CP0 MEMORY SUB-SYSTEM can be accessed by all A24 Busmasters (and also by other CP0, when acting as an A24 Busmaster).

SHORT ADDRESSING (1)

1 00 00		
FX XX	STANDARD A16 SLAVES	
F0 00		
EX XX	STANDARD A16 SLAVES	
E0 00		
DX XX	STANDARD A16 SLAVES	
D0 00		
CX XX	STANDARD A16 SLAVES	
C0 00		
BX XX	STANDARD A16 SLAVES	
B0 00		
AX XX	STANDARD A16 SLAVES	
A0 00		
9X XX	STANDARD A16 SLAVES	
90 00		
8X XX	STANDARD A16 SLAVES	
80 00		
7X XX	STANDARD A16 SLAVES	
70 00		
6X XX	STANDARD A16 SLAVES	
60 00		
5X XX	HISI A16 SLAVES	
50 00		
4X XX	MM (RFU)	
40 00		
3X XX	MM (RFU)	
30 00		
2X XX	MM2 (4kbyte WINDOW) (2)	
20 00		
1X XX	MM1 (4kbyte WINDOW) (2)	
10 00		-------
0X XX	RESERVED	4 kbyte
00 00		-------

TABLE A.4.3: VMEbus MAP (SHORT ADDRESSING)

(1) To address as an A16 Busmaster, CP0 issues an address in the range 50XX0000-50XXFFFF; the eight most significant bits are decoded to set a short addressing on address modifiers lines, while bits A15-A01 are used to drive the corresponding significant VMEbus address lines.

(2) The lowest 4 kbyte block of each CP0 MEMORY SUB-SYSTEM can be accessed by all A16 Busmasters (and also by other CP0, when acting as an A16 Busmaster).

A.2.4.4 DATA TRANSFER PATHS AND DECODE LOGIC

he CP0 board allows the following data paths:

* between MPU and internal resources
* between MPU and FPU
* between MPU and memory sub-system
* between MPU and VMEbus system resources (controllers/processors)
* between VME Busmasters and memory sub-system.

Decode logic provides all selection signals to enable the above data paths.

Table A.5 lists and summarizes all the selection signals.

DECODE LOGIC SIGNALS DESCRIPTION

SIGNAL NAME	DESCRIPTION
ACTYP1+00	SYNTHESIZE VADMD4+00; INVALID WHEN ADDRESSING A32 AND A16 SLAVES, ASSERTED IN ALL OTHER CASES (A24 SLAVES AND RFU SPACES)
ACTYP2+00	SYNTHESIZE VADMD5+00; INVALID WHEN ADDRESSING A32 SLAVES, ASSERTED IN ALL OTHER CASES (A24 AND A16 SLAVES, RFU SPACES)
ALLCYC+00	ASSERTED TO DECODE AN ALLOWED SPACE (SUPERVISOR DATA, SUPERVISOR PROGRAM, USER DATA, USER PROGRAM)
AVEC-00	ASSERTED TO DECODE AN AUTO-VECTORED INTERRUPT ACKNOWLEDGE CYCLE (LEVEL 6 OR 7)
CPU-00	ASSERTED ON LAS TO DECODE AN MPU ACCESS TO MEMORY SUB-SYSTEM
CPUSP-00	ASSERTED TO DECODE CPU SPACE
CSFPU-00	ASSERTED TO DECODE A CPU SPACE COMMUNICATION WITH FPU (COPROCESSOR IDENTIFIER N. 1)
CSMOVF-00	ASSERTED ON PAS TO DECODE PHYSICAL ADDRESS 0F600000-0F7FFFFF (RESET MONITOR OVERFLOW)
CSRMSK-00	ASSERTED ON PAS TO DECODE AN ACCESS TO PHYSICAL ADDRESS RANGE 0F400000-0F5FFFFF (WRITE ONLY MASK REGISTER)
DCEPRM-00	ASSERTED ON PAS TO DECODE A READ ONLY ACCESS TO EPROMS' BANK ADDRESS RANGE: 0F000000-0F1FFFFF WHEN ENCSEP+00(*) INVALID 00000000-001FFFFF WHEN ENCSEP+00 ASSERTED
DCIOCP-00	ASSERTED ON PAS TO DECODE AN ACCESS TO ADDRESS RANGE: 0F000000-0FFFFFFF WHEN ENCSEP+00 INVALID 00000000-03FFFFFF WHEN ENCSEP+00 ASSERTED
DCIOLC-00	ASSERTED TO DECODE AN ACCESS TO LOCAL RESOURCES ADDRESS RANGE: 04000000-0FFFFFFF WHEN ENCSEP+00 INVALID 00000000-03FFFFFF WHEN ENCSEP+00 ACTIVE
DCPIOT-00	ASSERTED ON PAS TO DECODE AN ACCESS TO ADDRESS RANGE 0F200000-0F3FFFFF (PI/T); ACTIVE ONLY IF ENCSEP+00 INVALID
DC64L-00	ASSERTED TO DECODE A LOGICAL ACCESS TO ADDRESS RANGE 00000000-03FFFFFF OF ALLOWED SPACES (IF CHPRES-00 AND CPOK+00 ASSERTED)

TABLE A.5: DECODE LOGIC SIGNALS DESCRIPTION

(*) ENCSEP+00 ENABLE CHIP SELECT EPROMs' BANK DURING THE FIRST FOUR READ CYCLES AFTER RESET

DECODE LOGIC SIGNALS DESCRIPTION	
SIGNAL NAME	DESCRIPTION
DC64P-00	ASSERTED TO DECODE A PHYSICAL ACCESS TO ADDRESS RANGE 00000000-03FFFFFF OF ALLOWED SPACES (IF CPUOK+00 ASSERTED AND ENCSEP+00 INVALID)
ENDAT1-00	ENABLE DATA TRANSCEIVERS WICH CONNECT SHARED (0-15) AND VMEbus (0-15)
ENDAT2-00	ENABLE DATA TRANSCEIVERS WICH CONNECT SHARED (16-31) AND VMEbus (0-15)
ENDT12-00	ASSERTED WHEN ONE OF TWO ABOVE IS ASSERTED
ENDAT3-00	ENABLE DATA TRANSCEIVERS WICH CONNECT SHARED (16-31) AND VMEbus (16-31)
ENEXT-VM	ASSERTED FOR EXTENDED ADDRESSING DATA TRANFER (FROM/TO VMEbus)
ENSTD-VM	ASSERTED FOR STANDARD ADDRESSING DATA TRANFER (FROM/TO VMEbus)
ENSHOR-VM	ASSERTED FOR SHORT ADDRESSING DATA TRANSFER (FROM/TO VMEbus)
IAKVME-00	ASSERTED TO DECODE A VECTORED INTERRUPT ACKNOWLEDGE CYCLE (LEVEL 1-5)
PREVME-00	ASSERTED TO DECODE A LOGICAL ADDRESS RANGE 10000000-FFFFFFFF (REF. TO VME RESOURCES)
SHORT-VM	ASSERTED FOR STANDARD OR SHORT ADDRESSING DATA TRANSFER
SLVREQ-00	ASSERTED WHEN A VME BUSMASTER WANT ACCESS MEMORY SUB-SYSTEM
VMEACC-00	ASSERTED TO DECODE A PHYSICAL ADDRESS RANGE 10000000-FFFFFFFF

TABLE A.5: DECODE LOGIC SIGNALS DESCRIPTION (CONT.)

A.2.5 SHARED & VMEbus CONTROL LOGIC (BUSCON)

SHARED & VMEbus CONTROL LOGIC (BUSCON), has two foundamental functions:

a) arbitrates the conflict on the MEMORY SUB-SYSTEM (connected to the board) and the associated SHARED BUS, between MPU and remote VME masters: this function is performed by a 2-stage ARBITER;

b) manages the requests to access VMEbus from CP0 PWA, and conflicts for VMEbus mastership with other concurrents (in this case arbiter resides on SYSTEM CONTROLLER PWA and CP0's REQUESTER asks for a DTB cycle in concurrence with other possible remote REQUESTERs): this function is performed by VMEbus REQUESTER described at paragraph A.2.6.3.

BUSCON controls or originates most of the signals that VMEbus INTERFACE transmits/receives to/from VMEbus (see par. A.2.6 for more details).

BUSCON also originates the signal grant to MEMORY SUB-SYSTEM to allow refresh.

A.2.5.1 GENERAL DESCRIPTION

The signals used to perform the above functions are described in table A.6.

CP0-BUSCON INTERFACE (INPUT SIGNALS)	
SIGNAL NAME	DESCRIPTION
CKBC+00	16.67 MHz CLOCK (COHERENT WITH MPU CLOCK)
CKBC-00	16.67 MHz CLOCK (INCOHERENT " " ")
VDSTBX-00; X=0-1	DATA STROBES FROM VMEbus
CCDSAK+00	ACTIVE IF CACHE HIT
CPULAS-10	LOGICAL ADDRESS STROBE FROM MPU
PAS-MU	PHYSICAL ADDRESS STROBE FROM CM1
REFREQ-10	REFRESH REQUEST TO BUSCON (ACTIVE UNTIL SERVICE ACCOMPLISHED)
IAKVME-00	INTERRUPT ACKNOWLEDGE (LEV. 1-5)
VMEACC-00	SEE TABLE A.5
VMEREQ-00	ACTIVE IF ON-BOARD MASTER WANT VMEbus
SLVREQ-00	ACTIVE IF REMOTE MASTERS ON VMEbus WANT ACCESS TO SHARED MEMORY
VMEAS+20	VALID ADDRESS STROBE (FROM VME)
RMC-00	READ-MODIFY-WRITE CYCLE
CPUSPC-00	CPU SPACE
ECS-00	EXTERNAL CYCLE
BG1IN-00	BUS GRANT INPUT (FOR PRIORITY REQUEST OF LEVEL ONE)

TABLE A.6: CP0-BUSCON INTERFACE

CP0-BUSCON INTERFACE (OUTPUT SIGNALS)	
SIGNAL NAME	DESCRIPTION
DATDIR-00	SELECT DIRECTION FOR TRANSCEIVERS WHICH CONNECT SHARED AND VMEbus (IF ACTIVE, DATA ARE RECEIVED FROM VME)
VMECYC-00	ASSERT/GRANT A VME CYCLE (FROM CP TO VME)
MPXSW1+00	SELECT WICH ONE CAN ACCESS MEMORY, BETWEEN MPU AND VME MASTERS (WHEN ASSERTED, VME IS SELECTED)
CPDEN-00	ENABLE DATA TRANSCEIVERS WICH CONNECT MPU TO SHARED BUS
RWMP+10	SELECT DIRECTION FOR ABOVE TRANSCEIVERS (IF ACTIVE, DATA ARE RECEIVED FROM SHARED BUS)
REFGR+00	REFRESH GRANT (WHEN ASSERTED REFRESH IS ALLOWED)
SLAVON+00	ASSERT/GRANT ALLOWING A SLAVE CYCLE (FROM VME ON MEMORY)
VIAME-00	START A MEMORY CYCLE FROM VME
START-00	CONFIRM A MEMORY CYCLE
MMSLAV+00	MEMORY SLAVE
SLVBLK+00	BLOCK MPU ACCESS TO MEMORY
VBURQ1-00	DTB REQUEST LEVEL ONE

TABLE A.6: CP0-BUSCON INTERFACE (CONT.)

A.2.5.2 ARBITER

The schematics of BUSCON ARBITER is depicted on fig. A.2; three possible requests to access the SHARED BUS and/or associated MEMORY SUB-SYSTEM are passed to ARBITER:

* FREERQ-00 this request, when asserted, notify the arbiter that CP0 PWA has obtained DTB on VME, thus MPU needs the SHARED BUS in order to transfer data to/from VME;

* SLVREQ-00 this request, when asserted, notify the ARBITER that a remote VME Busmaster is waiting for access to MEMORY SUB-SYSTEM and SHARED BUS;

* REFREQ-00 this request, when asserted, notify the ARBITER that MEMORY SUB-SYSTEM is waiting for a grant to begin a refresh cycle.

The arbitrating process is performed on a 60 ns cycle basis, that is the ARBITER accomplish his choice in a time period less than 60 ns (from the trailing edge of clock signal wich sample an active request).

The arbitrating process is disabled during an external MPU cycle, that is when ECS-00 is asserted, and during the entire cycle assigned by the previous iteration of decision process (DISABLE+00 active), that is if resources are not available because they are used by the winner of the last arbitration.
The ECS-00 conditioning of the decision process is accomplished o obtain a privileged path MPU to MEMORY SUB-SYSTEM, for which MPU doesn't request SHARED BUS but owns it for all the time that other concurrents do not.

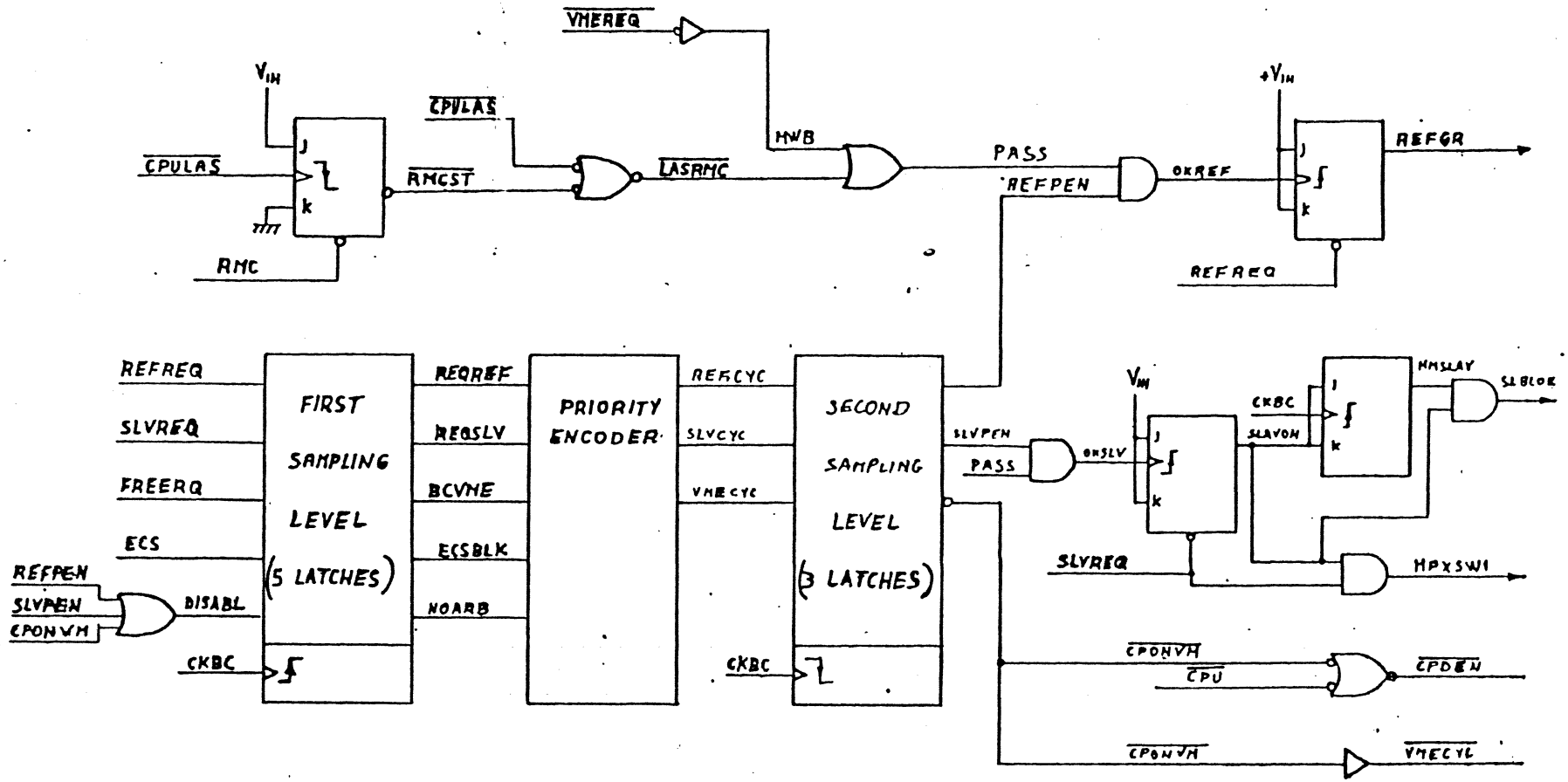
Status of request lines and disable lines is sampled on the trailing edge of the clock signal CKBC-00 by the first stage of latches and the outputs are encoded by a PAL to a second stage of latches with the following priority levels (only one asserted):

SLVREQ-00 HIGHEST PRIORITY LEVEL
REFREQ-00 MEDIUM PRIORITY LEVEL
FREERQ-00 LOWEST PRIORITY LEVEL

conditioned by inhibit signals:

DISABL+00 asserted during execution of previous assigned cycle
ECS-00 asserted when MPU is executing an external cycle.

Only one request can be asserted on the second stage outputs, after the falling edge of the clock signal CKBC-00, and when this occurs, a disable signal is feed-back to first stage input, to disable arbitration process, beginning with the next cycle.


VMEREQ
V_{IH}
CPULAS
CPULAS
RMC
RMCST
LASRMC
MWB
PASS
REFPEN
OKREF
$+V_{IH}$
REFGR
REFREQ
REFREQ
SLVREQ
FREERQ
ECS
REFPEN
SLVPEN
CPONVM
DISABL
CKBC
FIRST SAMPLING LEVEL (5 LATCHES)
REQREF
REQSLV
BCVME
ECSBLK
NOARB
PRIORITY ENCODER
REFCYC
SLVCYC
VMECYC
SECOND SAMPLING LEVEL (3 LATCHES)
CKBC
SLVPEN
PASS
OKSLV
V_{IH}
CKBC
SLAVON
HMSLAV
SLBLOK
SLVREQ
HPXSW1
CPONVM
CPU
CPUEN
CPONVM
VMECYL

.2.6 VMEbus INTERFACE

The VMEbus interface on CP0 board supports operations in a VMEbus based system and the full 4 Gbyte address range of the MC68020 MPU.

Access to the backplane address, data and control lines is provided by two triple row, 96-pin VME connectors on the rear of the module. Pin assignment, connectors physical characteristics and VMEbus signals and timing requirements are fully described in the VMEbus specification manual. CP0 can act on VMEbus as an A32:D32 Busmaster or as an A32:D32 Slave.

Definitions of A32 Master, A32 Slave, D32 master and D32 slave are here provided for completeness.

An A32 Master is capable of driving 31 address lines (A01-A31). It drives 31 lines with a valid address whenever it places an extended address modifier AM code on the bus, 23 lines whenever it places a standard AM code on the bus, and 15 lines when it places a short AM code on the bus.

An A32 Slave must be capable of decoding up to 31 address lines (A01-A31). It decodes 31 lines when an extended address modifier AM code is on the bus, 23 when a standard AM code is present, and 15 when a short AM code is present.

A D32 Master is capable of driving and monitoring 32 data lines (D00-D31). This allows it to do 32-bit data transfers on (D00-D31) while driving LWORD* low, and 16-bit data transfers on (D00-D15), or 8-bit data transfers on (D00-D07) or (D08-D15) while driving LWORD* high.

A D32 Slave is capable of driving and monitoring 32 data lines (D00-D31). When LWORD* is low, it transfers data on all 32 data lines. When LWORD* is high, it must be capable of transferring all internally stored data on the lower 16 data lines (D00-D15).

Table A.8 shows a cross-reference list between internal design names and standard mnnemonic for all possible VME signals on J1/P1 and J2/P2 connectors. Note also that most of User I/O pins of J2/P2 connector are used to connect CACHE PWA.

2.6.1 ADDRESS MODIFIER CODES

Of the 14 address modifier codes (AM) defined by the VMEbus specs, a subset of ten is supported by CP0 PWA: they are listed in table A.7.

AM CODE	AM bit 543210	FUNCTION
3E	111110	STANDARD SUPERVISORY PROGRAM ACCESS
3D	111101	STANDARD SUPERVISORY DATA ACCESS
3A	111010	STANDARD NON-PRIVILEGED PROGRAM ACCESS
39	111001	STANDARD NON-PRIVILEGED DATA ACCESS
2D	101101	SHORT SUPERVISORY I/O DATA ACCESS
29	101001	SHORT NON-PRIVILEGED I/O DATA ACCESS
0E	001110	EXTENDED SUPERVISORY PROGRAM ACCESS
0D	001101	EXTENDED SUPERVISORY DATA ACCESS
0A	001010	EXTENDED NON-PRIVILEGED PROGRAM ACCESS
09	001001	EXTENDED NON-PRIVILEGED DATA ACCES

TABLE A.7 ADDRESS MODIFIER CODES

A.2.6.2 UTILITY SIGNAL LINES

The following utility signal lines are supported/used by CP0 PWA:

* SYSRESET* (system reset), open collector line driven by a POWER MONITOR module and/or by a manual switch (such as from an operator's panel); whenever SYSRESET* is driven to low, it must be held there for a minimum of 200 millisecond; this signal is used to reset MPU, FPP and all remaining control logic.
* SYSFAIL* (failure of a system test), is shown via the SYSTEM FAIL line; it is recommended but not required that all boards within the VMEbus system drive this SYSTEM FAIL line low upon power up, and mantain it low until they have passed their respective self-test; whenever this line is driven to low by the VMEbus environment, a level 7 interrupt request is issued to MPU (maskable).
* ACFAIL* line, provided by POWER MONITOR MODULE, signals a power down: whenever this line is driven to low by the VMEbus environment, a level 7 interrupt request is issued to MPU (maskable).

Two different conflicts on resources are arbitrated and/or managed by SHARED & VMEbus CONTROL LOGIC (see par. A.2.5):

a) on the VMEbus, between CP0 and other concurrents for bus mastership (in this case CP0 wants a VME cycle and the on board requester ask for it, arbiter will reside on SYSTEM CONTROLLER PWA)
b) on the shared memory (controlled by on board DYNAMIC RAM CONTROLLER), between MPU and remote VME masters: in this case arbitration is performed by BUSCON, wich can allows to remote master a memory access to close the pending VME cycle (board looks like a slave device)

SHARED & VMEbus CONTROL LOGIC generates the signal handshaking and timing required by the VMEbus transfer protocol and provides all other control signals required to handle the physical link between shared and VMEbus.

When acting as slave CP0 will respond to A32 (four privileged types), A24 (four privileged types) and A16 (two privileged types).

VMEbus INTERFACE will provide a swapping configuration of transceivers to match D16 and D32.

A.2.6.3 VME REQUESTER

The CP0's REQUESTER will ask on BR1 line and both CP0s will be chained on BG1 line. The REQUESTER will be RELEASE WHEN DONE(RWD) type but VMEbus will be released in any case after completion of the single VME cycle (e.g. READ, WRITE or READ-MODIFY-WRITE cycle); a new request must be issued from the REQUESTER to obtain another VME cycle. The priority level of the request from CP0 board is one.

VMEbus REQUESTER provides the following logic functions:

a) asserts VBURQ1-00 to request VME DTB, when physical address originated by MPU/CM1 refers to VME SPACE (e.g. they are in the range 10000000 throught FFFFFFFF and VMEACC-00 signal is asserted) or when an interrupt acknowledge cycle is executed by CP0 to service a VME interrupter (e.g. IAKVME-00 is asserted). Assertion of VBURQ1-00 continues until VBG1IN-00 is asserted and becomes invalid when VBUSY-00 is driven to

b) blocks the grant propagation on the BG1 daisy-chain after recognition that CP0 has a DTB request pending on VBURQ1-00 line ; if not, REQUESTER pass on the grant, enabling lower priority REQUESTERs on the same VBURQ1-00 line.

c) after completion of the current cycle on VMEbus, REQUESTER asserts VMEBC+00 (FREERQ-00), to notify BUSCON that MPU needs the SHARED BUS to access to VMEbus, and then asserts VBUSY-00.

Time-out on bus request and data transfer will be provided by SYSTEM CONTROLLER PWA.

When acting as Master, CP0 PWA can talk to VME as one of the following:

* A32:D32 to connect to other CP0 and to D32 slaves
* A32:D16 to connect to DP and D16 slaves
* A24:D16/D32 to connect to standard slaves

A.2.6.4 VME INTERRUPT HANDLER

VME IRQ1 to IRQ5 lines can be managed by the CP0 PWA and individually masked via a programmable register mapped as I/O of the on-board MPU. This provision are made to allow the operating system to configure the same CP0 as CP01 or CP02 and to manage the fault of a single CP0 in a dual-IDP configuration.

During an interrupt acknowledge cycle (LEV. 1-5), the IAKVME-00 signal is asserted and actives a DTB request to VMEbus; when the cycle is allowed, a vector is fetched from the interrupter, wich is used to point the service routine.

A.2.6.5 VME INTERRUPTER

The CP0 PWA can interrupt on the lines IRQ1 to IRQ7, by programming the BIM residing on SYSTEM CONTROLLER PWA, or, in the case of STATION PROCESSOR (see SP0 PDD), by writing on an hardware register located at each SP0 address space, wich substains a single attention bit to SP0 interrupt logic.

In the dual-IDP configuration, CP01 and CP02 can also interrupt each other, by progamming the BIM residing on SYSTEM CONTROLLER PWA to interrupt on line IRQ2 and IRQ1 respectively.

A.2.6.6 SWAPPER

A transceivers arrangement allows routing of BDAT31+MM-BDAT16+MM section of SHARED-BUS to VDAT00+00-VDAT15+00 section of VMEbus.

ENDAT3-00 CONNECTS BDAT31+MM-BDAT16+MM TO VDAT31+00-VDAT16+00

ENDAT1-00 CONNECTS BDAT15+MM-BDAT00+MM TO VDAT15+00-VDAT00+00

ENDAT2-00 CONNECTS BDAT31+MM-BDAT16+MM TO VDAT15+00-VDAT00+00.

VMEbus J1/P1(*) PIN ASSIGNMENT (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	DO3
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	N.C.	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	VIAKIO-00	IACKIN*
A22	VIAKIO-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	N.C.	-12 V
A32	ZVP05	+5 V

TABLE A.8: CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES AND STANDARD VME MNEMONIC

VMEbus J1/P1(*) PIN ASSIGNMENT (ROW B)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
B 1	VBUBSY-00	BBSY*
B 2	N.C.	BCLR*
B 3	VACFAL-00	ACFAIL*
B 4	VBG0IO-00	BG0IN*
B 5	VBG0IO-00	BG0OUT*
B 6	VBG1IN-00	BG1IN*
B 7	VBG1OU-00	BG1OUT*
B 8	VBG2IO-00	BG2IN*
B 9	VBG2IO-00	BG2OUT*
B10	VBG3IO-00	BG3IN*
B11	VBG3IO-00	BG3OUT*
B12	N.C.	BR0*
B13	VBURQ1-00	BR1*
B14	N.C.	BR2*
B15	N.C.	BR3*
B16	VADMD0+00	AM0
B17	VADMD1+00	AM1
B18	VADMD2+00	AM2
B19	VADMD3+00	AM3
B20	ZGND	GND
B21	N.C.	SERCLK
B22	N.C.	SERDAT
B23	ZGND	GND
B24	N.C.	IRQ7*
B25	N.C.	IRQ6*
B26	VINRQ5-00	IRQ5*
B27	VINRQ4-00	IRQ4*
B28	VINRQ3-00	IRQ3*
B29	VINRQ2-00	IRQ2*
B30	VINRQ1-00	IRQ1*
B31	N.C.	+ 5 V STDBY
B32	ZVP05	+ 5 V

TABLE A.8: CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES (CONT.) AND STANDARD VME MNEMONIC

(*) Z01 IN THE DESIGN DOCUMENTATION

VMEbus J1/P1(*) PIN ASSIGNMENT (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	N.C.	+ 12 V
C32	ZVP05	+ 5 V

TABLE A.8: CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES (CONT.) AND STANDARD VME MNEMONIC

VMEbus J2/P2(*) PIN ASSIGNMENT (ROW A)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
A 1	N.C.	USER I/O (+ 5 V)
A 2	ZGND	USER I/O (GND)
A 3	N.C.	USER I/O
A 4	CDMOWF-00	USER I/O (MONITOR OVERFOLW)
A 5	CHPRES-00	USER I/O (CACHE PRESENT)
A 6	CNTRG0+00	USER I/O (CACHE DISABLE)
A 7	ZGND	USER I/O (GND)
A 8	PADD11+CH	USER I/O (PHYSICAL ADDRESS 11)
A 9	PADD12+CH	USER I/O (PHYSICAL ADDRESS 12)
A10	PADD13+CH	USER I/O (PHYSICAL ADDRESS 13)
A11	PADD14+CH	USER I/O (PHYSICAL ADDRESS 14)
A12	PADD15+CH	USER I/O (PHYSICAL ADDRESS 15)
A13	ZVP05	USER I/O (GND)
A14	PADD16+CH	USER I/O (PHYSICAL ADDRESS 16)
A15	PADD17+CH	USER I/O (PHYSICAL ADDRESS 17)
A16	PADD18+CH	USER I/O (PHYSICAL ADDRESS 18)
A17	N.C.	USER I/O
A18	VPNUM0+00	PROC.IDENTIFIER BIT 0
A19	VPNUM1+00	PROC.IDENTIFIER BIT 1
A20	VPNUM2+00	PROC.IDENTIFIER BIT 2
A21	N.C.	USER I/O
A22	ZGND	USER I/O
A23	N.C.	USER I/O
A24	ZGND	USER I/O
A25	N.C.	USER I/O
A26	ZVP05	USER I/O
A27	ZVP05	USER I/O
A28	ZVP05	USER I/O
A29	ZVP05	USER I/O
A30	ZVP05	USER I/O
A31	ZGND	USER I/O (GND)
A32	ZGND	USER I/O (GND)

TABLE A.8: CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES (CONT.) AND STANDARD VME MNEMONIC

VMEbus J2/P2 (*) PIN ASSIGNMENT (ROW B)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
B 1	ZVP05	+ 5 V
B 2	ZGND	GND
B 3	RESERV+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	VDAT16+00	D16
B15	VDAT17+00	D17
B16	VDAT18+00	D18
B17	VDAT19+00	D19
B18	VDAT20+00	D20
B19	VDAT21+00	D21
B20	VDAT22+00	D22
B21	VDAT23+00	D23
B22	ZGND	GND
B23	VDAT24+00	D24
B24	VDAT25+00	D25
B25	VDAT26+00	D26
B26	VDAT27+00	D27
B27	VDAT28+00	D28
B28	VDAT29+00	D29
B29	VDAT30+00	D30
B30	VDAT31+00	D31
B31	ZGND	GND
B32	ZVP05	+ 5 V

TABLE A.8: CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES (CONT.) AND STANDARD VME MNEMONIC

VMEbus J2/P2 (*) PIN ASSIGNMENT (ROW C)

PIN NAME	SIGNAL DESIGN NAME	STANDARD VME MNEMONIC
C 1	N.C.	USER I/O (+ 5 V)
C 2	ZGND	USER I/O (GND)
C 3	N.C.	USER I/O
C 4	CNTRG1+00	USER I/O (CONTROL REG. BIT 1)
C 5	STSRG2+00	USER I/O (STATUS REG. BIT 2)
C 6	STSRG0+00	USER I/O (STATUS REG. BIT 0)
C 7	STSRG1+00	USER I/O (STATUS REG. BIT 1)
C 8	ZGND	USER I/O (GND)
C 9	PADD19+CH	USER I/O (PHYSICAL ADDRESS 19)
C10	PADD20+CH	USER I/O (PHYSICAL ADDRESS 20)
C11	PADD21+CH	USER I/O (PHYSICAL ADDRESS 21)
C12	PADD22+CH	USER I/O (PHYSICAL ADDRESS 22)
C13	ZVP05	USER I/O (+5 V)
C14	PADD23+CH	USER I/O (PHYSICAL ADDRESS 23)
C15	PADD24+CH	USER I/O (PHYSICAL ADDRESS 24)
C16	PADD25+CH	USER I/O (PHYSICAL ADDRESS 25)
C17	DC64P-00	USER I/O (SEE TABLE 13)
C18	STSRG3+00	USER I/O (STATUS REG. BIT 3)
C19	N.C.	USER I/O
C20	N.C.	USER I/O
C21	N.C.	USER I/O
C22	ZGND	USER I/O
C23	N.C.	USER I/O
C24	ZGND	USER I/O
C25	N.C.	USER I/O
C26	ZVP05	USER I/O
C27	ZVP05	USER I/O
C28	ZVP05	USER I/O
C29	ZVP05	USER I/O
C30	ZVP05	USER I/O
C31	ZGND	USER I/O (GND)
C32	ZGND	USER I/O (+ 5 V)

TABLE A.8: CROSS-REFERENCE BETWEEN VME SIGNAL DESIGN NAMES (CONT.) AND STANDARD VME MNEMONIC

A.2.7 CACHE INTERFACE

This interface provides on the CP0's central front connector Y02 and on the J2/P2(Z02) VMEbus connector all the I/O signals to support an optional cache memory.

A detailed signal pin assignment is provided by the above table A.8 (Z02) and the next table A.9 (Y02).

Table A.10 provides a more detailed description of the cache interface signals.

CACHE KEY FUNCTIONS AND CHARACTERISTICS

* physical cache addressing
* 4-byte block size
* caching function performed for both data and instruction (user and supervisor)
* size 16 kbyte (SRAM chips)
* sector associative (2 kbyte X 8 sectors)
* write trough policy
* FIFO replacement algorithm
* cache to memory path: 4 bytes
* parity checked at byte level on data RAM
* EARLY HIT technique
* 68020 instruction rerun on missing occurrence
* VMEbus traffic monitor to invalidate a cache block after a VME Busmaster has modified the equivalent block in memory sub-system
* no wait state @ 16.67 MHz

For more details see also SGM2 CACHE MEMORY (CH0) PDD (SECTION B).

Y02 CACHE CONNECTOR PIN ASSIGNMENT (ROW A)

PIN NAME	SIGNAL NAME	DESCRIPTION
A 1	ZGND	
A 2	DATA00+CH	DATA FROM/TO MPU DATA BUS
A 3	DATA01+CH	" " " " "
A 4	DATA02+CH	" " " " "
A 5	DATA03+CH	" " " " "
A 6	DATA04+CH	" " " " "
A 7	DATA05+CH	" " " " "
A 8	DATA06+CH	" " " " "
A 9	DATA07+CH	" " " " "
A10	ZGND	
A11	DATA08+CH	DATA FROM/TO MPU DATA BUS
A12	DATA09 CH	" " " " "
A13	DATA10+CH	" " " " "
A14	DATA11+CH	" " " " "
A15	DATA12+CH	" " " " "
A16	DATA13+CH	" " " " "
A17	DATA14+CH	" " " " "
A18	DATA15+CH	" " " " "
A19	ZGND	
A20	CDSAK0-00	\ DATA & SIZE TRANSFER ACKNOWLED-
A21	CDSAK1-00	/ GE (TO MPU)
A22	ZGND	
A23	CHALT-00	HALT TO MPU (O.C. DRIVEN)
A24	CBERR-00	BUS ERROR TO MPU (O.C. DRIVEN)
A25	ZGND	
A26	FCODE0+00	MPU FUNCTION CODE 0 (FROM MPU)
A27	FCODE1+00	MPU FUNCTION CODE 1 (FROM MPU)
A28	FCODE2+00	MPU FUNCTION CODE 2 (FROM MPU)
A29	ZGND	
A30	CCDSAK-10	CACHE EARLY HIT
A31	ZGND	
A32	BCLK2-CH	MPU CLOCK (INCOHERENT)

TABLE A.9: Y02 CACHE CONNECTOR PIN ASSIGNMENT

Y02 CACHE CONNECTOR PIN ASSIGNMENT (ROW B)

PIN NAME	SIGNAL NAME	DESCRIPTION
B 1	CPULDS-01	RFU
B 2	ZGND	
B 3	N.C.	
B 4	ZGND	
B 5	ZGND	
B 6	ZGND	
B 7	ZGND	
B 8	ZGND	
B 9	ZGND	
B10	N.C.	
B11	ZGND	
B12	ZGND	
B13	ZGND	
B14	ZGND	
B15	ZGND	
B16	ZGND	
B17	N.C.	
B18	ZGND	
B19	SIZE0+00	MPU TRANSFER SIZE BIT 0
B20	SIZE1+00	MPU TRANSFER SIZE BIT 0
B21	RMC-00	READ-MODIFY-WRITE CYCLE
B22	RWMP+00	READ/WRITE
B23	ZGND	
B24	PAS-MU	PHYSICAL ADDRESS STROBE
B25	ZGND	
B26	BCLK1+CH	MPU CLOCK (COHERENT)
B27	ZGND	
B28	BCLK1-CH	MPU CLOCK (INCOHERENT)
B29	ZGND	
B30	CPULAS-01	RFU
B31	CS2CPU-00	STATICIZED LAS (ACTIVE LOW) (1)
B32	CS2CPU+00	STATICIZED LAS (ACTIVE HIGH) (1)

TABLE A.9: Y02 CACHE CONNECTOR PIN ASSIGNMENT (CONT.)

(1) ONLY WHEN CACHE ACTIVE AND DC64L-00 ASSERTED

Y02 CACHE CONNECTOR PIN ASSIGNMENT (ROW C)

PIN NAME	SIGNAL NAME	DESCRIPTION
C 1	ZGND	
C 2	DATA16+CH	FROM/TO MPU DATA BUS
C 3	DATA17+CH	" " " "
C 4	DATA18+CH	" " " "
C 5	DATA19+CH	" " " "
C 6	DATA20+CH	" " " "
C 7	DATA21+CH	" " " "
C 8	DATA22+CH	" " " "
C 9	DATA23+CH	" " " "
C10	ZGND	
C11	DATA24+CH	FROM/TO MPU DATA BUS
C12	DATA25+CH	" " " "
C13	DATA26+CH	" " " "
C14	DATA27+CH	" " " "
C15	DATA28+CH	" " " "
C16	DATA29+CH	" " " "
C17	DATA30+CH	" " " "
C18	DATA31+CH	" " " "
C19	ZGND	
C20	PADD10+00	PHYSICAL ADDRESS
C21	PADD09+00	" "
C22	PADD08+00	" "
C23	PADD07+00	" "
C24	PADD06+00	" "
C25	PADD05+00	" "
C26	PADD04+00	" "
C27	PADD03+00	" "
C28	PADD02+00	" "
C29	PADD01+00	" "
C30	PADD00+00	" "
C31	MEMDSK+00	MEMORY DATA TRANSFER ACKNOWLEDGE
C32	CASACT+00	CACHE ASYNCHRONOUS SERVICE

TABLE A.9: Y02 CACHE CONNECTOR PIN ASSIGNMENT (CONT.)

CACHE INTERFACE DETAILED SIGNALS DESCRIPTION

SIGNAL NAME	DESCRIPTION
BCLK1+CH	16.67 MHz COHERENT MPU CLOCK
BCLK1-CH	16.67 MHz INCOHERENT MPU CLOCK
BCLK2-CH	16.67 MHz INCOHERENT MPU CLOCK
CASACT+00	CACHE ASYNCHRONOUS SERVICE ACTIVE (ASSERTED DURING FLUSHING OPERATIONS)
CBERR-00	BUS ERROR TO MPU (O.C. DRIVEN SIGNAL)
CCDSAK-00	CACHE EARLY HIT; ASSERTED ON STATISTICAL PREVISION OF HIT
CDMOWF-OO	MONITOR OVERFLOW (ASSERTED WHEN VME MONITOR CAN'T FOLLOW WRITE ACCESS FROM VME TO MEMORY SUB-SYSTEM THAT IT MUST CACHE)
CDSAKX-00; X=0-1	CACHE DATA&SIZE TRANSF. ACKNOWLEDGE TO MPU
CHALT-00	HALT TO MPU (O.C. DRIVEN SIGNAL)
CHPRES-00	ASSERTED IF OPTIONAL CACHE IS PRESENT
CS2CPU+00	MPU S2-STATE DECODE
CS2CPU-00	MPU S2-STATE DECODE
CNTRG0+00	CACHE DISABLE (FROM CP0) (*)
CNTRG1+00	NORMAL/DIAGNOSTIC MODE WHEN VALID/INVALID
DATAXX+CH; X=0-31	DATA FROM/TO MPU DATA BUS
DC64L-00	EARLY DECODE OF LOWEST 64 M ADDRESS SPACE
DC64P-00	EARLY DECODE OF LOWEST 64 M ADDRESS SPACE
FCODEX+00; X=0-2	MPU FUNCTION CODES
PADDXX+00; X=0-10	PHYSICAL ADDRESS
PADDXX+CH; X=11-25	PHYSICAL ADDRESS
PAS-MU	PHYSICAL ADDRESS STROBE
RMC-00	READ-MODIFY-WRITE CYCLE
RWMP+00	READ/WRITE (ASSERTED IF READ)
SIZEX+00; X=0-1	TRANSFER SIZE
STSRGX+00; X=1-3	STATUS REGISTER.LINES (TO MPU)

TABLE A.10: CP0-CACHE INTERFACE

(*) The signal CNTRG0+00 allows inactivation/activation when HIGH/LOW of the cache on software control, for purposes as diagnostics, performance evaluation and others. During inactive status cache control logic invalids (flush) all entries. At reset time CACHE waits for activation from the associated CP0.

A.2.8 MEMORY SUB-SYSTEM INTERFACE

Table A.11 lists and describes all signals and controls provided to MEMORY SUB-SYSTEM by CP0 board. MEMORY SUB-SYSTEM will reside on up to three boards.

CP0-MEMORY SUBSYSTEM INTERFACE	
SIGNAL NAME	DESCRIPTION
BCLK1+MM	16.67 MHz CLOCK (COHERENT WITH MPU CLOCK)
MADDXX+00; X=2-25	MULTIPLEXED ADDRESS (ON SHARED BUS)
PADDXX+00; X=28-31	HIGHEST MPU PHYSICAL ADDRESS
PADD00+10-PADD01+01	LOWEST MPU PHYSICAL ADDRESS
VMAD01+00	LOWEST VME ADDRESS SIGNAL
SIZEX+10; X=0-1	MPU TRANSFER SIZE
RWMP+10	DATA TRANSFER DIRECTION (FOR MPU CYCLES)
DTACK-MM	DATA TRANSFER ACKNOWLEDGE (TO MPU OR VME)
BERR-MM	BUS ERROR (TO MPU OR VME)
HALT-MM	HALT (TO MPU)
VIACPU-00	ENABLE A MEMORY CYCLE FROM MPU
LWORD-MM	ASSERT A LONG-WORD TRANSFER (FROM/TO VME)
DSX+MM; X=0-1	VME DATA STROBES (ACTIVE HIGH)
MEPAS+10	PHYSICAL ADDRESS STROBE (FROM CMMU)
MDIRW-00	DATA TRANSFER DIRECTION (FOR SLAVE CYCLES)
ENSLV-MM	ENABLE A MEMORY CYCLE FROM VME
START-00	START A MEMORY CYCLE
REFREQ-10	REFRESH REQUEST, ASSERTED TO ASK A REFRESH CYCLE UNTIL CYCLE ACCOMPLISHED
REFGR+00	REFRESH GRANT, WHEN ASSERTED ALLOW A REFRESH CYCLE TO DRC
CPUOK+00	ASSERTED IF CP0 IS RIGHT INSTALLED
STOPCK-00	ASSERTED TO STOP CLOCK DISTRIBUTION
LOCK-00	ASSERTED TO DISABLE A NEW CYCLE
OEBDAT-00	ENABLE DATA TRANSFER FROM/TO MEMORY
ACFAIL-MM	ASSERTED ON POWER DOWN
ENDME-MM	STROBE WHICH RE-ENABLE A VME CYCLE
NREJO-00	ASSERTED DURING A REFRESH CYCLE

TABLE A.11: CP0-MEMORY SUB-SYSTEM INTERFACE

MEMORY SUB-SYSTEM KEY FUNCTIONS

* dual-ported memory: path to IDP (private bus) and to System bus (VMEbus)
* IDP memory addressability up to 16 Mbyte
* 4-byte parallelism (longword)
* data path to IDP (private bus) and to System bus (VMEbus): 4-byte
* operations' type:
  - read
  - write (1-, 2-, 3- and 4-byte)
  - read-modify-write
  - refresh
* cycle time:
  - read: 300 ns
  - write: 4-byte 300 ns
  - read-modify-write: 480 ns (this time is applicable also for 1-, 2- and 3-byte write operations)
* refreshing: 16 cycles (300 ns each) every 240 us
* EDAC (SEC-DEC) control: 7 bits on 4 bytes

A more detailed description will be provided by SGM2 MEMORY SUB-SYSTEM PDD (SECTION G).

A.2.9 EPROM

Two 16 kbyte packages (27128) have been utilized: these two packages are connected in such a way to constitute one 16 kword bank, mapped in the range (2 Mbyte):

0F 00 00 00 through 0F 1F FF FF

The EPROM contains EXCEPTION VECTOR TABLE at initialization time, resident diagnostic routines, HW initialization and operating systyem boot-strap routine.

A special circuit will force, after reset, the effective physical address of the reset vector in the EPROM; this forcing remains valid until ENCSEP+00 is valid (during first four read cycles after reset).

Access to EPROM requires 4 wait-state (420 ns).

A.2.10 PARALLEL I/O AND TIMER

Those functions are handled via a PARALLEL INTERFACE/TIMER (PI/T 68230), mapped in the range 0F200000 through 0F3FFFFF (2 Mbyte), whom main features are:

- 24-bit programmable timer with 5-bit prescaler

- (16+8)-bit parallel I/O.

The first 16 I/O lines are those directly handled by port A and port B of the 68230, the remaining 8 lines are those of port C: then the use of these lines as a simple I/O lines depends on exact utilization of 68230.

Table A.12 describes the assignment of CP0 signals to 68230 pins.

PI/T 68230 PIN ASSIGNMENT

PIN NAME	DIR	SIGNAL NAME	DESCRIPTION
PA0	I	CPUOK+00	ASSERTED WHEN CP0 IS IN RIGHT SLOT
PA1	I	VPNUM0+00	PIN LEAST SIGNIFICANT BIT
PA2	I	CHPRES-00	CACHE PRESENT
PA3	I	MONOVF-00	LATCHED MONITOR OVERFLOW
PA4	I	SYSFAL-00	SYSTEM FAIL (FROM VME)
PA5	I	ACFAIL-00	POWER DOWN (FROM VME)
PA6	I	MMUTP0+00	\ DEFINE THE CURRENT MMU TYPE
PA7	I	MMUTP1+00	/ NOTE (1)
PB0	I	DIAG1-00	RFU
PB1	I	DIAG2-00	RFU
PB2	O	CNTRG0+00	ENABLE/DISABLE CACHE
PB3	O	CNTRG1+00	NORMAL/DIAGNOSTIC MODE
PB4	I	STSRG0+00	CACHE STATUS REGISTER BIT 0
PB5	I	STSRG1+00	CACHE STATUS REGISTER BIT 1
PB6	I	STSRG2+00	CACHE STATUS REGISTER BIT 2
PB7	I	STSRG3+00	CACHE STATUS REGISTER BIT 3
PC0	O	MASKPT+00	PROFILE TIMER MASK BIT
PC1	O	RESEPT-00	PROFILE TIMER INTERRUPT RESET
PC2	N.A.	N.C.	TBD
PC3	O	TIMINT-00	TIMER INTERRUPT
PC4	N.A.	N.C.	TBD
PC5	N.A.	N.C.	TBD
PC6	N.A.	N.C.	TBD
PC7	N.A.	N.C.	TBD
H1	N.A.	N.C.	TBD
H2	N.A.	N.C.	TBD
H3	N.A.	N.C.	TBD
H4	N.A.	N.C.	TBD

TABLE A.12: MC68230 (PI/T) PIN ASSIGNMENT

NOTE

(1)

PA7	PA6	
0	0	CM1 (SUN-like)
0	1	RFU
1	0	RFU
1	1	RFU

A.2.11 OTHER I/O REGISTERS

MASK REGISTER

It is implemented by two 74F175 (quad D-type Flip-Flop) mapped im the range 0F400000 through 0F5FFFFF and selected by CSRMSK-00. By means of this register it is possible enable/disable selectively each VME INTERRUPT REQUEST lines and other internal generated/conditioned interrupts lines under SW control.
Access to MASK REGISTER requires one wait-state (240 ns).

RESET TO MONITOR OVERFLOW LATCH

This latch is setted by the monitor overflow signal passed by CACHE (MONOVF-00); when reference is made to 2 Mbyte range 0F800000 through 0F9FFFFF, that is CSMOVF-00 is asserted, latch is cleared.

A complete map of the local I/O space is shown in table A.13.

CS NAME	FROM	TO	DESCRIPTION
DCEPRM-00	0F 00 00 00	0F 1F FF FF	SELECT EPROMS' BANK
DCPIOT-00	0F 20 00 00	0F 3F FF FF	SELECT PI/T
CSRMSK-00	0F 40 00 00	0F 5F FF FF	SELECT MASK REGISTER
CSMOVF-00	0F 60 00 00	0F 7F FF FF	RESET OVFLW LATCH
TBD	0F 80 00 00	0F 9F FF FF	TBD
TBD	0F A0 00 00	0F BF FF FF	TBD
TBD	0F C0 00 00	0F DF FF FF	TBD
TBD	0F E0 00 00	0F FF FF FF	TBD

TABLE A.13: INTERNAL RESOURCES ADDRESS MAP

A.2.12 INTERRUPT HANDLER LOGIC

Table A.14 lists all interrupt request lines handled CP0 PWA,, showing priority level assigned to each one and, if available, the mask bit wich can disable it under software control.

INTERRUPT REQUEST	PRIORITY LEVEL	MASK BIT	INTERRUPT FROM
VINRQ1-00	1	MIRQ1-00	VME REQUEST LINE 1
VINRQ2-00	2	MIRQ2-00	VME REQUEST LINE 2
VINRQ3-00	3	MIRQ3-00	VME REQUEST LINE 3
VINRQ4-00	4	MIRQ4-00	VME REQUEST LINE 4
VINRQ5-00	5	MIRQ5-00	VME REQUEST LINE 5
TIMINT-00	6	MIRQ6-00	PI/T TIMER
MONOVF-00	7	---	LATCHED MONITOR OVERFLOW
SYSFAL-00	7	MSYFL-00	VME SYSFAL LINE ASSERTED
ACFAIL-00	7	MACFL-00	VME AC FAIL LINE ASSERTED
PROFIL-00	7	MASKPT+00 (*)	ASTABLE MULTIVIBRATOR

TABLE A.14: INTERRUPT REQUEST LINES HANDLED BY CP0

Mask bits are controlled by means of a write-only MASK REGISTER, mapped in the physical address range 0F400000 throught 0F5FFFFF; by handling this register, each interrupt request line (from VMEbus), can be individually enabled/disabled under software control.

When MPU executes a write operation referred to MASK REGISTER, the lines DATA24+00-DATA29+00 are latched on the MIRQ1-00-MIRQ6-00 bits, while DATA30+00-DATA31+00 are latched on MACFL-00 and MACFL-00 bits; this allows to disable all interrupt request lines whose mask bit is 0.

During set REGISTER MASK write operation, values of signal lines different by DATA24+00-DATA31+00 are not significant.

At power up of the system, all mask bits are resetted to 0.

Status of seven interrupt request lines (**) is sampled by a latch at any clock cycle and level of highest priority request latched is encoded to IPL0-00-IPL2-00 MPU lines.

When an interrupt acknowledge cycle occurs, it is recognized by dedicated decode, wich asserts:

IAKVME-00 for interrupts of level 1 throught 5 (thus enabling the interrupting VME board to answer with a vector)

AVEC-00 for interrupt of level 6 or 7 (thus enabling MPU to provide vector).

All interrupt request lines assigned to priority level 7 (except PROFIL-00), are also connected to port A of PI/T, wich allows to interrupt 7 service routine to identify wich one is interrupting.

(*) this mask bit (active HIGH) is setted on PI/T PC0 line

(**) a single line collects all interrupt request lines

SECTION B

CACHE MEMORY (CH0)

PRODUCT DESIGN DESCRIPTION

PREPARED BY: C. FIACCONI

REVIEWED BY: C. FIACCONI

INDEX OF SECTION B

		PAGE
B.1	CACHE MEMORY OVERVIEW	58
B.2	CACHE MEMORY FEATURES	59
B.3	CACHE MEMORY OPERATIONS	64
B.3.1	Hit Cycle	68
B.3.2	Rerun Cycle	71
B.3.3	Cache Replace Cycle	73
B.3.4	Update Indicator Cycle	77
B.3.5	Write Through Cycle	79
B.3.6	Cache Nop Cycle	81
B.3.7	Cache Wait Cycle	83
B.3.8	Monitor Check Cycle	85
B.3.9	Cache Flush Cycle	89
B.4	CACHE DIAGNOSTIC FEATURES	92
B.5	CACHE ARCHITECTURE BLOCK DIAGRAM	97
B.5.1	Cache Address Multiplexing	99
B.5.2	Cache Content-Addressable Memory	100
B.5.3	Cache Valid Bit Memory	103
B.5.4	Cache Comparator	107
B.5.5	Cache Hit/Miss Detection	110
B.5.6	Cache Hit/Miss Prediction	112
B.5.7	Cache Data Memory	114
B.5.8	VME Bus Monitor	117
B.5.9	Cache Activity Control and Timing Logic	122
B.6	CACHE INTERFACE DESCRIPTION	123
B.7	CACHE BOARD PHYSICAL DESCRIPTION	137

INTENTIONALLY LEFT BLANK

B.1 CACHE MEMORY OVERVIEW

Cache memories are small, high-speed buffer memories used in modern computer system to hold temporarily those portions of the contents of main memory which are (believed to be) currently in use. Since instructions and data in cache memories may be accessed in much less time than located in main memory, cache memories permit the execution rate of the machine to be substantially increased.

Independently how the cache memory is configured or what it is made of, its concept is based on "property-of-locality" probability principles, which experience has shown to have the following caracteristics.

The property-of-locality has two aspects, temporal and spatial. Over short periods of time, a program distributes its program references nonuniformly over its address space, and which portions of the address space are favored remain largely the same for long periods of time. This first property, called temporal locality, or locality by time, means that the information which will be in use in the near future is likely to be in use already. This type of behavior can be expected from program loops in which both data and instruction are reused. The second property, locality by space, means that portions of the address space which are in use generally consist of a fairly small number of individually contiguous segments of that address space. Locality by space, then, means that the loci of reference of the program in the near future are likely to be near the current loci of reference. This type of behavior can be expected from common knowledge of the programs: the instructions are mostly executed sequentially and the relative data (as variables, array) are usually stored together. Since the cache memory buffers segments of information that have been recently used, the property of locality implies that needed information is also likely to be found in the cache.

The main objects of a cache memory are: a) permit the system CPU to run at its maximum speed; b) reduce the processor's utilization of the available memory bandwidth, allowing other devices on the system bus to use the memory without interfering with the processor; c) assure consistency of data in cache memory and in main memory at any time, and d) maximize the probability of finding a memory reference's target in the cache (the hit ratio).

B.2 CACHE MEMORY FEATURES

The SGM2 cache memory represents an optional board of the system configuration. The SGM2 system connects up to two optional cache memory boards, each one coupled to its own CPU board.

Below, the SGM2 cache memory design choices are point out.

- No Wait State Cache Memory. The MPU68020 running up to 16.67 MHz can access the cache memory without wait states.
- 16 KByte Cache Size.
- Data/Instruction Storing Capability. The MPU68020 can refer the cache memory to obtain information like instruction (op-words) and/or data (operands) either supervisor or user type.
- Physical Cache Addressing. This means that the logical address generated by the MPU68020 must first be translated by the MMU to produce the physical address used to access the cache memory.
- Write-Through Main Memory Updating Technique. To avoid discrepancies, the cache must be updated when the MPU68020 writes the main memory. As each write operation is performed, data is immediately written to the main memory, but, if the cache contains that reference as well, the same data is also written into that cache location.
- FIFO Replacement Algorithm. When information is requested by the MPU68020 from main memory and the cache is full, the FIFO algorithm is employed to select the resources in the cache (the Sector Base Address Register, the Block Valid-Bit Memory, etc.) that must be replaced.
- 4 Byte Block Size. This parameter represents the number of bytes fetched from main memory after a cache miss.

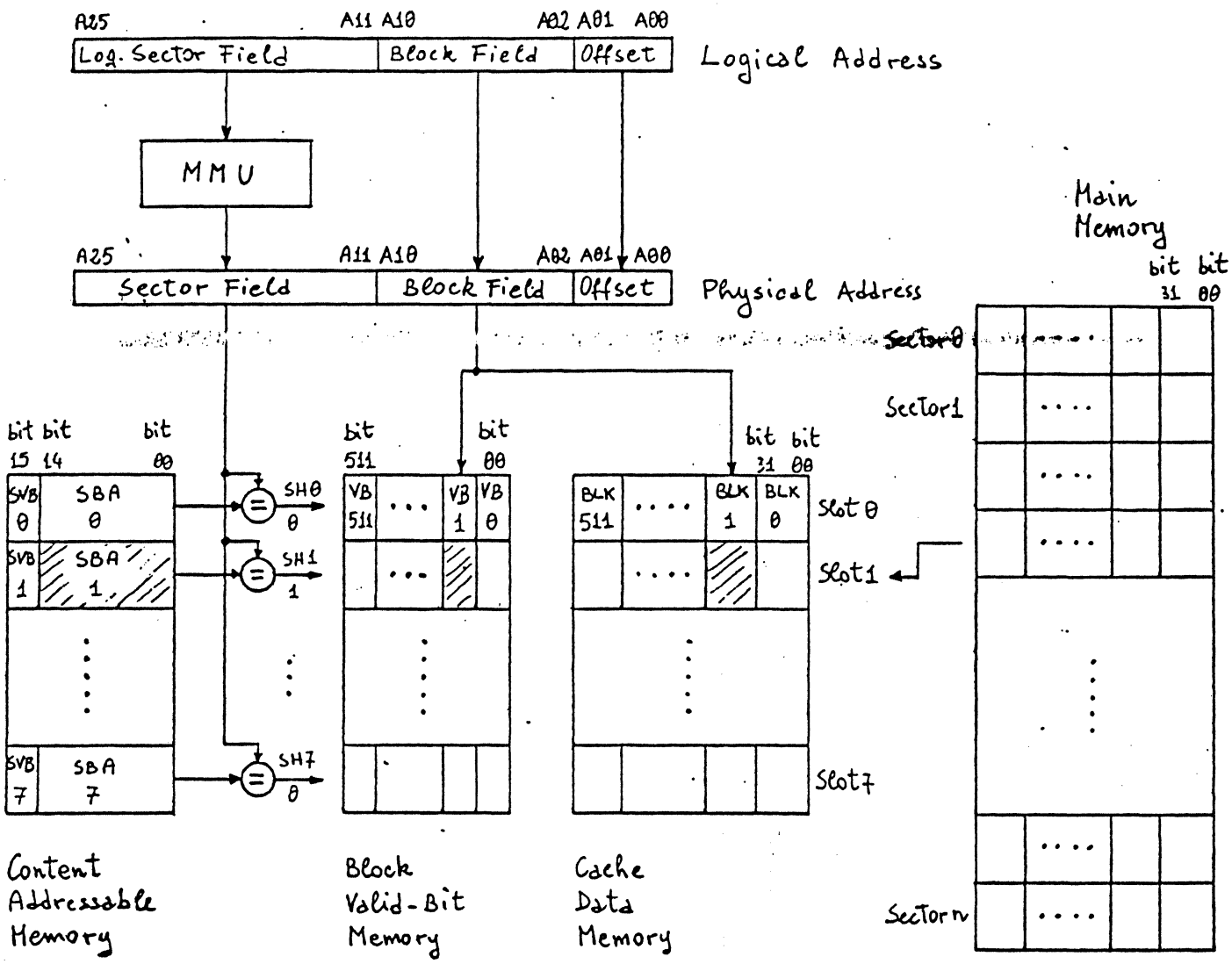
- Sector-Associative Cache. The technique employed in the SGM2 cache memory is called sector-associative. In this scheme, the main memory is partitioned into equal-size sectors (each one of 2k bytes), and the cache Data Memory is partitioned into 8 sector size slots (see Fig. B.2.1). Thus any main memory sector can occupy one of these 8 slots. However, since there are too main memory to fit each one into its own cache slots, the cache Control Logic employs a mapping procedure using 8-position fully-associative Content Addressable Memory (CAM) to identify those sectors in the cache.

  To find out if a sector is in the cache, the physical address received by the cache is broken into sector, block, and offset fields. If the sector field matches the contents of one of the CAM's positions, the requested sector is in that position's corresponding cache slot. A cache sector hit occurs when the sought sector is found in the cache; a cache sector miss occurs when the sector is not found.

  Because sectors are 2 kbyte large, filling a 16 kbyte cache one sector at a time is not at all pratical. For this reason, sectors are partitioned into 512 blocks made of 4 byte.

  To address a block, the logical address contains a block field and the offset field locates the exact byte sought. However, because not every block in a sector is necessarily in the cache, a set of valid bit is needed to say whether or not the addressed block is there.

  If during a read operatin, the valid bit for a requested block is not set, a miss occurs, and the main memory is accessed. In that case, the block (4 bytes) is transferred into the cache and the valid bit is set.



Note: SVB → Sector Valid Bit (1 = Not Valid, 0 = Valid)
SBA → Sector Base Address
VB → Block Valid Bit (1 = Valid, 0 = Not Valid)
SHn → Sector Hit n (1 = Sector Hit, 0 = Sector Miss)
Dashed Areas → Cache Information Used During an Hit Cycle

Fig. B.2.1 SGM2 Cache Memory General Stucture

- Cachable Resources. The SGM2 system is able to work in a biprocessor environment: the system structure is illustrated in Fig. B.2.2. The communications between the Central Processors and the I/O Processors are performed by the system bus that is the VME bus.

  The Central Processor architecture permits to the MPU68020 to work in cache memory and, at the same time, its main memory can be accessed by any master of the system bus. Then, each Central Processor can access (in read and write) the main memory of the other Central Processor and any I/O Processor is able to read/write any one of the two Main Memories. In a such system architecture each Central Processor is tightly coupled with its own Main Memory and Cache Memory. Each cache memory is able to cache only the information stored into its own coupled Main Memory: so all the accesses performed by the MPU68020 out of its Main memory space (that is, accesses on the internal resources, such as Eprom, Timer, etc., or accesses on the VME bus resources) are not cached.

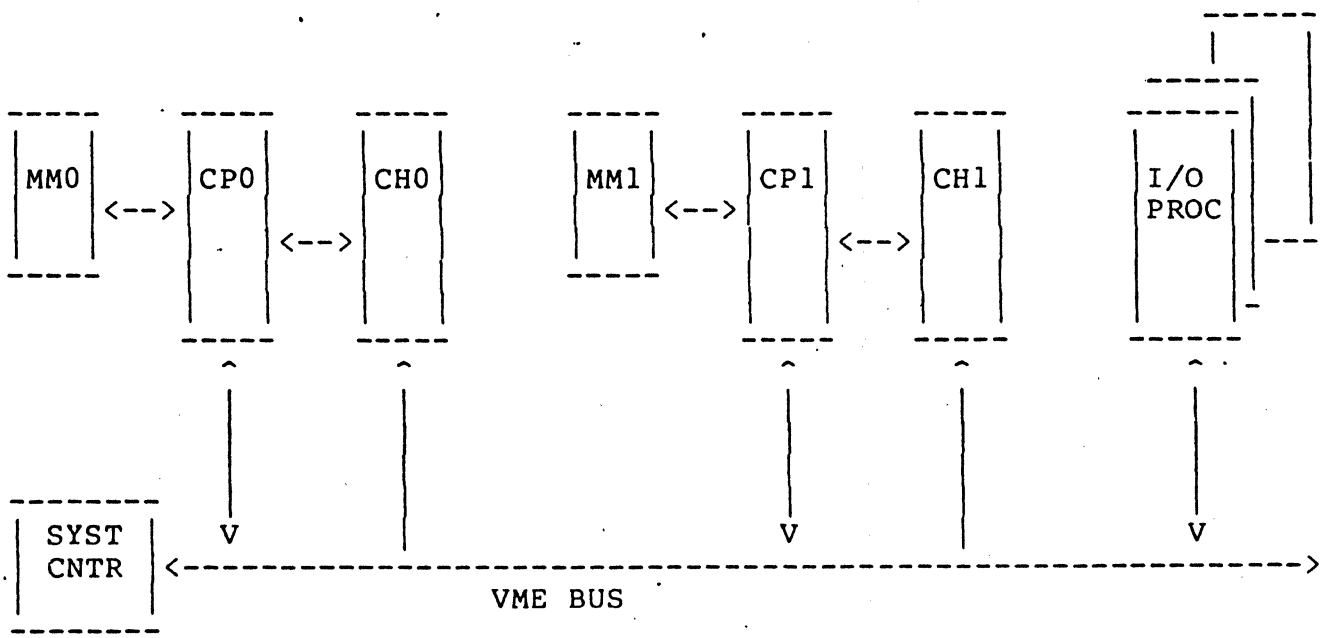


Fig. B.2.2 SGM2 System General Structure

- VME Monitor. To prevent the MPU68020 from using stale data, each cache memory is provided of a bus watching mechanism, the VME Monitor (see Fig. B.2.2) that ensure software transparent data consistency in multimaster systems. The VME Monitor logs all VME bus write activity related only to its coupled Main Memory and initiated by other VME bus masters. The VME bus addresses are latched and the Monitor requests a cache check cycle. The VME address is compared to those contained into the cache and if a match occurs at a valid block, that block is invalidated.

B.3 CACHE MEMORY OPERATIONS

This Section provides a general description of the cache memory operations, as well as detailed descriptions of all cache memory cycles.

The basic concept is connected to the behaviour of a typical program: the main characteristics are its sequentiality associated with its looping capability. These interesting program features are used by the cache controller in a probabilistic manner to handle the miss/hit indicator. A program is executed instruction after instruction and each instruction is fetched and executed by a certain number of bus cycles: so, a qualifier in term of hit/miss can be assigned to each bus cycle. In particular, the information associated with the hit/miss indicator is referred to the previous bus cycle and it is used in the current bus cycle.
So, if in a generic bus cycle the state of the indicator is miss this implies that in the previous bus cycle the searched information was not into the cache memory and, for the program characteristics above illustrated, with a very high probability also the present searched information will not be in cache. In this case a Main Memory cycle is started and contemporarily is checked the presence of the searched information in the cache: if present, the state of the indicator will be changed in hit, if absent, the state of the indicator remains miss and the read information will be also stored in cache. On the other hand, if in a generic bus cycle the indicator is hit this signifies that in the previous bus cycle the searched information was provided by the cache memory and, probably, also the present information is in cache. So, in this cycle type the cache provides the DSACK0,1 signals without to know really if the searched information is in cache: this permits to execute the present bus cycle without wait-states. Then, if a real hit is found out, the cache provides the information to the MPU68020 and the hit/miss indicator is unchanged: when the information is not in cache, HALT and BERR signals will be asserted (the MPU68020 rerun the same cycle) and the indicator assumes the miss state. Because the indicator is put in miss state, when the MPU68020 reruns the cycle, the Main Memory will provide the searched information. It must be pointed that the rerun cycle is payed only when after a series of hit cycles in cache memory occurs the miss condition. During a sequence of miss cycles (Main Memory cycles) the rerun mechanism does not work. In reality the previous hit/miss indicator is splitted into the following hit/miss indicators: Supervisor Program, Supervisor Data, User Program and User Data. Each indicator is used by the cache controller to menage the related current bus cycle on the base of the behaviour detected in the previous cycle.

The above cache operations can be summarized with the control-state diagram showed in Fig. B.3.1. Each state is entered under specific conditions, performs specific cache activities, and decides which is the next state to reach. The cache memory control-state diagram is made up of 13 states: when the cache memory has no activities running, then it is in the CACHE IDLE state. It must be pointed out that a cache memory cycle is constituted by some cache states. The following 9 cache cycle are illustrated in Fig. B.3.1:

- HIT cycle,
- RERUN cycle,
- CACHE REPLACE cycle,
- UPDATE INDICATOR cycle,
- WRITE THROUGHT cycle,
- CACHE NOP cycle,
- CACHE WAIT cycle,
- MONITOR CHECK cycle,
- CACHE FLUSH cycle.

The Table B.3.1 summarizes in detail the conditions and the operations relative to each cache memory cycles.

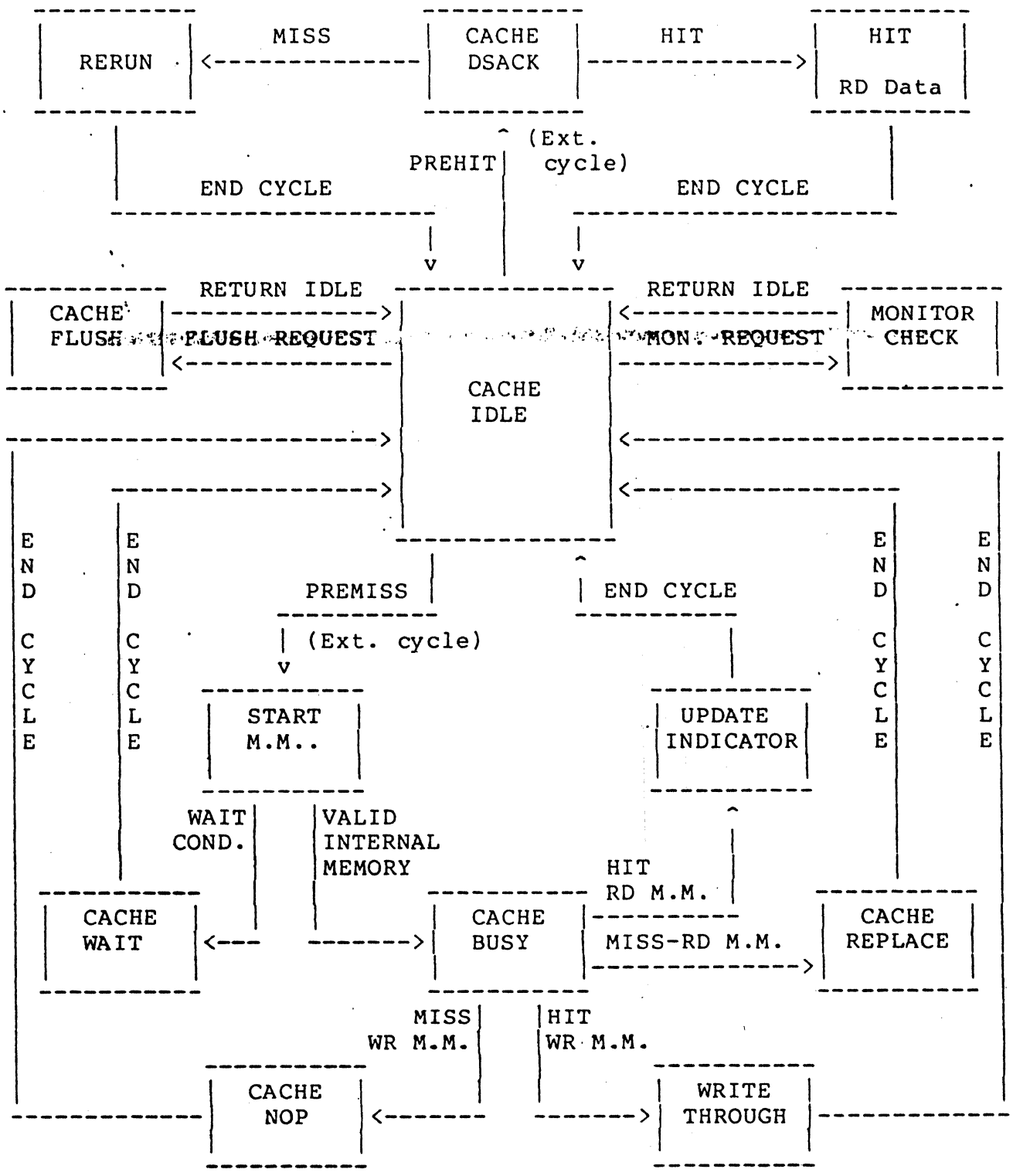


Fig. B.3.1 Cache Memory Control-State Diagram

Table B.3.1 Cache Cycle Operations Summary

Cache Cycle Type	CACHE /CT.	ASACT (Asy. Serv. ACT.)	RD/WT	RTC	DOSAL MM space	PTE MMT	DOSAP Int. MM	PAS	SCT HIT	BLK MMT	Operations
CACHE DISABLE	0	0	x	x	x	x	x	x	x	x	No operation performed
IDLE STATE	1	0	x	x	1	x	x	x	x	x	No operation performed but, if an ASYNC. REQ. occurs, set ASACT (ASYNC. SERVICE ACTIVE)
MONITOR or FLUSH	1	1	x	x	x	x	x	x	x	x	Monitor or Flush cycle started only after the END CYCLE detection (LAS negation by MPU68020).
WAIT	1	0/1	1	1	0	0	1	x	x	x	A S /=1 WAIT \ A / END \ ENABLE If pending set ASACT C \ / ,START , after END CYCLE T \=0 —> / MM
	1	0/1	1	0	0	x	x	x	x	x	
	1	0/1	0	x	0	x	1	x	x	x	
REPLACE	1	0/1	1	1	0	0	0	0/1	[0 1]	[x 0]	A S /=1 WAIT \ A / END \ ENABLE C \ / ,START , T \=0 —> / MM ; SBB PHHE AIHR STTR -+-+ ; 1xxx —> ; 00x1 ->ABCDEF ; 00x0 ->ABCDEF ; 0101 ->DE ; 0100 ->EF ; If pending set ASACT ,after END CYCLE
CHANGE INDIC.	1	1/0	1	1	0	0	0	0/1	1	1	A S /=1 WAIT \ A / END \ ENABLE C \ / ,START , T \=0 —> / MM ; P A S - ; 0 —> ; 1 set specific PREHIT Ind. ; If pending set ASACT ,after END CYCLE
RERUN	1	0/1	1	1	0	1	1	x	x	x	A S /=1 WAIT \ A / END \ set C \ / ,CACHE , T \=0 —> / DSACK ; set BERR and, HALT ; reset specific PREHIT indicat. ; If pending set ASACT after END CYCLE
	1	0/1	1	1	0	1	0	x	[0 1]	[x 0]	
READ HIT	1	0/1	1	1	0	1	0	x	1	1	A S /=1 WAIT \ A / END \ set C \ / ,CACHE , T \=0 —> / DSACK ; need CACHE, DATA ; set HALT if CHECK ERROR ; If pending set ASACT , after END CYCLE
NOP	1	0/1	0	x	0	x	0	x	[0 1]	[x 0]	A S /=1 WAIT \ A / END \ ENABLE If pending set ASACT C \ / ,START , after END CYCLE T \=0 —> / MM
WRITE THROUGH	1	0/1	0	x	0	x	0	0/1	1	1	A S /=1 WAIT \ A / END \ ENABLE C \ / ,START , T \=0 —> / MM ; T PHAST -+ ; 1x —> ; 01 write DATA CACHE as specif. (B,M,3D,LM) ; If pending set ASACT ,after END CYCLE

ASYNCHR. SERVICE ACTIVE can be set for:
- RESET.
- DISABLE CACHE.
- MONITOR OVERFLOW.
- MONITOR CYCLE.

A = Increment FIFO pointer (SHIFT RIGHT)
B = Store selected SECTOR REGISTER
C = Flush selected BLOCK MEMORY
D = Store VALID BIT (1) on selected block memory
E = Store INVALID BIT (0) on selected block memory
F = Store BLOCK (Long Word) on CACHE DATA MEMORY

B.3.1 HIT CYCLE

The cache HIT cycle is made up of CACHE IDLE, CACHE DSACK, and HIT states (see Fig. B.3.1.1):

The conditions under which the cache leaves the CACHE IDLE state and jumps to the CACHE DSACK state are the following:

d1. the cache is active (that is, CNTRG0+00 interface signal is driven to ZERO level by the MPU68020);
d2. the MPU68020 has started an external read cycle;
d3. the RMC signal is not asserted;
d4. the initiated external cycle is performed towards the Main Memory space (the coupled Main Memory or that one of the other CP) and it belongs to one of the following privileges: user program/data or supervisor program/data;
d5. the previous hit/miss indicator (selected by the Function Code signals) is in the HIT status.

In the CACHE DSACK state the assertion of DSACK0 and DSACK1 signals is performed without to know the real cache hit/miss status relative to the sought current block.

Then the passing to the HIT state is determined by the following events:

h1. the current Main Memory access is directed to the coupled memory;
h2. sector hit present, that is one of the eight sector registers matches with the current physical address;
h3. block hit present, that is the sought block is valid.

It must be added that the condition "PAS (Physical Address Strobe) asserted" is not necessary because if the MMU finds any error relative to the current cycle, it actives the BERR and/or HALT signals: in this case a delayed BERR or a RERUN condition is detected by the MPU68020 that ignores the current data provided by the cache.

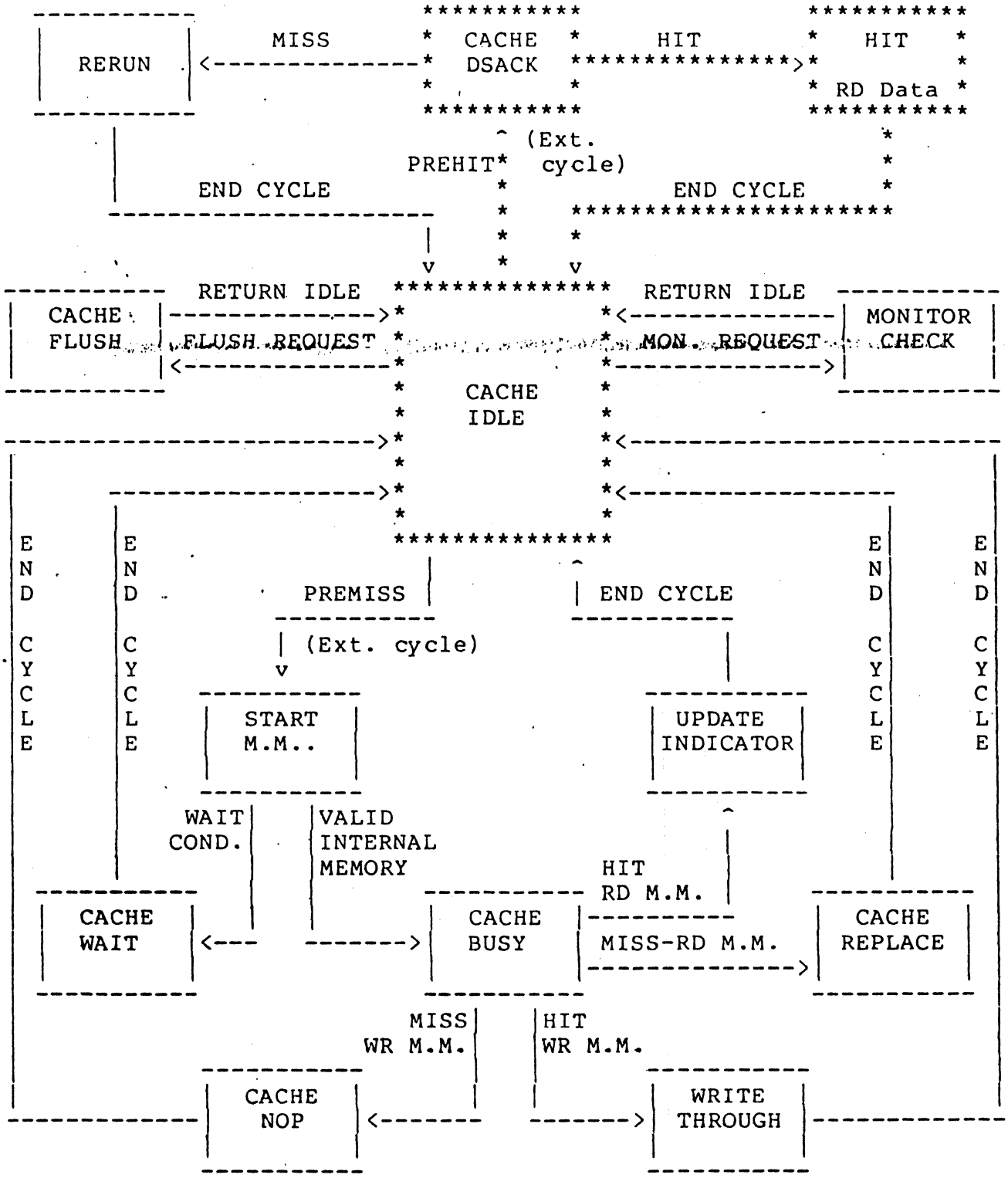


Fig. B.3.1.1 Cache Hit Cycle Diagram

The activities performed in the HIT state are the following:

a1. recovery of the block (4 bytes) from the cache Data Memory and put it on the MPU68020 data bus;
a2. check of the four parity error bits and assert the HALT signals if at least one is active: the coupled MPU68020 is halted and only a system reset or a power-on can restart it. This choise is determined by the impossibility to detect the presence of a parity error before to deliver the data to the MPU68020.

After the execution of these operations and when the "end cycle" condition is detected (LAS signal negated by the MPU68020) the return to the IDLE state is performed. The detection of the "end cycle" condition permits to set the ASYNCRONOUS SERVICE ACTIVE flip-flop if a general flush request or a monitor request is pending. In this case a CACHE FLUSH cycle or a MONITOR CHECK cycle will be started immediately and completely executed before to take into account an eventually new Main Memory cycle started by the MPU68020.

B.3.2 RERUN CYCLE

The involved states in the RERUN cycle are: CACHE IDLE, CACHE DSACK and RERUN (see Fig. B.3.2.1).

The passing from the CACHE IDLE state to the CACHE DSACK state is determined by the same conditions described for the HIT cycle (see points d1. through d5. of the Section B.3.1) and the DSACK0 and DSACK1 signals are asserted in the same manner.

The RERUN state can be reached when:

r1. the current Main Memory access is performed towards the memory of the other CP;

or if the following conditions are verified:

r2a. the coupled memory is accessed;
r2b. the sector miss or the block miss condition is asserted.

For both the above cases, in the RERUN state, the BERR and HALT signals are activated because it is not possible to provide to the MPU68020 the searched information in the time requested. Infact, the microprocessor ends the cycle a clock after the sampling of the DSACK0 and DSACK1 signals. Besides, the hit/miss indicator relative to the current cycle is set in the MISS state to perform the next rerun cycle directly in Main Memory. Then, the return to the IDLE state is performed after the detection of the "end cycle" condition (LAS signal negated by the MPU68020).

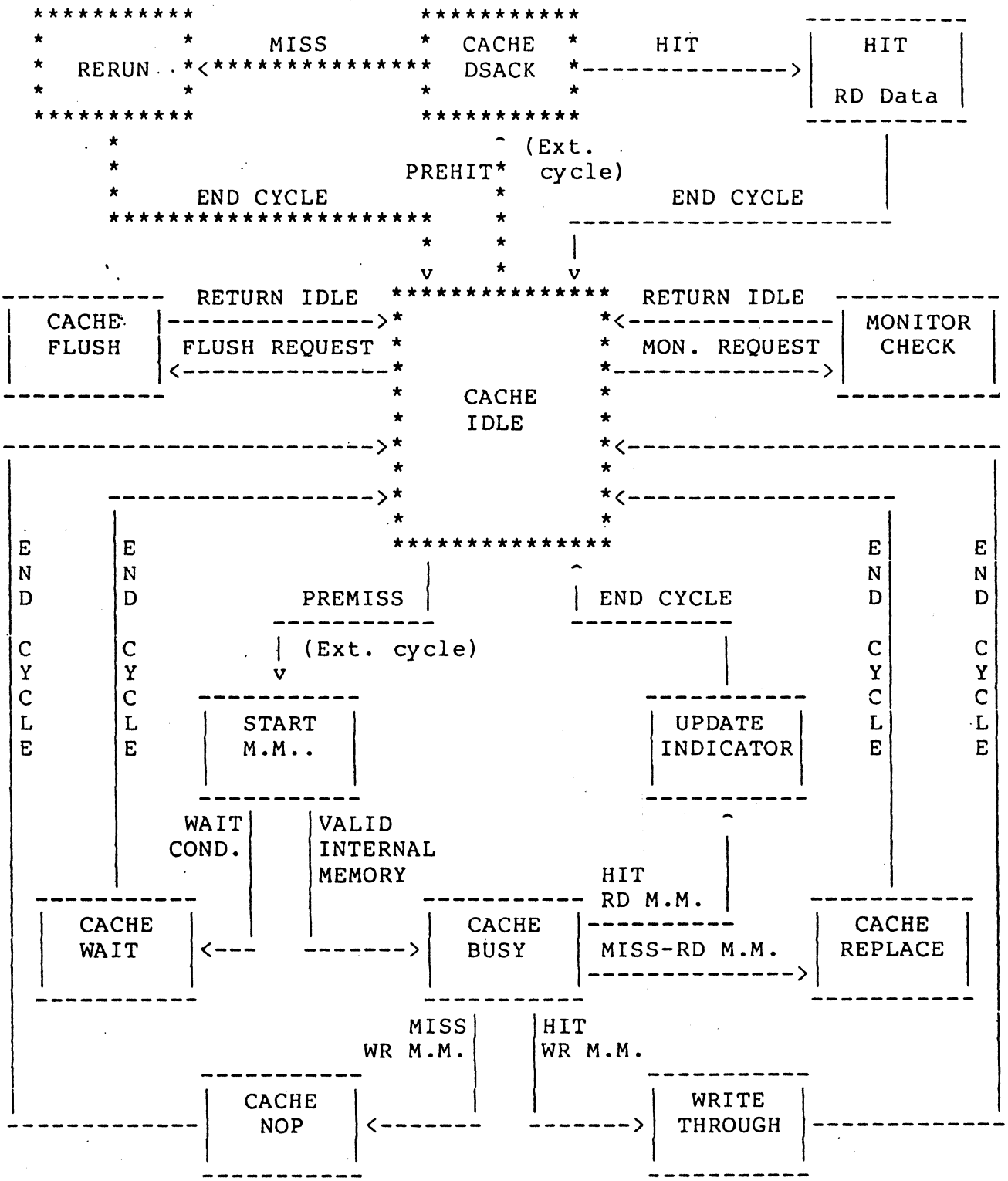


Fig. B.3.2.1 Rerun Cycle Diagram

B.3.3 CACHE REPLACE CYCLE

The CACHE REPLACE cycle is made up of the following states: CACHE IDLE, START Main Memory, CACHE BUSY, and CACHE REPLACE (see Fig. B.3.3.1).

This cycle begins from the CACHE IDLE state and the START Main Memory state is reached under the following conditions:

s1. the MPU68020 has started an external cycle (that is, LAS signal is asserted);
s2. the external cycle is performed towards the Main Memory space that means towards the coupled Main Memory or that one of the other CP: in any case only user program/data or supervisor program/data are taken into account;
s3. the cache is active,

and when almost one of the following events is verified:

s4a. RMC signal asserted (that is, the MPU68020 is executing a TAS or CAS instruction);
s4b. the previous hit/miss indicator is in the MISS condition.

In the START Main Memory state the cache enables the starting of the coupled Main Memory without to know which one of the two Main Memories will be accessed in this cycle. The physical address provided by the MMU decides which memory must handle the current cycle. If the memory of the other CP must execute the present cycle a VME bus cycle must be requested by the coupled CP: in this case the started coupled memory will close the cycle without provide the data.

It must be emphasized that the cache board controls the "Start Main Memory Mechanism" only when it is active; on the other hand, when the cache is present but "no active", the start of the coupled Main Memory is performed by the CP board and pratically follows the LAS signal assertion.

The CACHE BUSY state is reached when the following events are verified:

b1. the PAS signal is asserted;
b2. the MPU68020 has requested a cycle towards the coupled Main Memory (the signal DC64P-00 is active).

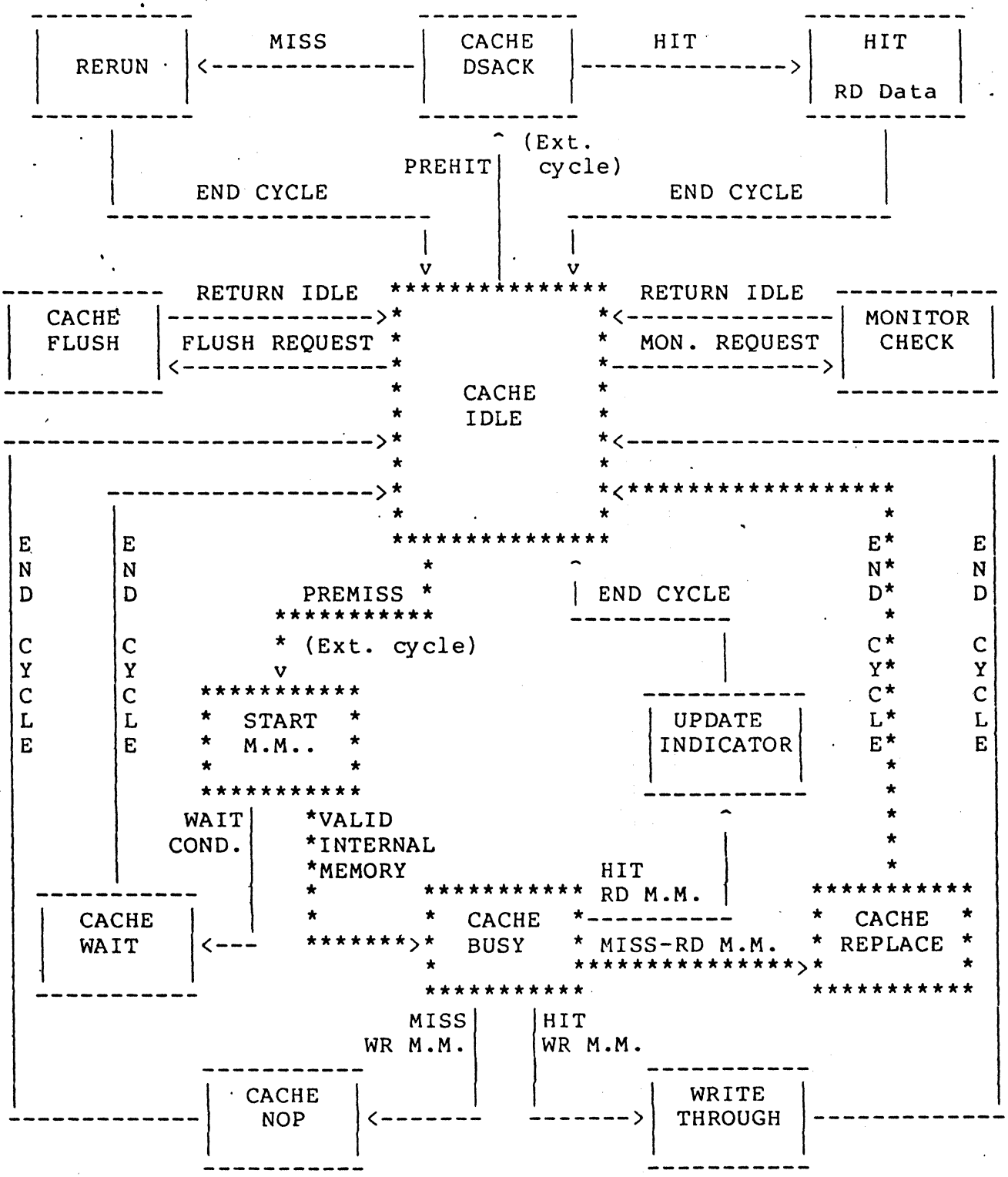


Fig. B.3.3.1 Cache Replace Cycle Diagram

In the CACHE BUSY state no specific action is performed: it depicts only an intermediate state between the START Main Memory state and the following four states: CACHE REPLACE, UPDATE INDICATOR, WRITE THROUGHT, and CACHE NOP.

At this point the passing to the CACHE REPLACE state is determined by the following conditions:

p1. the cycle started by the MPU68020 is a READ cycle;
p2. the RMC signal is not asserted;
p3. the selected previous hit/miss indicator is in the MISS state;
p4. the sector miss or the block miss condition is asserted.

When a sector miss occours the following activities are performed:

sm1. the FIFO Replacement Algorithm (see Section B.2) is pratically implemented by a shift register used to point the sector resources to be updated: so, first of all, the shift register is clocked one time, that is, a shift right operation is performed; at this point, the output at "level one" indicates which sector resources must be replaced;
sm2. the CAM sector register pointed by the shift register is loaded with the current Sector Base Address, that is, with the physical address 11 through 25; at the same time the validation of the relative Sector Valid Bit is performed;
sm3. the section of the Block Valid-Bit Memory actually pointed by the shift register is completely invalidate because now it is referred to a new Main Memory sector;
sm4. the block valid bit (addressed by the current block address field) of the Block Valid-Bit Memory actually pointed by the shift register is set (to logic ONE) if the Main Memory cycle is correctly terminated (BERR signal not asserted): otherwise it remains in the not valid status (logic ZERO);

sm5. the Cache Data Memory block (four bytes) addressed by the current block address field of the "actual pointed slot" is replaced with the long word (with the relative four computed check bits) provided by the Main Memory. The Cache Data Memory slot is pointed by a combinatory network that encodes the eight match outputs of the sector comparators into three signals that represents the three most significant Cache Data Memory addresses. The Sector Comparators and the Slot Encoding Logic are physically connected after the Sector Registers (CAM), are always active, and always work on the basis of the Sector Registers contents and of the actual sector address field value.

If only a block miss occurs, that is, the current sector field matches with one of the eigth sector addresses stored in the CAM, the following operations are executed:

bm1. storing of the sector number that has provided the sector hit condition in the Hit Sector Number Register which will point the sector resources to replace;
bm2. the block valid bit (addressed by the current block address field) of the Block Valid-Bit Memory actually pointed by the Hit Sector Number Register is set (to logic ONE) if the Main Memory cycle is correctly terminated (BERR signal not asserted): otherwise it remains in the not valid status (logic ZERO);
bm3. same operations described in point sm5.

When the "end cycle" condition is detected (LAS signal negated by the MPU68020) the IDLE state is still reached.

B.3.4 UPDATE INDICATOR CYCLE

The UPDATE INDICATOR cycle is composed by the following states: CACHE IDLE, START Main Memory, CACHE BUSY, and UPDATE INDICATOR (see Fig. B.3.4.1).

The START Main Memory and CACHE BUSY states are reached under the same conditions and perform the same operations explained for the CACHE REPLACE cycle (see Section B.3.3).

The UPDATE INDICATOR state is reached when:

u1. the cycle started by the MPU68020 is a READ cycle;
u2. the RMC signal is not asserted;
u3. the selected previous hit/miss indicator is in the MISS state;
u4. the total hit condition (sector hit and block hit) is asserted.

In this state the selected previous hit/miss indicator is positioned in the HIT status: the sought data is still delivered to the MPU68020 by the Main Memory.

At this point, the "end cycle" condition is waited (LAS signal negated by the MPU68020) and then the IDLE state is reached.

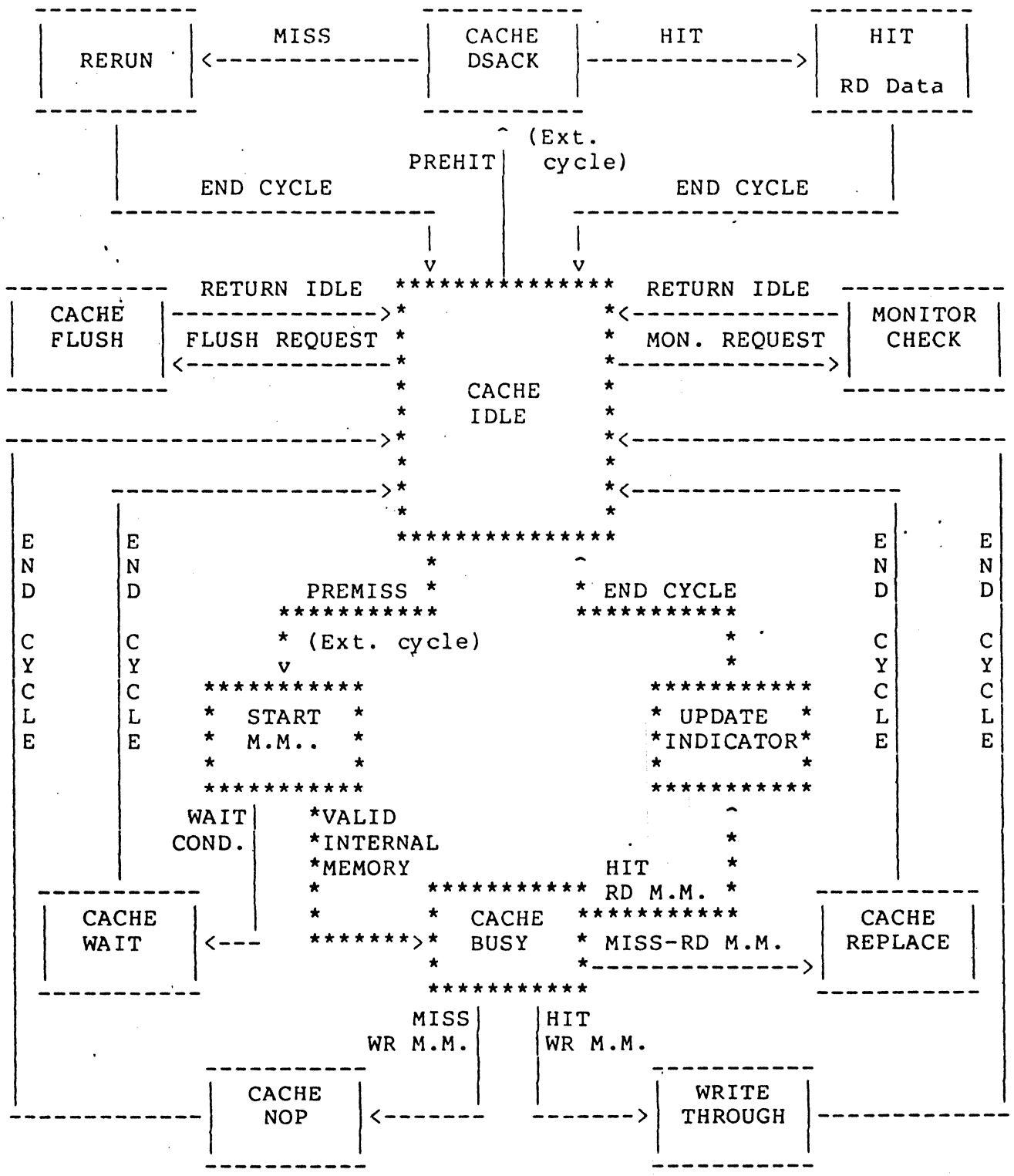


Fig. B.3.4.1 Update Indicator Cycle Diagram

B.3.5 WRITE THROUGH CYCLE

The WRITE THROUGH cycle is made up of the following states: CACHE IDLE, START Main Memory, CACHE BUSY, and WRITE THROUGH (see Fig. B.3.5.1).

The START Main Memory and CACHE BUSY states are reached under the same conditions and perform the same operations explained in the CACHE REPLACE cycle (see Section B.3.3).

The WRITE THROUGH state is reached when the following events are verified:

w1. the cycle started by the MPU68020 is a WRITE cycle;
w2. the total hit condition (sector hit and block hit) is asserted.

In this state, the current pointed block is updated with the data provided by the MPU68020. The block and related check bits updating can be done at one or two or three or four bytes level according to the status of the PADD00, PADD01, SIZE0, and SIZE1 signals driven by the MPU68020.

The IDLE state is reached when the "end cycle" condition (LAS signal negated by the MPU68020) is detected.

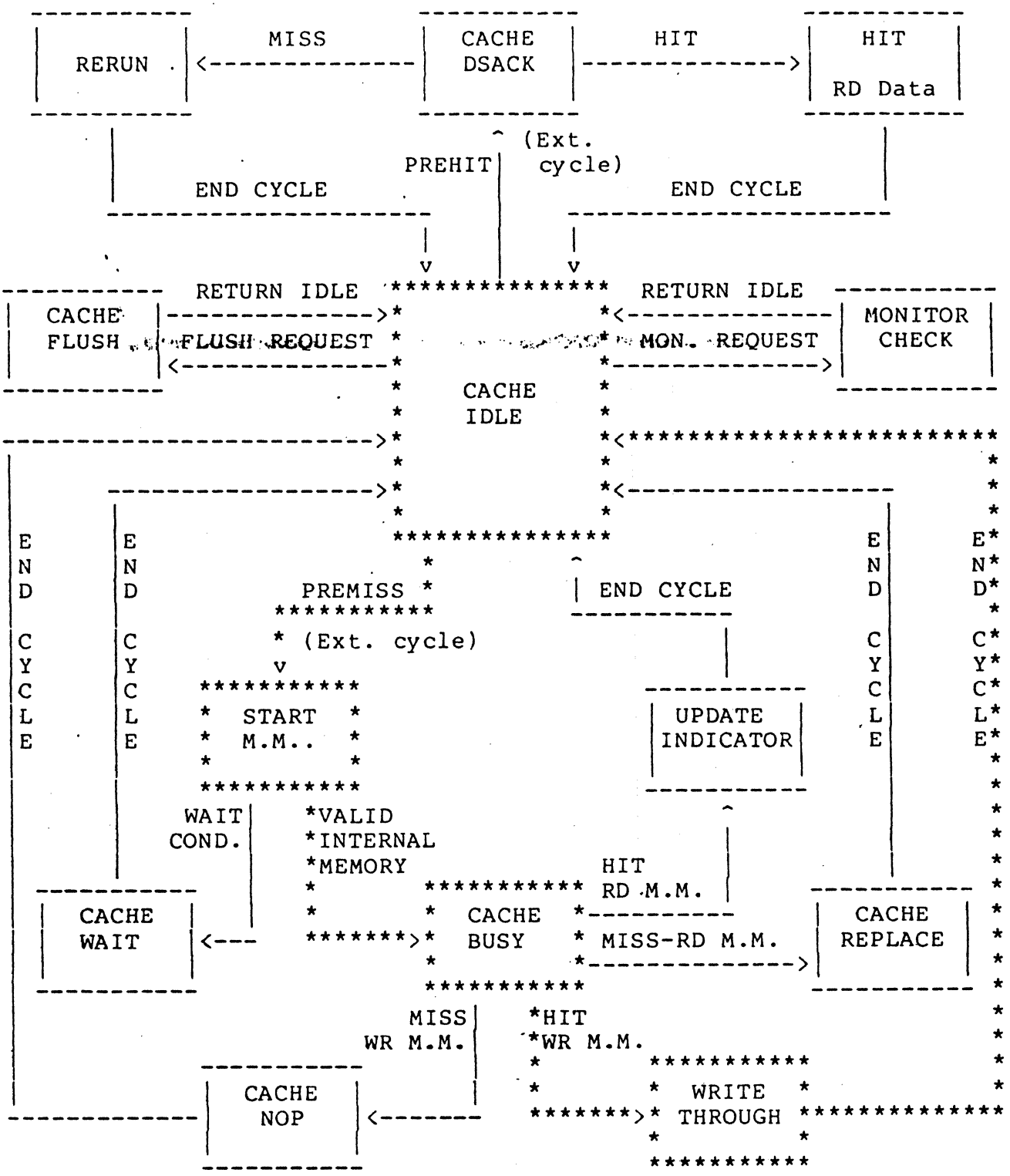


Fig. B.3.5.1 Write Through Cycle Diagram

B.3.6 CACHE NOP CYCLE

The CACHE NOP cycle is made up of the following states: CACHE IDLE, START Main Memory, CACHE BUSY, and CACHE NOP (see Fig. B.3.6.1).

The START Main Memory and CACHE BUSY states are reached under the same conditions and perform the same operations explained for the CACHE REPLACE cycle (see Section B.3.3).

The CACHE NOP state is reached when the following conditions occur:

n1. the cycle started by the MPU68020 is a WRITE cycle;
n2. the sector miss or the block miss condition is asserted.

In the CACHE NOP state no action is performed because the MPU68020 is updating an information not present in cache.

Also in this cycle, the "end cycle" condition is waited (LAS signal negated by the MPU68020) and then the IDLE state is reached.

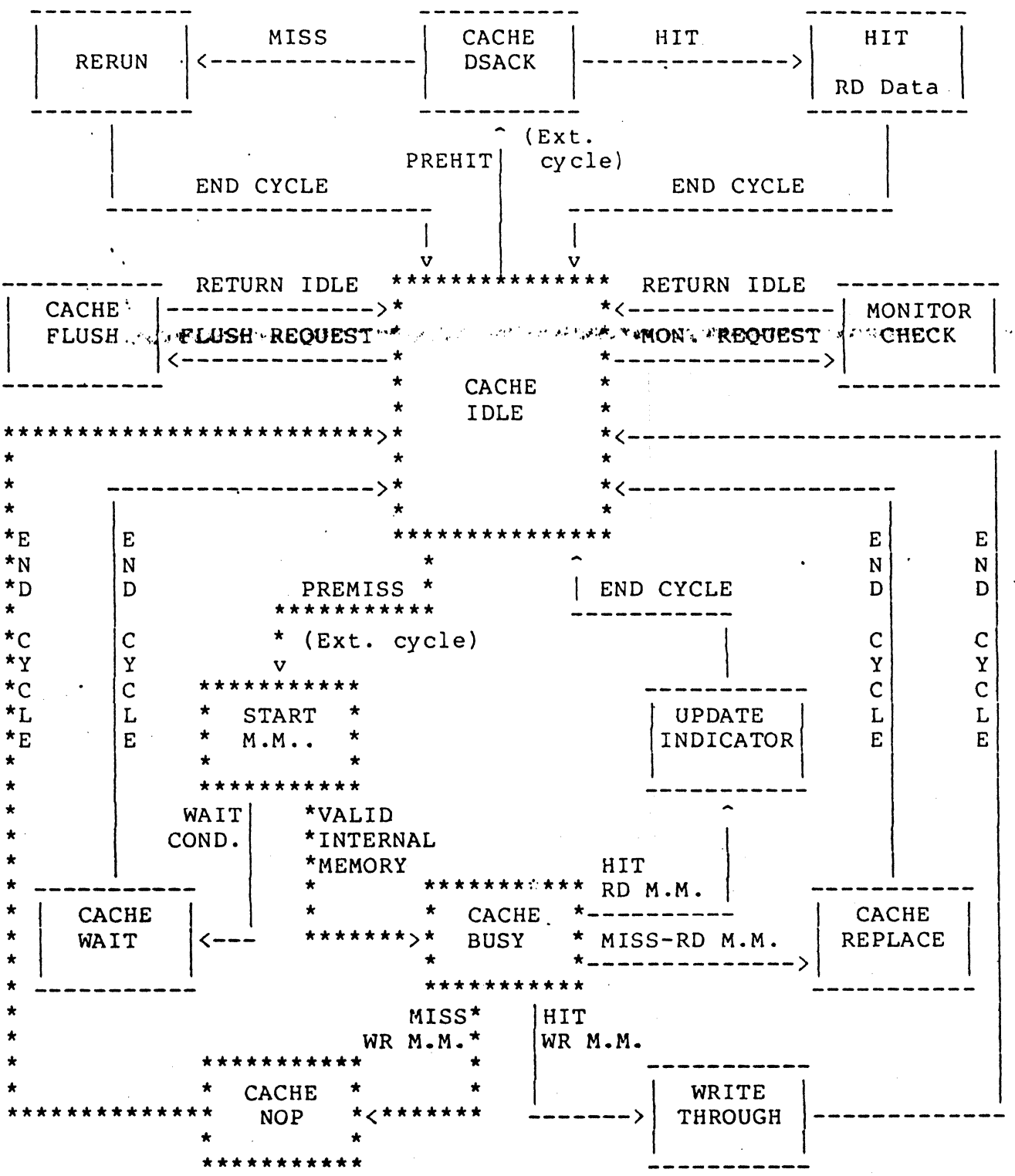


Fig. B.3.6.1 Cache Nop Cycle Diagram

B.3.7 CACHE WAIT CYCLE

The CACHE WAIT cycle is composed by CACHE IDLE, START Main Memory, and CACHE WAIT states (see Fig. B.3.7.1).

The START Main Memory state is reached under the same conditions and performs the same operations described for the CACHE REPLACE cycle (see Section B.3.3).

Then the passing to the CACHE WAIT state occurs when at least one of the following events is verified:

w1. the PAS signal is not asserted (that is, the present cycle is not valid and probably the MMU will assert the BERR or BERR and HALT signals);
w2. the MPU68020 has requested a cycle towards the Main Memory of the other CP (the signal DC64P-00 is not asserted);
w3. the RMC signal is asserted and the cycle started by the MPU68020 is a READ cycle.

The CACHE WAIT state does not perform operations in the cache resources and the cache waits for the "end cycle" condition (that is, LAS signal negated by the MPU68020) that permits the return to the IDLE state.

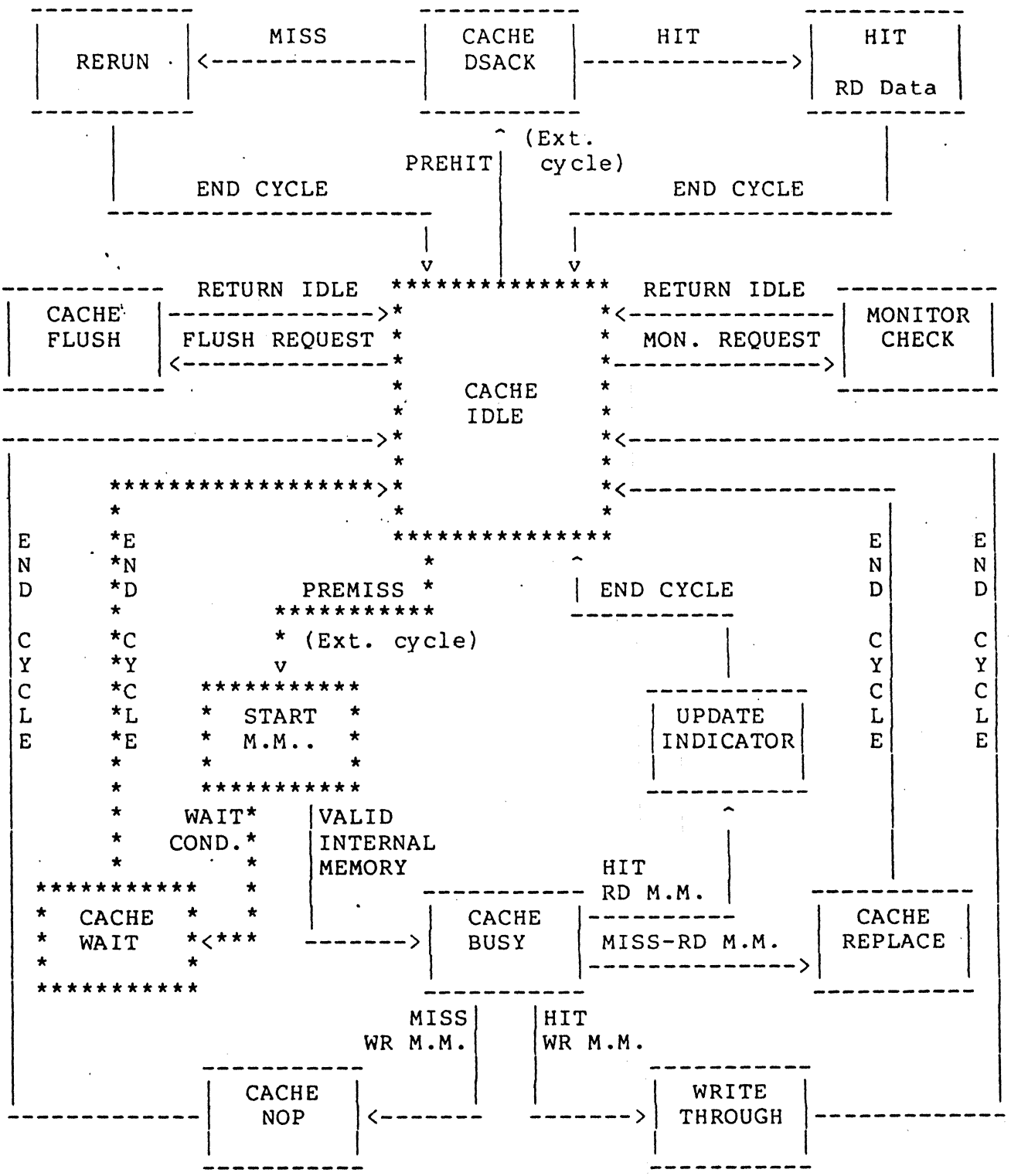


Fig. B.3.7.1 Cache Wait Cycle Diagram

B.3.8 MONITOR CHECK CYCLE

This cycle is made up only of two states: CACHE IDLE and MONITOR CHECK (see Fig. B.3.8.1).

The MONITOR CHECK state is reached when a MONITOR request is present and only from the CACHE IDLE state. This means that the ASYNCHRONOUS SERVICE ACTIVE flip-flop is set, that is, the MONITOR request is immediately serviced if the MPU68020 is not accessing the Main Memory Space. On the other hand, when it is addressing the Main Memory Space the MONITOR request is pending until the detection of the "end cycle" condition and only now the ASYNCHRONOUS SERVICE ACTIVE flip-flop is set.

The MONITOR request is asserted when (see also Table B.5.8.1):

- m1. the cache is active;
- m2. the current VME Bus master is performing a WRITE cycle;
- m3. the write VME cycle is directed towards the coupled Main Memory;
- m4. the Monitor FIFO is not full;
- m5. the MONITOR request is not already asserted that means a MONITOR request is still pending and only when it will be serviced, the actual request will be newly asserted.

If the m4 condition is not met, the MONITOR OVERFLOW condition is set, then a CACHE FLUSH cycle will be activated (see Section B.3.9) and an interrupt of level 7 is provided to the MPU68020; instead of, when it is verified the current VME address is stored into the Monitor FIFO at the receiving of the VDTACK-00 signal (that means the actual VME SLAVE has not found errors of any type executing the write operation).

In the MONITOR CHECK state the following activities are performed:

- mc1. changing the selection input, the SECTOR and BLOCK ADDRESS MULTIPLEXERS drive the MONITOR addresses;
- mc2. the Hit Sector Number Register is clocked to store the sector number that eventually has matched the sector address field of the current MONITOR address;

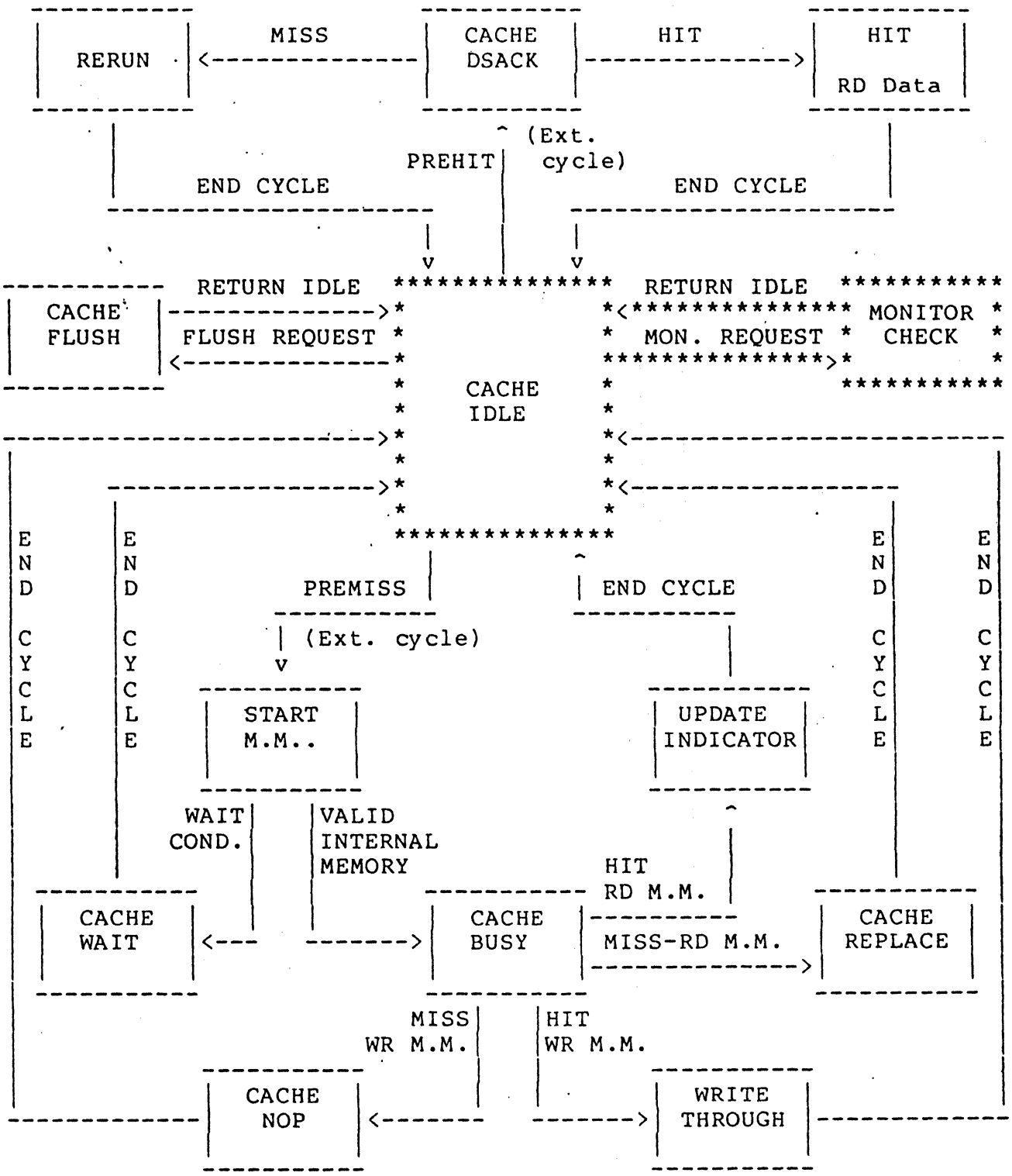


Fig. B.3.8.1 Monitor Check Cycle Diagram

mc3. if the total hit (sector hit and block hit) condition is verified the block valid-bit addressed by the current MONITOR block address field of the Block Memory pointed by the Hit Sector Number Register is reset to zero; in this manner the corrisponding data block in the Cache Data Memory is not valid. If a miss condition occurs no block valid-bit is updated;
mc4. reset the MONITOR request;
mc5. reset the content of the Hit Sector Number Register;
mc6. change the selection of the address multiplexers that now return to drive the MPU68020 addresses;
mc7. reset the ASYNCHRONOUS SERVICE ACTIVE flip-flop.

At this point the return to the IDLE state is performed.

The time spent by the cache to perform a MONITOR CHECK cycle is three machine clocks (180ns in the case of 16.67 MHz clock frequency).

During this time, and precisely in any of the previous three machine clocks, the MPU68020 can start an external cycle towards the Main Memory Space: in this case the Cache Controller stops the MMU operations and delays the Main Memory starting or the MPU68020 DSACK0-1 signals asserting until the MONITOR CHECK cycle is terminated (at maximum for three machine clocks). It must be added that, if the MPU68020 performs an external cycle towards a local resources (EPROM, TIMER, etc.) this cycle and the MONITOR CHECK cycle run in parallel.

The Table B.3.8.1 summarizes the activities of the cache ASYNCHRONOUS SERVICES (that is, the MONITOR CHECK cycle and the CACHE FLUSH cycle) originated by the MONITOR request and the GENERAL FLUSH request.

Table B.3.8.1 Asynchronous Service Activities

GENERAL FLUSH REQUEST	MONITOR REQUEST	TOTAL HIT	Operations
0	0	X	No Action
0	1	0	Hit Network activation, load Hit Sector Number Reg., reset Monitor req., reset Hit Sector Number Reg.
0	1	1	Hit Network activation, load Hit Sector Number Reg., reset Block Valid-Bit to the specified address, reset Monitor req., reset Hit Sector Number Reg.
1	X	X	Reset Sector Valid Bits, reset Shift Reg., reset Block Valid-Bit Memory, reset Pre-Hit indicators, reset General Flush req., reset Monitor Logic (also an eventually Monitor req. pending), reset Block and Data parity error registers.

X --> don't care

GENERAL FLUSH REQUEST:
- Power On Reset;
- Operator Panel Reset;
- Disable Cache;
- Monitor Overflow.

B.3.9 CACHE FLUSH CYCLE

The CACHE IDLE and the CACHE FLUSH are the two states of the CACHE FLUSH cycle (see Fig. B.3.9.1).

The CACHE FLUSH state is reached when a GENERAL FLUSH request is present and only from the CACHE IDLE state. The ASYNCHRONOUS SERVICE ACTIVE flip-flop is set and the GENERAL FLUSH request is immediately serviced if the MPU68020 is not accessing the Main Memory Space. Instead of, if a Main Memory Space access is running, the GENERAL FLUSH request is pending until the "end cycle" condition is detected and only now the ASYNCHRONOUS SERVICE ACTIVE flip-flop is set.

The GENERAL FLUSH request arises when one of the following conditions is detected:

f1. power-on RESET;
f2. push-button RESET;
f3. cache disactivation (CNTRG0+00 signal goes high);
f4. VME Monitor FIFO overflow.

It must be pointed that if the MONITOR request and the GENERAL FLUSH request occur at the same time the GENERAL FLUSH request has the highest priority.

The operations performed in the CACHE FLUSH cycle are the following (see also Table B.3.8.1):

cf1. the eight Sector-register Valid Bit are put in the "Not Valid" status (logical ONE level);
cf2. "shift register" initialization (at the beginning it points the sector #0 resources);
cf3. reset all the Block Valid-Bits, that is, reset the entire Block Valid-Bit Memory;
cf4. the four previous hit/miss indicators are initialized in the MISS status;
cf5. reset the block and data parity-check error registers;
cf6. the STSRG0+00, STSRG1+00, STSRG2+00, and STSRG3+00 interface signals are put at ZERO level (see also Section B.4);
cf7. reset the VME Monitor logic (reset also an eventually MONITOR request pending);
cf8. reset the GENERAL FLUSH request;
cf9. reset the ASYNCHRONOUS SERVICE ACTIVE flip-flop.

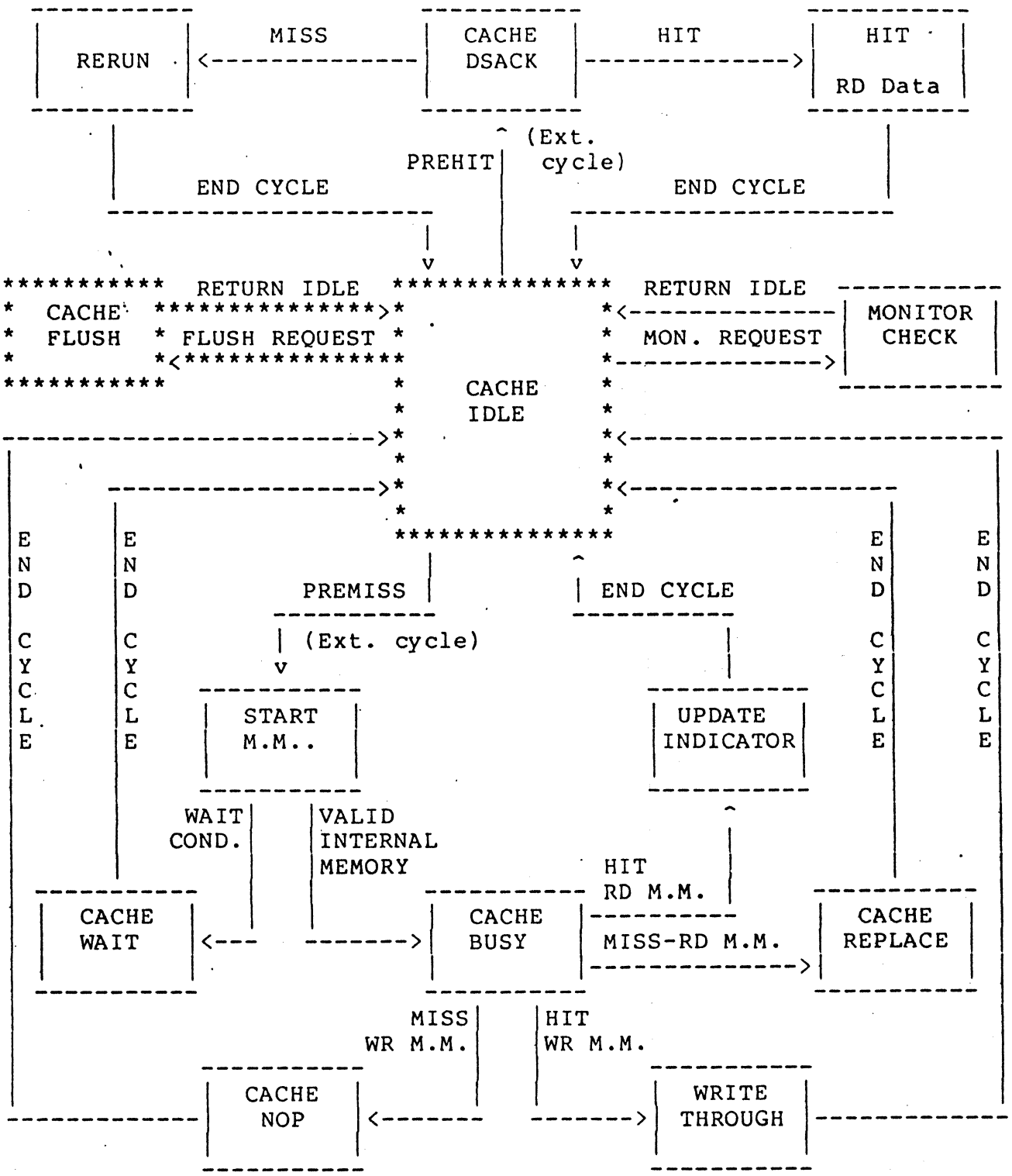


Fig. B.3.9.1 Monitor Check Cycle Diagram

After this, the CACHE IDLE state is still reached.

The above operations are accomplished with four machine clocks (240ns in the case of 16.67 MHz clock frequency).

As described in the previous Section, in any of this four machine clocks the MPU68020 can start an external cycle towards the Main Memory Space: in this case the Cache Controller stops the MMU operations and delays the Main Memory starting or the MPU68020 DSACK0-1 signals asserting until the CACHE FLUSH cycle is completed (at maximum for four machine clocks). It must be added that, if the MPU68020 performs an external cycle towards a local resources (EPROM, TIMER, etc.) this cycle and the CACHE FLUSH cycle run in parallel

B.4 CACHE DIAGNOSTIC FEATURES

In this Section will be described the diagnostic features provided by the cache memory and a detailed description of the cache command and status signals.

Two interface signals driveable by the CP board have the nature of commands (see also Section B.6) and precisely:

- CNTRG0+00
  - when it is at level ONE the cache board is "not active": in this condition only the CACHE FLUSH cycle can be executed;
  - when it is at level ZERO the cache board is "active" and any one of the cycles described in Section B.3 can be performed;

  after a RESET this signal arises at level ONE;

- CNTRG1+00
  - when it is at level ONE the cache board is enabled to work in NORMAL mode;
  - when it is driven to level ZERO the cache board is enabled to work in DIAGNOSTIC mode;

  also when the DIAGNOSTIC mode is selected, the cache is able to execute any cache cycle; after a RESET the NORMAL mode is activated.

On the other hand, six interface signals provided by the cache can be read by the CP board; they have the meaning of status signals (see Section B.6) and have the following description:

- CHPRES-00
  - this signal is at level ONE when the cache board is not inserted into the system;
  - a level ZERO is forced when the cache board is inserted into the system;

- CDMOWF-00
  - when an overflow condition occurs on the two level FIFO of the VME Monitor logic a low impulse is provided on this line and a level 7 MPU68020 interrupt is activated; after a RESET this signal is inactive (level ONE);

- STSRG0+00 . this signal gives to the MPU68020 the knowledge of the DSACK0,1 signals activation by the cache board at cycle level. After a CACHE FLUSH cycle this line is at level ZERO: then, each time that the cache board provides the DSACK0,1 signals (on the HIT cycle and on the RERUN cycle) this signal changes its status in toggle manner. In other words, it must be remembered the last STSRG0+00 signal status to compare it with the current value and if a match occurs the cache board has started the Main Memory; instead of, if a mismatch occurs, the cache board has provided the DSACK0,1 signals and hence the data;

- STSRG1+00 . this line informs at cycle level the MPU68020 when the cache board requests a RERUN cycle. After a CACHE FLUSH cycle this signal is at level ZERO: then, each time that a MISS condition is detected after a DSACK0,1 signal assertion this line toggle its status. No change from the previous status means "RERUN cycle not requested"; if a change occurs a "RERUN cycle is requested";

- STSRG2+00 . this signal has the meaning of cache data parity error and is driven to level ONE when a parity error is detected during a read operation of the Cache Data Memory; the parity-check bits control is performed only during the HIT cycle. After a CACHE FLUSH cycle this line is driven at level ZERO;

- STSRG3+00 . this signal is the logical or of the cache data parity error and the cache block parity error and goes to level ONE when almost one of these two errors arises. The block check bit control is performed each time the Block Valid-Bit Memory is read, that is in all cache cycles except in the CACHE FLUSH cycle. This line is driven to level ZERO after a CACHE FLUSH cycle execution. When this signal is used in combination with the STSRG2+00 signal is possible to detect which memory has provided the parity error. When a cache parity error occurs the HALT signal of the coupled MPU68020 is asserted and only a push-button RESET or a power-off power-on sequence can restart the halted MPU68020.

It must be pointed that the STSRG0,1,2,3 signals can be software resettable: infact, a CACHE FLUSH cycle can be originated by the MPU68020 performing a "cache active and cache not active" sequence, that is, driving the CNTRG0+00 signal first to level ZERO and then to level ONE.

In the Table B.4.1 is summarized the operations and the meanings of the above cache command and status signals.

Table B.4.1 Cache Command and Status Signals

	Signal Name	Logical Value	Meaning/Operation	Value After RESET
COMMANDS	CNTRG0+00	0	Cache Board Active	1
		1	Cache Board Not Active	
	CNTRG1+00	0	Diagnostic Mode	1
		1	Normal Mode	
STATUS	CHPRES-00	0	Cache Present	N.A.
		1	Cache Not Present	
	CDMOWF-00	0	VME Monitor Overflow (low impulse)	1
		1	VME Monitor Overflow Not Active	
	STSRG0+00	CHANGE	Cache DSACK0,1 Asserted	0
		NO CHANGE	Cache DSACK0,1 Not Asserted	
	STSRG1+00	CHANGE	Cache RERUN Cycle Requested	0
		NO CHANGE	Cache RERUN Cycle not Requested	
	STSRG2+00	0	No Cache Data Parity Error	0
		1	Cache Data Parity Error	
	STSRG3+00	0	No Cache Data/Block Parity Error	0
		1	Cache Data/Block Parity Error	

N.A. --> Not Applicable

It must be added that the cache operation mode is determined by the CNTRG0,1 signals coding. The Table B.4.2 shows how the above signals must be set to select the particular cache operation mode chosen.

Table B.4.2 Cache Operation Modes Summary

CNTRG1+00	CNTRG0+00	Cache Operation Modes
[illegible]	[illegible]	Normal Mode
0	0	Diagnostic Mode
0	1	Monitor Overflow Diagnostic Mode
1	1	No Action

The Monitor Overflow Diagnostic Mode can be activated only in a MONO-CP environment and allows the testing of the Monitor Overflow generation logic. This check is performed accesing the not existing second Main Memory: in this manner the VME Bus addresses is registered when the System Controller drives the VBUERR-00 signal for time-out expiration: then, the Monitor request is blocked and so on the third access the FIFO overflow must arise producing a level 7 MPU68020 interrupt.

When the DIAGNOSTIC MODE is selected the following conditions are forced:

a. the Data Check Bits are always loaded (during a CACHE REPLACE cycle or a WRITE THROUGH cycle) into the Data Check Rams with a level ONE;

b. for the Block Valid-Bit Memory the inverty-check feature is implemented, so during a CACHE REPLACE cycle or a MONITOR CHECK cycle the Block Check Bits are built to perform the ODD parity (in NORMAL MODE the EVEN parity is provided);

c. the generation of the MPU68020 HALT signal is disabled to permits the testing of the cache memories and of the relative parity-check circuitries avoiding to halt the MPU68020;

d. the MONITOR request is asserted only when the MPU68020 accesses (in read or write, preferably in read) the Main Memory Space of the not existing second Main Memory: in this case the VME Bus cycle will terminate with a bus error (VBUERR signal assertion) that stores the current VME Bus address in the MONITOR FIFO (obviously if the MONITOR FIFO is not full). This feature can be used, in a MONO-CP environment, to perform block invalidations accessing the not existing second Main Memory at the same location address of the previous cached information: in this manner the VME Bus interface and the MONITOR logic can be tested. Instead of, in a DUAL-CP environment the above mechanism can not be used but the same cache area can be tested by the other CP or by any other VME Bus master that performs VME Bus write cycles towards specific address locations of the first Main Memory.

Therefore, all the cache areas can be tested driving and checking adequately the cache command and status signals and performing opportune cache cycle sequence.

B.5 CACHE ARCHITECTURE BLOCK DIAGRAM

The cache memory architecture consists of the following modules (see Fig. B.5.1):

- Cache Address Multiplexing;
- Cache Content-Addressable Memory;
- Cache Valid Bit Memory;
- Cache Comparator;
- Cache Hit/Miss Detection;
- Cache Hit/Miss Prediction;
- Cache Data Memory;
- VME Bus Monitor;
- Cache Activity Control and Timing Logic.

The following Sections describe in detail the above modules from the hardware point of view.

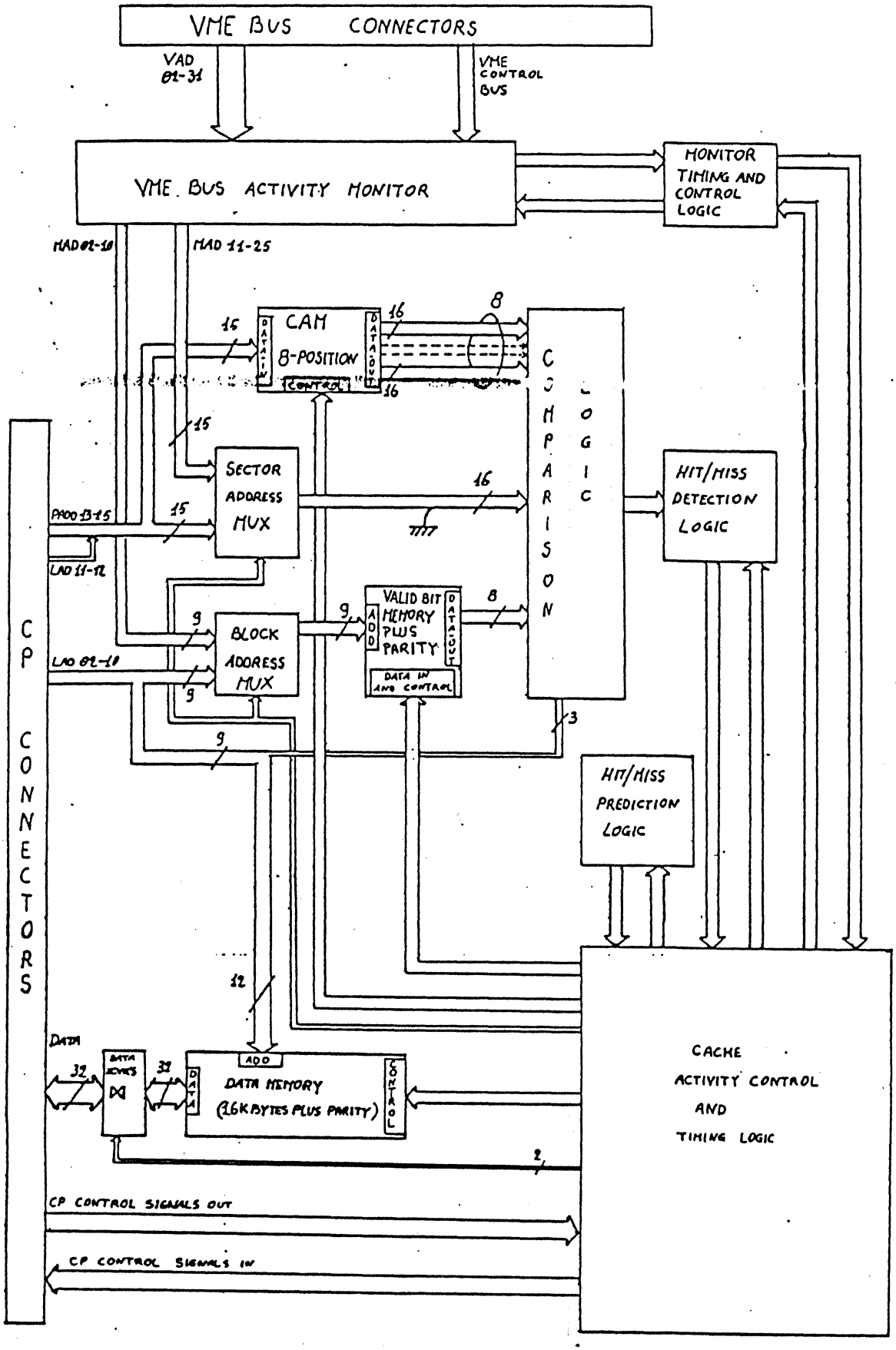


Fig. B.5-1 Cache Main Block Diagram

B.5.1 CACHE ADDRESS MULTIPLEXING

This module consists essentially of two multiplexer groups (SECTOR mux and BLOCK mux) of the 2-->1 type as shown in Fig. B.5.1.1.

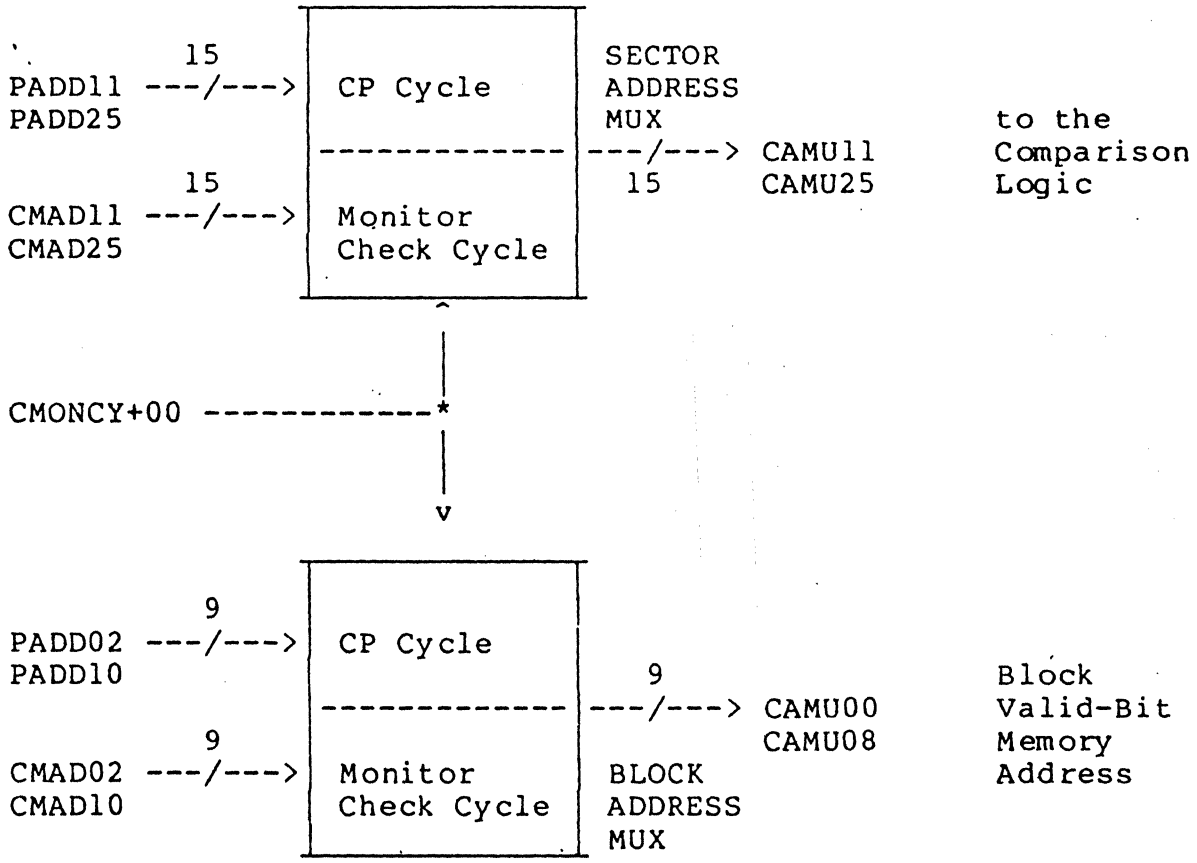
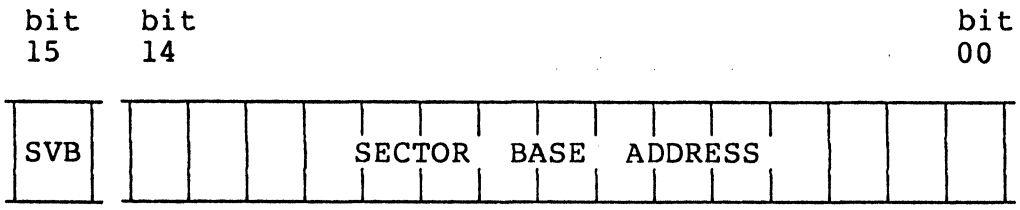


Fig. B.5.1.1 Address Mux Block Diagram

Only during a MONITOR CHECK cycle the CMONCY signal is at level ONE and so the latched VME addresses from the MONITOR FIFO are used by the COMPARISON logic and to address the Block Valid-Bit Memory: normally, CP cycles are selected and PADD02-PADD25 are muxed through.

B.5.2 CACHE CONTENT-ADDRESSABLE MEMORY

The cache Content-Addressable Memory is organized in eight positions each one has the following layout (see Fig. B.5.2.1): fifteen bits are used to store a Main Memory Sector Base address (that is the physical address group PADD25-PADD11) and one bit has the meaning of CAM-position content validation bit (1 --> CAM-position content Not Valid; 0 --> CAM-position content Valid).



SVB --> Sector Valid Bit

Fig. B.5.2.1. CAM Position Layout

Each CAM-position is implemented with two 8-bit registers (74F374) and one flip-flop as shown in Fig. B.5.2.2.

The input signals of this module are the following:

- PADD11
  PADD25 — the fifteen physical addresses provided by the MMU are connected on the inputs of each CAM positions;

- CLSERx-00 — only one signal at a time is active, precisely that individuated by the Fifo Replacement Logic (Shift Register outputs), and it performs the storing of the actual PADD11-PADD25 addresses into the relative CAM-position registers and the validation of the relative Sector Valid Bit: this signals can be asserted during the CACHE REPLACE cycles only;

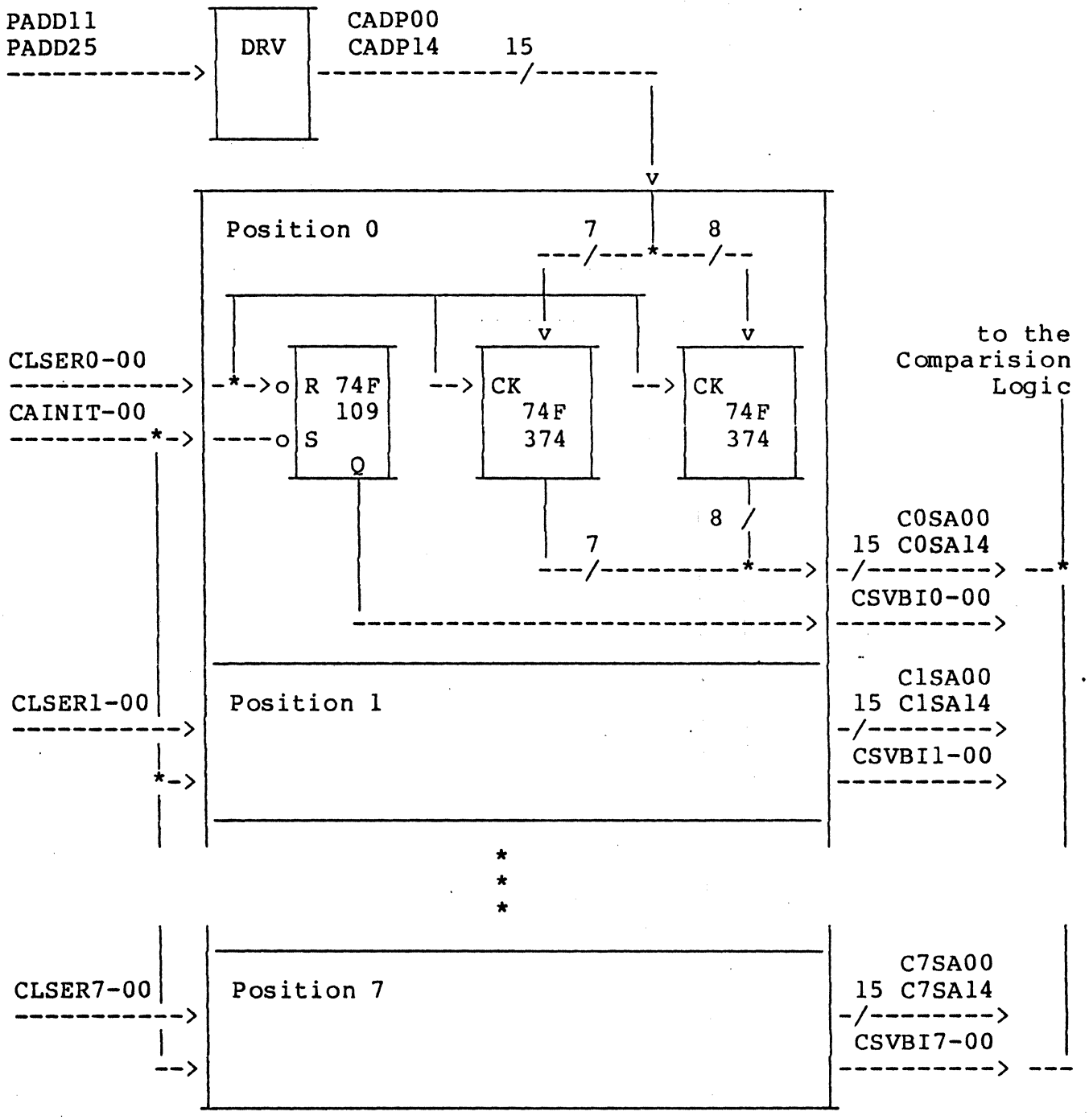


Fig. B.5.2.2 CAM Block Diagram

- CAINIT-00 this signal initialize all the Sector Valid Bit (the CSVBIx-00 lines) to level ONE when a CACHE FLUSH cycle is requested,

and the output signals are sixteen lines for each CAM-position: the fifteen registered physical addresses (CxSA00-CxSA14) and the associated Sector Valid Bit (CSVBIx-00). At cycle level, each one of these eight signal groups are all compared with the current physical address to search a match (that is the "sector hit" condition).

B.5.3 CACHE VALID BIT MEMORY

The Block Valid-Bit memory is organized in eight 512x2 memories each one relative to a specific sector-slot of the Cache Data memory (see also Fig. B.2.1). The size of each Valid-Bit memory is determined by the block number in a sector-slot, that is 512, and the layout of a Valid-Bit memory word is shown in Fig. B.5.3.1, where:

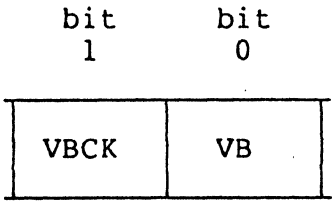


Fig. B.5.3.1 Valid-Bit Memory Word Layout

- VB Valid Bit that when at level ZERO means "block not valid" and when at level ONE means "block valid";

- VBCK Valid Bit Check that is the EVEN parity bit of the VB bit when the cache works in normal mode; in diagnostic mode (see also Section B.4) the ODD parity (that is the inverty-check feature) is provided.

The Valid-Bit memory is physically implemented by eight SRAM AM9150, 1Kx4, 25ns access time, with RESET feature connected as shown in Fig. B.5.3.2. The input signals of this module are the following:

- CAMU00
  CAMU08 are the nine addresses provided by the Block Address Mux, so the Valid-Bit memory is dual ported between CPU cycles and the VME Bus Monitor cycles;

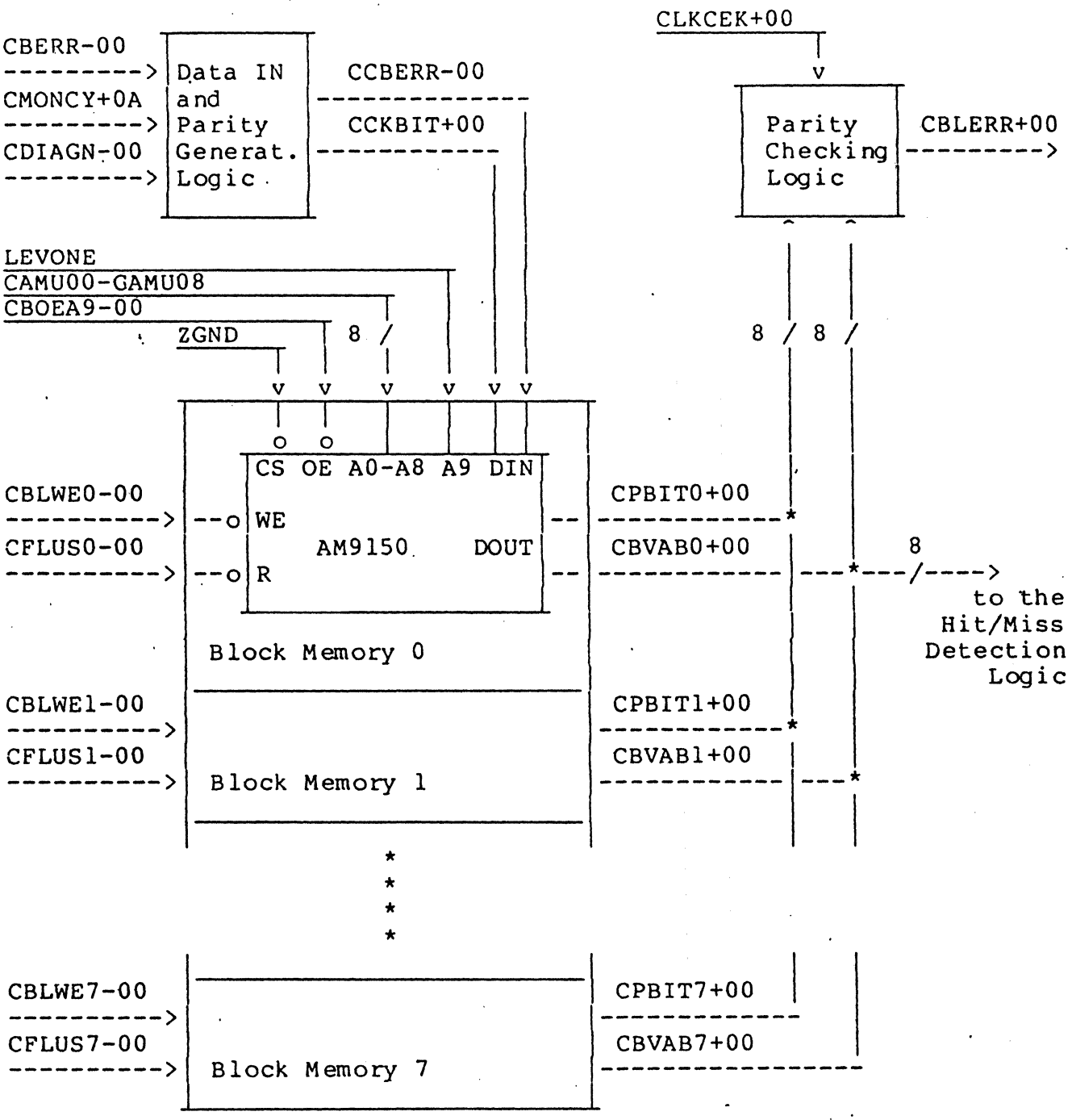


Fig. B.5.3.2 Valid-Bit Memory Block Diagram

- CCBERR-00
  CCKBIT+00

the Data In to the memory is CCBERR signal and CCKBIT signal is the associated parity bit. These signals come from a combinatory logic, the "Data In and Parity Generation Logic", that recevies three signals: CBERR (the MPU68020 BERR line), CMONCY (when at level ONE indicates Monitor Check cycle in execution), and CDIAGN that decides the cache operation (Normal/Diagnostic). The above logic works as described in the following truth table:

CDIAGN-00	CMONCY+0A	CBERR-00	CCBERR-00	CCKBIT+00
1	0	0	0	0
1	0	1	1	1
1	1	X	0	0
0	0	0	0	1
0	0	1	1	0
0	1	X	0	1

X --> Meaningless

that is, for the MPU68020 cycles, the Block Memory Data-In is represented by the status of the CBERR line (when 1, Main Memory BERR not occurred, the block is validated; when 0, Main Memory BERR occurred, the block is always stored but not validated) and during a Monitor Check cycle the Data-In is forced to ZERO to invalidate the relative block. The CCKBIT represents the parity bit that in normal mode is the EVEN parity bit and in diagnostic mode the inverty-check feature is provided;

- CBOEA9-00 this line is the block SRAM output enable; it is always at ZERO level except during a RESET operation where is recommended to drive it at ONE level (this occurs executing a Cache Flush cycle or a Cache Replace cycle with a sector miss condition);

- CBLWEx-00 these signals are tied to the SRAM write enables; these lines are driven by the "Cache Activity Control and Timing Logic" one at a time and are selected:

  . during a Cache Replace cycle by the Shift Register outputs on a sector miss condition or by the Hit Sector Number Register outputs when only a block miss condition occurs;
  . during a Monitor Check cycle by the Hit Sector Number Register outputs;

- CFLUSx-00 these signals are the SRAM RESET inputs. A low impulse on this SRAM input clear all the SRAM locations. These lines are driven by the "Cache Activity Control and Timing Logic": during a Cache Replace cycle with sector miss is asserted a line at a time selected by the Shift Register outputs, instead of, in a Cache Flush cycle all the lines are contemporarily asserted.

It must be remembered that all the SRAM chip select inputs are tied to ground and so the data outputs availability depends only on the address inputs time. The rationales of this choice are that the handling of the chip selects determines a big delay on the data outputs availability and then the package power dissipation is very similar in the case of package selected or not selected.

The output signals of this module are the eight Valid Bit signals CBVABx+00 and the relative eight parity-check-bit signals CPBITx+00. The Valid Bit signals are received by the Hit/Miss Detection Logic and concur to determine the total hit condition. All the sixteen outputs are handled by the Parity Check Logic to determine the presence of the Block Valid-Bit memory parity error. This error is checked each time the Valid-Bit memory is used and when an error is found the coupled MPU68020 HALT signal is asserted and the microprocessor restarts its operations only after a button RESET or a new Power-on.

B.5.4 CACHE COMPARATOR

This module consists of the comparator group organized as shown in Fig. B.5.4.1 and of the Encoding Logic for the three most significant addresses of the Cache Data Memory.

Each comparator section performs the comparision between the current physical address (CAMU11-CAMU25) provided by the MMU and the relative sector registered addresses (CxSA00-CxSA14) and contemporarily is controlled the validity of the sector registered addresses.

On the other hand, each comparator section provides two comparision match signals, one active LOW and the other active HIGH: when these signals (CORxAB+/-00) are asserted a Sector Hit condition is occurred: this event is registered into the Hit Sector Number Register, is used to find the Total Hit condition, and also to produce the high address of the Cache Data Memory: this Encoding Logic works by the followiong truth table:

COR7AB-	COR6AB-	COR5AB-	COR4AB-	COR3AB-	COR2AB-	COR1AB-	COR0AB-	CDRA11+	CDRA10+	CDRA09+
0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	0
1	1	0	1	1	1	1	1	1	0	1
1	1	1	0	1	1	1	1	1	0	0
1	1	1	1	0	1	1	1	0	1	1
1	1	1	1	1	0	1	1	0	1	0
1	1	1	1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	0	0	0	0

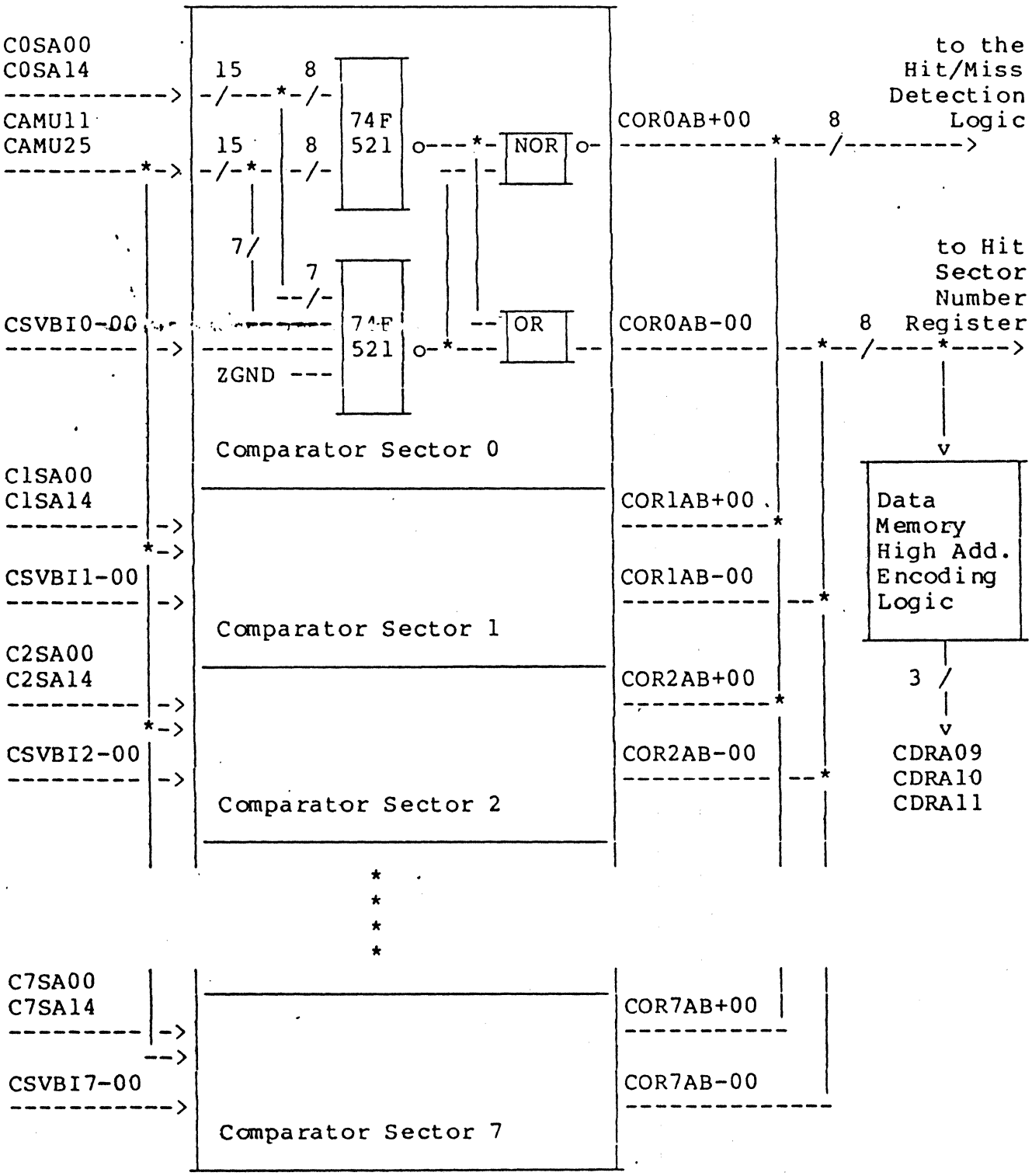


Fig. B.5.4.1 Cache Comparator Block Diagram

It must be pointed that the CORxAB combinations permitted are only eight because only a sector hit at a time (or none) can occur during an MPU68020 cycle and besides that the comparator module is a combinatory logic that works at any time.

B.5.5 CACHE HIT/MISS DETECTION

This combinatory module is used to detect the total Hit/Miss condition. This circuitry is implemented as shown in Fig. B.5.5.1.

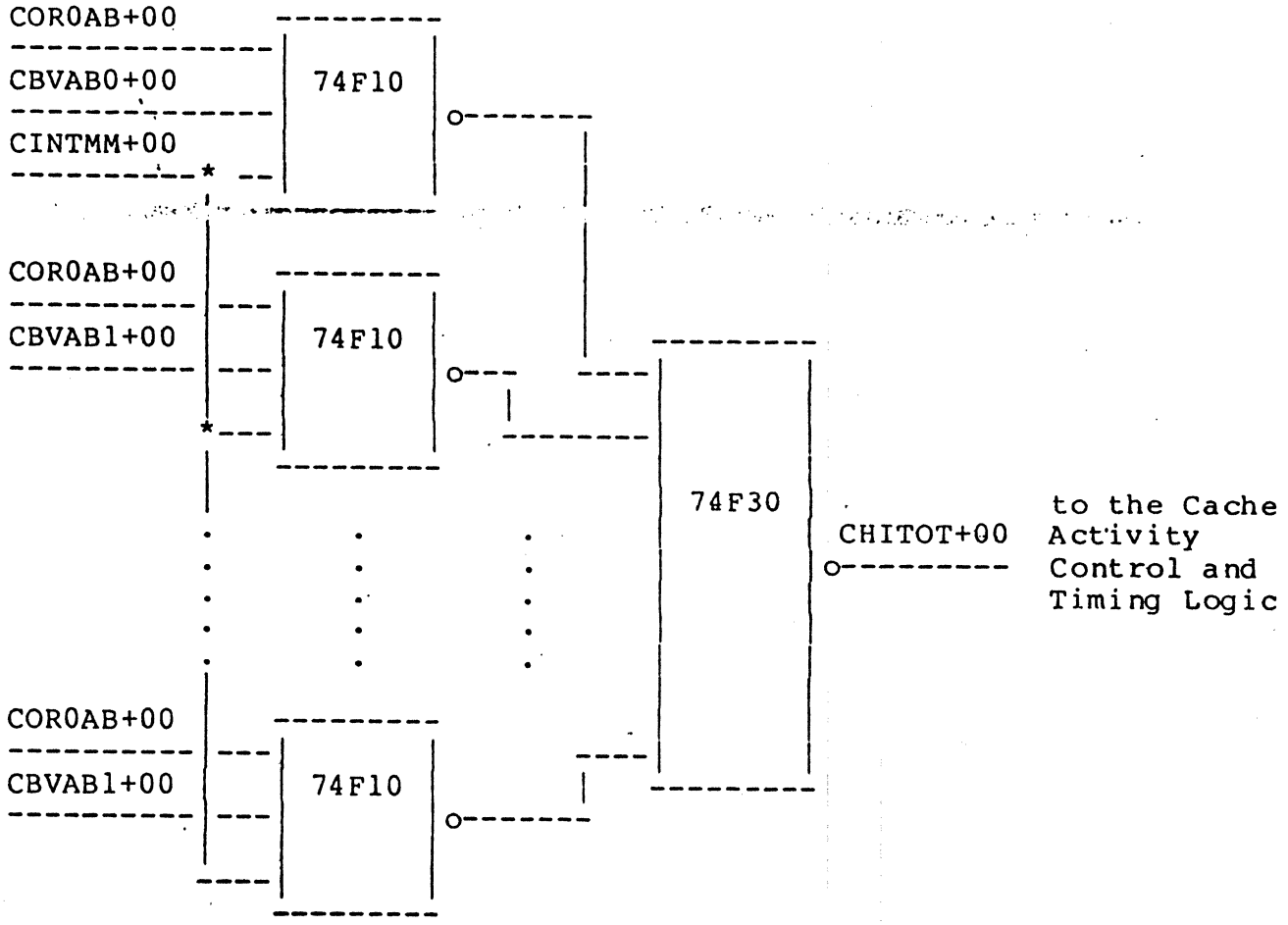


Fig. B.5.5.1 Hit/Miss Detection Logic Diagram

When a Sector Hit condition occurs (CORxAB signal at level ONE), a Block Hit condition occurs (CBVABx signal at level ONE), and the current MPU68020 cycle is performed towards the coupled Main Memory the combinatory Total Hit condition is asserted (CHITOT+00 signal goes to level ONE) and will be used by the Cache Activity Control Logic to decide the cache cycle to execute.

It must be clear that the above logic as well as the CAM Memory, the Valid-Bit Memory, and the Cache Comparator are shared between the MPU68020 cycles and the Monitor Check cycles. During a Monitor Check cycle execution the CINTMM signal is always asserted because the VME Monitor logic captures only VME write cycles directed towards the coupled Main Memory.

B.5.6 CACHE HIT/MISS PREDICTION

This module provides the previous hit information on the basis of previous cycle hit status (relative at the same privilege): if it was an hit a previous hit will be provided, if it was a miss a previous miss will be provided and the coupled Main Memory will be started. The above functions are implemented as shown in Fig. B.5.6.1.

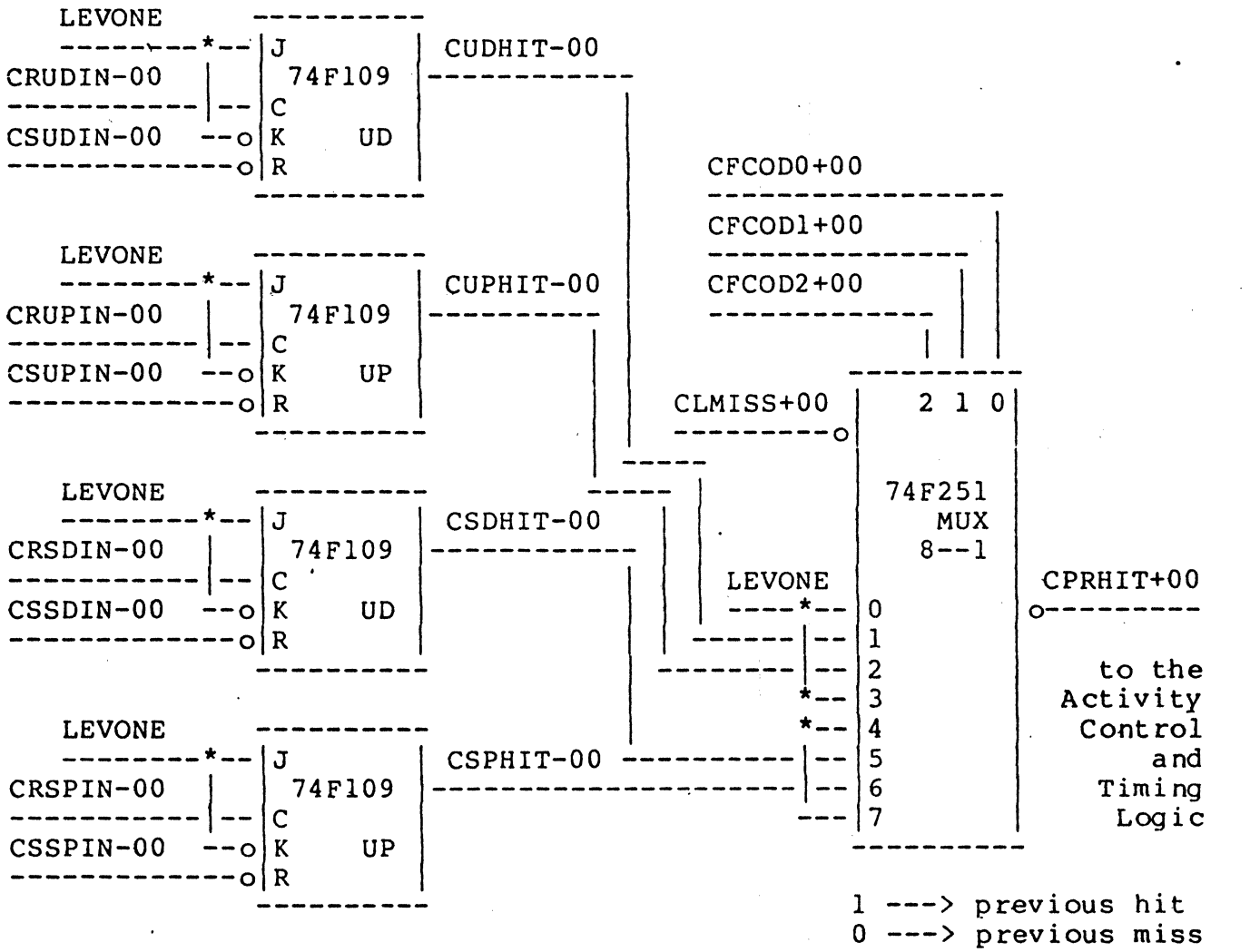


Fig. B.5.6.1 Hit/Miss Prediction Logic Diagram

The four previous hit/miss indicators are handled as follows:

- after a Cache Flush cycle all the indicators will produce the previous MISS condition (assertion of all CRxyIN-00 signals); after a Rerun cycle only the indicator selected by the current Function Code signals will be put in the MISS condition with the assertion of the relative CRxyIN-00 signal;

- after an Update Indicator cycle the indicator selected by the current Function Code signals will be put in the HIT condition asserting the corresponding CSxyIN-00 signal.

During an MPU68020 cycle, a multiplexer 8 to 1 selects the indicator relative to the current cycle by the Function Code signals: so the mux provides the CPRHIT+00 signal that represents a qualifier of the current cycle.

B.5.7 CACHE DATA MEMORY

The Cache Data Memory contains a 4K by 32 bit (16 Kbytes) static RAM plus a byte level parity for store data and program information: this memory is accessed only for MPU68020 cycles, differently by the CAM and Block Valid-Bit memories.

The "cache data block" organization and structure is illustrated in Fig. B.5.7.1.

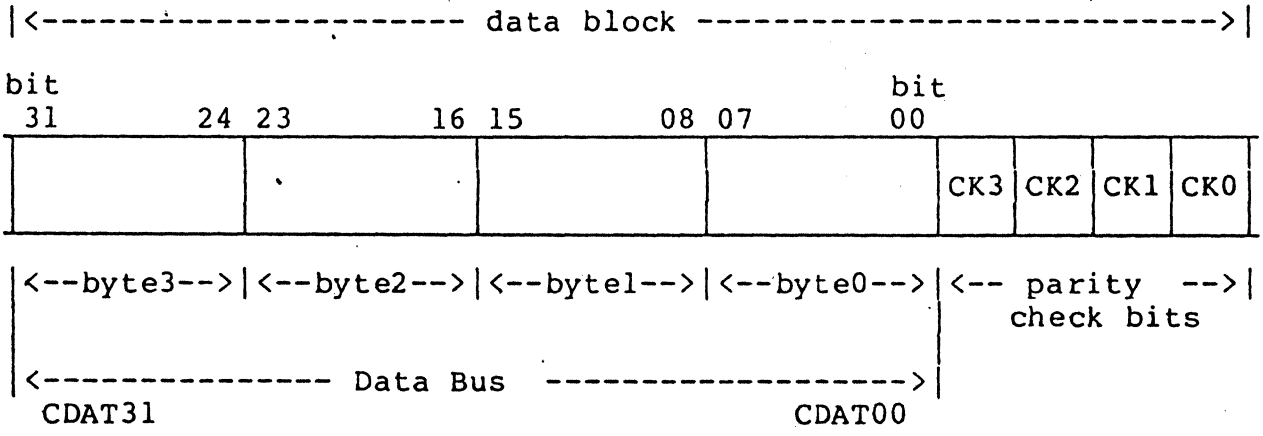


Fig. B.5.7.1 Data Block Structure

The Cache Data Memory is physically implemented by twelve high speed (25 ns access time) Static RAM 4Kx4 of which eight store data and program information and the other four are used as SRAM 4Kx1 for the parity bit storing: the organization of the Cache Data Memory is shown in Fig. B.5.7.2. This module also includes two other hardware blocks: the "Data Transceivers" and the "Parity Generator and Checking Logic".

The Cache Data Memory is accessed in READ, during the Hit and Rerun cycles, always at Long Word (32 bits) level: the data provides by the Data SRAM is delivered to the MPU68020 through the Data Transceivers and contemporarily are received by the Parity Checking Logic with the four Parity Check signals provided by the Check SRAM. The Parity Checking Logic produces four combinatory parity error signals (CDCERx+00) active at level ONE: the CDCERx signals are strobed only in the Hit cycles and even if only one is active, the MPU68020 HALT signal is asserted because an incorrect data was latched by the microprocessor.

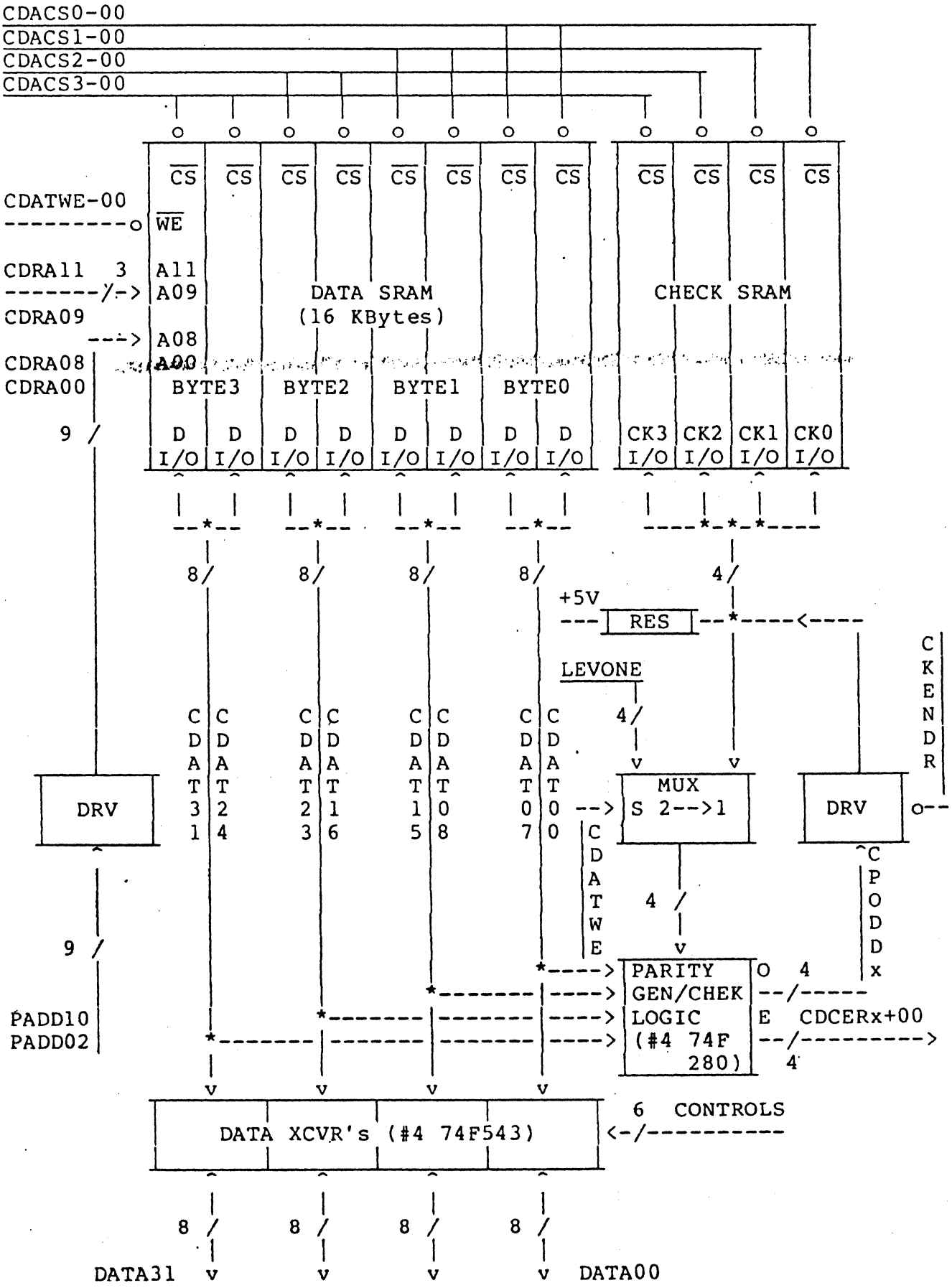


Fig. B.5.7.2 Cache Data Memory Block Diagram

The Cache Data Memory is accessed in WRITE in the Cache Replace and Write Through cycles. During the Cache Replace cycle, that is a Read cycle for the MPU68020, the Long Word (32 bits) provided by the coupled Main Memory is also stored into the cache Data SRAM with the four odd parity check bits provided by the Parity Generator Logic and stored into the cache Check SRAM. During the Write Through cycle with the hit condition present the data provided by the MPU68020 update the coupled Main Memory contents and also the cache memory contents: this updating can occur at byte, word, three bytes, and long word level and the parity check bits are generated in correlated manner.

The Cache Data Memory is addressed by twelve addresses: the nine least significant are the PADD08-PADD02 addresses provided directly by the MPU68020 (only a DRIVER circuit is present). The three most significant come from the High Address Encoding Logic (see Section B.5.4) that produces a coding (on three bits) of the actual hit sector number.

B.5.8 VME BUS MONITOR

When a VME Bus master executes a write cycle towards the coupled Main Memory, the VME Monitor latches in the dual buffer FIFO the associated address and requests a Monitor Check cycle. If cached (valid), that entry is invalidated to eliminate stale data in the cache. Only very fast VME Bus write activity (cycle time faster than 180 nanoseconds) may overrun the Monitor dual buffer FIFO: actually any VME Bus cycle is forecasted longer than any MPU68020 cycle. In any case, the overrun condition will clear the cache (a Cache Flush cycle is executed) and assert the CDMOWF-00 signal (Monitor Overflow) that will produce a software handled level 7 MPU68020 interrupt.

The VME Bus Monitor (refer to Fig. B.5.8.1) consists of the VME Bus signal receiver circuitry, the dual buffer FIFO used to latch the VME Bus address and the Monitor Timing and Control Logic.

The VME Bus addresses (VADD31-VADD02) are latched at the beginning of any VME Bus cycle by the rising edge of the CADSTB+00 signal and so the CADD31-CADD02 addresses do not change until the next VME Bus cycle; this latching operation is performed on any type of VME Bus Read/Write cycle. The CADD11-CADD02 addresses go directly to the first FIFO buffer; instead the group CADD25-CADD12 can be modified (see Table B.5.8.1) depending on the VME Bus Master type and according to the VME Bus maps (see CP0 PDD). The VME Bus Master type is detected coding the VME Bus address modifiers (CADMD5 and CADMD4 signals; see also Table B.5.8.1).

Table B.5.8.1 Handling of the CADD25-CADD12 Addresses

CADMD5	CADMD4	VME MASTER TYPE	CADM25 CADM20	CADM19 CADM16	CADM15 CADM12
1	0	SHORT (16)	0	0	0
1	1	STANDARD (24)	0	CADD19 CADD16	CADD15 CADD12
0	0	EXTENDED (32)	CADD25 CADD20	CADD19 CADD16	CADD15 CADD12

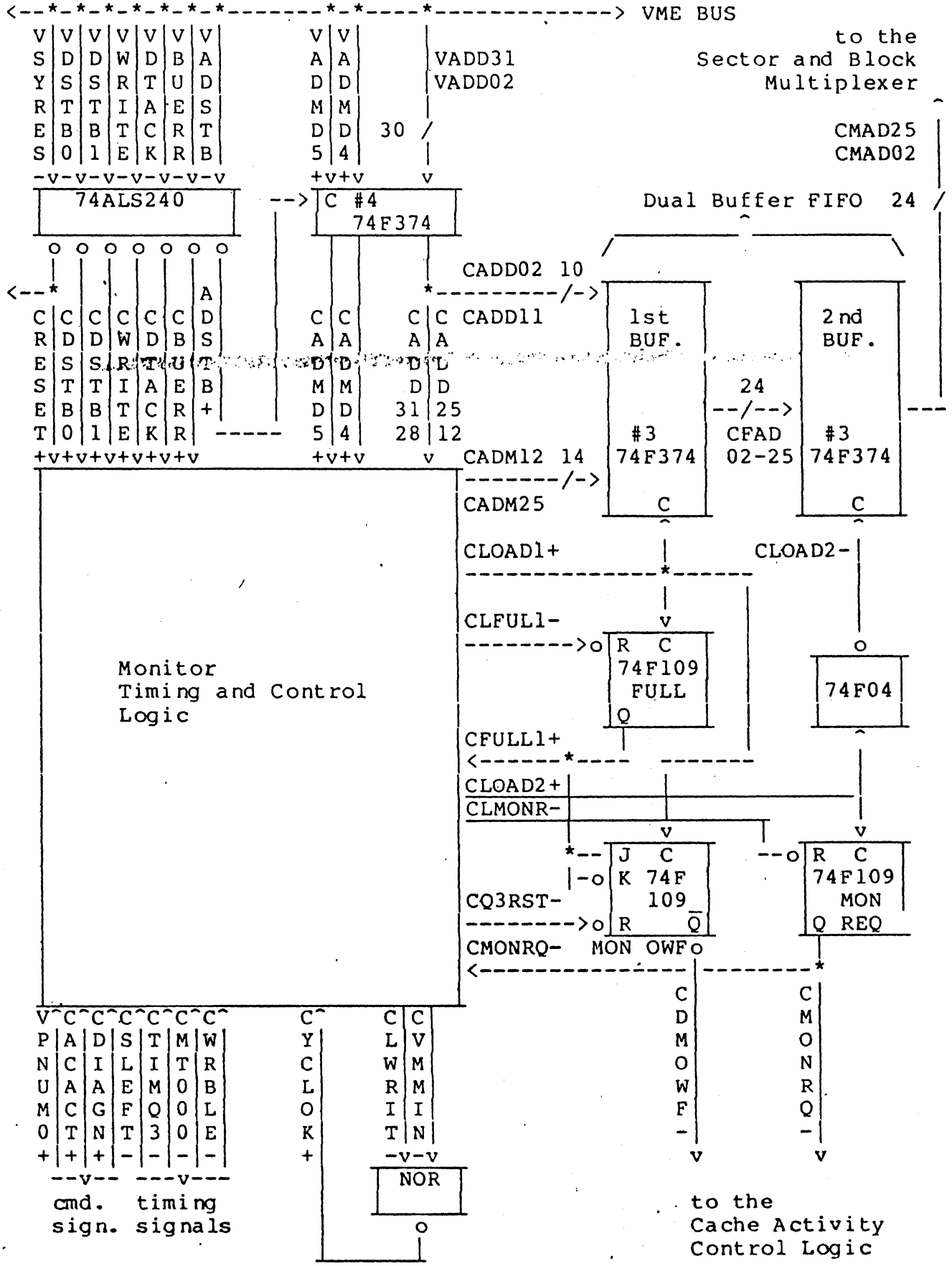


Fig. B.5.8.1 VME Bus Monitor Block Diagram

It must be pointed that is requested to load in the VME FIFO only the VME addresses corresponding to the Sector Address Field (stored into the CAM memory; CADM25-CADM12,CADD11) and the Block Address Field (CADD10-CADD02) to detect the Block Valid-Bit. Actually, the VADD27 and VADD26 addresses are received but not used in anywhere.

A VME Bus master performs a "Cache VME Valid Main Memory Access" (that is, the CVMMIN-00 signal is asserted) under the conditions described in the Table B.5.8.2. Only the accesses toward the coupled Main Memory are take into account because the other Main Memory contents are not cachable.

Table B.5.8.2 Cache VME Valid Main Memory Access

		CADMD5	CADMD4	CADDM32-17	CADDM22-30	CADDM11-52	VPNUM0+	CACT+	CVMMIN-
		x	x	x	x	x	x	x	1
VME MASTER	SHORT	1	0	NA	NA	0001 / 0010	0 / 1	1	0
	STANDARD	1	1	NA	0001 / 0010	NA	0 / 1	1	0
	EXTENDED	0	0	0001 / 0010	NA	NA	0 / 1	1	0

x --> don't care
NA --> Not Applicable

If the CVMMIN-00 signal is asserted and the VME cycle ia a WRITE cycle (CLWRIT-00 signal asserted; this signal is latched at each VME cycle by a strobe that is the ored VME Data Strobes) the CYCLOK+00 signal is asserted and only in this case when the coupled Main Memory drives the VDTACK-00 signal the CLOAD1+00 clock is activated (see Table B.5.8.3). This means that, in normal mode, if the coupled Main Memory detects an error and drives the VBUERR-00 signal, the CLOAD1+00 clock is not asserted and no real VME Monitor activities are performed.

Table B.5.8.3 Handling of the CLOAD1+00 Signal

	CACACT+	CDIAGN+	VBUERR-	VDTACK-	CLWRIT-	CVMMIN-	CLOAD1+	Description
	0	0	x	x	x	x	0	Cache Disabled (No Action)
Normal Mode	1	0	NA	v	1	x	0	VME READ Cycle (No Action)
	1	0	NA	v	0	1	0	Coupled M.M. not Accessed (No Action)
	1	0	NA	v	0	0	^	Valid Access (Monitor FIFO Logic Started)
Diagnostic Mode	1	1	v	NA	(x)	x	^	Diagnostic Access (Monitor FIFO Logic Started)
	0	1	v	NA	(x)	x	^	Diagnostic Access to Verify the VME Monitor Overflow Condition (Monitor FIFO Logic Started)

x --> don't care
(x) --> don't care but a READ access is preferred
NA --> Not Applicable

In diagnostic mode and in MONO-CP environment, the CLOAD1+00 signal is asserted only on the VBUERR-00 signal activation: this can be performed, for VME Monitor diagnostic purposes only, accessing the address space of the not-present second Main Memory. The started VME cycle is terminated by the System Controller asserting the VBUERR-00 signal for time-out expiration.

The Monitor dual buffer FIFO operations are:

- storing the VME address in the first FIFO buffer;
- shift the first FIFO buffer contents into the second FIFO buffer;
- set the VME Monitor request;
- set (if the case) the VME Monitor Overflow condition.

The above operation are handled by three flip-flops: fifo FULL, MONitor REQuest, and MONitor OVerFlow (see Fig. B.5.8.1). The operation starting comes on the rising edge of the CLOAD1+00 signal as shown in Fig. B.5.8.2.

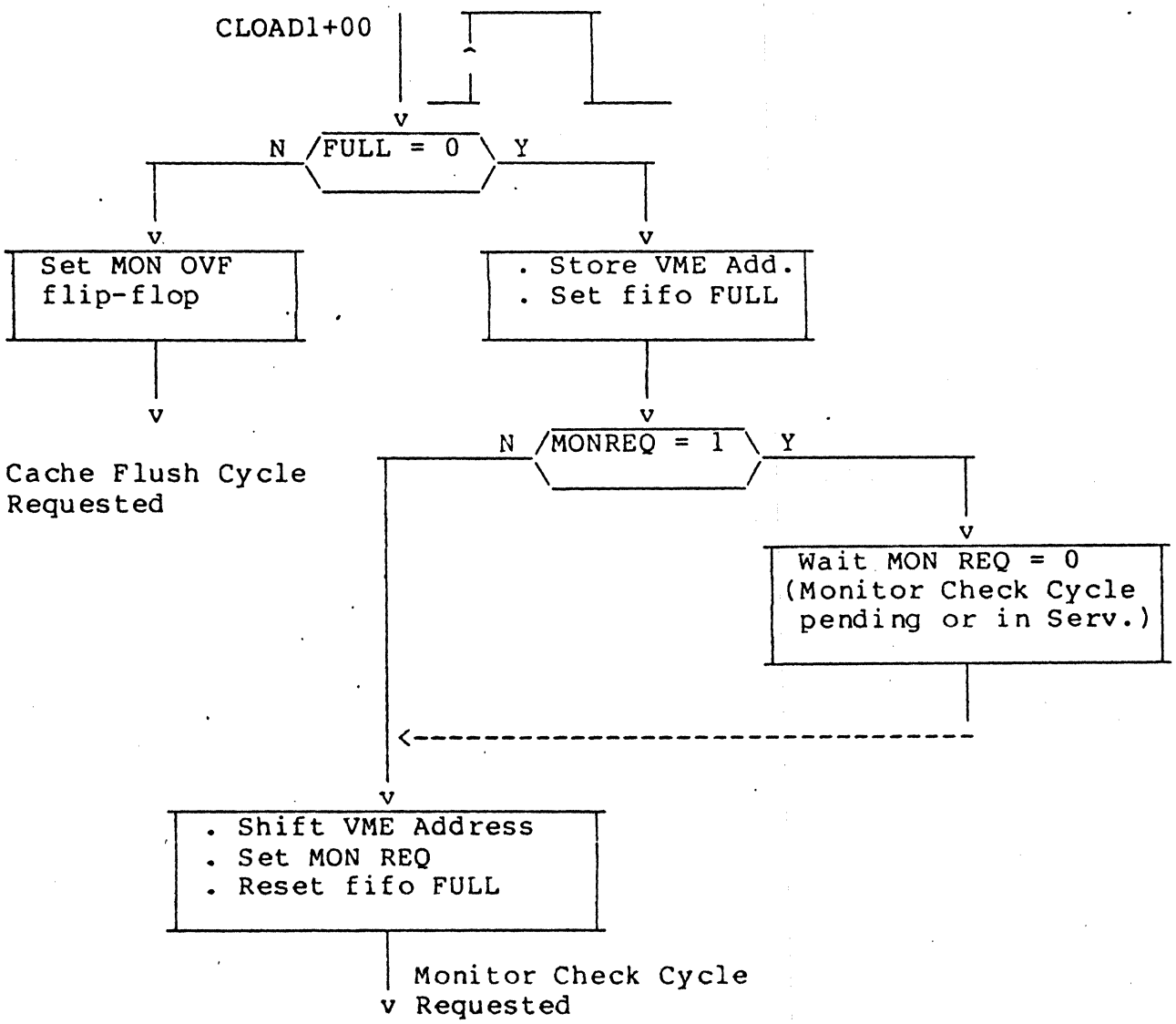


Fig. B.5.8.2 Monitor Dual Buffer FIFO Activities

B.5.9 CACHE ACTIVITY CONTROL AND TIMING LOGIC

This module represent the basic cache circuitry used to decide and manage all the cache activities. This combinatory and sequential logic can be divided in Timing Logic and Control Logic.

All the cache timing is synchronous with the MPU68020 machine clock (16.67 MHz) except the Monitor Dual Buffer FIFO timing that is controlled by the VME Bus strobes. The Timing Logic is made up of four timing circuitries that handle respectively the following cache cycles:

1 ---> . Cache Replace,
. Update Indicator,
. Write Through,
. Cache Nop,
. Cache Wait,
2 ---> . Hit,
. Rerun,
> - MPU68020 cycles

3 ---> Monitor Check, - VME Bus cycles

4 ---> Cache Flush. - Asynchronous cycles

The Control Logic on the basis of the CP interface signals and cache internal signals detects the current cache cycle, actives the relative timing logic and manages the signal driving towards the cache resources interested to the current cache cycle. The Control Logic manages the following cache areas:

- Cache Data Memory,
- Data Transceivers and Latches,
- Block Valid-Bit Memory,
- CAM Registers,
- Previous Hit/Miss Indicators,
- Sector and Block Address Multiplexers,
- Hit Sector Number Register,
- Shift Register (FIFO Replacement Logic),
- Synchronization Logic of the Cache Asynchronous Requests,
- Block and Data Parity-Check Error Logic,

and handles the CP control signals.

B.6 CACHE INTERFACE DESCRIPTION

The cache board has two logic interface: the CP Interface and the VME Bus Interface. The first is physically implemented by the Y02 dedicated connector and by a portion of the Z02 VME Bus connector user pins. The VME Bus Interface is implemented by the two VME Bus connectors Z01 and Z02: the Fig. B.6.1 shows the three connectors present on the cache board with the associated logic interface.

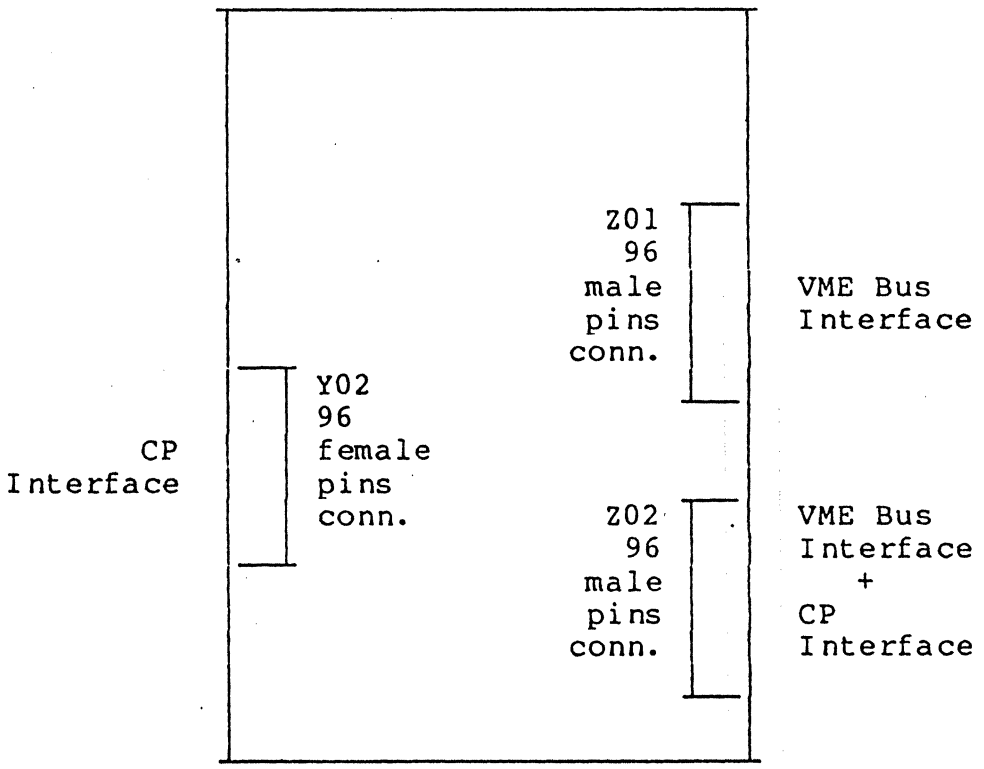


Fig. B.6.1 Cache Connectors and Interfaces

The following tables (Table B.6.1, B.6.2, B.6.3 and B.6.4) provide a detailed description of each interface signal.

Table B.6.1 CP Interface Signals - Y02 Connector Pin Assignment and Description

Pin N.	Signal Name	Signal Source	Description
A01-A10 A19-A22 A25-A29 A31-B02 B04-B05 B06-B07 B08-B09 B11-B12 B13-B14 B15-B16 B18-B23 B25-B27 B29-C01 C10-C19	ZGND	---	Logic GND level.
B01-B03 B10-B17 B30	---	---	Pins not used.
A02	DATA00+CH	CP0 CH0	Data Bus (bidirectional signals) Byte 0 (least significant).
A03	DATA01+CH	CP0 CH0	
A04	DATA02+CH	CP0 CH0	
A05	DATA03+CH	CP0 CH0	
A06	DATA04+CH	CP0 CH0	
A07	DATA05+CH	CP0 CH0	
A08	DATA06+CH	CP0 CH0	
A09	DATA07+CH	CP0 CH0	

Table B.6.1 CP Interface Signals - Y02 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
A11	DATA08+CH	CP0 CH0	Data Bus (bidirectional signals) Byte 1.
A12	DATA09+CH	CP0 CH0	
A13	DATA10+CH	CP0 CH0	
A14	DATA11+CH	CP0 CH0	
A15	DATA12+CH	CP0 CH0	
A16	DATA13+CH	CP0 CH0	
A17	DATA14+CH	CP0 CH0	
A18	DATA15+CH	CP0 CH0	
C02	DATA16+CH	CP0 CH0	Data Bus (bidirectional signals) Byte 2.
C03	DATA17+CH	CP0 CH0	
C04	DATA18+CH	CP0 CH0	
C05	DATA19+CH	CP0 CH0	
C06	DATA20+CH	CP0 CH0	
C07	DATA21+CH	CP0 CH0	
C08	DATA22+CH	CP0 CH0	
C09	DATA23+CH	CP0 CH0	

Table B.6.1 CP Interface Signals - Y02 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
C11	DATA24+CH	CP0 CH0	Data Bus (bidirectional signals) Byte 3 (most significant).
C12	DATA25+CH	CP0 CH0	
C13	DATA26+CH	CP0 CH0	
C14	DATA27+CH	CP0 CH0	
C15	DATA28+CH	CP0 CH0	
C16	DATA29+CH	CP0 CH0	
C17	DATA30+CH	CP0 CH0	
C18	DATA31+CH	CP0 CH0	
C20	PADD10+00	CP0	Address Bus. These nine addresses are directly provided by the MPU68020 and are the least significant logical addresses.
C21	PADD09+00	CP0	
C22	PADD08+00	CP0	
C23	PADD07+00	CP0	
C24	PADD06+00	CP0	
C25	PADD05+00	CP0	
C26	PADD04+00	CP0	
C27	PADD03+00	CP0	
C28	PADD02+00	CP0	

Table B.6.1 CP Interface Signals - Y02 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
C29	PADD01+00	CP0	Logical address 1.
C30	PADD00+00	CP0	Logical address 0.
C31	MEMDSK+00	CP0	Main Memory Data Acknowledge signal active at level ONE.
C32	CASACT+00	CH0	Cache Asynchronous Service ACTive. Cache busy for Monitor cycle or Flush cycle.
A20	CDSAK0-00	CH0	Cache Data transfer and Size AcKnowledge. Bus response signals that indicate the requested data transfer operation is completed. In addition, these two lines indicate the size of the external bus port on a cycle-by-cycle basis.
A21	CDSAK1-00	CH0	
A23	CHALT-00	CH0	Cache HALT signal. This line is provided during a Rerun cycle or to suspend the MPU68020 bus activity when a cache parity error occurs.
A24	CBERR-00	CH0	Cache BERR signal. This line is provided only during a Rerun cycle.
A26	FCODE0+00	CP0	Function CODEs. These 3-bit function codes are used to identify the address space of each bus cycle.
A27	FCODE1+00	CP0	
A28	FCODE2+00	CP0	
A30	CCDSAK-10	CH0	Cache Combinatory Data transfer and Size AcKnowledge. This signal informs the CP0 that the CH0 will provide the CDSAK0,1 signals.

Table B.6.1 CP Interface Signals - Y02 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
B19	SIZE0+00	CP0	SIZE signals indicate the number of bytes remaining to be transferred for this cycle. These lines, together with PADD00 and PADD01, define the active sections of the data bus.
B20	SIZE1+00	CP0	
B21	RMC-00	CP0	Read Modify write Cycle provides an indicator that the current bus cycle is part of an indivisible read-modify-write operation.
B22	RWMP+00	CP0	Read Write MicroProcessor signal defines the MPU68020 data bus transfer direction.
B24	PAS-00	CP0	Physical Address Strobe indicates that a valid physical address is on the bus.
B26	BCLK1+CH	CP0	16.67 MHz positive clock. This clock is in phase with that one of the MPU68020.
B28	BCLK1-CH	CP0	16.67 MHz negative clocks. These clocks are in phase opposition with that one of the MPU68020.
A32	BCLK2-CH	CP0	

Table B.6.2 CP Interface Signals - Z02 Connector Pin Assignment and Description

Pin N.	Signal Name	Signal Source	Description
A07-C08	ZGND	---	Logic GND level.
A08	PADD11+00	CP0	Logical address 11. This line is provided by the MPU68020.
A09	PADD12+00	CP0	Physical Address Bus. This fourteen adresses are provided by the Memory Management Unit.
A10	PADD13+00	CP0	
A11	PADD14+00	CP0	
A12	PADD15+00	CP0	
A14	PADD16+00	CP0	
A15	PADD17+00	CP0	
A16	PADD18+00	CP0	
C09	PADD19+00	CP0	
C10	PADD20+00	CP0	
C11	PADD21+00	CP0	
C12	PADD22+00	CP0	
C14	PADD23+00	CP0	
C15	PADD24+00	CP0	
C16	PADD25+00	CP0	

Table B.6.2 CP Interface Signals - Z02 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
A06	CNTRG0+00	CP0	Cache Control signals (for the signal descriptions see the Section B.4).
C04	CNTRG1+00	CP0	
C06	STSRG0+00	CH0	Cache Status signals (for the signal descriptions see the Section B.4).
C07	STSRG1+00	CH0	
C05	STSRG2+00	CH0	
C18	STSRG3+00	CH0	
C17	DC64P-00	CP0	This signal is active only when an MPU68020 cycle is referred to the coupled Main Memory.
A04	CDMOWF-00	CH0	Cache Data Monitor OverFloW is provided by the CH0 when an overflow of the monitor FIFO occurs (see also Section B.4).
A05	ZGND	CH0	This ground has the meaning of "cache present" (see also Section B.4).
A03-A17 C03-C19 C20	---	---	Pins not used.

Table B.6.3 VME Bus Interface Signals - Z01 Connector Pin Assignment and Description

Pin N.	Signal Name	Signal Source	Description
A09-A11 A15-A17 A19-B20 B23-C09	ZGND	---	Logic GND level.
A32-B32 C32	ZVP05	---	+5V Voltage Supplay.
B31	ZVP5SB	---	+5V Voltage STANDBY (not used by CH0).
A01-A02 A03-A04 A05-A06 A07-A08 A10-A20 A30-A31 B01-B02 B03-B12 B13-B14 B15-B16 B17-B18 B19-B21 B22-B24 B25-B26 B27-B28 B29-B30 C01-C02 C03-C04 C05-C06 C07-C08 C10-C13 C31	---	---	Pins not used.
A12	VDSTB1-00	VME	Data STroBe 1. Three-state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus lines D08-D15.
A13	VDSTB0-00	VME	Data STroBe 0. Three-state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus

Table B.6.3 VME Bus Interface Signals - Z01 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
A14	VWRITE-00	VME	Three-state driven signal that specifies the data transfer cycle in progress to be either read or write. A HIGH level indicates a READ operation; a LOW level indicates a WRITE operation.
A16	VDTACK-00	VME	Data Transfer ACKnowledge. Open-collector driven signal generated by a VME Bus slave. The falling edge of this signal indicates that valid data is available on the data bus during a read cycle, or that data has been accepted from the data bus during a write cycle.
A18	VADSTB-00	VME	ADdress STroBe. Three-state driven signal that indicates a valid address is on the address bus.
C11	VBUERR-00	VME	BUs ERRor. Open-collector driven signal generated by a slave. This signal indicates that an unrecoverable error has occurred and the bus cycle must be aborted.
C12	VSYRES-00	VME	SYstem RESet. Open-collector driven signal which, when low, will cause the system to be reset.
A21	CIAKCC-00	CH0	This signal puts in short circuit the Interrupt Acknowledge IN and the Interrupt Acknowledge OUT VME signals.
A22	CIAKCC-00	CH0	
B04	CBG0CC-00	CH0	This signal puts in short circuit the Bus Grant 0 IN and the Bus Grant 0 OUT VME signals.
B05	CBG0CC-00	CH0	

Table B.6.3 VME Bus Interface Signals - Z01 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
B06	CBG1CC-00	CH0	This signal puts in short circuit the Bus Grant 1 IN and the Bus Grant 1 OUT VME signals.
B07	CBG1CC-00	CH0	
B08	CBG2CC-00	CH0	This signal puts in short circuit the Bus Grant 2 IN and the Bus Grant 2 OUT VME signals.
B09	CBG2CC-00	CH0	
B10	CBG3CC-00	CH0	This signal puts in short circuit the Bus Grant 3 IN and the Bus Grant 3 OUT VME signals.
B11	CBG3CC-00	CH0	
A23	VADMD4+00	VME	ADdress MoDifier (bits 4 and 5). Three-state driven lines that provide to the CH0 board the VME Bus master type identification.
C14	VADMD5+00	VME	
A29	VADD02+00	VME	VME Address Bus.
A28	VADD03+00	VME	
A27	VADD04+00	VME	
A26	VADD05+00	VME	
A25	VADD06+00	VME	
A24	VADD07+00	VME	
C30	VADD08+00	VME	
C29	VADD09+00	VME	

Table B.6.3 VME Bus Interface Signals - Z01 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
C28	VADD10+00	VME	
C27	VADD11+00	VME	
C26	VADD12+00	VME	
C25	VADD13+00	VME	
C24	VADD14+00	VME	
C23	VADD15+00	VME	
C22	VADD16+00	VME	
C21	VADD17+00	VME	
C20	VADD18+00	VME	VME Address Bus.
C19	VADD19+00	VME	
C18	VADD20+00	VME	
C17	VADD21+00	VME	
C16	VADD22+00	VME	
C15	VADD23+00	VME	

Table B.6.4 VME Bus Interface Signals - Z02 Connector Pin Assignment and Description

Pin N.	Signal Name	Signal Source	Description
A02-A22 A24-A31 A32-B02 B12-B22 B31-C02 C22-C24 C31-C32	ZGND	---	Logic GND level.
A13-A29 A30-B01 B13-B32 C13-C29 C30	ZVP05	---	+5V Voltage Supplay.
A01-A19 A20-A21 A23-A25 A26-A27 A28-B14 B15-B16 B17-B18 B19-B20 B21-B23 B24-B25 B26-B27 B28-B29 B30-C01 C21-C23 C25-C26 C27-C28	---	---	Pins not used.
B03	VRESERV+01	---	Signal line reserved for future VME bus enhancements.

Table B.6.4 VME Bus Interface Signals - Z02 Connector Pin Assignment and Description (Cont'd.)

Pin N.	Signal Name	Signal Source	Description
B04	VADD24+00	VME	VME Extended Address Bus.
B05	VADD25-00	VME	
B06	VADD26+00	VME	
B07	VADD27-00	VME	
B08	VADD28+00	VME	
B09	VADD29+00	VME	
B10	VADD30+00	VME	
B11	VADD31-00	VME	

B.7 CACHE BOARD PHYSICAL DESCRIPTION

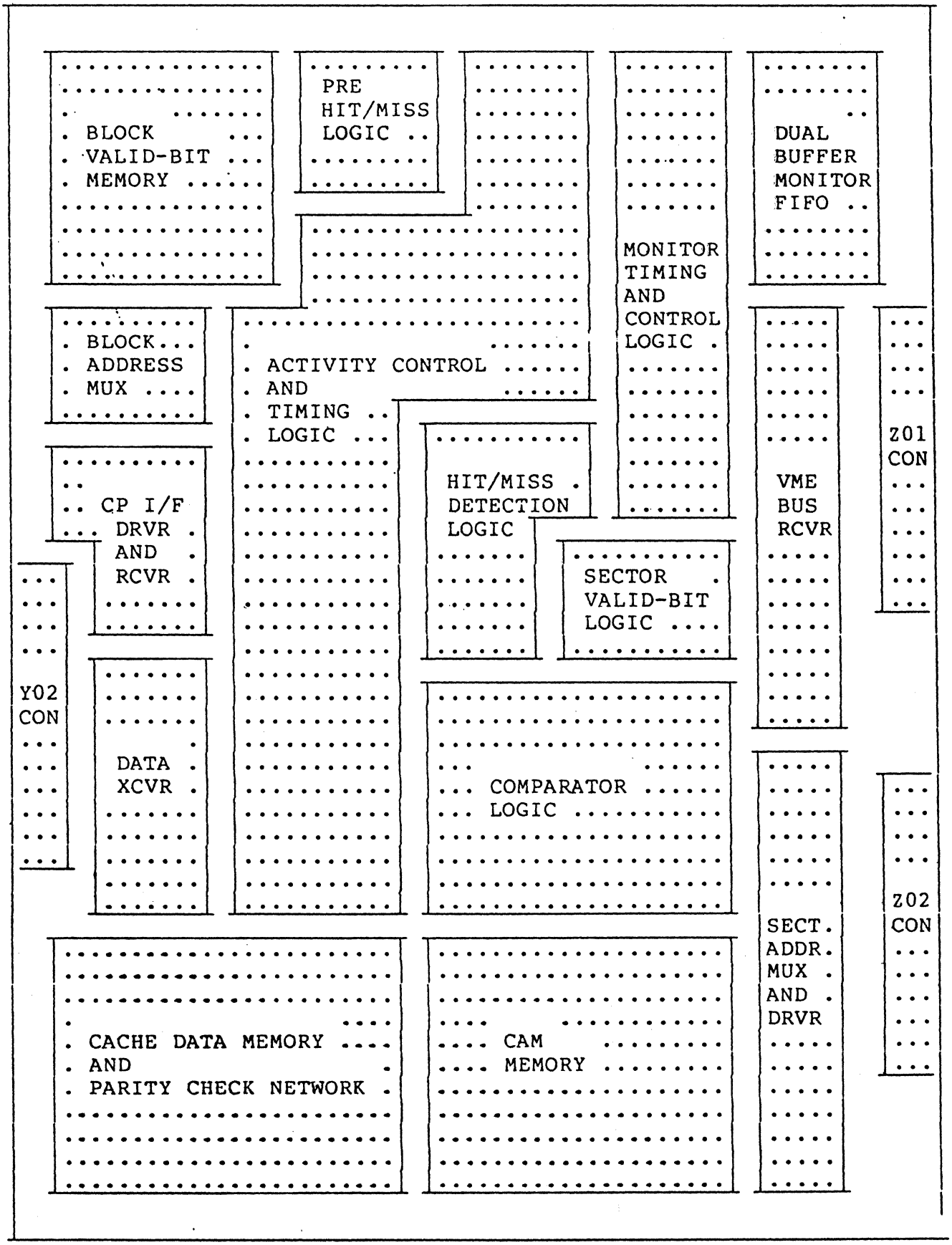
The cache board is an "optional" board of the SGM2 system: up two cache boards can be present in the maximum system configuration.

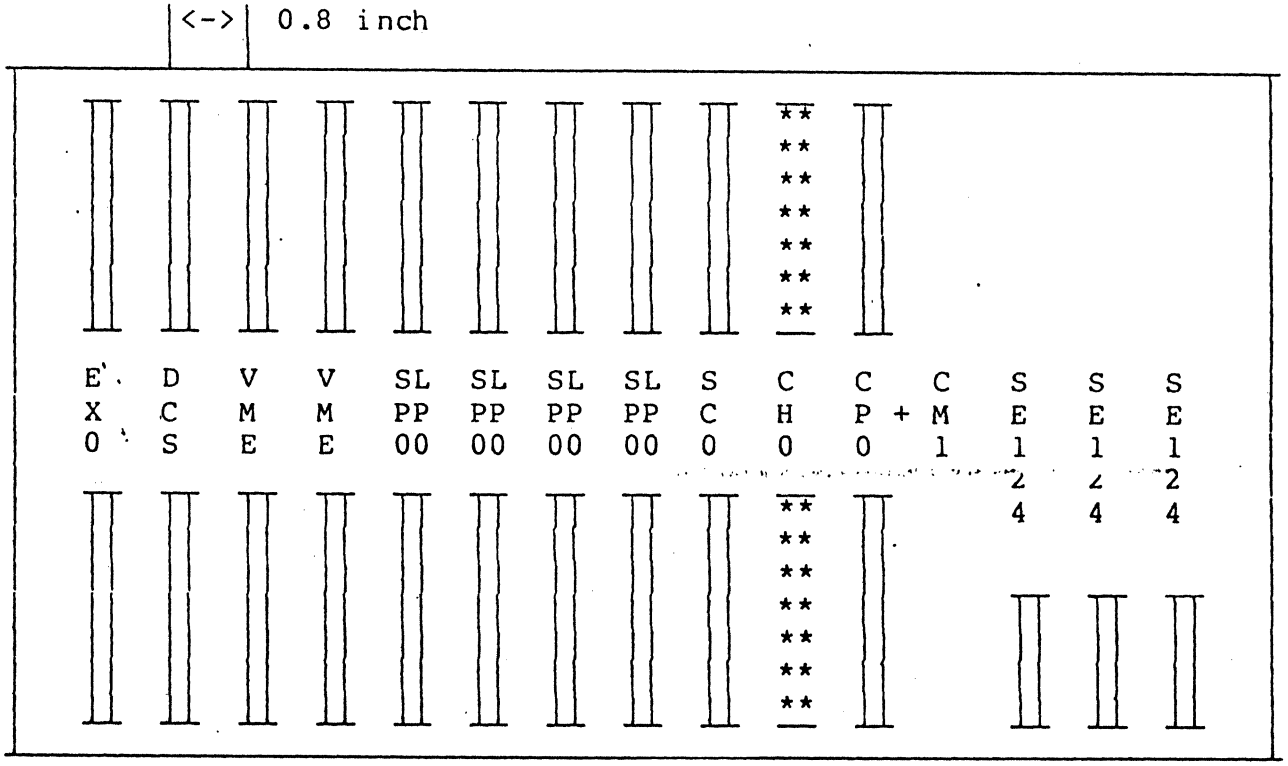
The main physical features of the CH0 board are the following:

- four printed layers;
- about 900 networks;
- 174 physical Dual In line Packages, plus dicrete resistors, capacitors and plus three 96 pin EURODIN connectors;
- about 45 W of power dissipation;
- 14.4 inch X 10.4 inch size board.

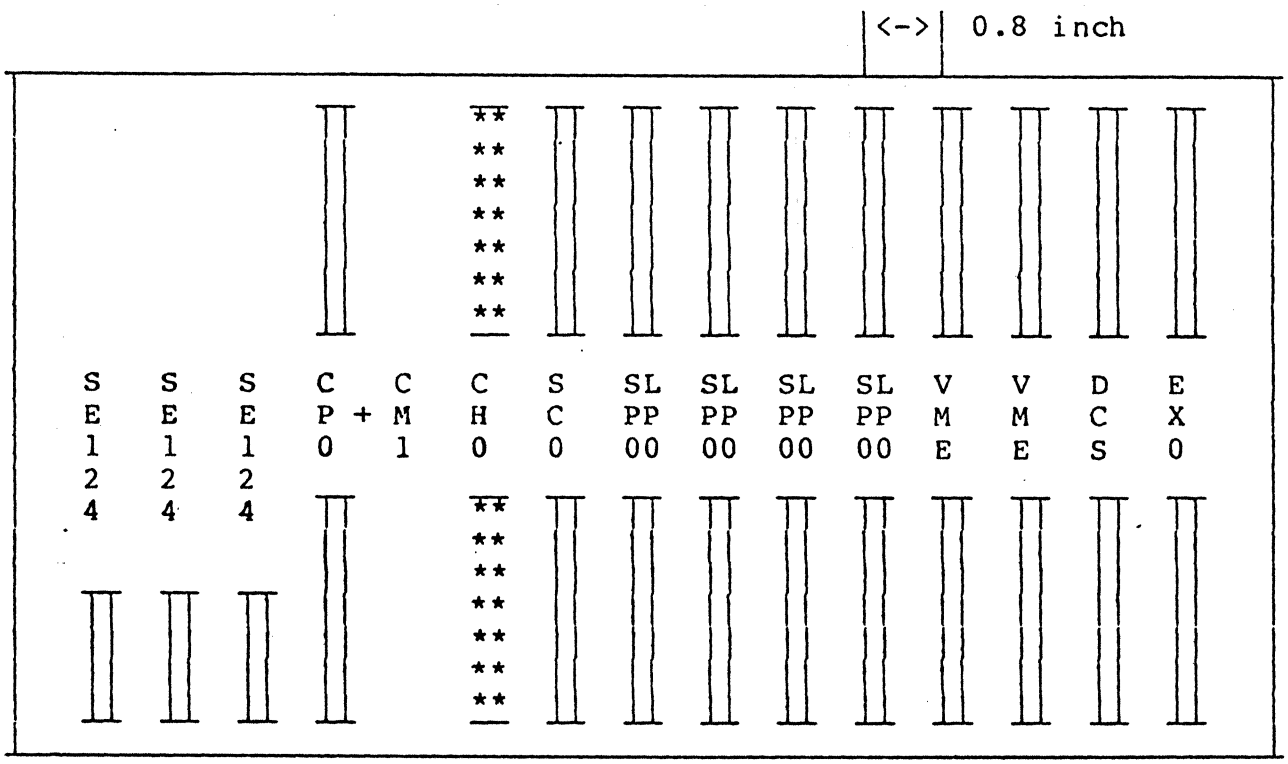
In Fig. B.7.1 is shown the placement of the main logic areas of the cache board.

The cache board is physically connected to the system backplane by the Z01 and Z02 VME Bus connectors from which the board receives also the power supply (+5V). The Fig. B.7.2 illustrates a semplified layout of the SGM2 basic and extended backplanes to show the fixed positions to insert the cache boards. Further, the cache board is connected to the CP board by a jumper printed board that is plugged into the Y02 connectors.


BLOCK VALID-BIT MEMORY
PRE HIT/MISS LOGIC
DUAL BUFFER MONITOR FIFO
MONITOR TIMING AND CONTROL LOGIC
BLOCK ADDRESS MUX
ACTIVITY CONTROL AND TIMING LOGIC
Z01 CON
CP I/F DRVR AND RCVR
HIT/MISS DETECTION LOGIC
VME BUS RCVR
SECTOR VALID-BIT LOGIC
Y02 CON
DATA XCVR
COMPARATOR LOGIC
Z02 CON
SECT. ADDR. MUX AND DRVR
CACHE DATA MEMORY AND PARITY CHECK NETWORK
CAM MEMORY



a) Basic (PWA insertion view)



b) Extended (PWA insertion view)

Fig. B.7.2 SGM2 Basic and Extended Backplanes

SECTION C

SYSTEM CONTROLLER (SC0)

PRODUCT DESIGN DESCRIPTION

PREPARED BY: M. FOLDES

REVIEWED BY: A. LAZZARI

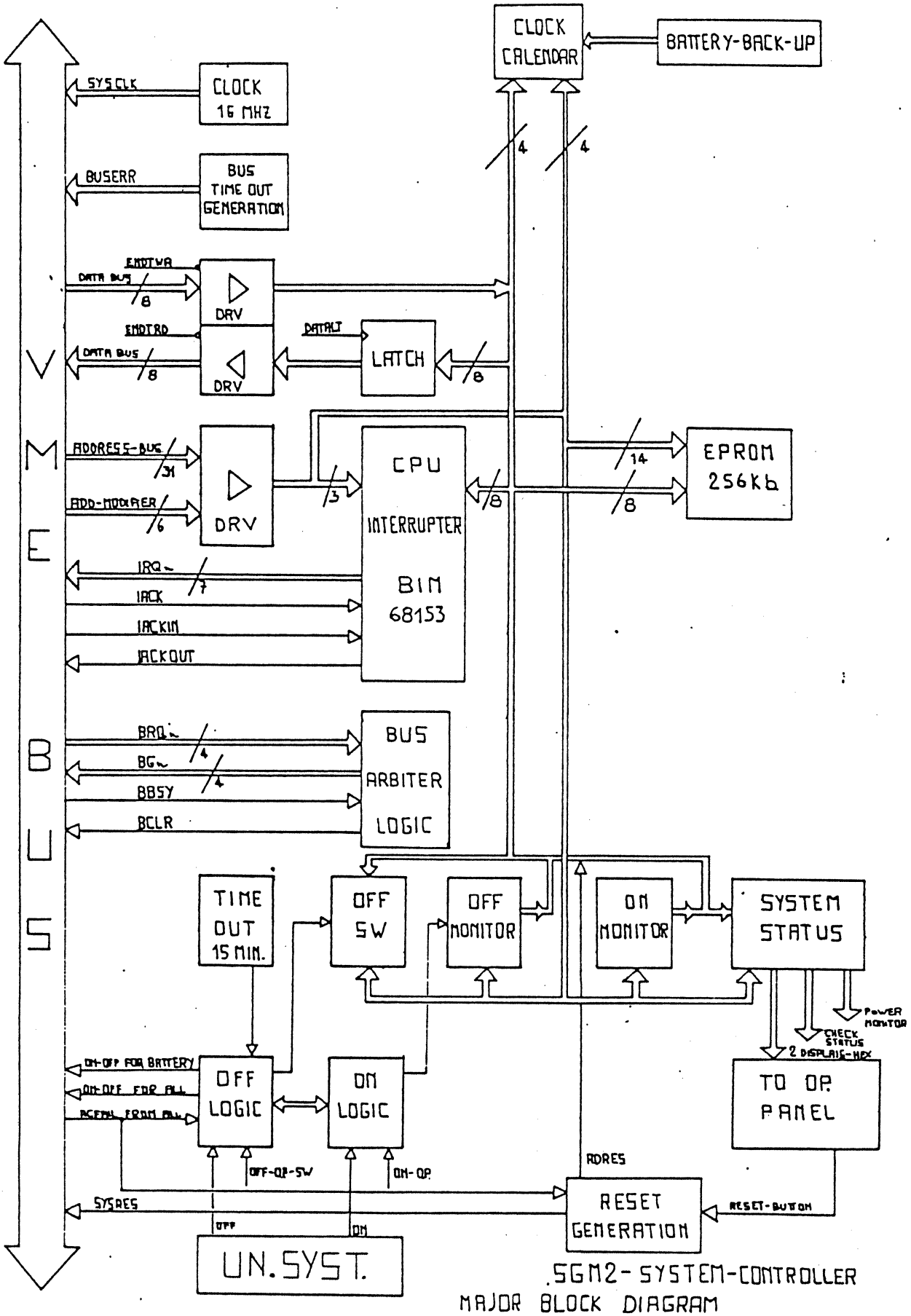
INDEX OF SECTION C

		PAGE
C.1	SYSTEM CONTROLLER OVERVIEW	142
C.2	MAJOR BLOCKS DESCRIPTION	144
C.2.1	16.00-MHz SYMMETRICAL SYSTEM CLOCK	144
C.2.2	SYSTEM RESET GENERATION	144
C.2.3	AC-FAIL GENERATION	144
C.2.4	BUS TIME-OUT GENERATION	145
C.2.5	BUS ARBITER IN FIXED-PRIORITY	145
C.2.6	CPU INTERRUPTER	145
C.2.7	CLOCK CALENDAR	145
C.2.8	OPERATOR PANEL INTERFACE	146
C.2.9	UNATTENDED SYSTEM INTERFACE	146
C.2.10	POWER SUPPLY INTERFACE	146
C.2.10.1	ON GENERATION	146
C.2.10.2	OFF GENERATION	147
C.2.11	PORTABLE EPROM	148
C.2.12	INTERNAL MEMORY MAPS	148
C.2.12.1	BUS INTERRUPTER REGISTER	149
C.2.12.2	HEX DISPLAY	149
C.2.12.3	SW REGISTER	149
C.2.12.4	ON OFF STATUS REGISTER	150
C.2.12.5	CLOCK CALENDAR REGISTERS	151
C.2.12.6	32 KBYTES PORTABLE EPROM	151

C.1 SYSTEM CONTROLLER OVERVIEW

The SC0 is a single-module system controller that provides general system utilities necessary for VMEbus-compatible modules.
In addition the controller provides the connection with the Operator Panel, the Power Supply Module and the Unattended System.

The SC0 was specially designed to be the System Controller Board of the HISI SGM2 Microcomputer.



SGM2-SYSTEM-CONTROLLER
MAJOR BLOCK DIAGRAM

C.2 MAJOR BLOCKS DESCRIPTION

The figure C.1 shows a Major Blocks Diagram of the SC0 System Controller.

C.2.1 16.00 MHz SYMMETRICAL SYSTEM CLOCK

The clock signal is provided per the VMEbus specification. SYSCLK is a 16.00-MHz symmetrical clock signal used for general system timing.

C.2.2 SYSTEM RESET GENERATION

An active-low system reset (SYSRES*) signal, approximately 300 milliseconds in duration, is generated by any of the following conditions:

- Power up;
- Closing of a manually operated switch located on the operator panel;
- Before a power-off system request.

C.2.3 AC-FAIL GENERATION

The external AC-FAIL input (from now on referred as EX-FAIL*) is provided on connector P2 of the VMEbus and is driven by the system Power Supply.

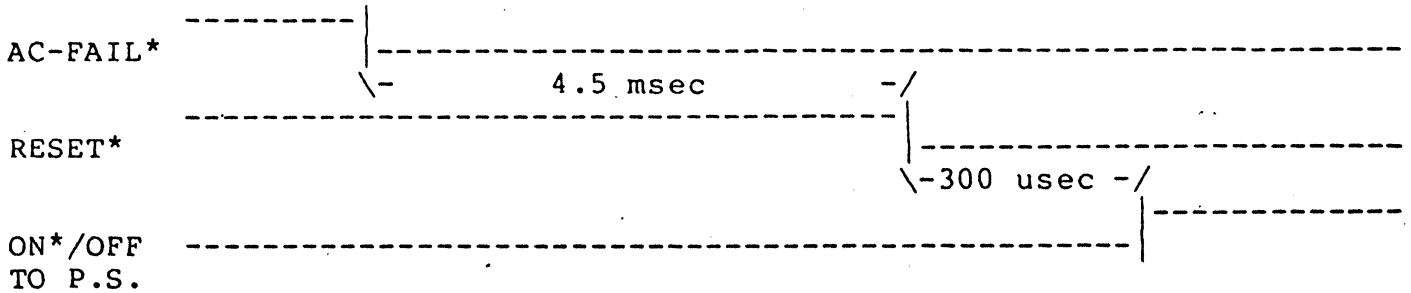
When the EX-FAIL* goes true, or after a power-off request, the SC0 will respond by driving the VMEbus AC-FAIL* line low.

On AC-FAIL* true, caused by an EX-FAIL*, after 4.5 milliseconds the SYSRES* signal will be driven low by the SC0 and after 300 usec a power-off command will be asserted to the Power Supply.

When the cause is a power-off request the time between AC-FAIL* low and SYSRES* low can be greater (see chapter C.2.10).

It is possible to read in a proper register the AC-FAIL cause.

The AC-FAIL timing is the following:



C.2.4 BUS TIME-OUT GENERATION

The SC0 contains a bus timer that controls if a bus-cycle is longer then 60 microseconds.
At the end of this time, BERR is driven low and will remain low until AS*, DS1* and DS0* are all inactive, regardless the state of DTACK*.

C.2.5 BUS ARBITER IN FIXED PRIORITY

The bus arbitration subsystem is designed to prevent simultaneous access of the VMEbus by two or more masters; it schedules request from multiple masters for optimum bus usage.
The arbiter supports all four levels of bus request, with a typical turnaround time of 100 nsec.
The arbitration is in fixed-priority (PRI).

In PRI mode, the arbiter grants the bus to the highest pending level of request, where BR3* is the highest and BR0* is the lowest.
When a master has been granted the bus, the master indicates that it has control of the bus by driving BBSY* low. To prevent a master from locking the bus, the arbiter will drive the BCLR* signal low upon receipt of a request on a level higher than that of the current bus master. The present master should then relinquish the bus in a reasonable period of time.

C.2.6 CPU INTERRUPTER

The SC0 contains also the CPU-interrupter module.
For this purpose the MC68153 Bus Interrupter Module (BIM) is used. It handles up to 4 independent sources of interrupt requests on four different VMEbus interrupt levels. When the system interrupt handler begins an interrupt acknowledge cycle, the BIM can answer supplying an interrupt vector and handling all the handshake.
The SC0 uses the MC68153 in a daisy-chain configuration.

C.2.7 CLOCK CALENDAR

The VMEbus clock calendar is made using a MM58274 Microprocessor Real Time Clock. It has the following features:

- Timekeeping from tenths of seconds to tens of years in independently accesible registers;
- Leap year register;
- Hours counter programmable for 12 or 24 hour operation;
- Buffered crystal frequency output in test mode for easy oscillator setting.

A battery back-up is provided to grant calendar supply during the system stand-by.
At this cause a 3V lithium battery is installed on SC0 board.

C.2.8 OPERATOR PANEL INTERFACE

SC0 has a connector for operator panel interface.
It handles:

- Eight data for two hex-display (TIL 311) and their blinking pin;
- The On and Stand-by buttons;
- The informations for AC present LED, DC on LED, BATTERY on LED, STND-BY in progress LED, HW check LED and SW check LED;
- The reset button.

C.2.9 UNATTENDED SYSTEM INTERFACE

A proper connector is provided to connect the SC0 to a remote station that can give an ON or STAND BY command to the system.

C.2.10 POWER SUPPLY INTERFACE

The SC0 System Controller provides the Power Supply Module interface.

The interface signals provided on the VMEbus SC0 P2 connector are:

- EX-FAIL*
- ON*/OFF
- 5VS (5 Volts DC, present with AC on)

C.2.10.1 ON GENERATION

The causes of an ON request to Power Supply can be two:

a- from button on operator panel
b- from Unattended System

When one of the previous causes has occurred, the ON*/OFF line is driven low. The power goes on and when the 5V DC is stabilized, the EX-FAIL* is driven high by the Power Supply.

The CPU can then read in a proper register the cause of the ON command. A three seconds time-out controls the correct system power-up, and if in this time the DC does not go on, the system must be reset putting the AC breaker in the off position and then it is possible to retry the start-up. A failure on the retried start-up can be caused by a Power Supply malfuction.

The status of the voltages is monitored on the operator panel Leds.

C.2.10.2 OFF GENERATION

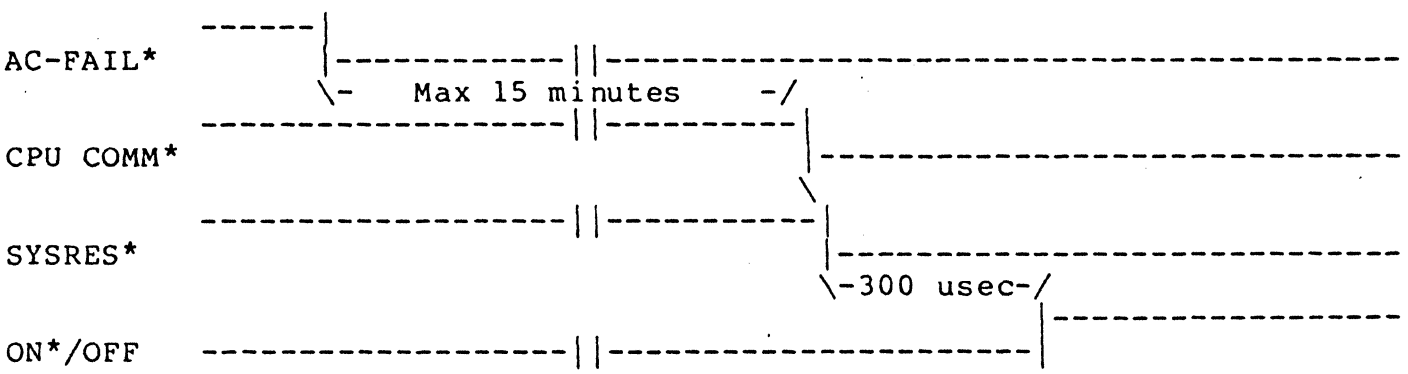
The causes of an OFF request to Power Supply can be three:

a- from button on operator panel
b- from US
c- from an AC failure

When one of the previous causes has occurred, the VMEbus ACFAIL* line is driven low. The CPU can read a register to identify who generated the Stand-by request. In a AC failure case there are 4.5 msec. before the SYSRES* is generated, and after 300 usec a power-off command will be generated to the Power Supply. In the other cases the available time is greater (up to 15 minutes) and the real off command is given by the CPU (CPU COMM*), writing to a proper register when a graceful shut-down is made. In any case, after 15 minutes from the OFF request a Time-out logic gives however the OFF command to the Power Supply.

The information that a Stand-by command is pending is monitored on a proper Led on the operator panel.

The OFF command for the points a- and b- is the following:



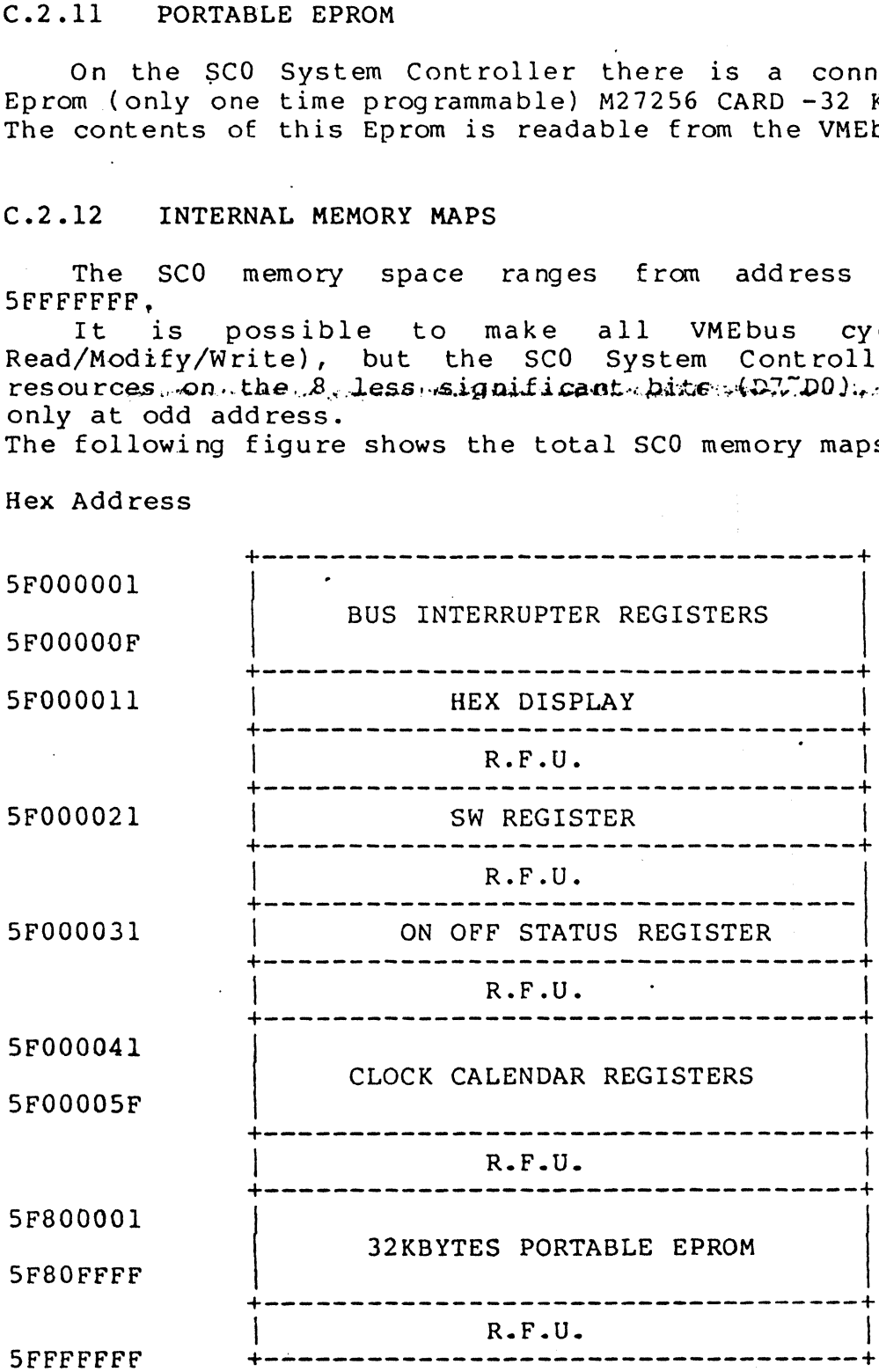
C.2.11 PORTABLE EPROM

On the SC0 System Controller there is a connector for a portable Eprom (only one time programmable) M27256 CARD -32 KBytes-.
The contents of this Eprom is readable from the VMEbus.

C.2.12 INTERNAL MEMORY MAPS

The SC0 memory space ranges from address 5F000000 to address 5FFFFFFF.

It is possible to make all VMEbus cycles (Read, Write, Read/Modify/Write), but the SC0 System Controller presents all its resources on the 8 less significant bits (D7~D0), so it is addressable only at odd address.
The following figure shows the total SC0 memory maps:



C.2.12.1 BUS INTERRUPTER REGISTERS

ADDRESS	REGISTER NAME	ACC	INIT. VAL.
$5F000001	CONTROL REGISTER 0	R/W	0000 0000
$5F000003	CONTROL REGISTER 1	R/W	0000 0000
$5F000005	CONTROL REGISTER 2	R/W	0000 0000
$5F000007	CONTROL REGISTER 3	R/W	0000 0000
$5F000009	VECTOR REGISTER 0	R/W	0000 1111
$5F00000B	VECTOR REGISTER 1	R/W	0000 1111
$5F00000D	VECTOR REGISTER 2	R/W	0000 1111
$5F00000F	VECTOR REGISTER 3	R/W	0000 1111

C.2.12.2 HEX DISPLAY

ADDRESS $5F000011

D7	D6	D5	D4	D3	D2	D1	D0
most significant hex-display value				less significant hex-display value			

Write only register

C.2.12.3 SW REGISTER

ADDRESS $5F000021

XX	XX	XX	XX	D3	D2	D1	D0

D3 = SPARE
D2 = SPARE
D1 = SW CHECK (0=on 1=off)
D0 = DISPLAY BLINKING (0=on 1=off)

Initial value XXXX XX00
Write only register

C.2.12.4 ON OFF STATUS REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
$5F000031								

IN READ

D7 = RESET CAUSE (1=button 0=power-on)

D6 = OFF REQUEST BY AC-FAIL (1=true 0=false)
D5 = OFF REQUEST BY UNATTENDED SYSTEM (1=true 0=false)
D4 = OFF REQUEST BY BUTTON (1=true 0=false)

D3 = ON REQUEST STATUS BUTTON (1=open 0=closed)

D2 = R.F.U.

D1 = ON REQUEST BY UNATTENDED SYSTEM (1=true 0=false)
D0 = ON REQUEST BY BUTTON (1=true 0=false)

IN WRITE

D7~D0 = MUST BE ZERO

****
A WRITE CYCLE TO THIS ADDRESS CAUSE A SYSTEM SHUT-OFF.
****

C.2.12.5 CLOCK CALENDAR REGISTERS

ADDRESS	REGISTER NAME	ACC	SIGN. BIT
$5F000041	CONTROL REGISTER	R	D3 DX DX D0
$5F000041	CONTROL REGISTER	W	D3 D2 D1 D0
$5F000043	TENTHS OF SECONDS	R	D3 D2 D1 D0
$5F000045	UNITS OF SECONDS	R/W	D3 D2 D1 D0
$5F000047	TENS OF SECONDS	R/W	D3 D2 D1 D0
$5F000049	UNITS OF MINUTES	R/W	D3 D2 D1 D0
$5F00004B	TENS OF MINUTES	R/W	D3 D2 D1 D0
$5F00004D	UNITS OF HOURS	R/W	D3 D2 D1 D0
$5F00004F	TENS OF HOURS	R/W	D3 D2 D1 D0
$5F000051	UNITS OF DAYS	R/W	D3 D2 D1 D0
$5F000053	TENS OF DAYS	R/W	D3 D2 D1 D0
$5F000055	UNITS OF MONTHS	R/W	D3 D2 D1 D0
$5F000057	TENS OF MONTHS	R/W	D3 D2 D1 D0
$5F000059	UNITS OF YEARS	R/W	D3 D2 D1 D0
$5F00005B	TENS OF YEARS	R/W	D3 D2 D1 D0
$5F00005D	DAY OF WEEK	R/W	D3 D2 D1 D0
$5F00005F	CLOCK SETTING/INT. REGISTER	R/W	D3 D2 D1 D0

NB: DX = MUST BE ZERO

C.2.12.6 32KBYTES PORTABLE EPROM

FIRST ADDRESS $5F800001
SECOND ADDRESS $5F800003
.....
LAST ADDRESS $5F80FFFF

SECTION D

STATION PROCESSOR (SP0)

PRODUCT DESIGN DESCRIPTION

PREPARED BY: P. GONELLLA

REVIEWED BY: A. GRASSI

INDEX OF SECTION D

STATION PROCESSOR

		PAGE
D.1	GENERAL DESCRIPTION	154
D.2	HARDWARE DESCRIPTION	155
D.2.1	I/O MICROPROCESSOR	164
D.2.2	INT & INTA LOGIC	165
D.2.3	SERIAL INPUT OUTPUT	168
D.2.3.1	CONFIGURATION	169
D.2.3.2	ELECTRICAL INTERFACE	176
D.2.4	TIMER AND PARALLEL PRINTER INTERFACE	177
D.2.5	VME BUS INTERRUPTER LOGIC (BIL)	187
D.2.6	ATTENTION LOGIC AND SYSTEM FAIL DETECTION	190
D.2.7	PROCESSOR NUMBER AND BOARD TYPE DETECTION	192
D.3	VME BUS PIN ASSIGNMENT	194

D.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.

D.2 HARDWARE DESCRIPTION

The Major Block Diagram of the Station Processor 0 board is described in details in Fig. D.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. D.2.1);

- INT & INTA LOGIC

  this logic permits to handle the interrupt lines (see para. D.2.2.);

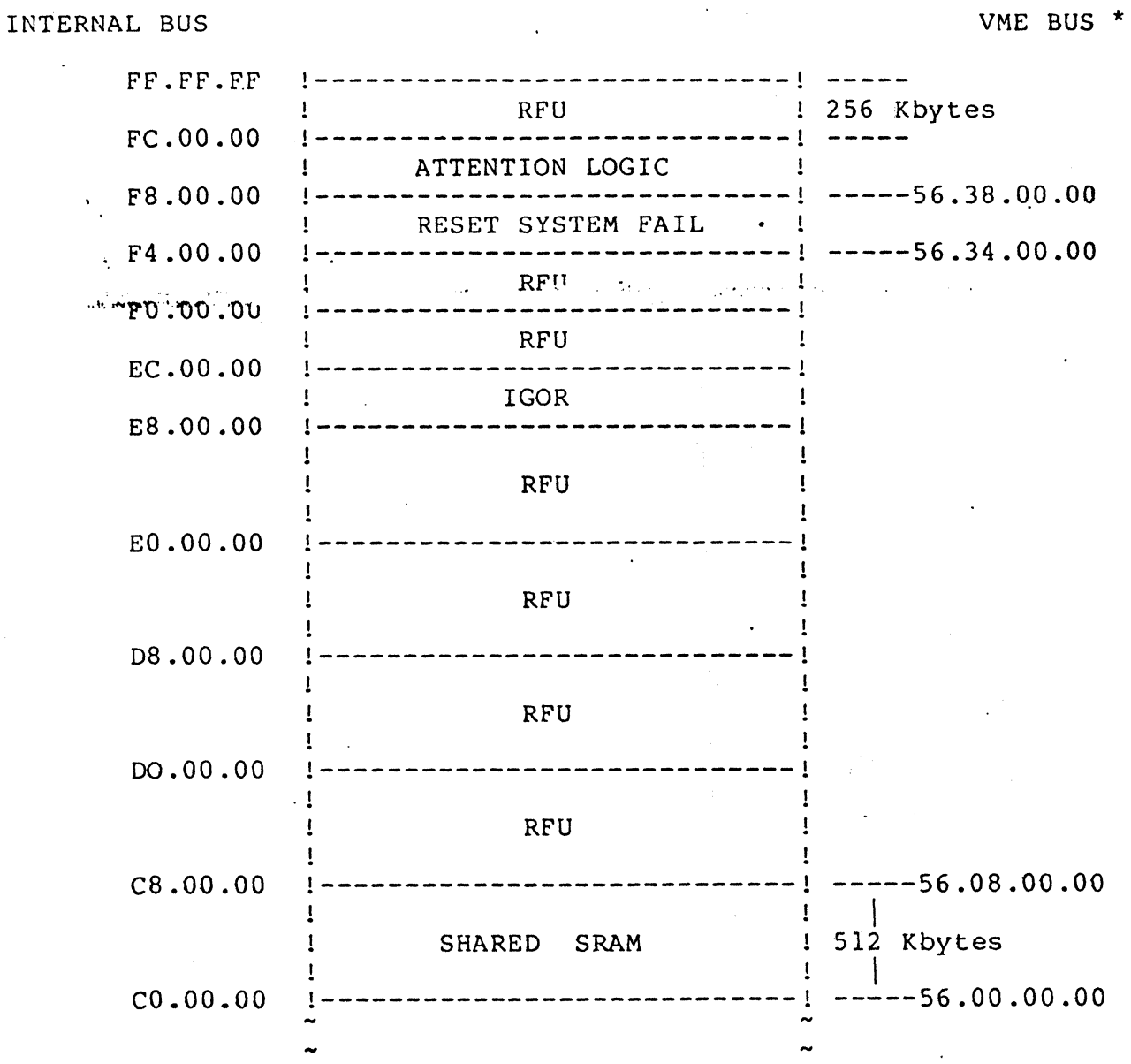
- CHIP SELECT AND CONTROL LOGIC FOR LOCAL/SHARED AREA

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

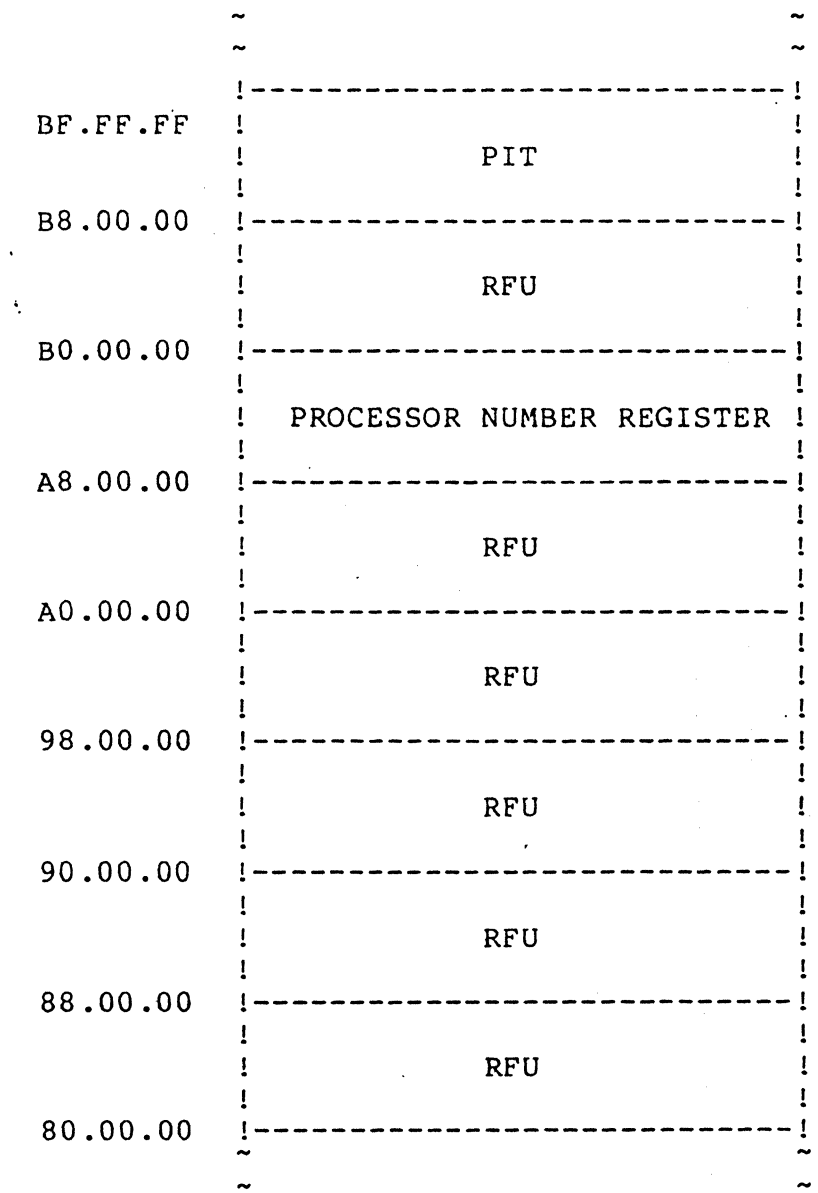
  - generation of the chip-selects;
  - generation of control signals (for example, the READ and WRITE clocks of the memory, etc);
  - generation of DATA TRANSFER ACKNOWLEDGE signal towards MC68000;
  - generation of a time-out signal (BERR line) every 16 us if on-board peripheral don't return DTACK signal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

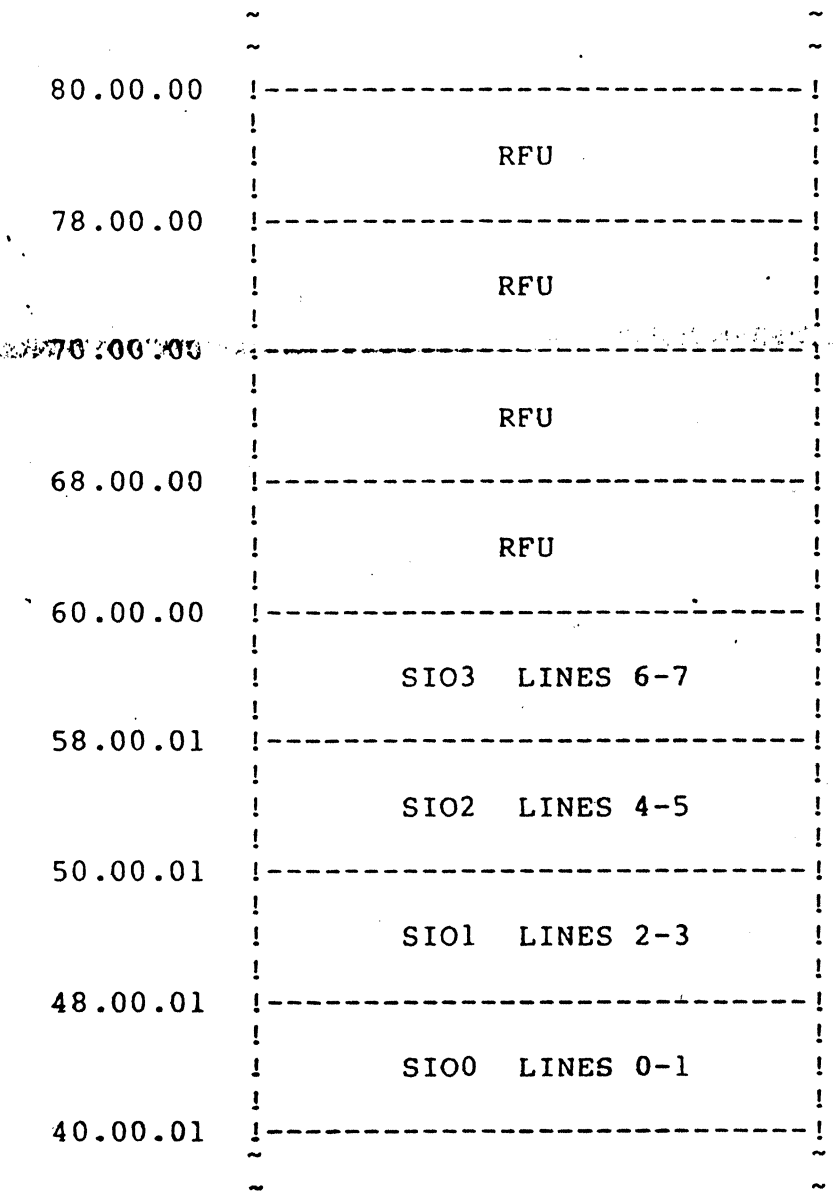
The 16 Mbyte addressing space is subdivided as shown below:

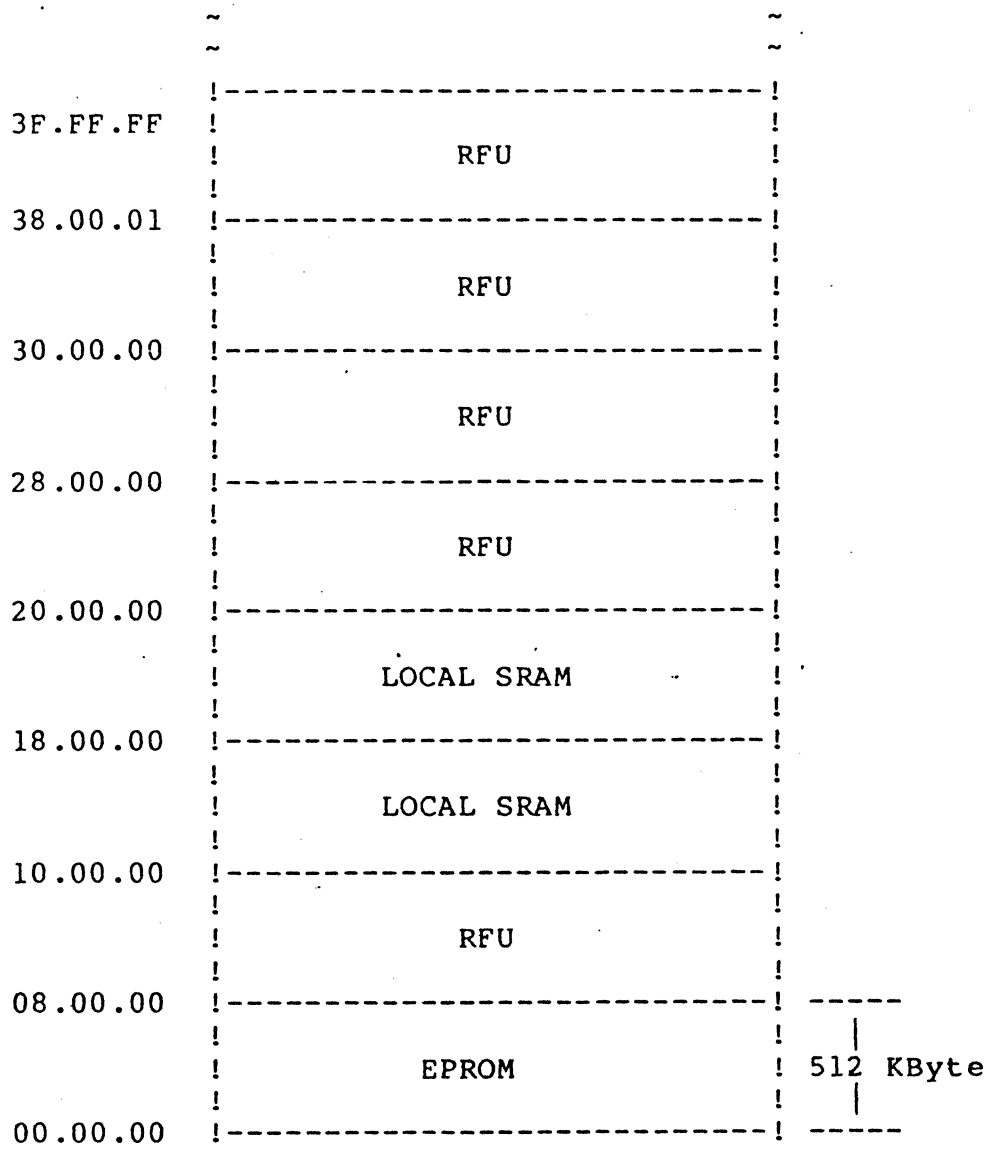
STATION PROCESSOR 0 - MASTER PROCESSOR MEMORY MAP



* Map for the first Station Processor with the Processor Number 0. For the others see para. D.2.7 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


80.00.00
RFU
78.00.00
RFU
70.00.00
RFU
68.00.00
RFU
60.00.00
SIO3 LINES 6-7
58.00.01
SIO2 LINES 4-5
50.00.01
SIO1 LINES 2-3
48.00.01
SIO0 LINES 0-1
40.00.01


3F.FF.FF
RFU
38.00.01
RFU
30.00.00
RFU
28.00.00
RFU
20.00.00
LOCAL SRAM
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 D.4 (MC68230 Register Model).

- SERIAL INPUT OUTPUT

four USART (SIO) have been used. Each of these controllers can handle two Serial Ports (see para D.2.3);

- 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. D.2.4). PIT is also utilized for diagnostics purposes;

- 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 D.4 (MC68230 Register Model).

- VME BUS INTERRUPTER LOGIC
  this logic acts as an interrupt requester on VMEbus (see para. D.2.5);

- ATTENTION LOGIC

  this logic permits the system CPUs to interrupt the SP (see para. D.2.6);

- 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. D.2.7).

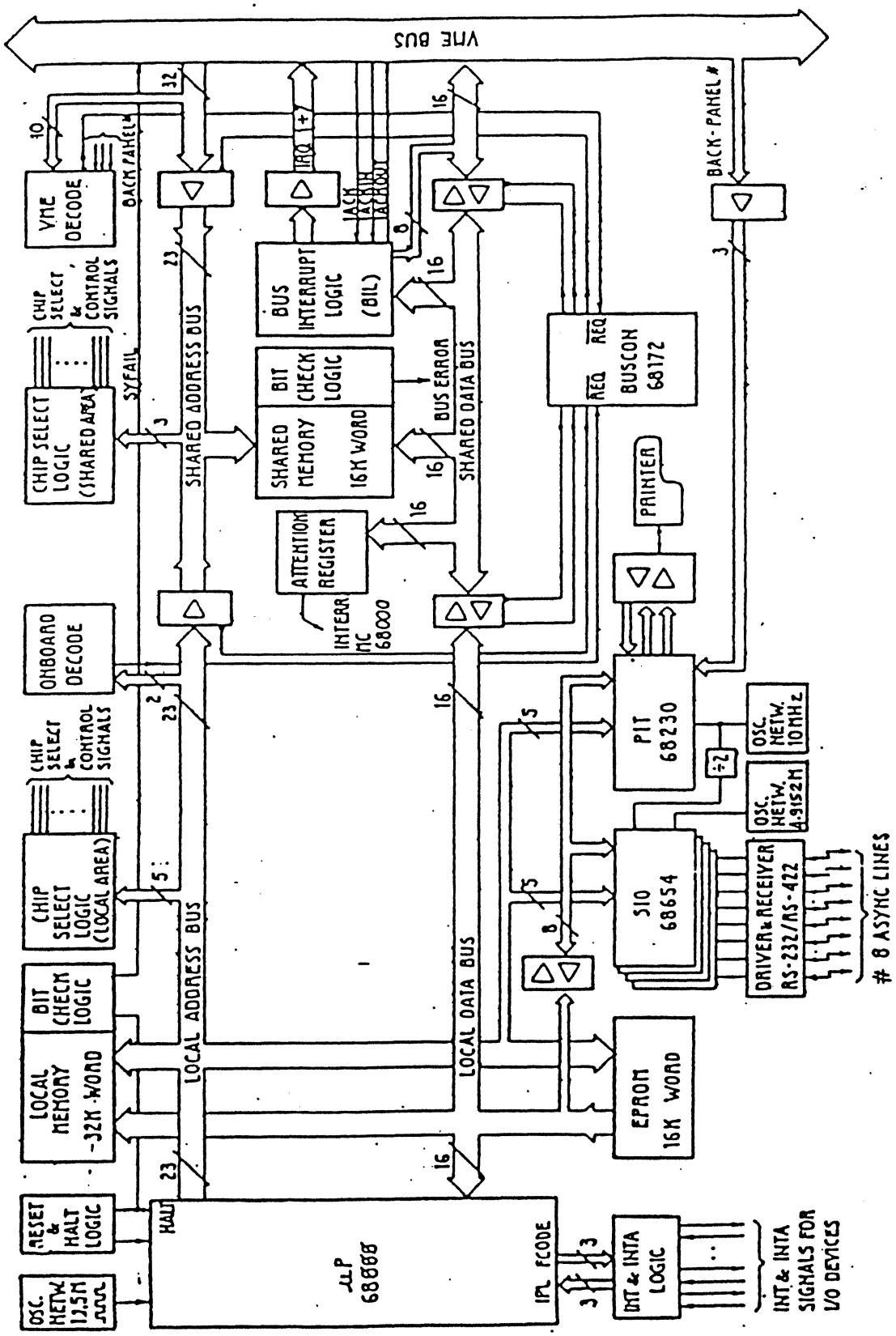


Fig. D.1 - SPO Major Block Diagram

D.2.1 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..

D.2.2 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		

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.

D.2.3 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

D.2.3.1 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 38.400 Bps)
          ! LINE#2 > ASYNCHRONOUS
          !                              \ RS-422A (up to 19.200 Bps)
    SIO1  <
          !                              / RS-232C (up to 38.400 Bps)
          ! LINE#3 > ASYNCHRONOUS
          \                              \ RS-422A (up to 19.200 Bps)

          /                              / RS-232C (up to 38.400 Bps)
          ! LINE#4 > ASYNCHRONOUS
          !                              \ RS-422A (up to 19.200 Bps)
    SIO2  <
          !                              / RS-232C (up to 38.400 Bps)
          ! LINE#5 > ASYNCHRONOUS
          \                              \ RS-422A (up to 19.200 Bps)

          /                              / RS-232C (up to 38.400 Bps)
          ! LINE#6 > ASYNCHRONOUS
          !                              \ RS-422A (up to 19.200 Bps)
    SIO3  <
          !                              / RS-232C (up to 38.400 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)
BIT RATE	76800	16	10	4	1	1	4
	38400	32	20	4	2	2	4
	19200	64	40	4	4	4	4
	9600	128	80	4	8	8	4
	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:

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

(*) 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.

D.2.3.2 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.

D.2.4 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.

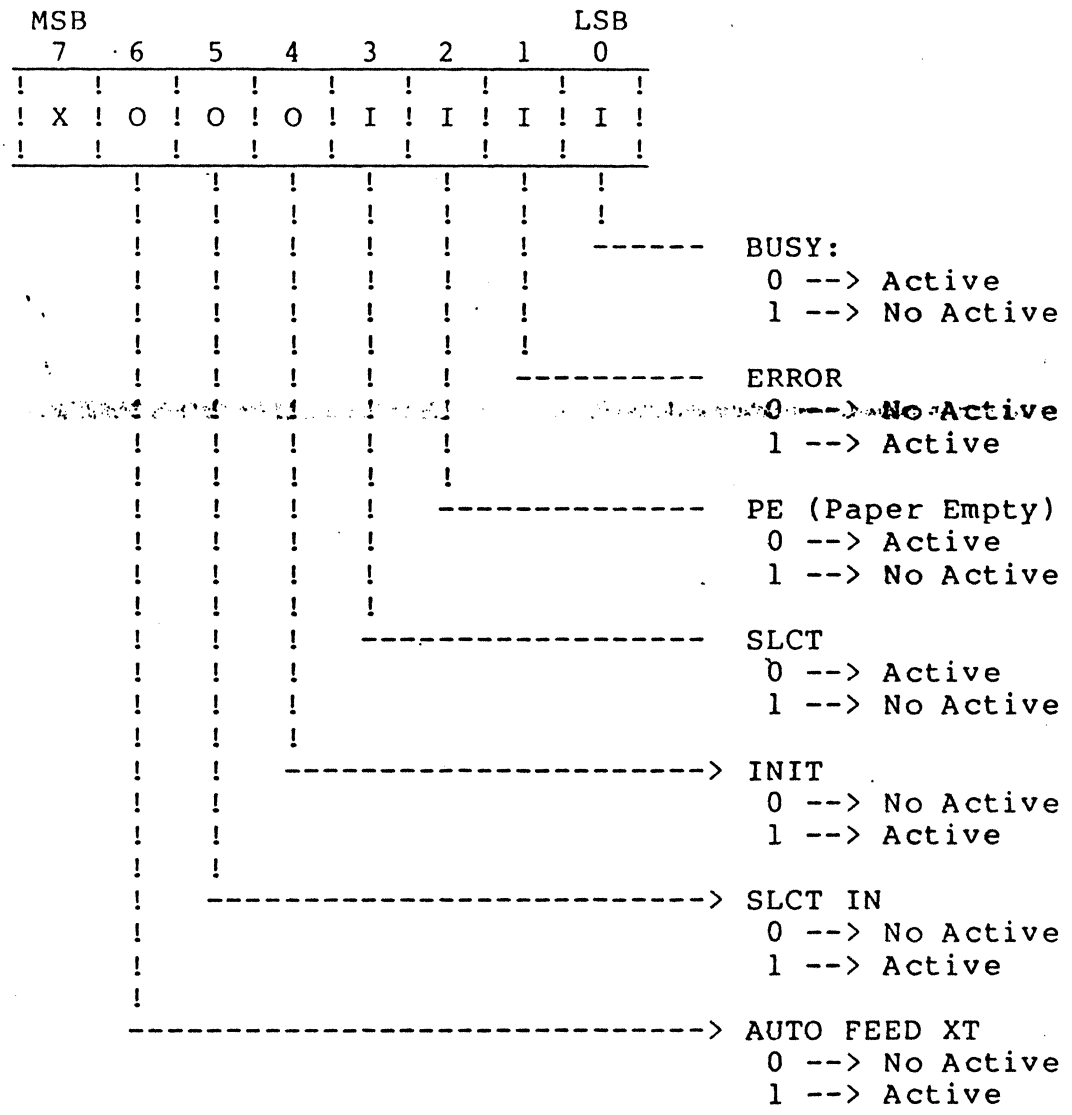
A register model that includes the corresponding Register Selects is shown in Fig. D.4. For further information refer to "16-BIT MICROPROCESSOR DATA MANUAL - 1983 MOTOROLA" pag. 4-509 - 4-537.

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. D.2 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. D.3 which shows the timings relevant to CENTRONICS and IBM interfaces).

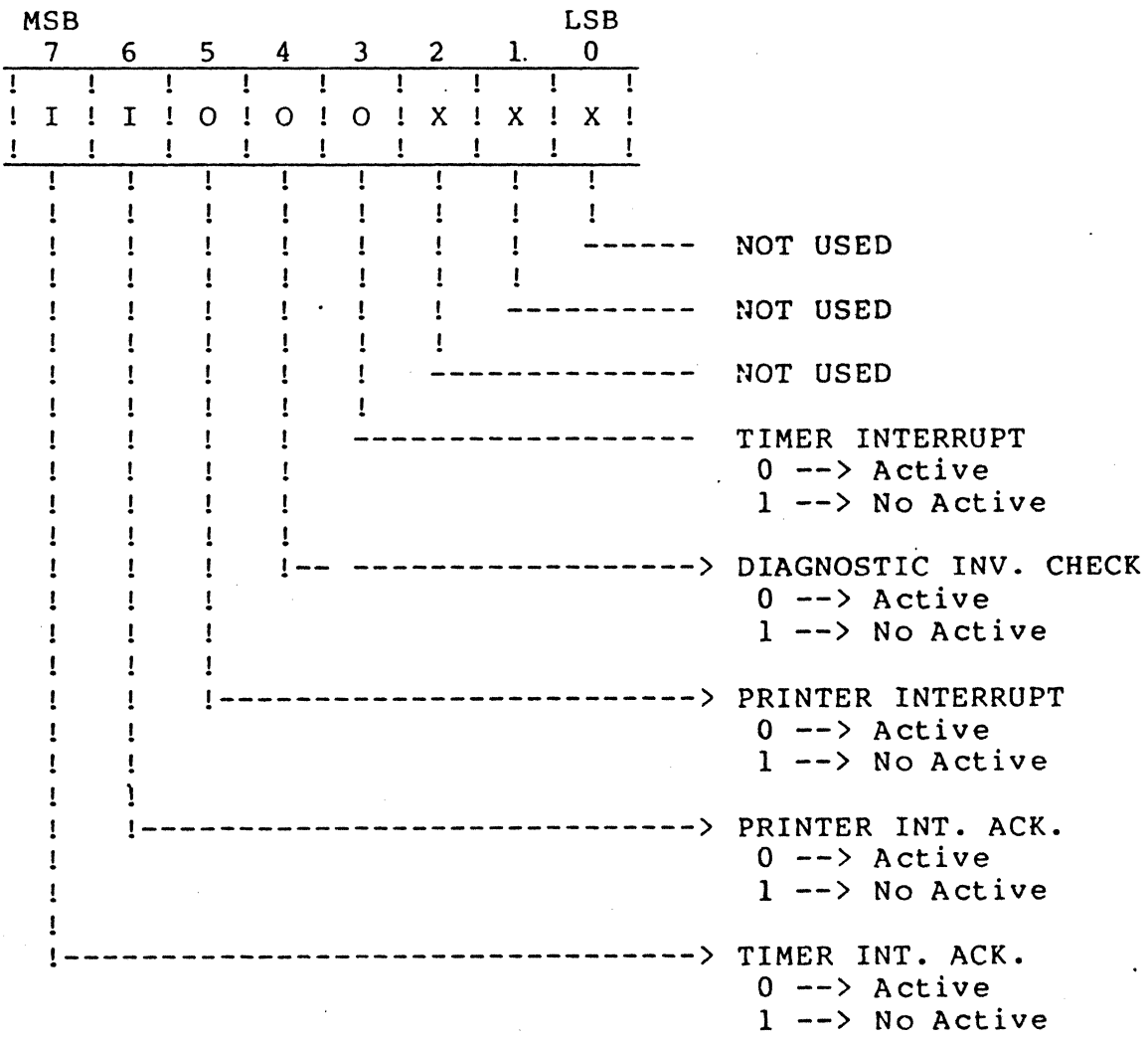
A register model that includes the corresponding Register Selects is shown in Fig. D.4. For further information refer to "16-BIT MICROPROCESSOR DATA MANUAL - 1983 MOTOROLA" pag. 4-509 - 4-537.

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 D.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)



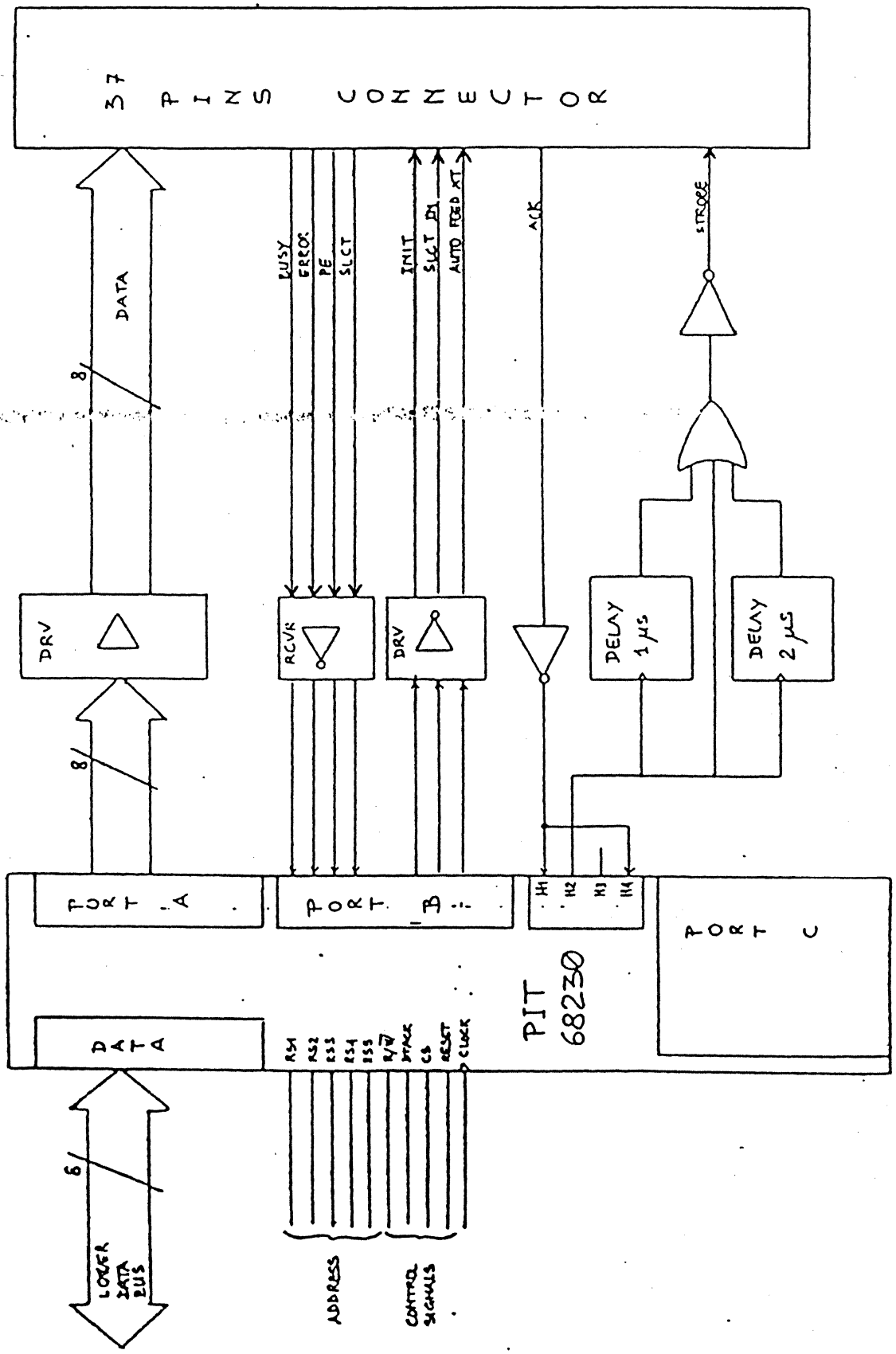
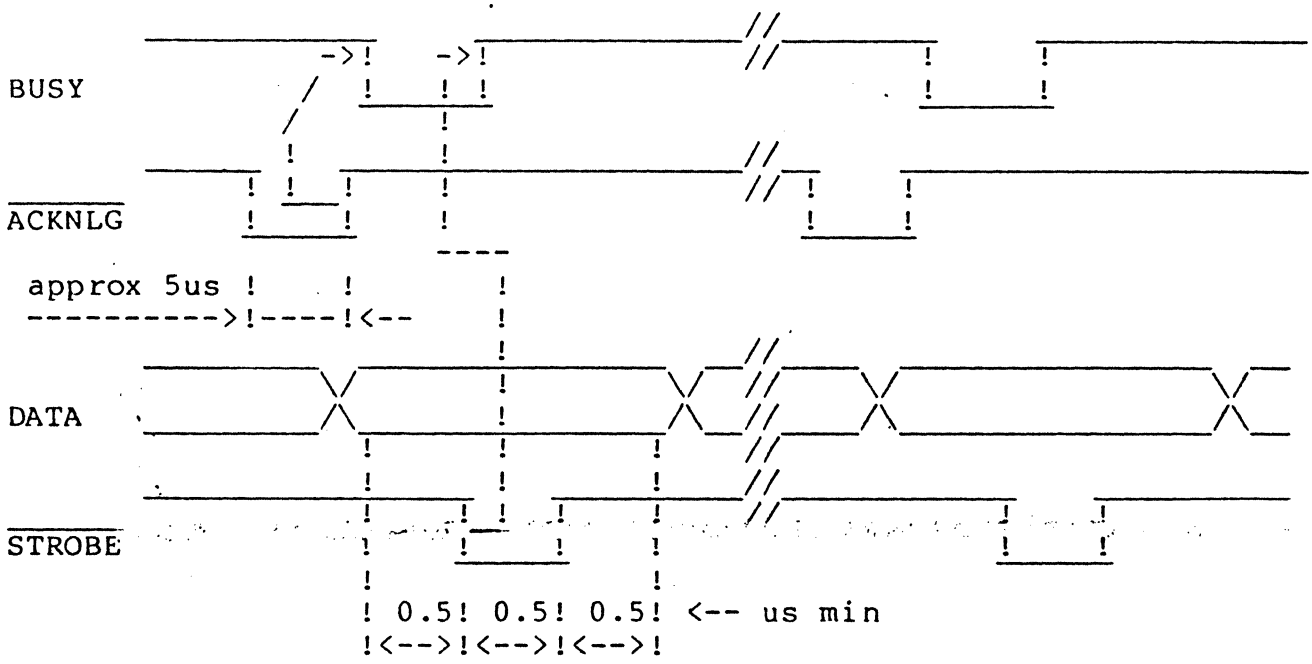
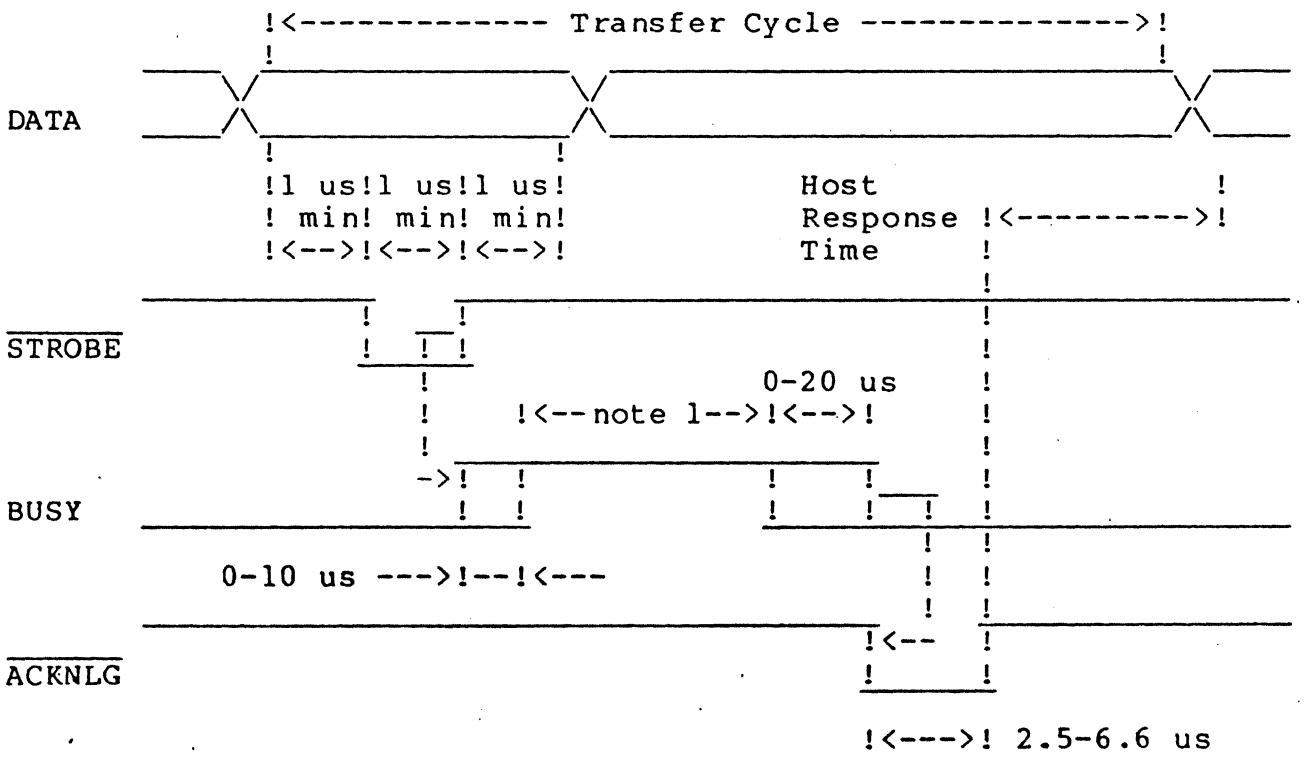


Fig. D.2 - Parallel Printer Interface Block Diagram



a) IBM Interface Timing



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

b) CENTRONICS Interface Timing

Fig. D.3 - Parallel Printer Interface Timing Diagram

ADDRESS (Hex)	7	6	5	4	3	2	1	0	
B8.00.01	Port Mode Control		H3= Enable	H2? Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	Port General Control Register
B8.00.03	*	SVCRO 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 SVCRO Enable	H1 Stat Ctrl	Port A Control Register
B8.00.0F	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRO 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/TIACK Control			Z O 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	*	*	*	*	*	*	*	ZOS	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. D-4 MC68230 Register Model

Table D.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. D.3 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. D.3 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. D.3 are shown timing considerations for IBM and CENTRONICS interfaces.

Table D.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. D.3 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 D.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 D.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.

D.2.5 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.

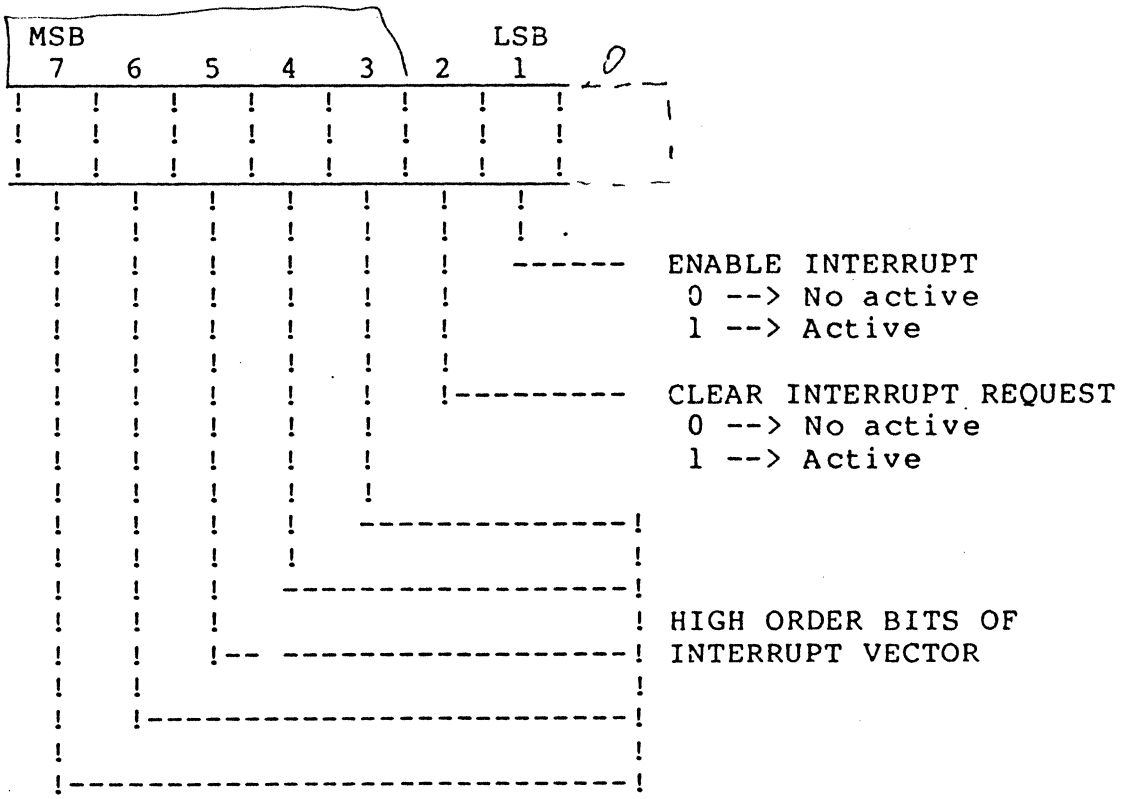
Inside it has two registers:

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

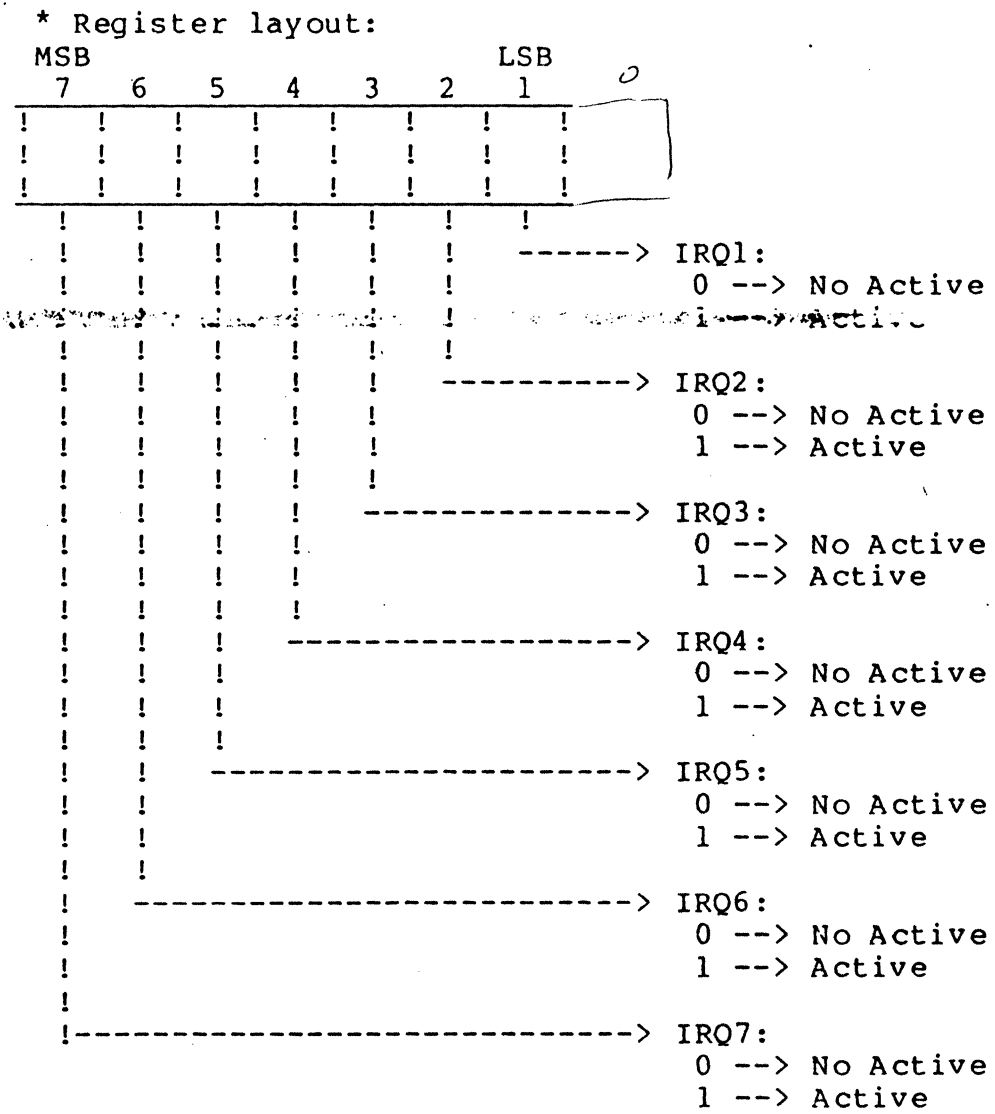
- the INTERRUPT REQUEST REGISTER R1 (located at Hex E8.00.05)

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).

D.2.6 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:

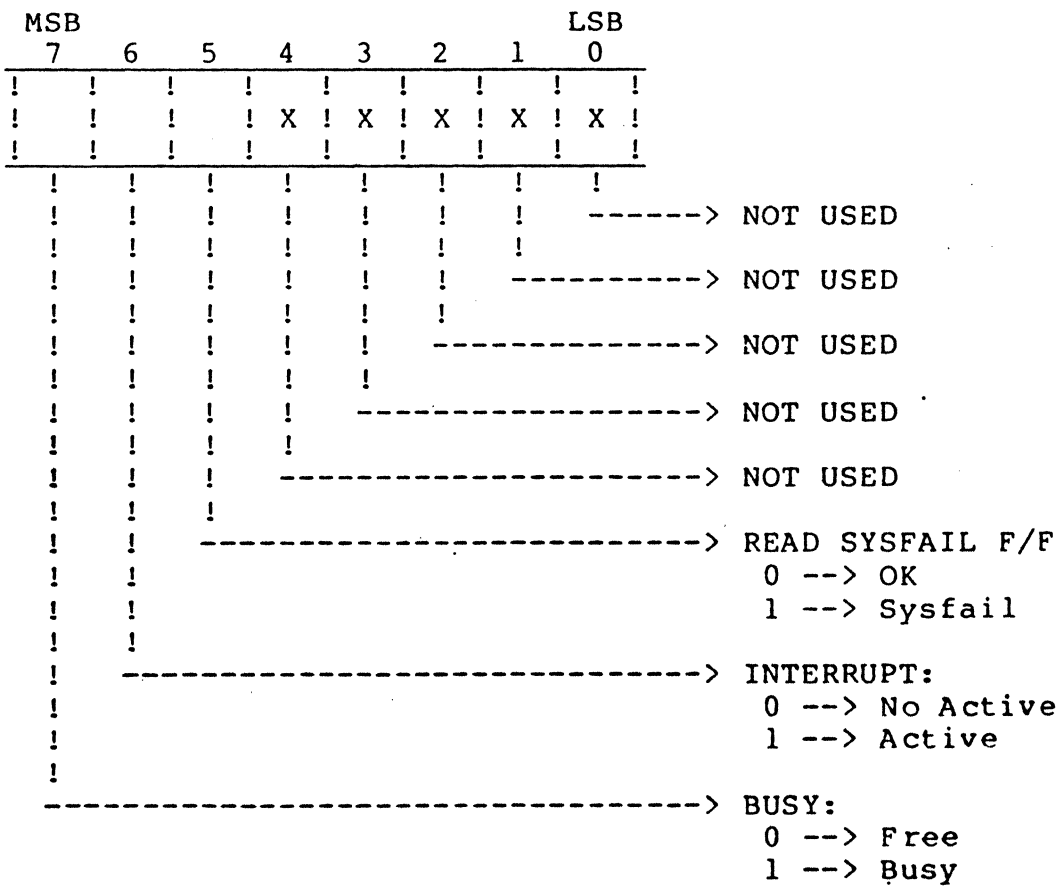
- ATTENTION INTERRUPT REGISTER:

  * Write/Read 8 bit register.

  * Access mode:

    ADDRESS: F8.00.00 Hex.
    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.

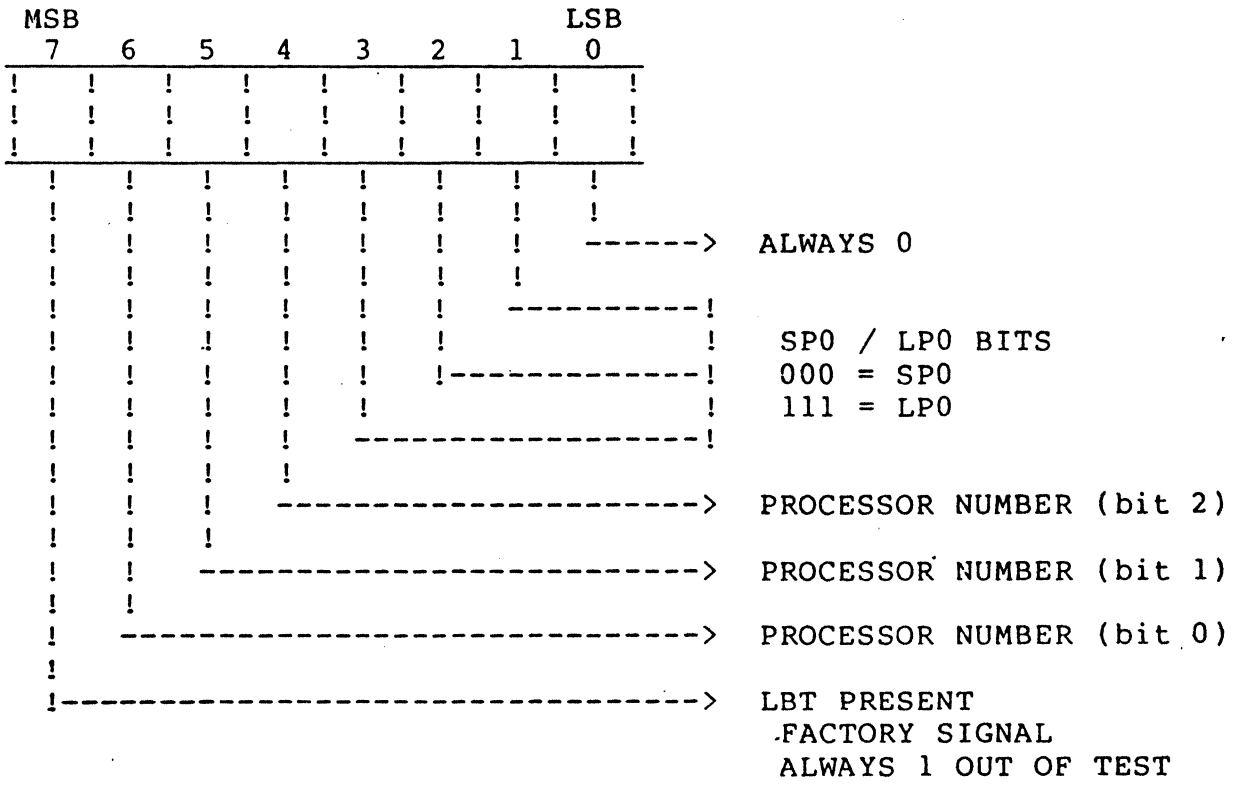
The system fail condition is detectable in the bit 5 of the ATTENTION INTERRUPT REGISTER . The software running in the SGM2 CPU can know which SP0 board in the VME Bus is in the system fail condition.

D.2.7 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);
- testing 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)

D.3 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	DO3
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

SECTION E

LINE PROCESSOR (LP0)

PRODUCT DESIGN DESCRIPTION

PREPARED BY: P. GONELLA

REVIEWED BY: A. GRASSI

INDEX OF SECTION E

		PAGE
E.1	GENERAL DESCRIPTION	202
E.2	HARDWARE DESCRIPTION	203
E.2.1	MASTER AND SLAVE MICROPROCESSORS	216
E.2.2	MASTER AND SLAVE INT & INTA LOGIC	218
E.2.3	SERIAL INPUT OUTPUT	224
E.2.3.1	CONFIGURATION	225
E.2.3.2	ELECTRICAL INTERFACE	232
E.2.3.3	MODEM SIGNALS	239
E.2.4	TIMER AND PARALLEL PRINTER INTERFACE	241
E.2.5	VME BUS INTERRUPTER LOGIC (BIL)	251
E.2.6	ATTENTION LOGIC AND SYSTEM FAIL DETECTION	254
E.2.7	PROCESSOR NUMBER AND BOARD TYPE DETECTION	256
E.3	DMA EMULATOR INTERFACE	258
E.4	VME BUS PIN ASSIGNEMENT	269

E.1 GENERAL DESCRIPTION

The Line Processor 0 is an intelligent communication processor with SGM2-VME bus interface, which supports up to 6 full duplex channels with RS-232C and RS-422A interfaces, for serial communications and one Centronics/IBM parallel printer interface.

In VME enviroment the Line 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 LP0 can take place in three ways:

-- from host to LP0 or vice versa by message interchange via a shared-RAM allocated on the LP0;

-- from host to LP0 by writing an 1-bit attention register to interrupt the LP0;

-- from LP0 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 LP consists of 32Kbytes EPROM area, 512Kbytes of Local Dinamic Memory, 64Kbytes of Shared Sram Memory and 16Kbytes of I/O Sram Memory.

E.2 HARDWARE DESCRIPTION

The Major Block Diagram of the LINE Processor 0 board is described in details in Fig. E.1 where it is possible to see the following functional areas:

- the local area;
- the I/O area;
- the shared area;

The main functional blocks are:

- MASTER AND SLAVE MICROPROCESSORS

  the microprocessors used are the 16 bit Motorola MC68000. They operate at 12.5 Mhz clock and they have an addressing capability up to 16 Mbytes (see para. E.2.1); in this board one is master, the other is slave and it emulates a DMA.

- MASTER AND SLAVE INT & INTA LOGIC

  this logic permits to handle the interrupt lines (see para. E.2.2.);

  NOTIFY INTERRUPTS.

  by this logic the slave processor can interrupt the master processor and viceversa. (see para. E.2.2);

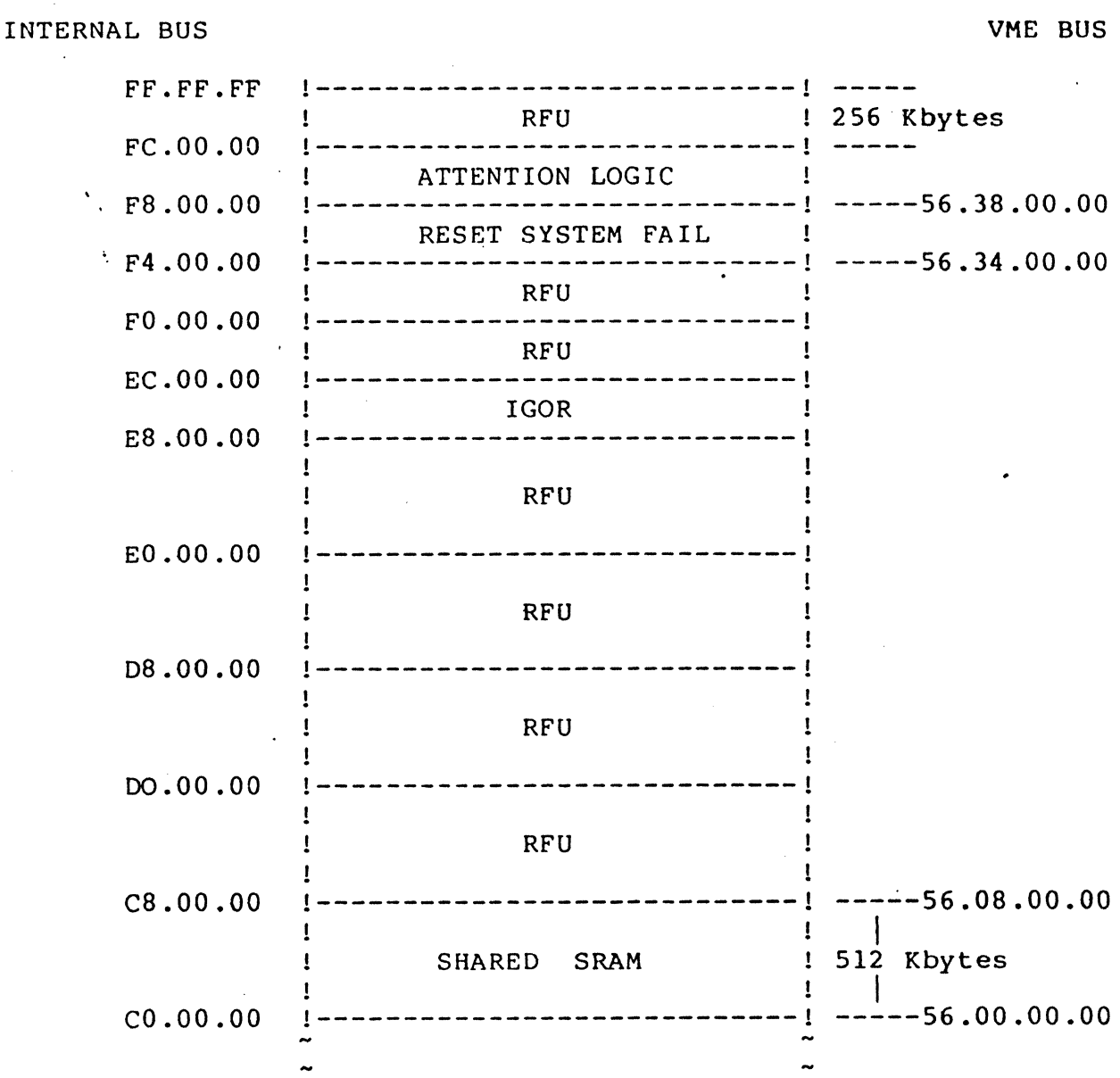
- CHIP SELECT AND CONTROL LOGIC FOR I/O, LOCAL AND SHARED AREA

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

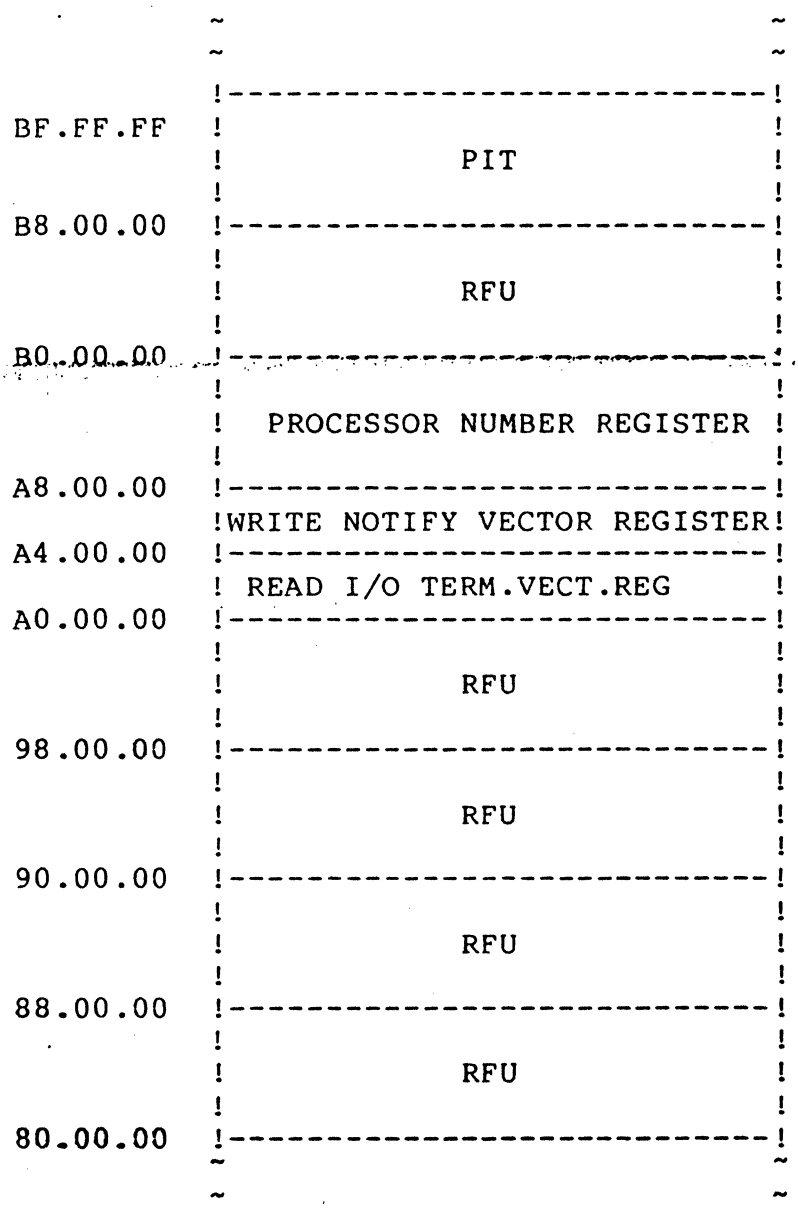
  - generation of the chip-selects;
  - generation of control signals (for example, the READ and WRITE clocks of the memory, etc);
  - generation of DATA TRANSFER ACKNOWLEDGE signal towards MC68000;
  - generation of a time-out signal (BERR line) every 16us if on-board peripheral don't return DTACK signal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

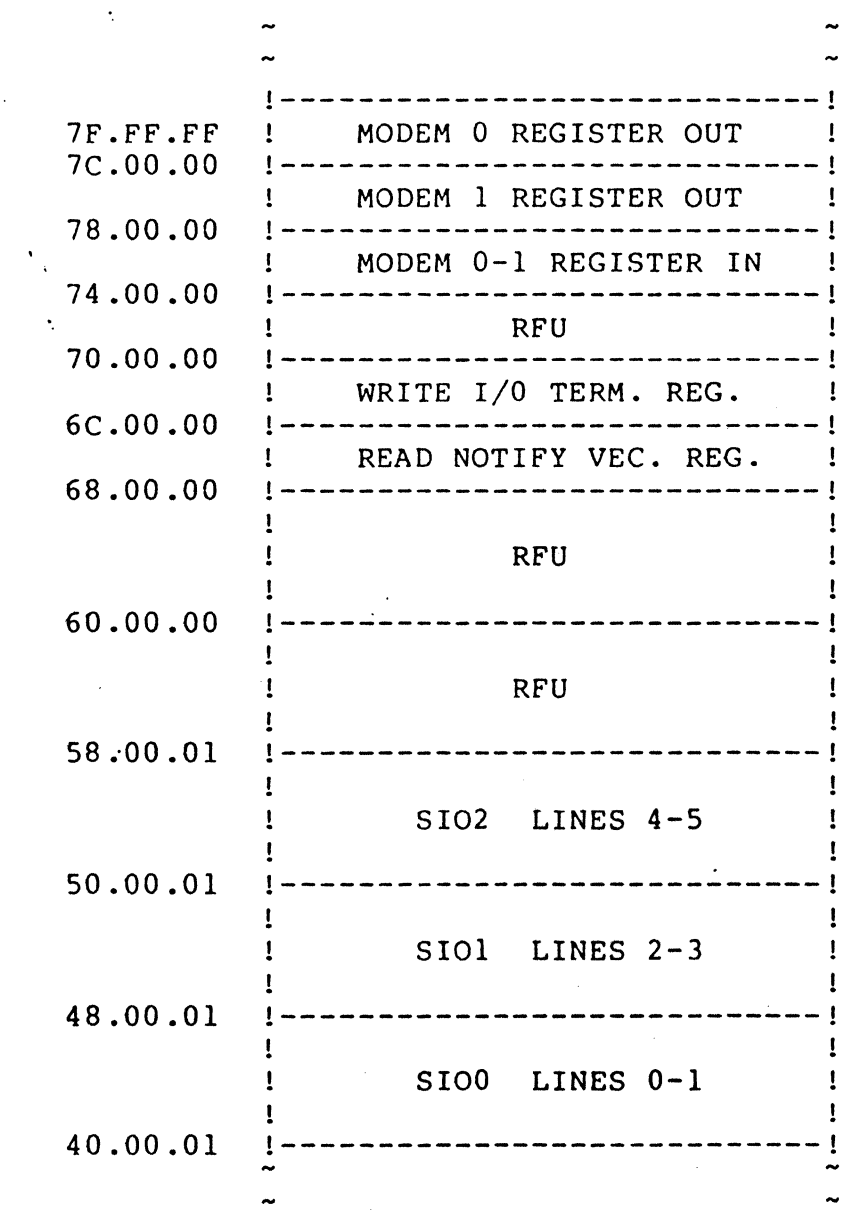
The 16 Mbyte addressing spaces of the processors are subdivided as

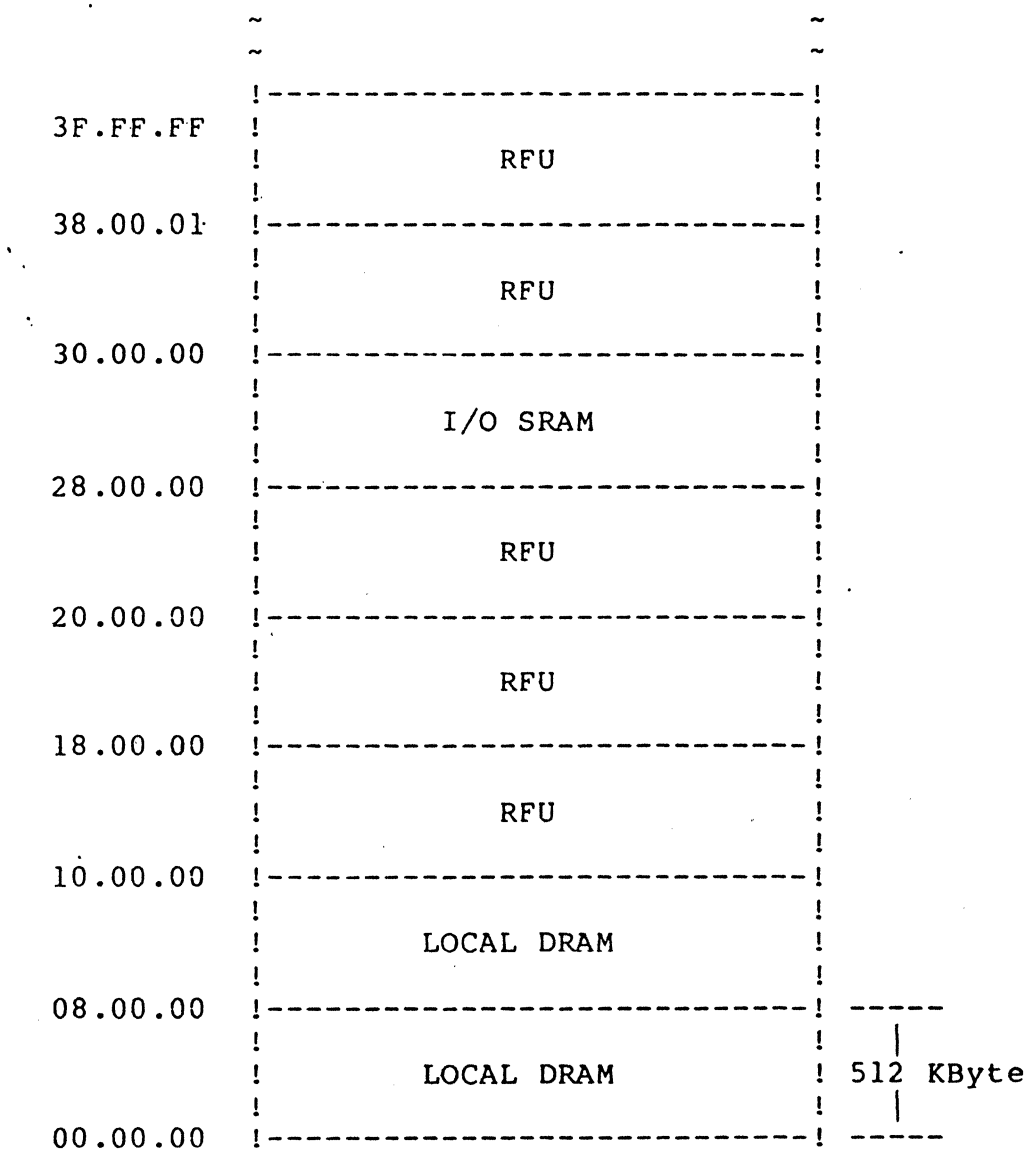
LINE PROCESSOR 0 - MASTER PROCESSOR MEMORY MAP



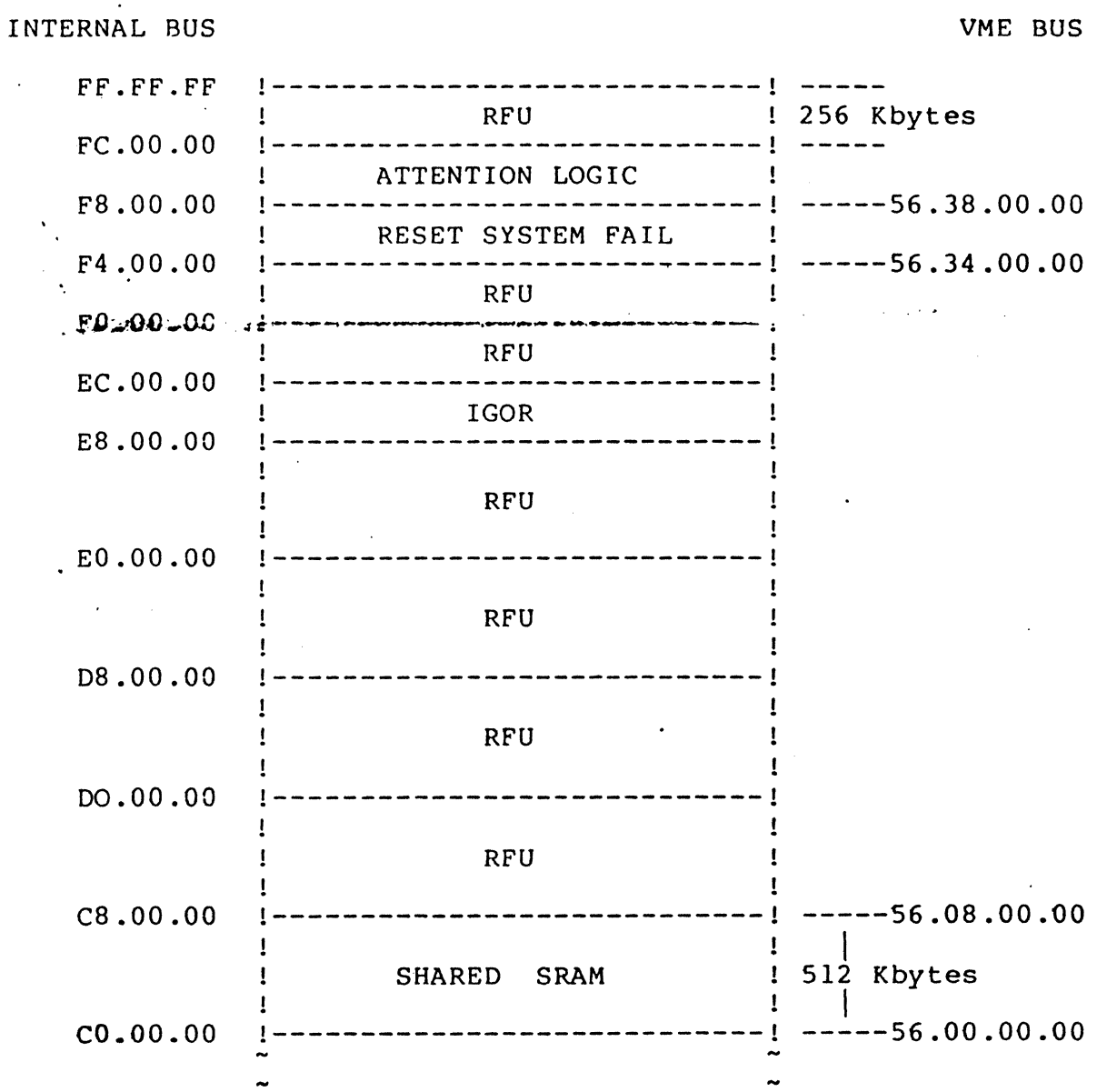
* Map for the first Line Processor with the Processor Number 0. For the others see para. E.2.7 processor number and board type detection.


BF.FF.FF
PIT
B8.00.00
RFU
B0.00.00
PROCESSOR NUMBER REGISTER
A8.00.00
WRITE NOTIFY VECTOR REGISTER
A4.00.00
READ I/O TERM.VECT.REG
A0.00.00
RFU
98.00.00
RFU
90.00.00
RFU
88.00.00
RFU
80.00.00

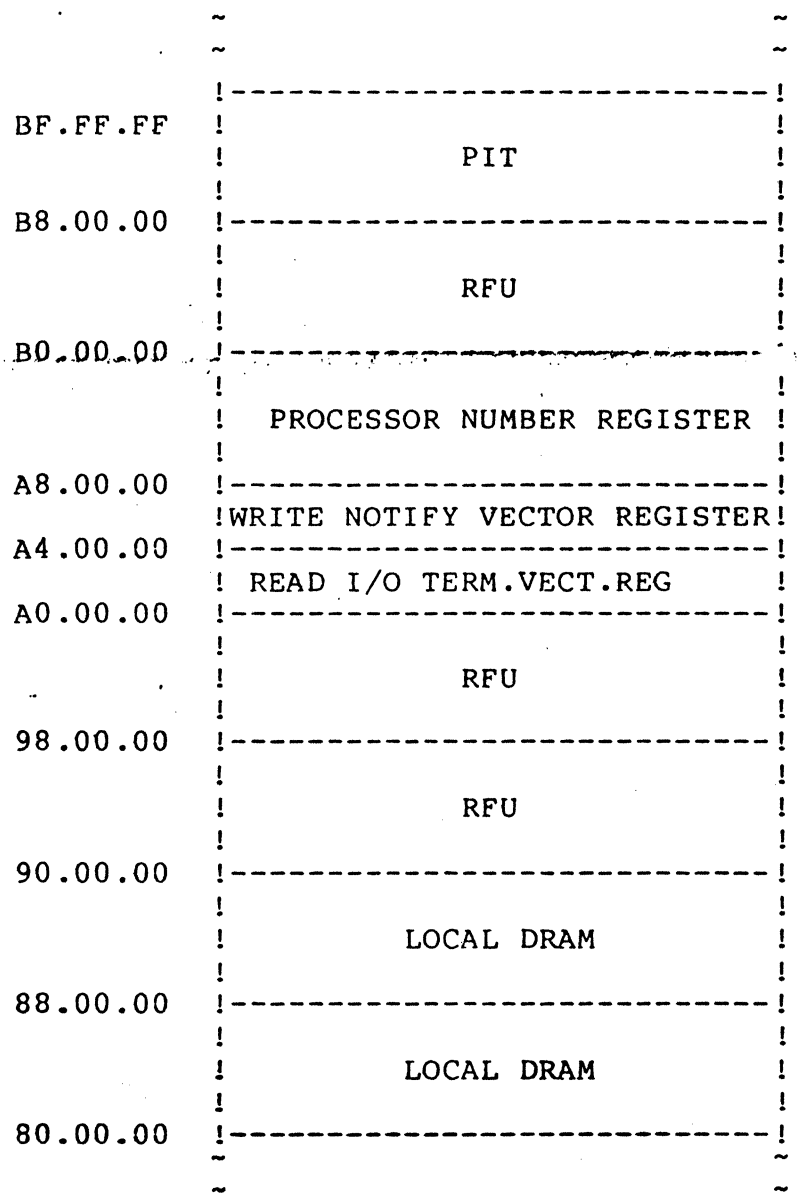

7F.FF.FF
MODEM 0 REGISTER OUT
7C.00.00
MODEM 1 REGISTER OUT
78.00.00
MODEM 0-1 REGISTER IN
74.00.00
RFU
70.00.00
WRITE I/0 TERM. REG.
6C.00.00
READ NOTIFY VEC. REG.
68.00.00
RFU
60.00.00
RFU
58.00.01
SIO2 LINES 4-5
50.00.01
SIO1 LINES 2-3
48.00.01
SIO0 LINES 0-1
40.00.01

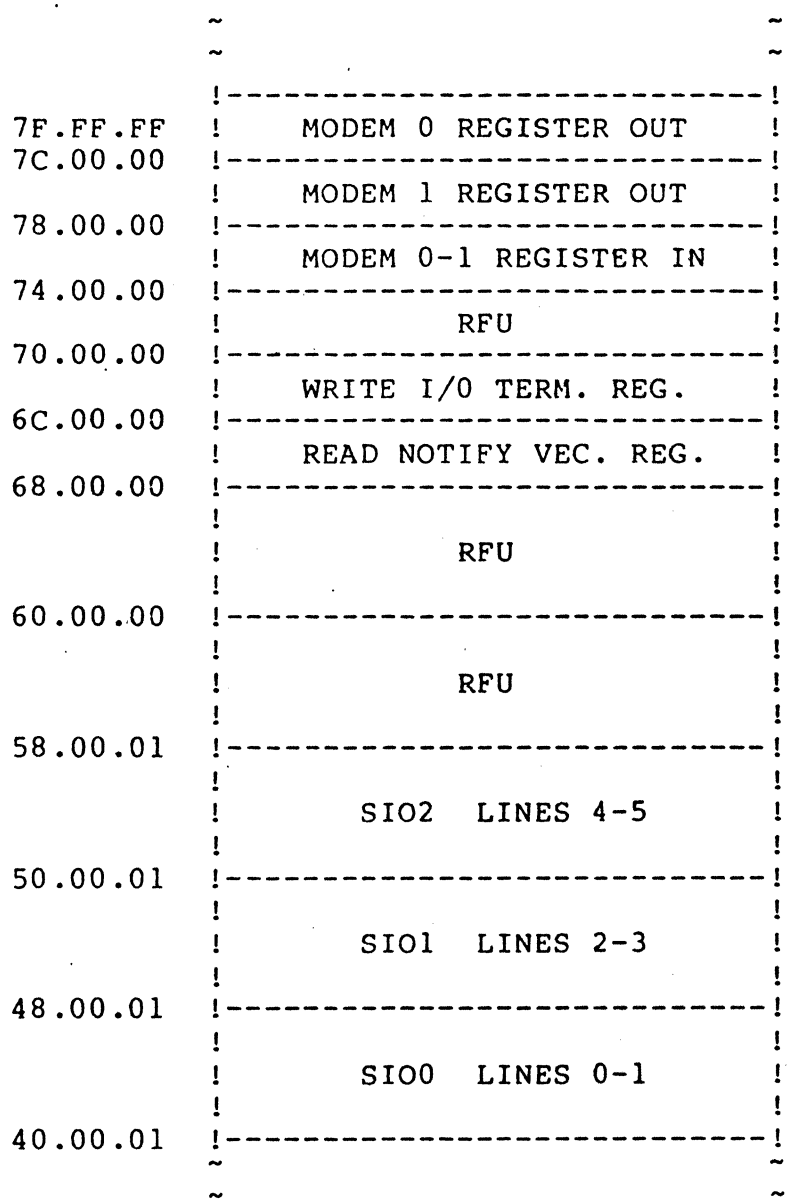

3F.FF.FF
RFU
38.00.01
RFU
30.00.00
I/O SRAM
28.00.00
RFU
20.00.00
RFU
18.00.00
RFU
10.00.00
LOCAL DRAM
08.00.00
LOCAL DRAM
512 KByte
00.00.00

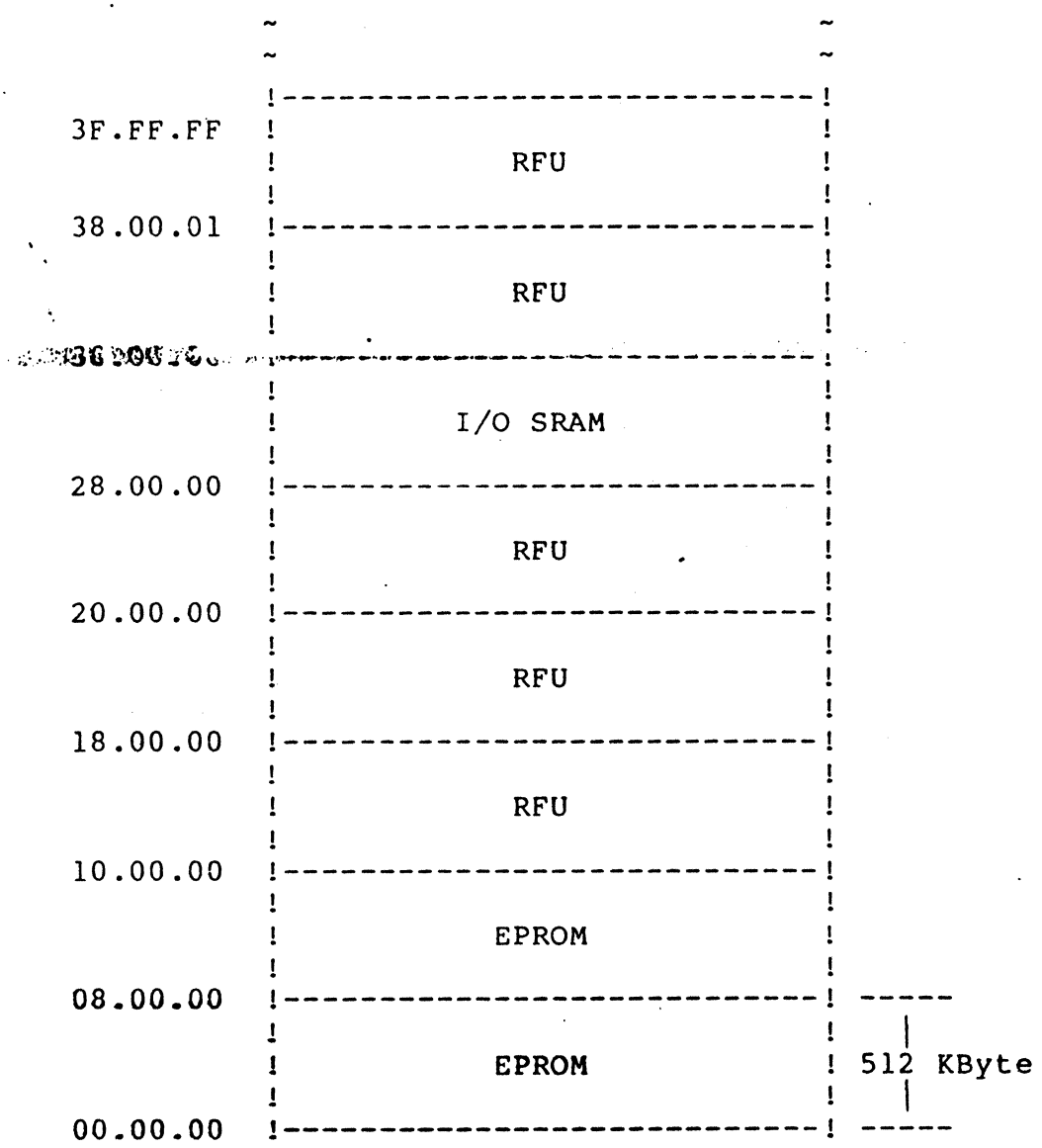
LINE PROCESSOR 0 - SLAVE PROCESSOR MEMORY MAP



* Map for the first Line Processor with the Processor Number 0. For the others see para. E.2.7 processor number and board type detection.


BF.FF.FF
PIT
B8.00.00
RFU
B0.00.00
PROCESSOR NUMBER REGISTER
A8.00.00
WRITE NOTIFY VECTOR REGISTER
A4.00.00
READ I/O TERM.VECT.REG
A0.00.00
RFU
98.00.00
RFU
90.00.00
LOCAL DRAM
88.00.00
LOCAL DRAM
80.00.00


7F.FF.FF
MODEM 0 REGISTER OUT
7C.00.00
MODEM 1 REGISTER OUT
78.00.00
MODEM 0-1 REGISTER IN
74.00.00
RFU
70.00.00
WRITE I/0 TERM. REG.
6C.00.00
READ NOTIFY VEC. REG.
68.00.00
RFU
60.00.00
RFU
58.00.01
SIO2 LINES 4-5
50.00.01
SIO1 LINES 2-3
48.00.01
SIO0 LINES 0-1
40.00.01


3F.FF.FF
RFU
38.00.01
RFU
I/O SRAM
28.00.00
RFU
20.00.00
RFU
18.00.00
RFU
10.00.00
EPROM
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;

- I/O MEMORY

The I/O memory consists of 2 8Kbytes static ram connected in such a way that it constitutes one 8Kword bank.
It is mapped in the following address range :

- 08 Kwords bank: 28.00.00 Hex --> 28.3F.FF Hex.

- EPROM

In the I/O area we find the EPROMS, two 16 Kbytes packages (Eprom 27128) have been used. These two packages are connected in such a way that they 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 8Kwords/16Kwords EPROM bank is mapped in the following address range of the slave 68000:

- 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 LP0 board;

- LOCAL MEMORY

the local memory consists of 16 256K x1 DRAM packages plus two 256K x1 DRAM packages utilized like check bit. The 256K x1 packages are connected in such that they constitute one 256 Kwords bank. This memory can be accessed at byte length (right byte and left byte) and at word length with zero or maximum one wait cycles.

The LOCAL MEMORY bank is mapped, for the slave processor, in the following address range:

80.00.00 Hex --> 87.FF.FF Hex;

On the other hand, the master processor sees the local memory in the following address range:

00.00.00 Hex --> 07.FF.FF

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 for the system bus

56.34.00.00 Hex for the VME bus (first processor number)

The diagnostics can generate an inverted memory check bit to test the error detection circuit. For this feature the PC4 Port C bit of the Parallel Interface Timer has been used.
Writing a 0 into the PC4 bit of the PIT the inversion of the check bit will be generated. For The Port C Addressing space, refer to figure E.6 (MC68230 Register Model).

- SERIAL INPUT OUTPUT

three USART (SIO) have been used. Each of these controllers can handle two Serial Ports (see para E.2.3);

- 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. E.2.4). PIT is also utilized for diagnostics purpouses, notify interrupts and it is located in the local area;

- VME BUS CONTROLLER (BUSCON)

it is an interface device that assures VMEbus compatibility, allowing either the LP0 microprocessor and the system CPUs to dialogue with the shared devices;

- SHARED MEMORY

the shared memory consists of eight 8Kx8 SRAM packages plus two 64Kx1 SRAM packages utilized like check bit. The 64 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 with at least four wait cycles.

The SHARED MEMORY bank is mapped in the following address range:

C0:00:00 Hex ---> C0:FF: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 generate an inverted memory check bit to test the error detection circuit. For this feature the PC4 Port C bit of the Parallel Interface Timer has been used.
Writing a 0 into the PC4 bit of the PIT the inversion of the check bit will be generated. For The Port C Addressing space, refer to figure E.6 (MC68230 Register Model).

- VME BUS INTERRUPTER LOGIC
this logic acts as an interrupt requester on VMEbus (see para. E.2.5);

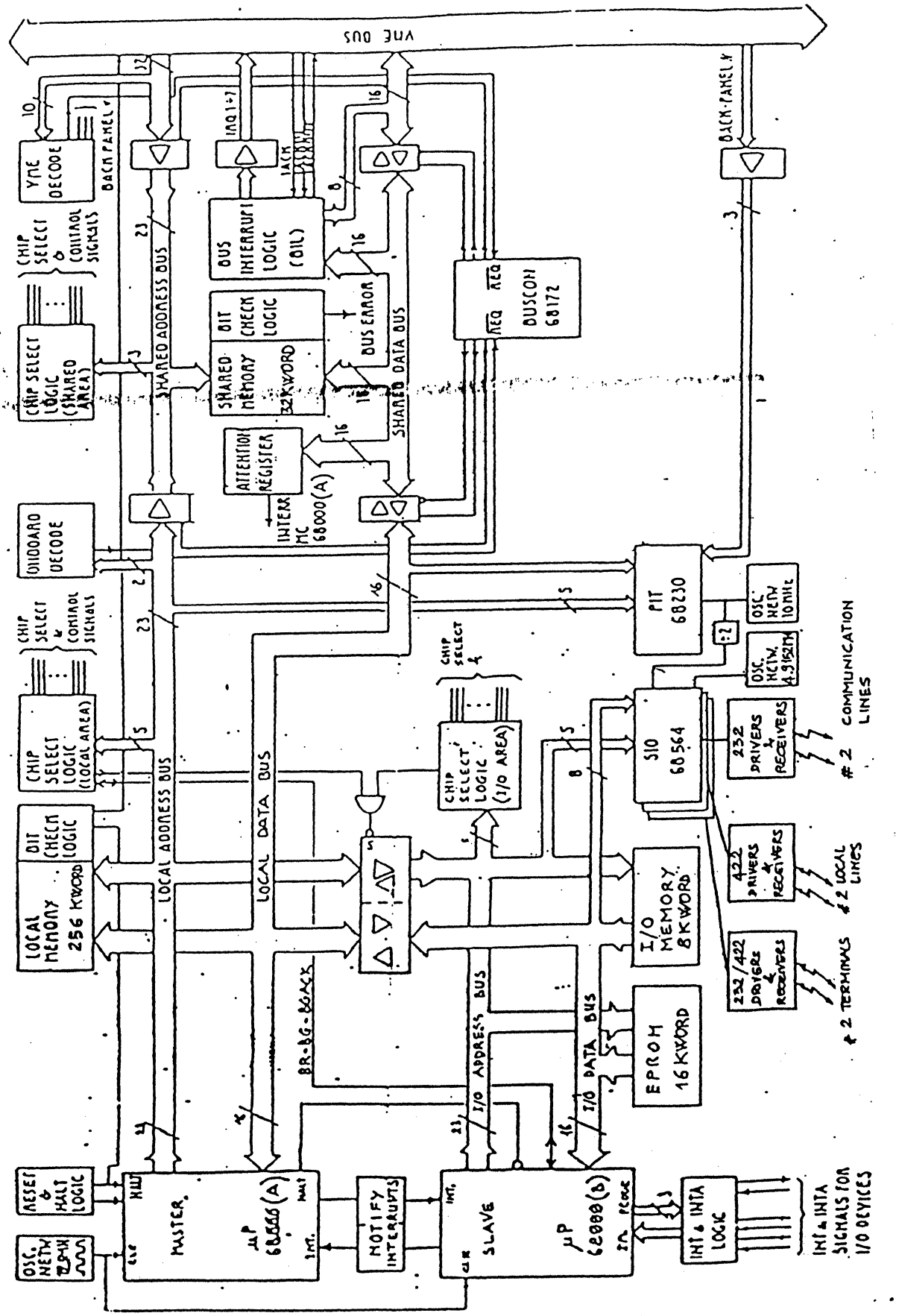
- ATTENTION LOGIC AND SYSTEM FAIL DETECTION

this logic permits the system CPUs to interrupt the LP and, in case of LP0 system fail, to recognize the system fail condition of the board (see para. E.2.6);

- 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 communication board was inserted (SP0 or LP0). (see para. E.2.7).

SGM2 - LINE COMMUNICATION PROCESSOR LP8
MAJOR BLOCK DIAGRAM



E.2.1 MASTER AND SLAVE MICROPROCESSORS

The LP0 uses two 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.

The two microprocessors share the same bus and memories and can communicate each other by the NOTIFY INTERRUPT and the TERMINATION I/O INTERRUPT (see para. E.2.2).
At the initial time the master is in HALT state until the slave initializes the LP0 board. When the slave processor finishes its job, (it prepares exception tables, it down-loads programs in the local and I/O memory from shared memory, etc.), it wakes up the master writing a logic zero in the PC2 bit of the PIT Port C. (see para. E.2.4).
Thus the master leaves its halt state and starts its routine while the slave keeps on waiting for commands from the master.
It is to be noted that, when needed, both microprocessors can have access to the all shared resources, but the access is much faster to the HW resources connected to its bus. For instance:

Master 68000 access to	LOCAL MEMORY	in 400 ns.
	I/O MEMORY	in 560 ns.
	SHARED MEMORY	in 640 ns.
Slave 68000 access to	LOCAL MEMORY	in 560 ns.
	I/O MEMORY	in 320 ns.
	SHARED MEMORY	in 800 ns.

If the MASTER processor wants to read or write the I/O memory and at the same time, the SLAVE processor wants to have an access to the LOCAL MEMORY, the MASTER processor re-runs the last instruction after the end of the slave access in the LOCAL MEMORY.

In case of TAS instruction on the I/O memory by the MASTER processor during the same condition upon described, the master will not re-run the instruction, but a BUS ERROR will happen. A software treatment of this BUS ERROR must be done.

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

E.2.2 MASTER AND SLAVE 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 acknowledges and handles the Interrupt Acknowledge cycle using FC0, FC1, FC2 lines (Processor Status) and the A1, A2, A3 address signals.

In the LP0 board have been realized two interrupt & interrupt acknowledgment logics: one for the master processor, the other for the slave.

The master handles the following interrupts:

Level		Interrupt		
+ with the highest priority				
7 (NMI)		NOT USED		
6	SIOs	SIO1	LINE#2	Special RX Cond. RX Data Request TX Data Request STS CHG Request
			LINE#3	Special RX Cond. RX Data Request TX Data Request STS CHG Request
		SIO0	LINE#0	Special RX Cond. RX Data Request TX Data Request STS CHG Request
			LINE#1	Special RX Cond. RX Data Request TX Data Request STS CHG Request
		SIO2	LINE#4	Special RX Cond. RX Data Request TX Data Request STS CHG Request
			LINE#5	Special RX Cond. RX Data Request TX Data Request STS CHG Request
5		NOT USED		
4		INTERRUPT FROM DMA EMULATOR (SLAVE PROCESSOR)		
3		TIMER		
2		ATTENTION REGISTER		
1		PARALLEL PRINTER		
0		NO INTERRUPTS		
- with the lowest priority				

When the SLAVE needs to communicate with the MASTER processor it must write the interrupt vector in the I/O TERMINATION REGISTER located at:

Hex 6C.00.01 for the write cycle

and

Hex A0.00.01 for the read cycle

Then writing a zero in the PC0 bit of the Pit Port C, it gives interrupt to the master at fourth level.

The right flow of the SLAVE operations is:

1) Test until the PC0 bit is set to one from the MASTER processor (NO INTERRUPT PENDING).

2) When the PC0 is one, write the interrupt vector on the I/O TERMINATION REGISTER.

3) Write a zero in the PC0 bit of the Pit Port C (INTERRUPT TO THE MASTER).

The MASTER processor, during the interrupt routine, will have to set the PC0 bit to one to allow other interrupts from the slave to the master.

On the contrary, the slave processor handles the following interrrupts:

Level	Interrupt
+ with the highest priority	
7 (NMI)	NOT USED
6	NOT USED
5	SIOs
4	NOT USED
3	NOT USED
2	NOT USED
1	NOTIFY INTERRUPT

SIO1: LINE#2 (RXRDY DMA signal / TXRDY DMA signal); LINE#3 (RXRDY DMA signal / TXRDY DMA signal)

SIO0: LINE#0 (RXRDY DMA signal / TXRDY DMA signal); LINE#1 (RXRDY DMA signal / TXRDY DMA signal)

All the interrupts are vectored

The Sios interrupt the slave processor using the TXRDY and RXRDY pins connected to a priority encoder that gives the following priority to each line (from highest to lowest):

Line 2 Receive Ready

Line 2 Transmit Ready

Line 3 Receive Ready

Line 3 Transmit Ready

Line 0 Receive Ready

Line 0 Transmit Ready

Line 1 Receive Ready

Line 1 Transmit Ready

The interrupt vector is automatically generated by a logic that provides different interrupt vectors:

Hex 40 for Line 2 Receive Ready interrupt

Hex 41 for Line 2 Transmit Ready interrupt

Hex 42 for Line 3 Receive Ready interrupt

Hex 43 for Line 3 Transmit Ready interrupt

Hex 44 for Line 0 Receive Ready interrupt

Hex 45 for Line 0 Transmit Ready interrupt

Hex 46 for Line 1 Receive Ready interrupt

Hex 47 for Line 1 Transmit Ready interrupt

The master processor to interrupt the slave, must write the interrupt vector in the NOTIFY REGISTER located at:

Hex A4.00.01 for the write cycle

and

Hex 68.00.01 for the read cycle

Then writing a zero in the PC1 bit of the PIT Port C it interrupts the slave processor.

The right flow of the MASTER operations is:

1) Test until the PC1 bit is set to one from the SLAVE processor (NO INTERRUPT PENDING).

2) When the PC1 is one, write the interrupt vector on the NOTIFY REGISTER.

3) Write a zero in the PC1 bit of the Pit Port C (INTERRUPT TO THE SLAVE).

The SLAVE processor, after having received the interrupt, will have to set the PC1 bit to one to allow other interrupts from the master to the slave.

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 for the slave microprocessor and in local DRAM for the master.

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

E.2.3 SERIAL INPUT OUTPUT

The LP0 serial ports allow Local and Remote Asynchronous Communications and Local and Remote Synchronous communication by means of RS-232C (V24/V28) and RS-422A (V11) electrical interface.

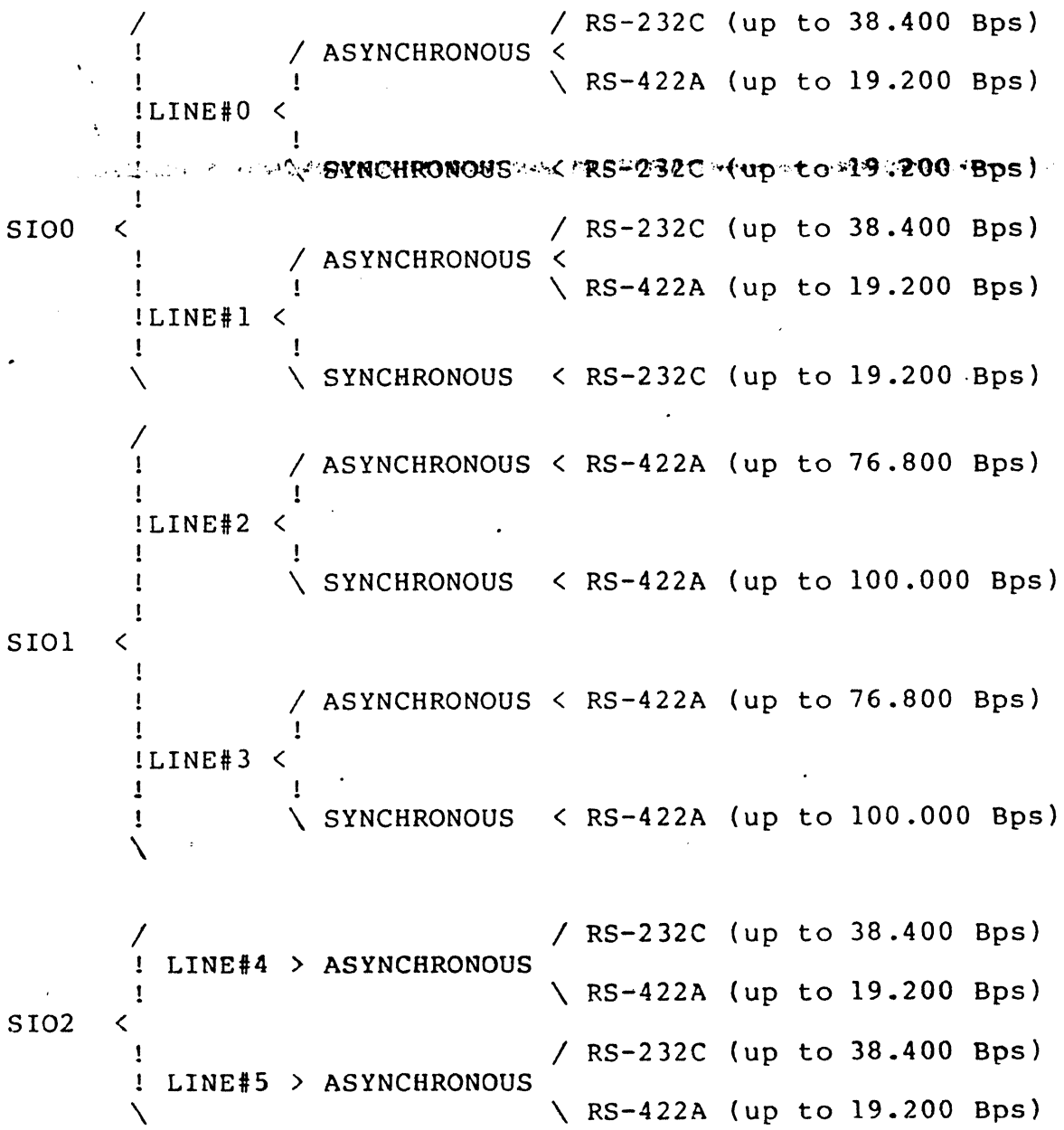
The handling of the serial ports are made by the Serial Input Output (SIO) 68564 which are LSI chips with 48 pins dual in line packages operating at 5 Mhz clock.

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;
- Dma pins;
- 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
- Byte Synchronous features:
  * internal or external character synchronization;
  * one or two sync characters in separate registers;
  * automatic sync characters insertion;
  * CRC 16 or CRC-CCITT block check generation and checking;
- Bit synchronous features:;
  * abort sequence generation and detection;
  * automatic zero insertion and detection;
  * automatic flag insertion between messages;
  * address field recognition;
  * I-field residue handling;
  * valid receive messages protected from overrun;
  * CRC-CCITT block check generation and checking;

E.2.3.1 CONFIGURATION

The configuration consists of 6 Serial Ports . These ports or lines are indicated as follows:



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

The maximum speeds reachable by each line are:

	RS-232C ASYNC	RS-232C SYNC	RS-422A ASYNC	RS-422A SYNC
LINE 0	38.400	19.200	19.200	
LINE 1	38.400	19.200	19.200	
LINE 2			76.800	100.000
LINE 3			76.800	100.000
LINE 4	38.400		19.200	
LINE 5	38.400		19.200	

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
up to 40KBps.	600 meters
up to 80KBps.	300 meters
up to 100KBps.	240 meters

The choice between the electrical interfaces RS-232C and RS-422A is made using different cables for lines 0/1/4/5. The lines 2/3 have only the RS-422A interface.

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.

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)
BIT RATE	100000	12	0C	4	N. A.	N. A.	N. A.
	76800	16	10	4	1	1	4
	38400	32	20	4	2	2	4
	19200	64	40	4	4	4	4
	9600	128	80	4	8	8	4
	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:

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

(*) 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.

E.2.3.2 ELECTRICAL INTERFACE AND MODEM SIGNALS

The Serial Ports use as interface three kinds of connectors:

LINE 0 25 pins connector with female pins;

LINE 1 25 pins connector with female pins;

LINE 2 15 pins connector with male pins;

LINE 3 15 pins connector with male pins;

LINE 4 9 pins connector with female pins;

LINE 5 9 pins connector with female pins;

The 25 pins connector includes both RS-232C and RS-422A interface signals. The table below shows the pin assignement:

Connec. Pin N.	Interface Circuit Name	Note
1	Cable Shield Ground	* $
2	Transmitted Data	*
3	Received Data	*
4	Request to Send	*
5	Clear to Send	*
6	Data Set Ready	*
7	Signal Ground	* $
8	Data Carrier Detect	*
9	Make Busy Out	*
10	Enable to receive Rx Clock	*
11	New Sync	*
12	Enable to receive Tx Clock	*
13	Transmitted Data A	$
14	Transmitted Data B	$
15	Transmitted Clock	*
16	Received Data A	$
17	Received Clock	*
18	Local Loop Back	*
19	Received Data B	$
20	Data Terminal Ready	*
21	Remote Loop Back	*
22	Ring Indicator	*
23	Data Rate Selector	*
24	Transmitted Internal Clock	*
25	Test Indicator	*

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

The ENABLE TO RECEIVE RX/TX CLOCK signals when in the low level, allow to receive the external clocks.

THe 15 pins connector includes only the RS-422A interface signals:

Connec. Pin N.	Interface Circuit Name	Note
1	Cable Shield Ground	$
2	Transmitted Data A	$
3	Not Used	$
4	Received Data A	$
5	Not Used	$
6	Received Clock A	$
7	Transmitted Clock A	$
8	Signal Ground	$
9	Transmitted Data B	$
10	Not Used	$
11	Received Data B	$
12	Not Used	$
13	Received Clock B	$
14	Transmitted Clock B	$
15	Not Used	$

In the figure E.2 are shown the layouts and the positions of the LP0 jumpers. These jumpers allow to configure the line 2 and 3 in synchronous or asynchronous mode.

For these lines, in synchronous mode the TxC (Tx Clock) pin provides the clock and RxC (Rx Clock) pin receives the clock. The SIO must be programmed in this way: TxC internal mode, RxC external mode.

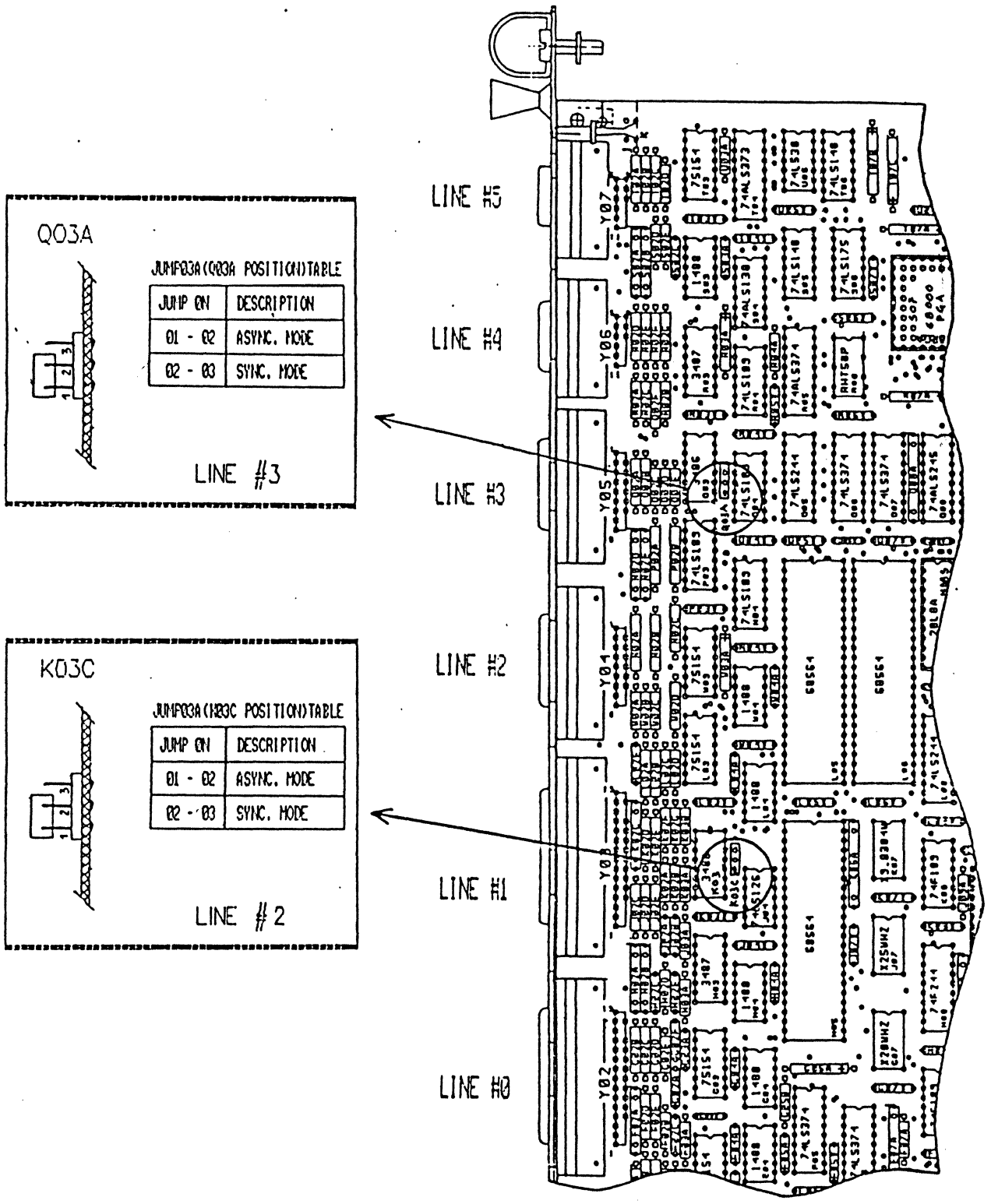


Fig. E.2 - Layouts and positions of the LP0 jumpers
(ETCHED BOARD)

In the figure E.3 are described the jumpers for the multiwire boards.

These jumpers configure the 0-1-2-3 lines in asynchronous or synchronous communications and the clocks direction.

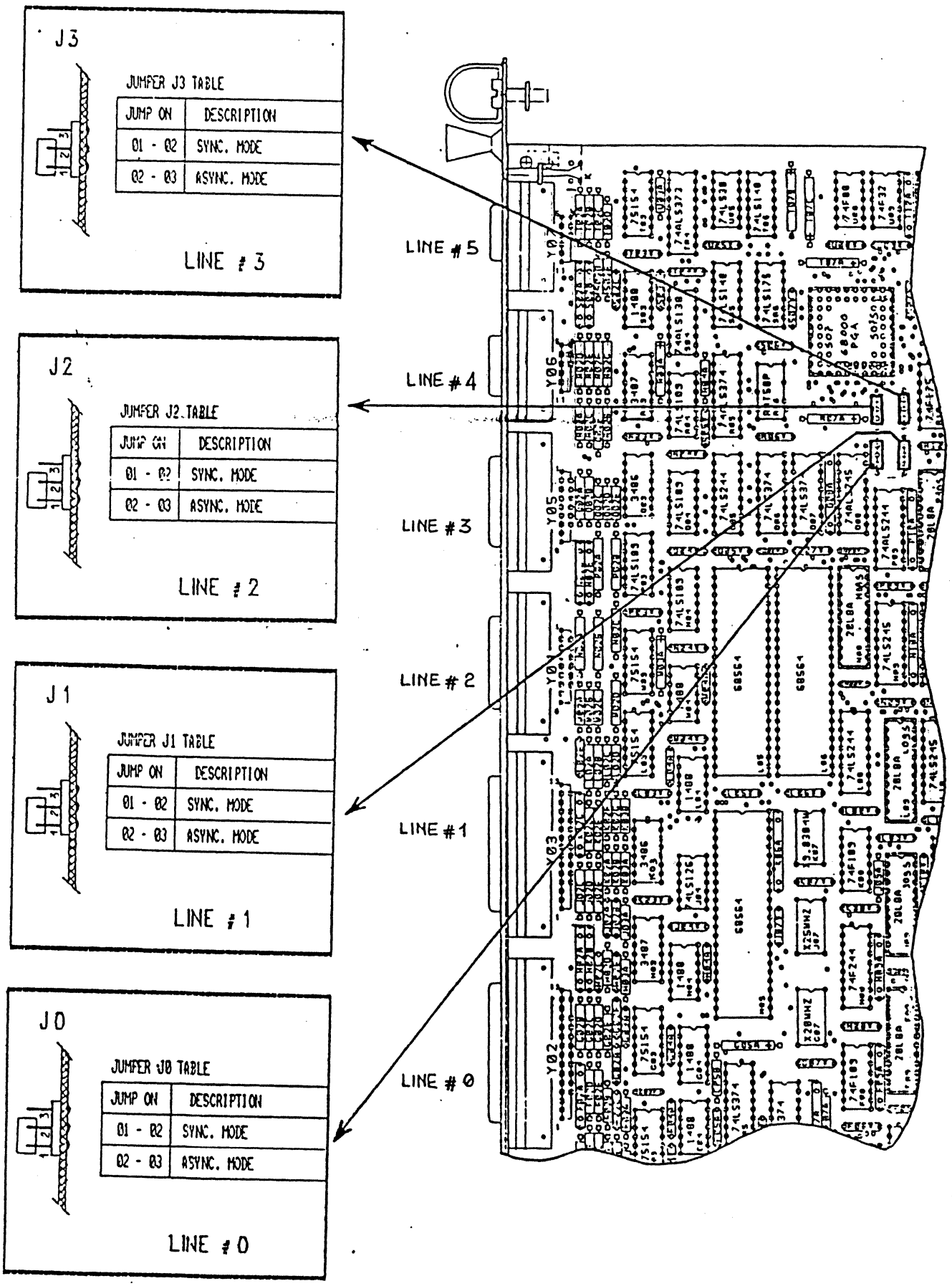


Fig. E.3 - Layouts and positions of the LP0 jumpers
(MULTIWIRE BOARD)

The 9 pins connector 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.

E.2.3.3 MODEM SIGNALS

Many modem signals are provided by the SIO, but not all. Thus there are three registers, two for the output, one for the input, which allow the connection of the two RS-232 lines.

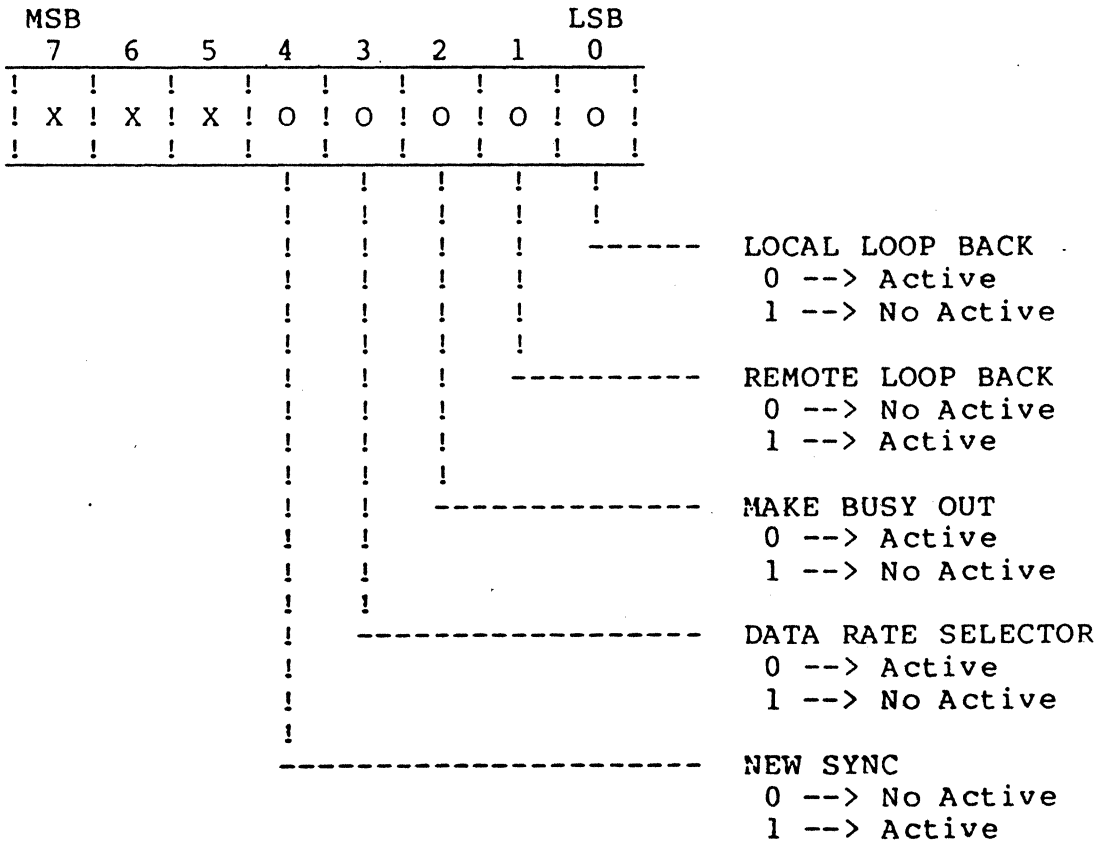
MODEM REGISTER LINE 0 OUTPUT is located at:

Hex 7C.00.01

MODEM REGISTER LINE 1 OUTPUT is located at:

Hex 78.00.01

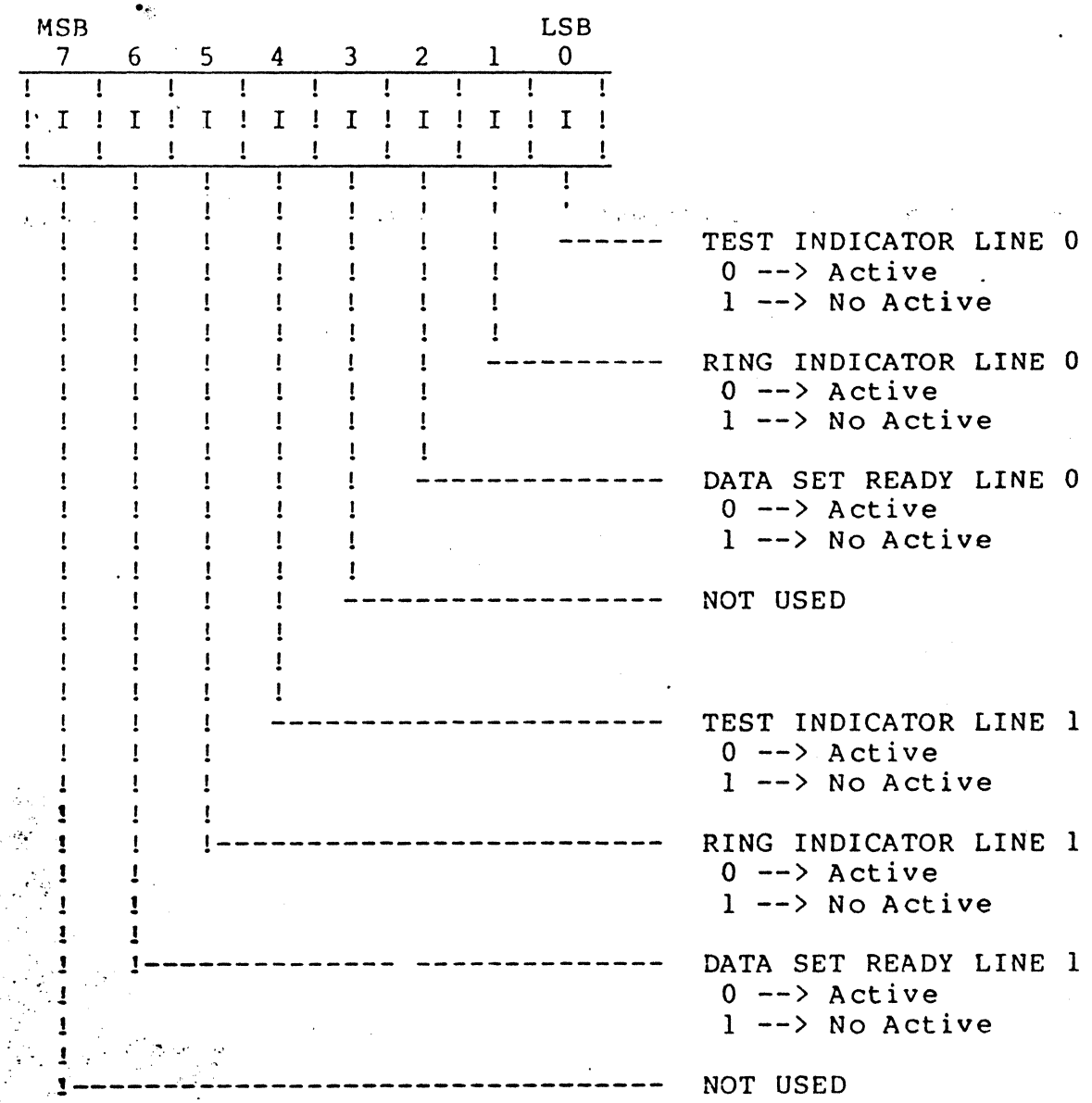
and they bring this signal:



MODEM REGISTER LINE 0/1 INPUT is located at:

Hex 74.00.01

and it brings this signal:



E.2.4 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.

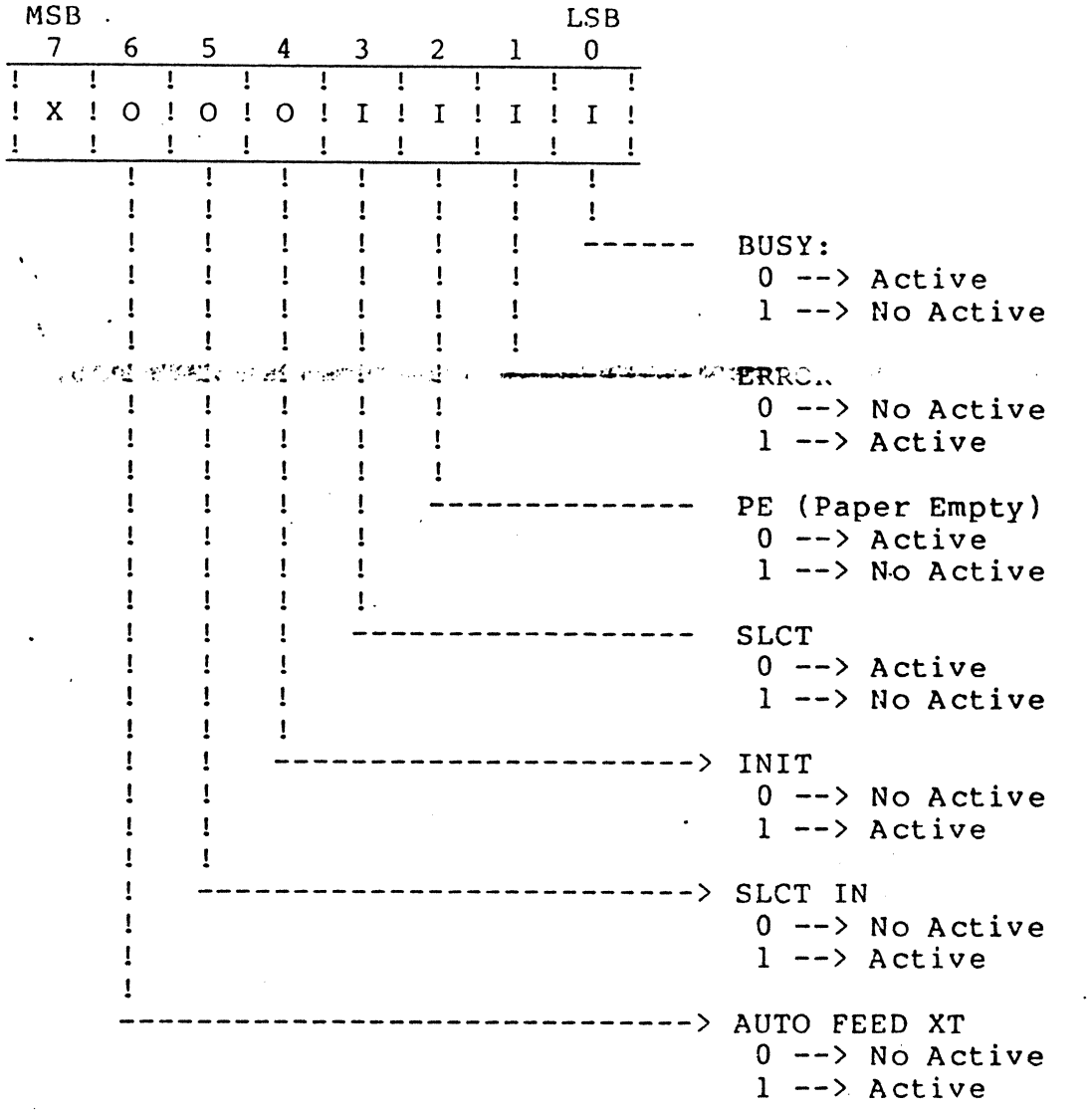
A register model that includes the corresponding Register Selects is shown in Fig. E.6. For further information refer to "16-BIT MICROPROCESSOR DATA MANUAL - 1983 MOTOROLA" pag. 4-509 - 4-537.

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. E.4 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. E.5 which shows the timings relevant to CENTRONICS and IBM interfaces).

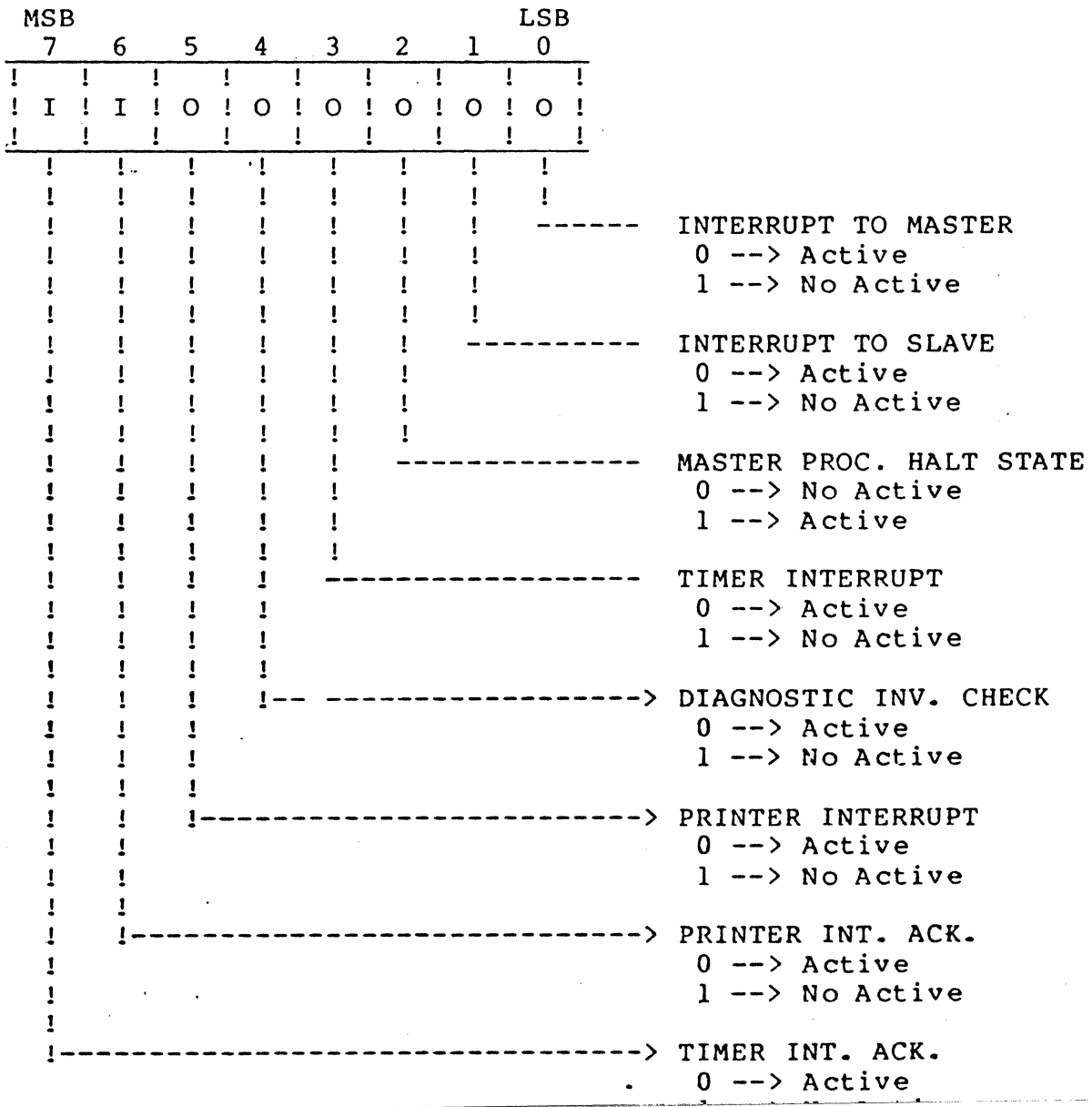
A register model that includes the corresponding Register Selects is shown in Fig. E.1.1.4.3. For further information refer to "16-BIT MICROPROCESSOR DATA MANUAL - 1983 MOTOROLA" pag. 4-509 - 4-537.

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 E.1.

The LP0 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)



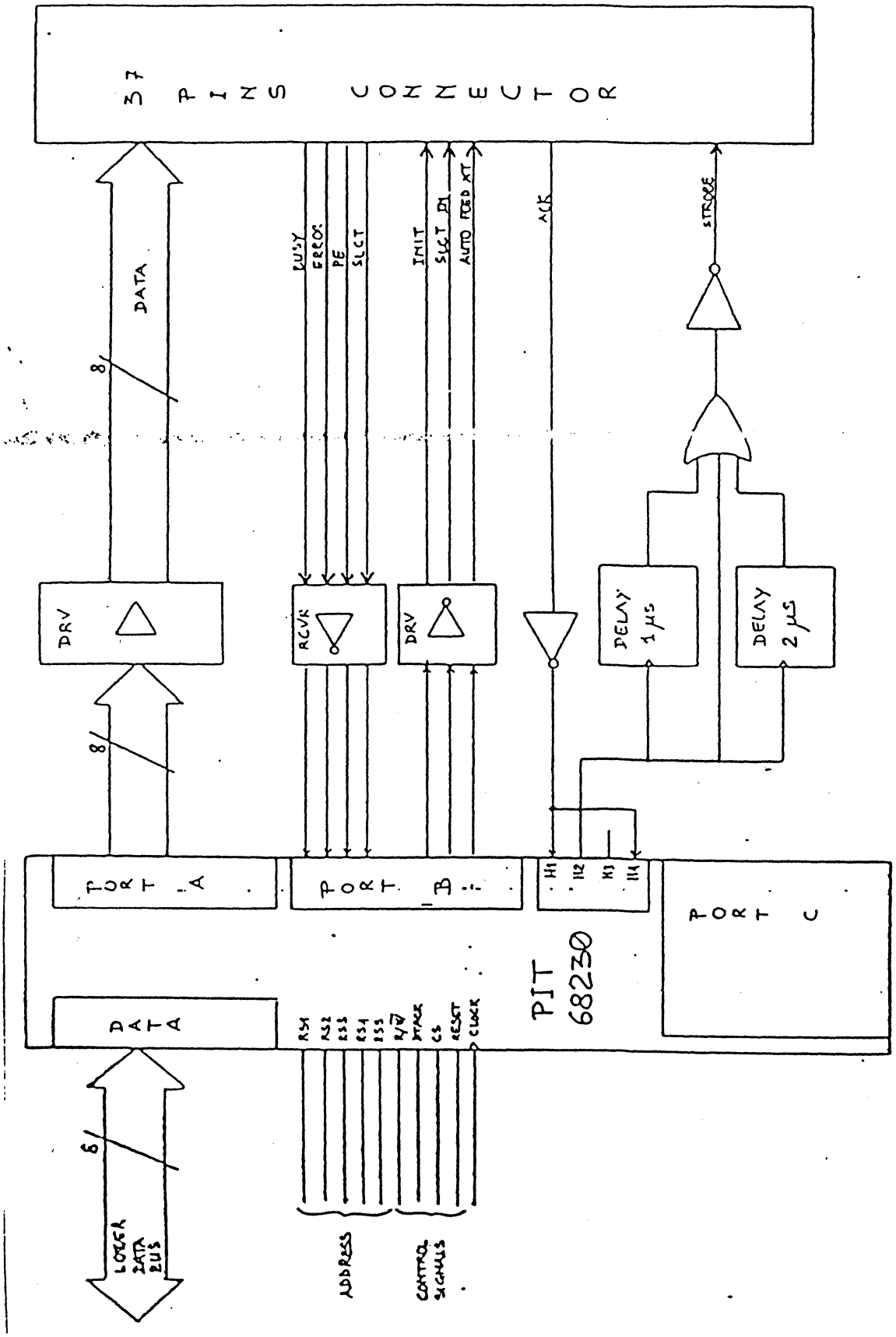
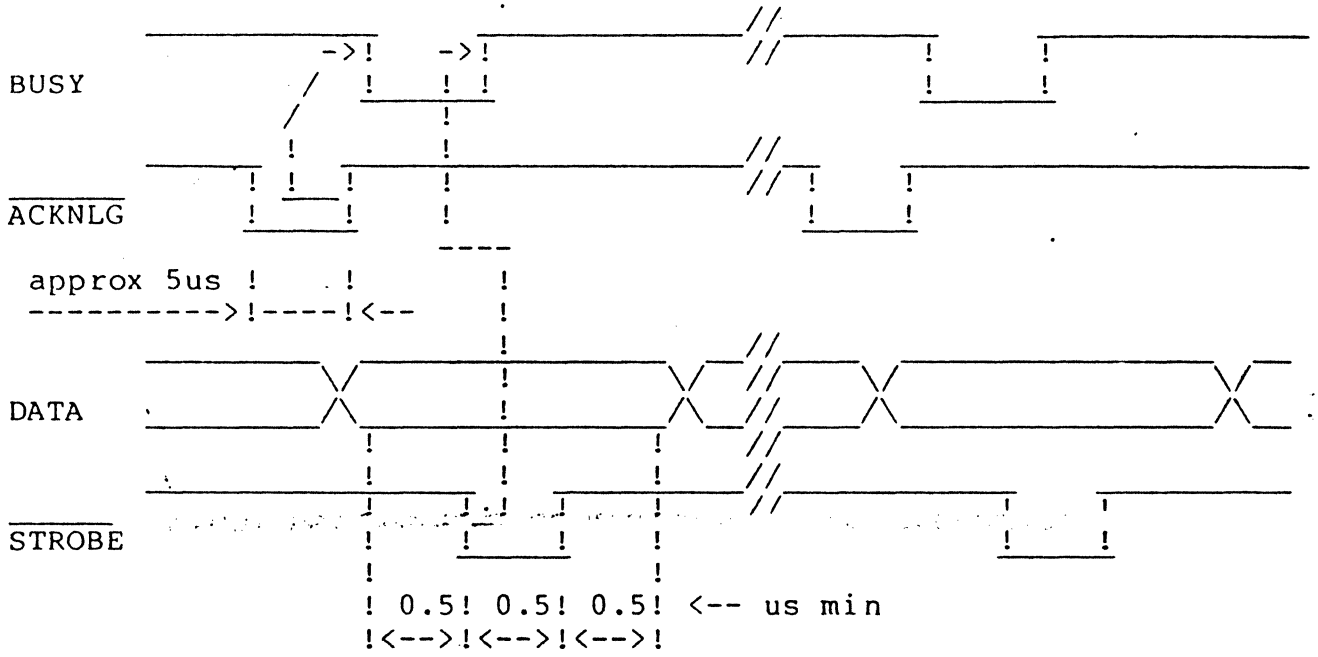
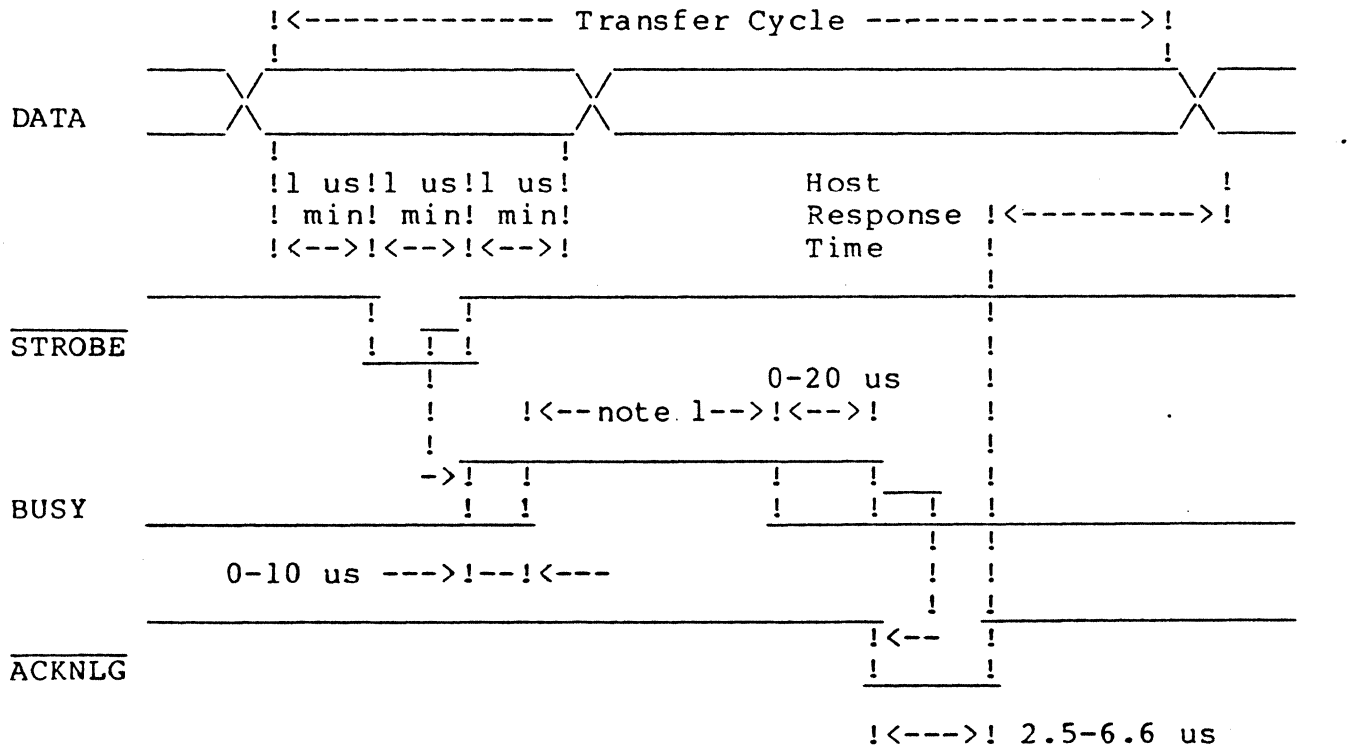


Fig. E.4 - Parallel Printer Interface Block Diagram



a) IBM Interface Timing



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

b) CENTRONICS Interface Timing

Fig. E.5 - Parallel Printer Interface Timing Diagram

ADDRESS (Hex)	7	6	5	4	3	2	1	0	
B8.00.01	Port Mode Control		H3= Enable	H12 Enable	H4= Sense	H3 Sense	H2 Sense	H1= Sense	Port General Control Register
B8.00.03	*		SVCRO 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 SVCRO Enable	H1 Stat Ctrl	Port A Control Register
B8.00.0F	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRO 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/TIACK Control			ZD 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. E.6 - MC68230 Register Model

Table E.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. E.5 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. E.5 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. E.5 are shown timing considerations for IBM and CENTRONICS interfaces.

Table E.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. E.5 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 E.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 E.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.

E.2.5 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.

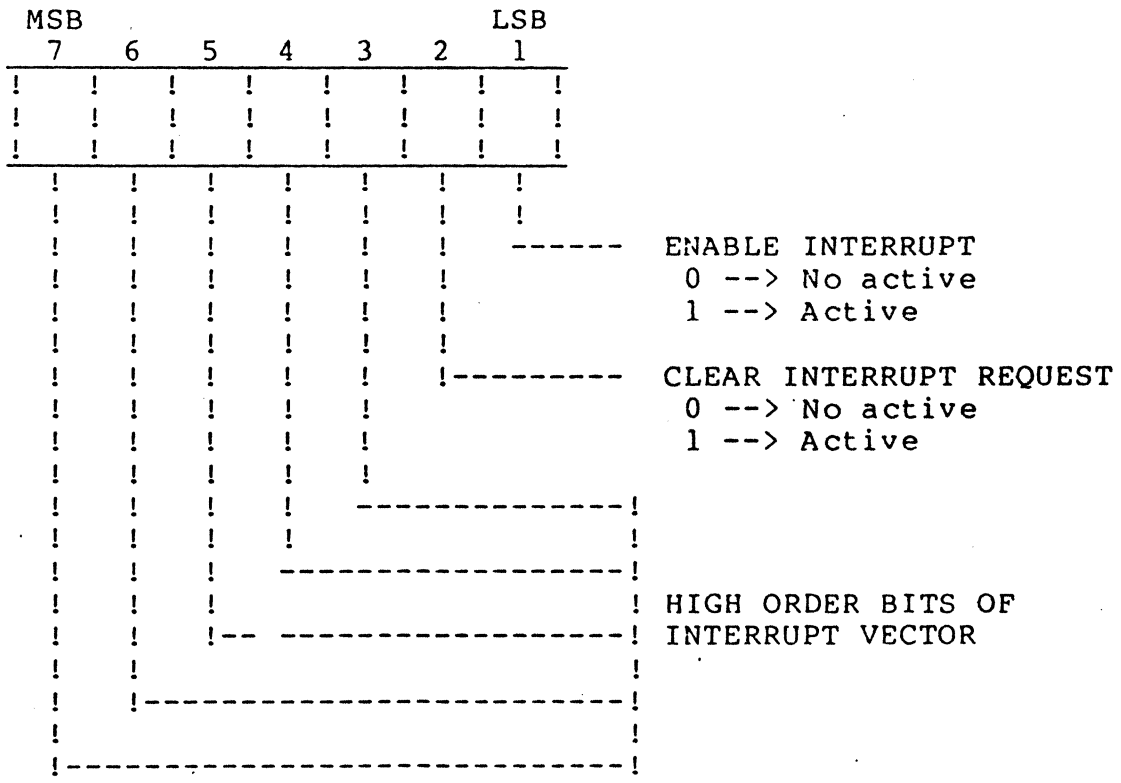
Inside it has two registers:

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

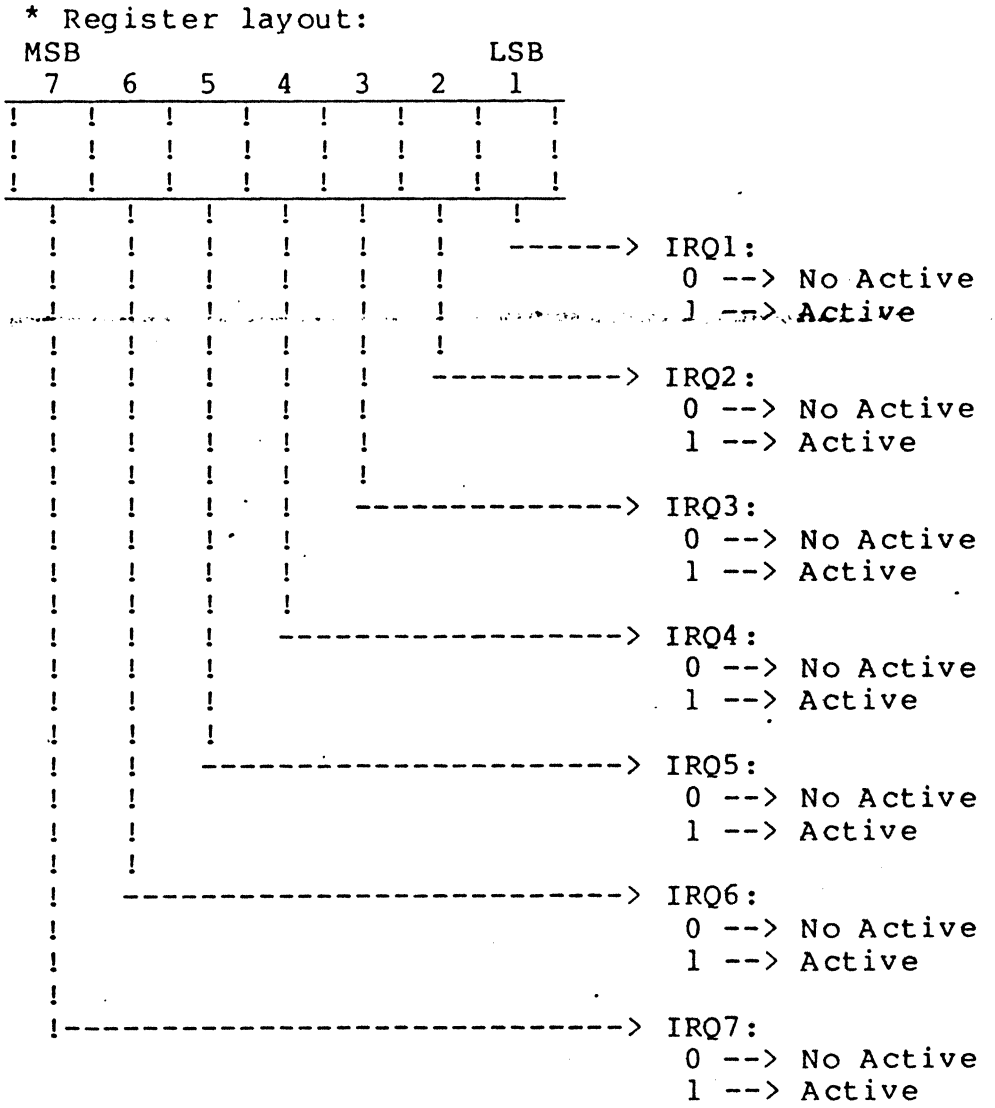
- the INTERRUPT REQUEST REGISTER R1 (located at Hex E8.00.05)

The local processor writes the interrupt vector register to enable interrupts and to program the high order bits of the interrupt vector 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).

E.2.6 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:

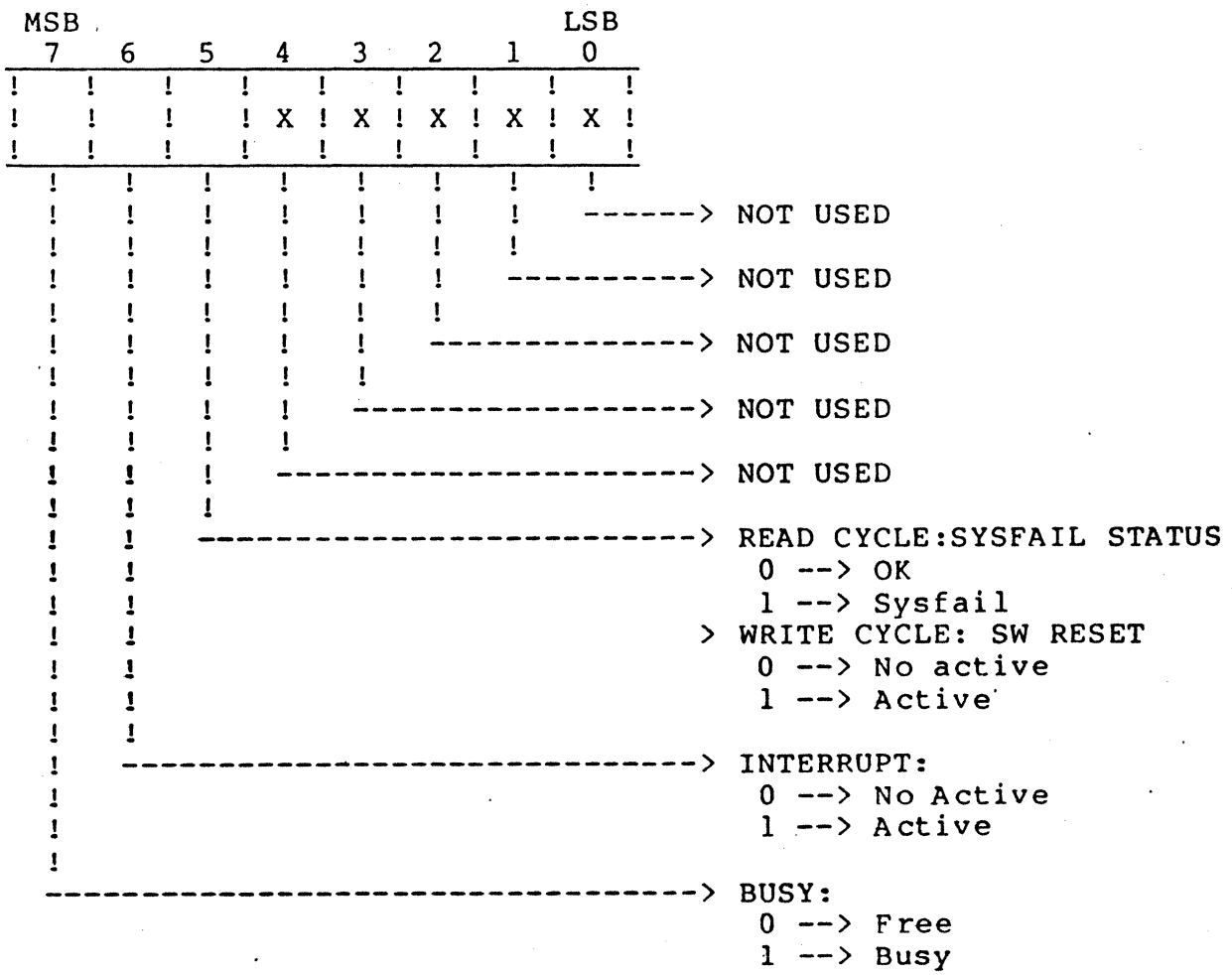
- ATTENTION INTERRUPT REGISTER:

  * Write/Read 8 bit register.

  * Access mode:

    ADDRESS: F8.00.00 Hex.
    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 LSB

7	6	5	4	3	2	1	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 LP0 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 LP0 board in the VME Bus is in the system fail condition.

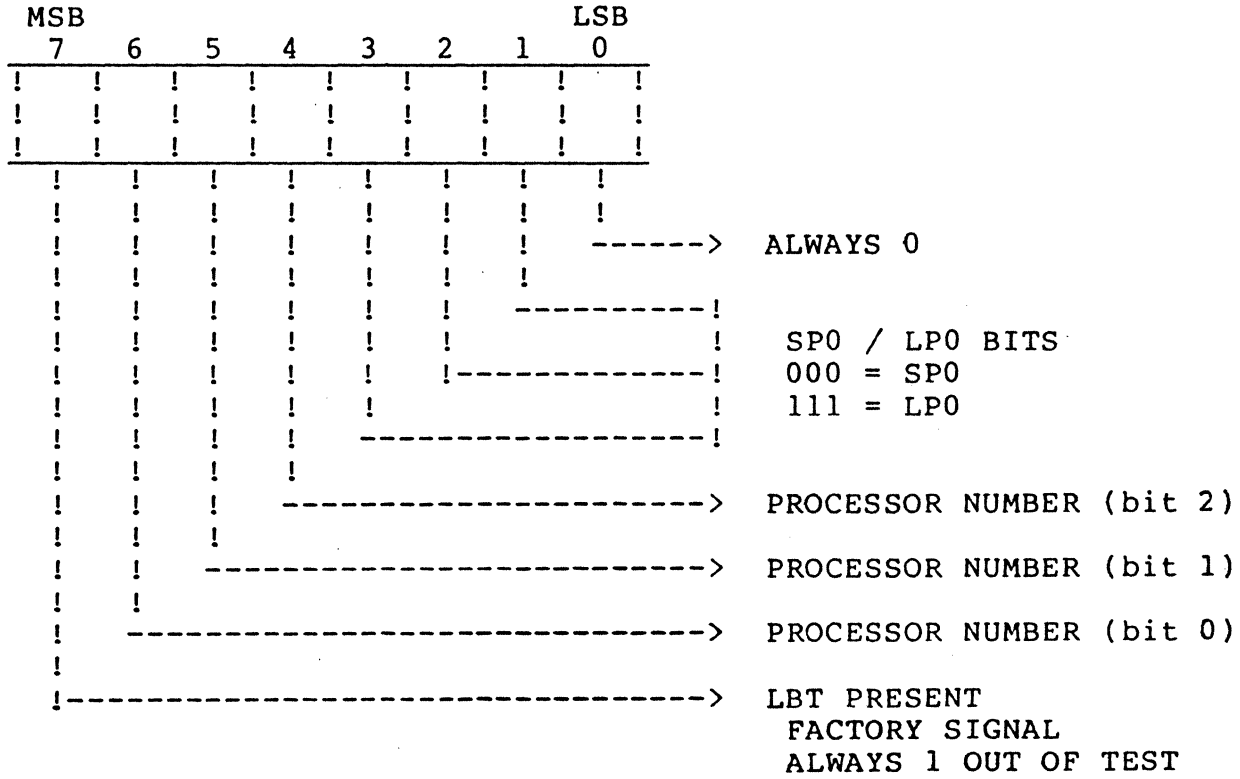
With the same bit, in write mode, the SGM2 CPU's can reset the LP0 board writing a 1 in the bit 5 for 1 msec., and after this reset period, the bit 5 must be cleared.

E.2.7 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 LP0 board was inserted;
- the type of the board inserted (SP0 or LP0);
- testing enviroment;

The layout of this register is:



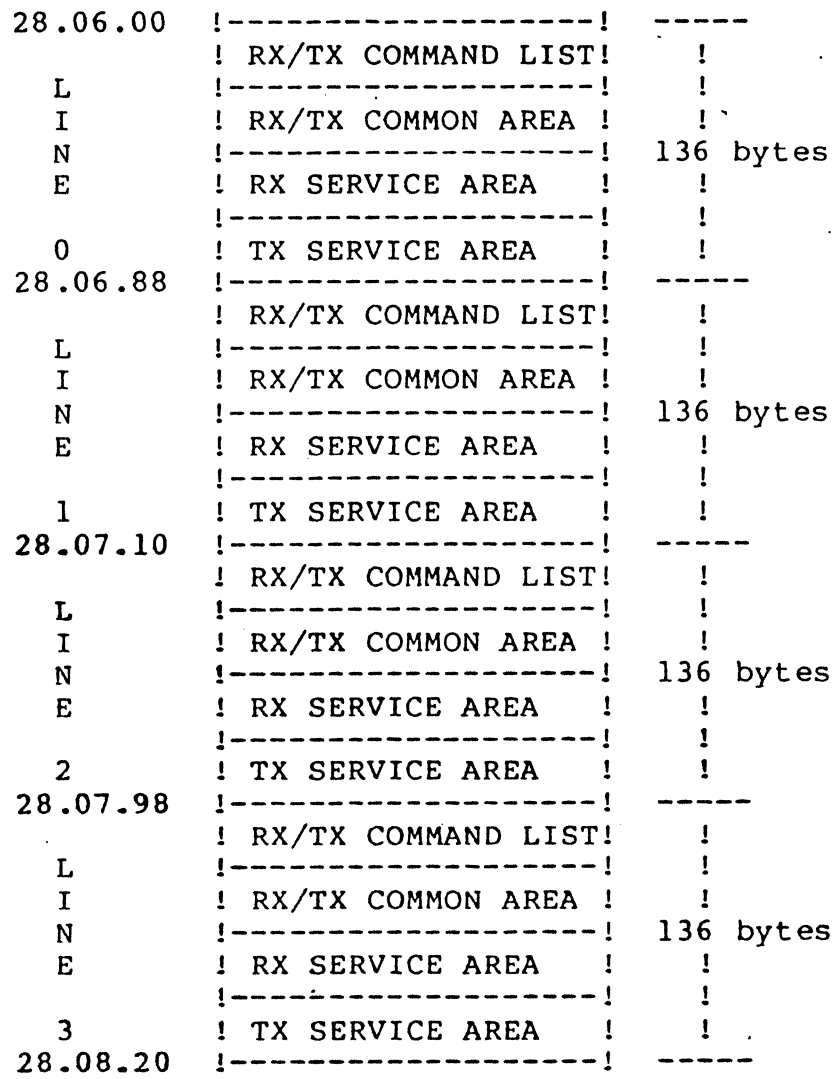
The table below describes the VME bus addressing space of each board with the processor number bits equal to:

6	5	4	
0	0	0	PROCESSOR NUMBER 0 - LP0 located at Hex 56.00.00.00 (VME BUS ADDRESSING SPACE)
1	0	0	PROCESSOR NUMBER 1 - LP0 located at Hex 56.40.00.00 (VME BUS ADDRESSING SPACE)
0	1	0	PROCESSOR NUMBER 2 - LP0 located at Hex 56.80.00.00 (VME BUS ADDRESSING SPACE)
1	1	0	PROCESSOR NUMBER 3 - LP0 located at Hex 56.C0.00.00 (VME BUS ADDRESSING SPACE)
0	0	1	PROCESSOR NUMBER 4 - LP0 located at Hex 57.00.00.00 (VME BUS ADDRESSING SPACE)
1	0	1	PROCESSOR NUMBER 5 - LP0 located at Hex 57.40.00.00 (VME BUS ADDRESSING SPACE)
0	1	1	PROCESSOR NUMBER 6 - LP0 located at Hex 57.80.00.00 (VME BUS ADDRESSING SPACE)
1	1	1	PROCESSOR NUMBER 7 - LP0 located at Hex 57.C0.00.00 (VME BUS ADDRESSING SPACE)

E.3 DMA EMULATOR INTERFACE

The interface model of the DMA is located, starting at 28.06.00 HEX, in the I/O memory.

For each line there is a structure of 136 bytes divided in 4 parts.

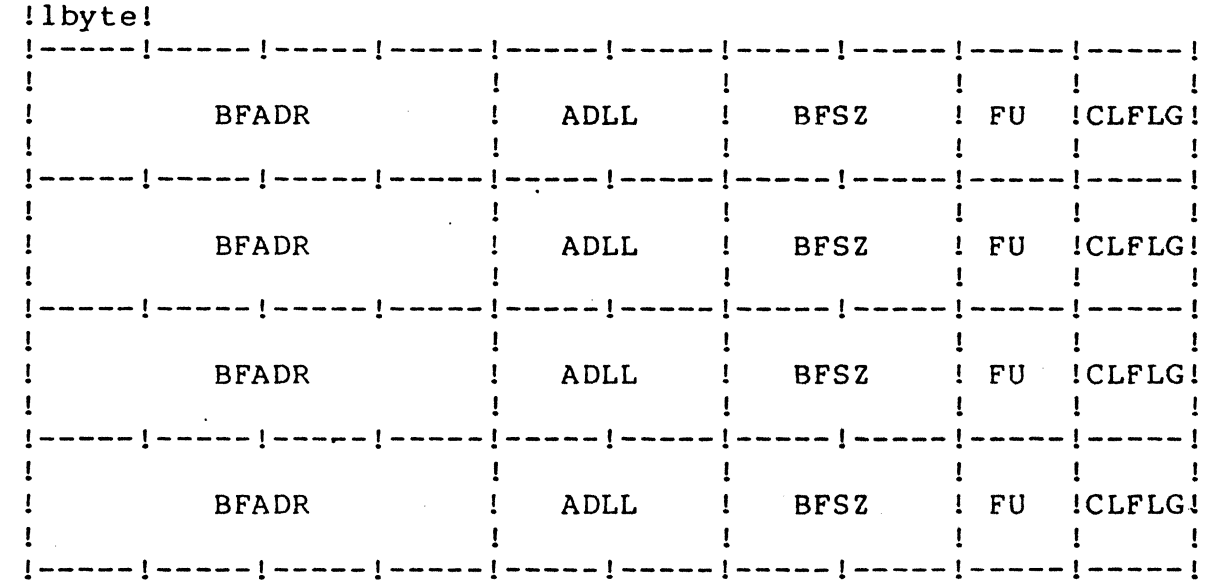


The first part is the COMMAND LIST (80 bytes):

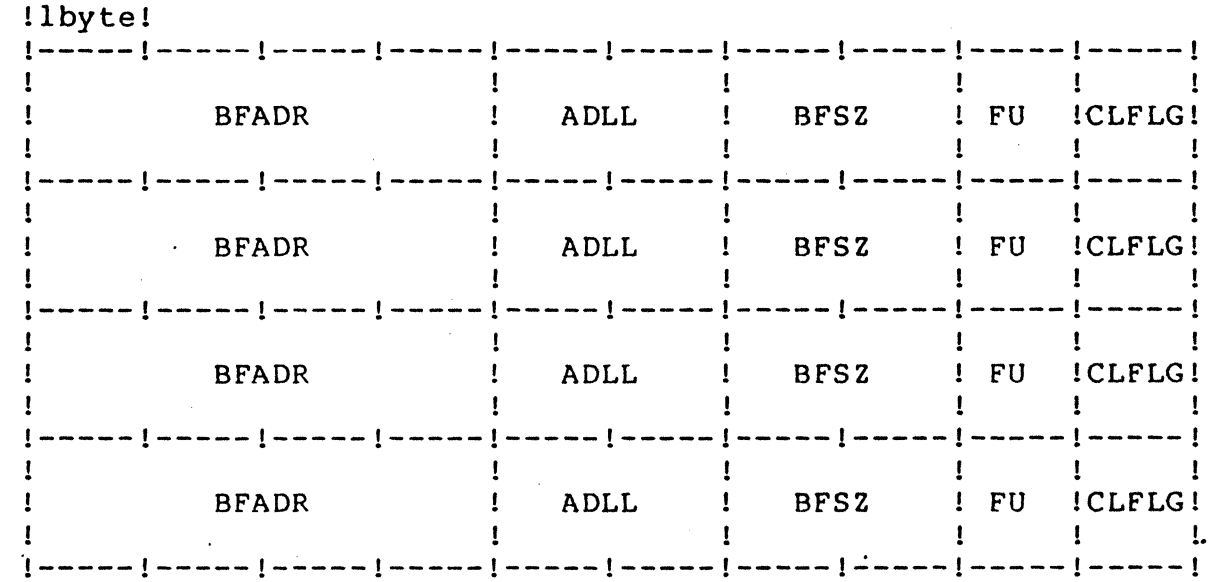
Divided in RX and TX section, consists of 4 entries for each channel. The dma software allows a mechanism of circular list of the four entries.

Each entry consists of 10 bytes.

RX COMMAND LIST (4 ENTRIES)



TX COMMAND LIST (4 ENTRIES)



- buffer address (4 bytes): BFADR - set by MASTER

- actual data lenght (2 bytes): ADLL
  (remaining bytes to be transferred)
  (number of transferred bytes = BFSZ - ADLL)
  ADLL will be valid after an abort command or at the end of the entry execution.

- RX/TX : set by DMA
  reset by MASTER

- buffer size (2 bytes): BFSZ - set by MASTER
  (number of bytes to be transferred)

- FU (1 byte)

- command list flag (1 byte): CLFLG

bit(0) = 1 intermediate command list
= 0 last command list
set by MASTER

bit(1) = 1 command already executed
set by DMA after INT..
reset by MASTER

bit(2) = 1 slave must interrupt master
= 0 no interrupt to master
set and reset by SLAVE

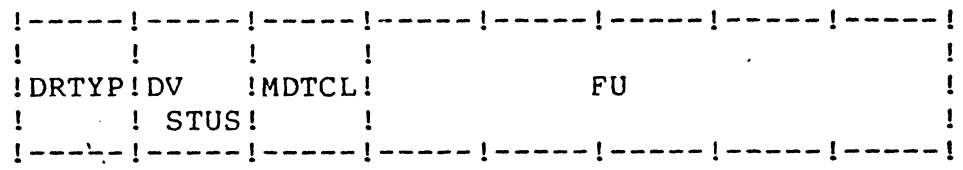
bit(3) = 1 intermediate interrupt
= 0 no intermediate interrupt
set and reset by SLAVE

bit(4) = 1 channel error interrupt
= 0 no channel error interrupt
set and reset by SLAVE

bit(5) = 1 event interrupt
= 0 no event interrupt
set and reset by SLAVE

bit(6) = 1 command interrupt
= 0 no command interrupt
set and reset by SLAVE

The second part is the RX/TX COMMON AREA (8 bytes) and consists of:



- driver type (1 byte): DRTYP

value:

VIP Hex 04
BSC Hex 08
SNA Hex 09
X25 Hex 0A

- device status (1 byte): DVSTUS

value:

opened Hex 01
closed Hex 00
set by MASTER

- mode control register (1 byte): MDCTL

bit(0) = 1 COMPLEX MODE RX CTL. TABLE character treatement with the possibility to link the buffers.

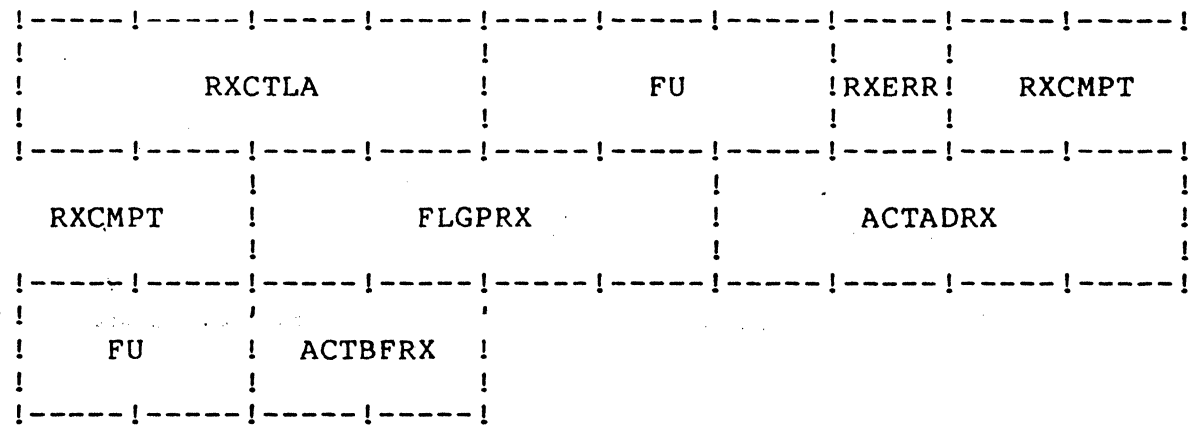
bit(1) = 1 COMPLEX MODE TX CTL. TABLE character treatement with the possibility to link the buffers.

bit(2) = 0 LINKED MODE no character treatement but with the possibility to link the buffers.

bit(3) = 1 FRAME MODE

These flags are set by MASTER at the open time; these status is kept until the close is requested.

The third part is the RX SERVICE AREA (24 bytes):



- Rx control table address (4 bytes): RXCTLA
  pointer to control table used by the channel in the complex routine.
  set by MASTER

- FU (3 byte)

- RX error status (1 byte): RXERR
  set by DMA

- RX current command list pointer(4 bytes): RXCMPT
  pointer to active RX entry.
  used by DMA

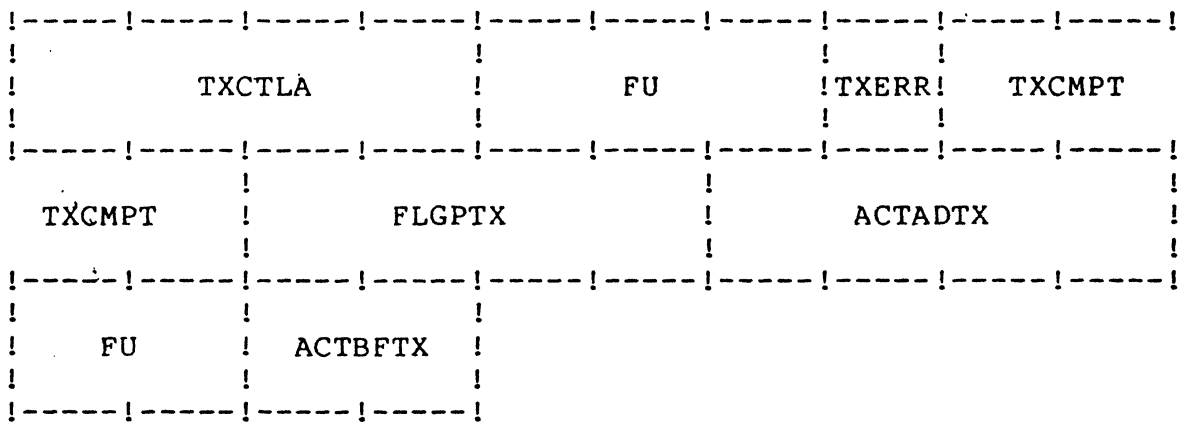
- RX command list flag pointer (4 BYTES): FLGPRX
  used by DMA

- RX active buffer address (4 BYTES): ACTADRX
  used by DMA

- FU (2 byte)

- RX active buffer size (2 BYTES): ACTBFRX
  used by DMA

The fourth group is the TX SERVICE AREA (24 bytes):



- TX control table address(4 bytes): TXCTLA
  pointer to TX control table used by the channel in the complex routine.
  set by MASTER

- FU (3 byte)

- TX error status (1 byte):TXERR
  set by DMA

- TX current command list pointer (4 bytes): TXCMPT
  pointer to active TX entry.

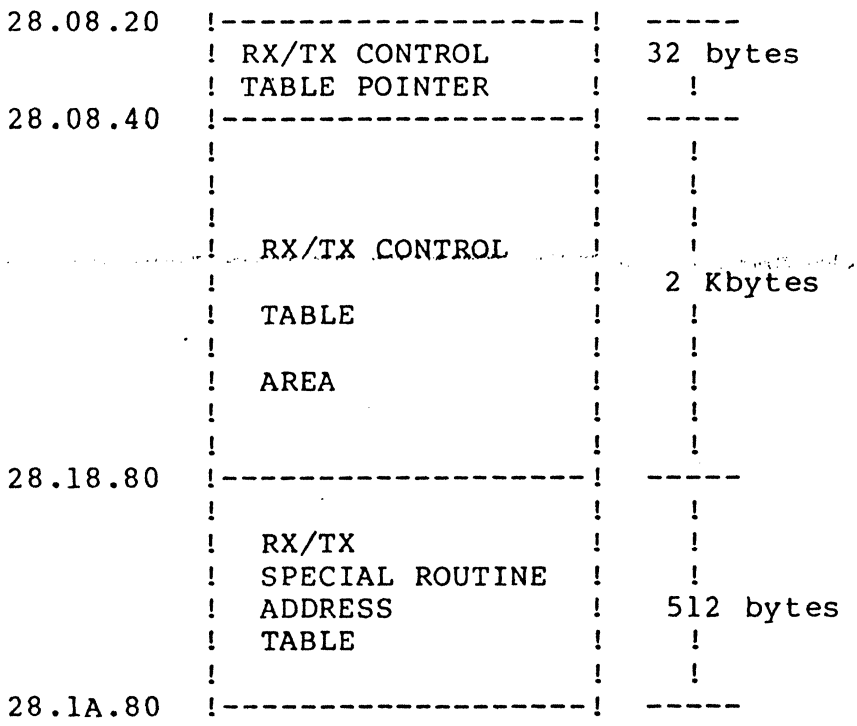
- TX command list flag pointer (4 BYTES): FLGPTX
  used by DMA

- TX active buffer address (4 BYTES): ACTADTX
  used by DMA

- FU (2 byte)

- TX active buffer size (2 BYTES): ACTBFTX
  used by DMA

At the end of the four channels there are three group of data.



The first group is the RX/TX CONTROL TABLE POINTER (32 bytes):
There are 8 pointers so dedicated:

POINTER 1
- VIP RX control table pointer (4 bytes):
  set by DMA
  copied by the MASTER in RXCTLA at open time of the channel.

POINTER 2
- VIP TX control table pointer (4 bytes):
  set by DMA
  copied by the MASTER in TXCTLA at open time of the channel.

POINTER 3
- FREE

POINTER 4
- FREE

POINTER 5
- FREE

POINTER 6
- FREE

POINTER 7
- FREE

POINTER 8
- FREE

The second group is the RX/TX CONTROL TABLE AREA (2 Kbytes), pointed by the pointers of the first group:

AREA 1 (256 bytes)
- RX VIP CONTROL TABLE
  This area is filled with the 08 value except the 04 position that contains the 04 value.

  0 is the displacement for the routine address table to jump at the subroutine with NORMAL RETURN.

  4 is the displacement for the routine address table to jump at the subroutine with EOT TREATEMENT.

  8 is the displacement for the routine address table to jump at the subroutine with NO PARITY DATA.

AREA 2 (256 bytes)
- TX VIP CONTROL TABLE
  This area is filled with 0.

AREA 3 (256 bytes)
- FREE
  This area is filled with 0.

AREA 4 (256 bytes)
- FREE
  This area is filled with 0.

AREA 5 (256 bytes)
- FREE
  This area is filled with 0.

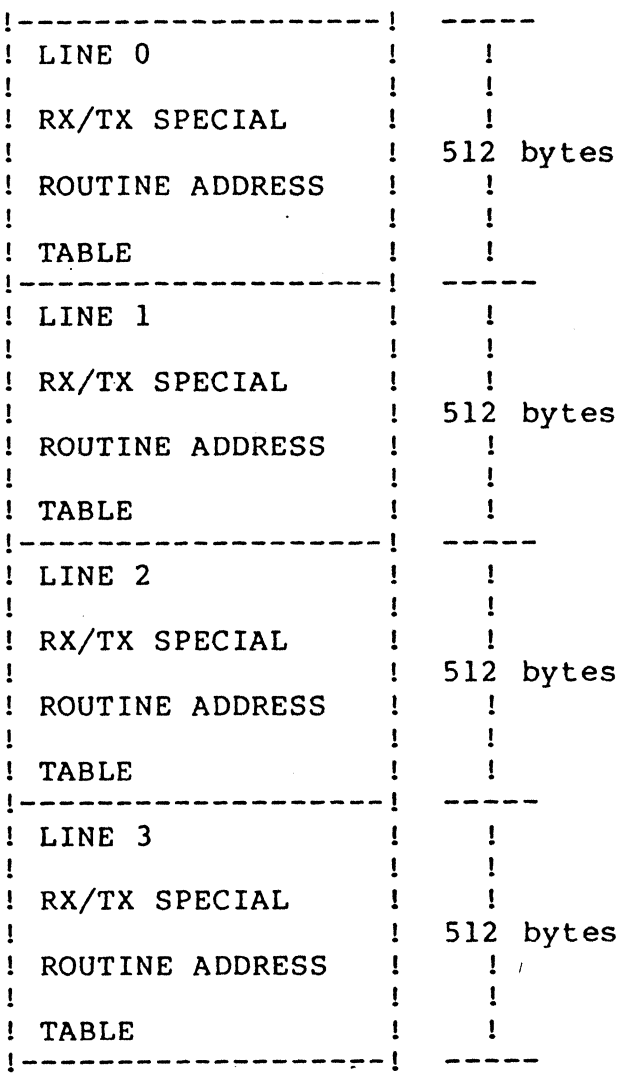
AREA 6 (256 bytes)
- FREE
  This area is filled with 0.

AREA 7 (256 bytes)
- FREE
  This area is filled with 0.

AREA 8 (256 bytes)
- FREE
  This area is filled with 0.

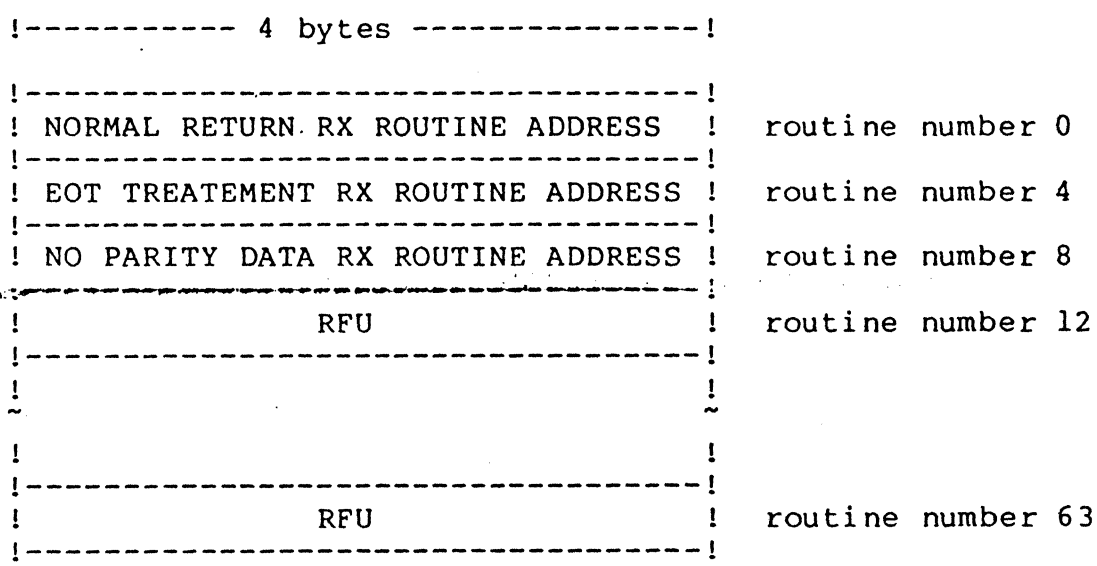
The third group is the RX/TX SPECIAL ROUTINE ADDRESS TABLES (2 Kbytes) used in complex mode.

For each line there is a structure of 512 bytes, 256 for the RX channel and 256 for the TX channel, that allows 63 different routines for each channel.:

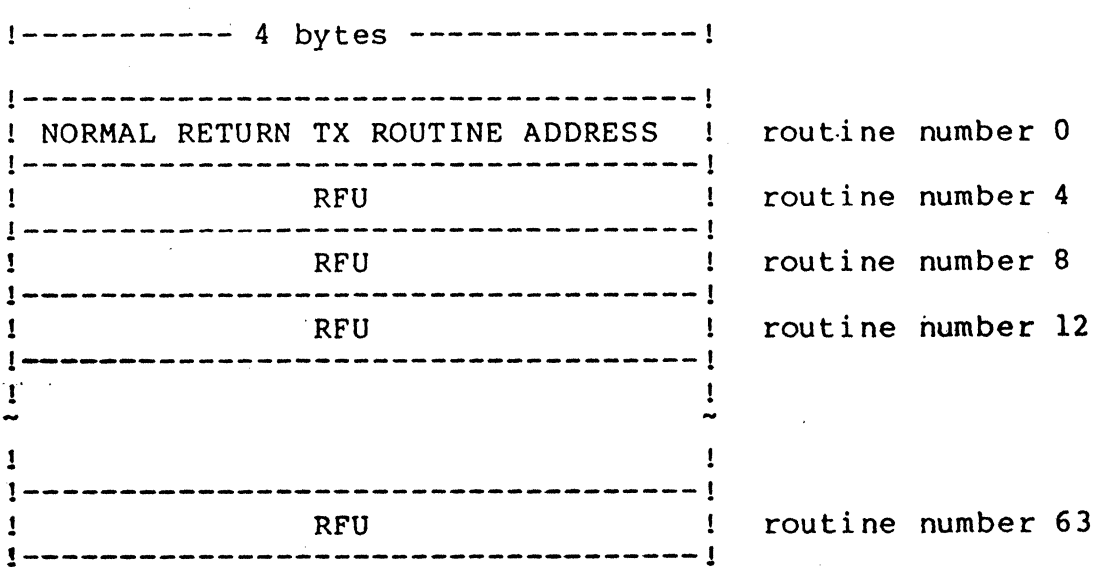


An example, for a line, is showed below:

RX ROUTINE ADDRESS (256 bytes):



TX ROUTINE ADDRESS (256 bytes):



PAY ATTENTION THAT TAS INSTRUCTION FOR THE MASTER PROCESSOR IN THE I/O BUS IS NOT ALLOWED !!

E.4 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	DO3
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 ASSIGNMENTS (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.)

N.B.- For J2/P2 VMEbus pin assignments refer to previous description.

SECTION F

DISK CONTROLLER (DCS)

PRODUCT DESIGN DESCRIPTION

PREPARED BY: R. POZZAN

REVIEWED BY: R. POZZAN

INDEX OF SECTION F

		PAGE
F.1	DISK CONTROLLER OVERVIEW	274
F.2	MAJOR BLOCKS DESCRIPTION	275
F.2.1	IMDC	276
F.2.2	DPLL	278
F.2.3	BIM	279
F.2.4	VMEbus INTERFACE	280
F.2.5	INTERNAL DATA PATH	281
F.3	INTERNAL REGISTER MAP	282
F.3.1	REGISTERS PROGRAMMING	284
F.3.2	CHANNELS CONFIGURATION	285
F.4	MAIN MEMORY TABLES	286
F.5	MASTER-SLAVE INTERRUPT FEATURES	292
F.6	THEORY OF OPERATION	293
F.6.1	OFF LINE COMMANDS	293
F.6.1.1	RECALIBRATE COMMAND	293
F.6.1.2	SEEK COMMAND	294
F.6.1.3	CHIP DIAGNOSTIC COMMAND	295
F.6.2	DATA TRANSFER COMMANDS	296
F.7	SW GUIDELINES	297
F.7.1	ECA TABLES PROGRAMMING	297
F.7.2	OPERATIONS STRUCTURE	299
F.7.3	ERRORS MANAGEMENT	300
F.7.3.1	SW ERRORS	301
F.7.3.2	HW ERRORS	302
F.7.3.3	MEDIA ERRORS	303
F.7.3.4	ECC HANDLING	304
F.8	FORMAT UTILITY	305
F.8.1	BAD SECTORS HANDLING	306
F.8.2	ERRORS RECOVERY PRCEDURE	307
F.8.3	STRUCTURE OF FORMAT UTILITY	308
F.8.4	STRUCTURE OF BAD BLOCK MAP	309
F.8.5	VENDORS INFORMATIONS COMPUTING	310
F.9	DENSITY MODE	311

F.1 D.C.S. CONTROLLER OVERVIEW

This section is a general specification intended to describe the features of the DISK/DKT STREAMER/TAPE CONTROLLER connected to the SGM2 system.

F.2 MAJOR BLOCKS DESCRIPTION

A specific hardware architecture performs the connection between the DISK CONTROLLER and the VMEbus side and provides all the hardware interfaces to the device side.
The general configuration of the D.C.S.CONTROLLER is depicted in Fig. F.1.

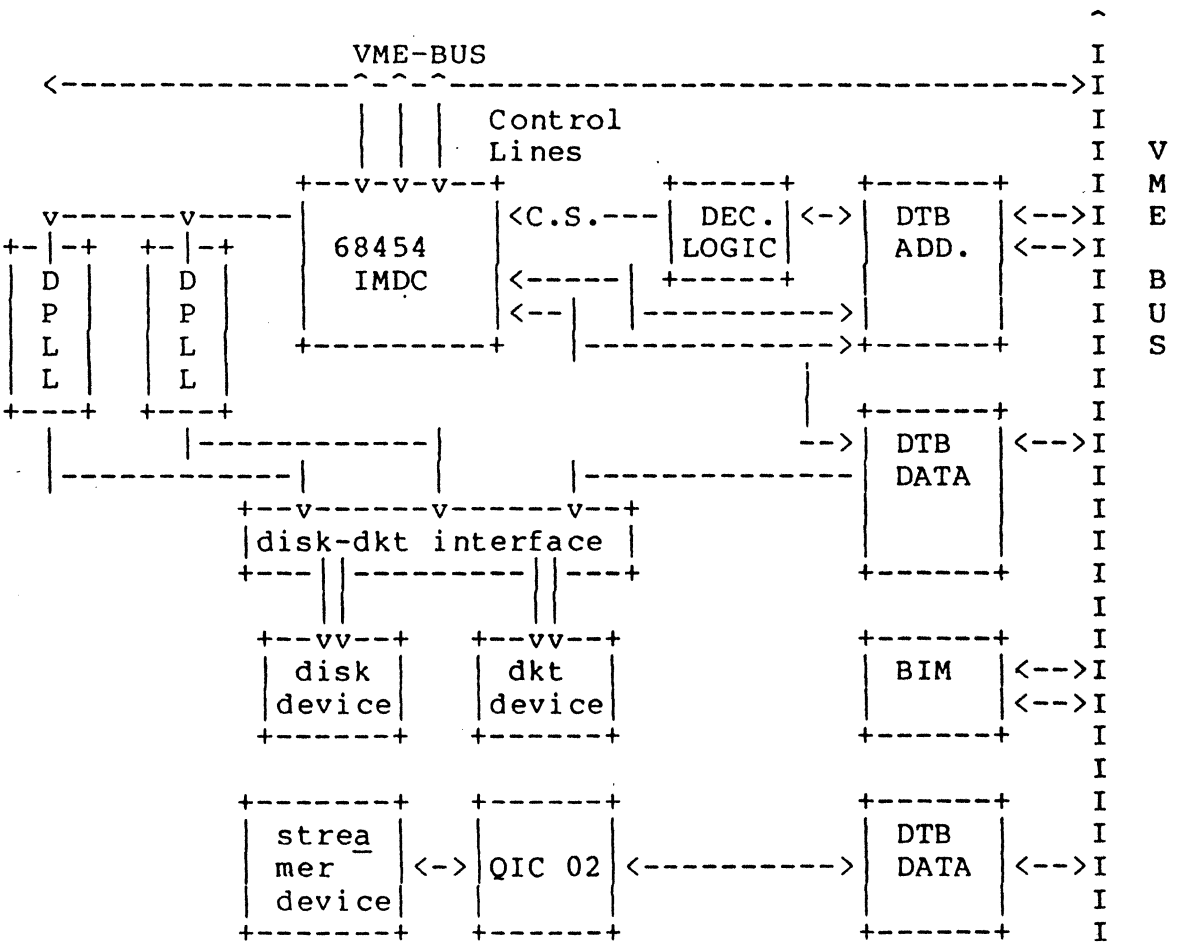


Fig. F.1 - DCS MAJOR BLOCK DIAGRAM

F.2.1 IMDC SCN68454

The SCN68454 Intelligent Multiple Disk Controller (IMDC) provides the traditional and andvanced features required to control Winchester rigid disk an Floppy disk drive. It provides for an internal data path two byte wide demultiplexed for address and data, all the control signals to handle bus dialogue and all the transfers to or from disk interface.

IMDC works with a 31 bit address counter and 16 bit of parallel data.

The controller is programmed via an external host processor by the use of high level commands, specific tables in Main Memory (E.C.A.) and the data transfer on the host data is 16 bit in parallel.

The IMDC supports Soft Sectored disk track format and standard track format for floppy disk (double density/side) and is capable of handling a serial data rate up to 10Mbit per second.

The IMDC is able to execute multiple sector READ/WRITE, with implied seek, and data transfer to/from main memory are executed via an internal DMA controller. In addition a temporary dinamical data storage is foreseen with an internal FIFO buffer 128 bytes long.

The IMDC provides four possible operating modes:

a) REGISTER MODE

This operating mode refers to the state when the IMDC is chip selected to allow READ/WRITE internal register

b) DMA MODE

The term DMA MODE refers to the state when the IMDC assumes ownership of the bus

c) LOCAL MODE

The term LOCAL MODE refers to the state when the IMDC is transferring command to the disk interface or receives control informations from disk interface

d) IDLE MODE

The term IDLE MODE refers to a wait state. For example the IMDC is in IDLE MODE at the reset time or when waits for a seek or recalibrate commands to be ended.

The IMDC provides also automatic error recovery procedures in case of BUS ERROR (automatic rerun);it handles automatically bad and alternative sectors and provides 32/40 bit ECC programmable polynomials or computer generated ECC polynomials.
To allow a complete user programmability the IMDC provides an internal set of accessible registers each 8 bits wide and some main memory tables named E.C.A.(Event Control Area).

F.2.2 DPLL SCB68459

The SCB68459 Disk Phase Locked Loop (DPLL) is a bipolar chip that used together with the IMDC 68454 provides all the functions to control disk/dkt devices with SA800, ST506, ST1000 interface types.
The DPLL uses an external VCO for the variable clock rate which tracks the read data from the disk unit.
Disk/dkt controller uses two DPLL chips because two different frequences are needed. In fact disk device runs at 5Mbit per second while dkt device runs at 250/500Kbit per second.
The SCB68459 operates producing an oscillating frequency to match the frequency of an input signal.
In this locked condition any slight change in the input frequency will appear as a change in phase between the input frequency and the VCO frequency. This phase shift then acts as an error signal to change the frequency of the local DPLL VCO to match the input frequency.
The DPLL SCB68459 is able to accept NRZ data and write clock information in parallel from SNC68454. It combines data and clock to obtain a resulting signal that becomes the write data signal which goes to the disk drive.
The DPLL also built write precompensation algorithm in order to adjust special clock data patterns that require compensation. An external delay line is used to generate the precision compensation needed.
With a PLL circuit built into a read chain data can be more accurately recovered. Aided by an external voltage controlled oscillator the DPLL function derives a clock signal from the read data streams and tracks it through a range of variation.
The decoding hardware uses the clock signal to couple the read data stream.

F.2.3 B.I.M. MC68153

The bipolar LSI MC68153 bus interrupter interfaces a micro computer system bus to multiple slave devices requiring interrupt capabilities.
It handles up to four independent sources of interrupt request and is fully programmable.
There is one control register for each interrupt source.

CR0 control INT0 not used
CR1 control INT1 interrupt of the IMDC
CR2 control INT2 interrupt of the ~~mass~~ (tape) really
CR3 control INT3 interrupt of the exception (tape)

Each interrupt has its own associated vector register (VR0-VR3).
These 8 read/write register are used by the host for full control of the interrupt request.

F.2.4 INTERFACE AND LOGICAL HANDLING VME BUS

This interface is composed in five functional blocks.

- DATA TRANSFER BUS ADDRESS

This functional block is capable to supply in output the main memory address (32 bit) and to accept in input some address lines to generate chip selection for read write register function (IMDC, BIM, TAPE).

- DATA TRANSFER BUS DATA

This functional block is capable to supply or to accept 16 bits of data to or from VME bus.

- DATA TRANSFER BUS CONTROL LINES

This functional block controls all the signals involved during a read/write bus cycle.

- INTERRUPT REQUESTER

This functional block is capable to generate an Interrupt request to the host and to recognize the proper Interrupt acknowledge response.

- ARBITER INTERFACE

This functional block controls all the signals involved during a request bus cycle.

F.2.5 INTERNAL DATA PATH

This is an internal data path 2 bytes wide capable to carry address and data to and from the VME bus. It also carries in output the commands to the devices and in input the status to the IMDC.

F.3 INTERNAL REGISTRERS MAP

HEX ADDRESS	54000000	EPH	Eca Pointer High
HEX ADDRESS	54000001	EPMH	Eca Pointer Middle High
HEX ADDRESS	54000002	EPML	Eca Pointer Middle Low
HEX ADDRESS	54000003	EPL	Eca Pointer Low

These four registers are used by the host system to direct the IMDC to a table of pointers.
At reset time these registers will be initialized to zero.

HEX ADDRESS 54000004 IVR Interrupt Vector Register

The Interrupt Vector Register contains the value that the IMDC will place on the data bus upon receipt of an interrupt acknowledge from the CPU.
The value for this register must be HEX VALUE xxxxxxxx (SW defined).
At reset time this register will be initialized to 0F hex

HEX ADDRESS 54000005 ISR Interrupt Source Register

For this register only the bits 7-6-5-4 are meaning.
These bits are used to indicate which drive was the source of a command completion interrupt.
Bit4 reflects drive 0 as source and bit 7 is for drive 3.
At reset time these four bits will be initialized to zero.

HEX ADDRESS 54000006 DSCR Drive Status And Configuration Register

Bit 0 of this register means 8-16 bits mode. This bit will be set to one to obtain 16 bits length of main memory and register data transfer. At reset time this bit is set to zero.
Bits 7-6-5-4 of this register mean drive busy.
These bits are sets by the host system to initiate an IMDC command operation for a particular drive.
Bit 4 is used for drive 0, bit 7 is used for drive 3.

HEX ADDRESS	54001001	CR0	control register 0
	54001003	CR1	control register 1
	54001005	CR2	control register 2
	54001007	CR3	control register 3

These four registers are used by the host system to control the operating mode.

HEX ADDRESS	54001009	VR0	vector register 0
	5400100B	VR1	vector register 1
	5400100D	VR2	vector register 2
	5400100F	VR3	vector register 3

These four registers contain the value that the B.I.M. will place on the data bus upon receipt of an interrupt acknowledge from the CPU.
The B.I.M. is programmed in traspàrent mode for of IMDC interrupter.

HEX ADDRESS	54003000	CRW	command register write
	54003002	CRR	command register read
	54003004	DRW	data register write
	54003006	DRR	data register read
	54003008	RFFR	reset flip flop ready
	5400300A	RFFE	reset flip flop exception

These are registers for the handshaking between host and streamer tape.
At reset time these registers are initialized randomly.

HEX ADDRESS	54005000	SDM	set single density mode
	54006000	DDM	set double density mode

These two registers are used by the host to select 250KHz or 500KHz floppy transfer rate.
At reset time, 250KHz is selected.

F.3.1 REGISTERS PROGRAMMING

The first register that the user must write is the DSCR register. The user must write this byte register as a word register: LOAD in DSCR 0101 HEX value.
At this point the D.C.S. controller is programmed with an interface Dma Data and Data Register for two bytes. It is suggested that the user, when needs to write the DSCR register, always uses a word instruction and always must sets the 16 bit mode, togheter the command bit, while for all the others registers the user can choose a single byte or word transfer.
The four control register (CR0 - CR3) are used by the host for the full control of the operation of the device.
The other four register (VR0 - VR3) are vector register that contain the vector data used diring an interrupt acknowledge cycle.
The first pair of registers (CR0 VR0) are not used.
The second pair of registers (CR1 VR1) are used by the host for control of IMDC interrupt, the user should write 34 HEX value in the CR1, in this case the VR1 is not used.
The other two control registers (CR2 CR3) are used by the host for controlling the interrupt of the tape.
The host should write 14 HEX value in the CR2 and CR3.
The other two vector registers contains the value that the BIM will place on the data bus upon receipt of an interrupt acknowledge from the CPU.
The value for these registers must be HEX value.

Code says X'04'
but that's for polling mode!

F.3.2 CHANNELS CONFIGURATION

The D.C.S.controller connects up to three hard Disk drive and one Floppy Disk drive double density.
The HW connection provides Disk Drive on channel number 1,2,3 and Floppy drive on channel number 4.
That means that Disk Commands will be issue writing in DSCR register an HEX value equal to 1111 or 2121 or 4141 and the commands for Floppy drive will be issue writing in DSCR register an HEX value equal to 8181.
The D.C.S. controller also accept overlapping command: the host sets same channel busy to obtain overlapping mode.

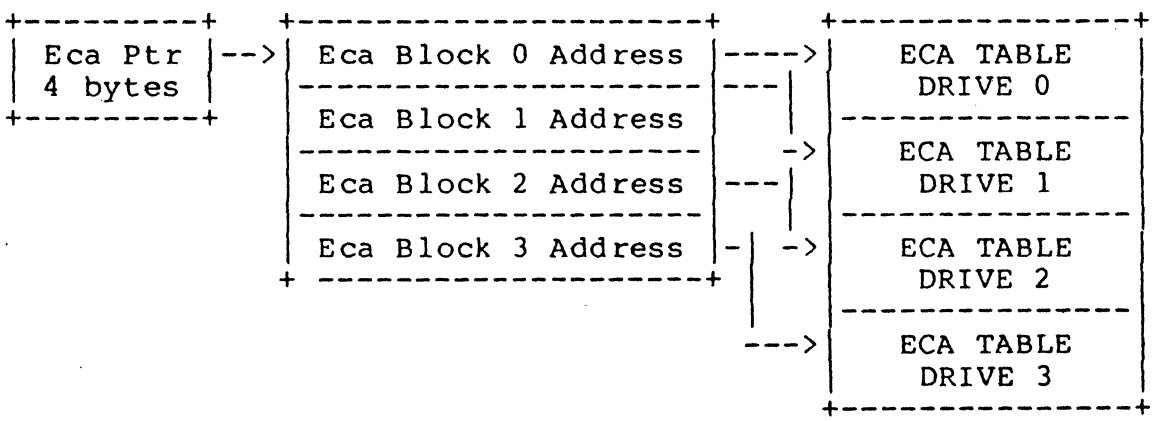
F.4 MAIN MEMORY TABLES

The IMDC needs some special tables located in system memory to be able to execute the commands required by the host system. There are no restrictions to locate these tables, the only condition that the IMDC requires is that the tables must be located to an even address boundary.
Three types of tables must be generated in system memory:

- Pointers Table
- Event Control Area Tables
- Format Tables

- POINTERS TABLE

This table is addressed by the content of the EPH-EPHL-EPML-EPL registers and consists of four 2 words addresses that point to the location of four ECA blocks.
The pointers must be arraged in ascending order by drive



- ECA TABLE

The IMDC needs the ECA tables (Event Control Area) to communicate with the host system. An ECA table must be set for each disk/dkt drive to be controlled.
The ECA table contains informations about the requested command and the disk drive. The internal microprogram will use these tables to generate the disk interface signals and performs the I/O operation.
The IMDC also will use these tables to return the status informations to the host system.
The disk drive to ECA block assignement is determined by the relative position of the pointer in the table.
Fig F.2 shows the ECA block format.

16 bit

Word #	15 — 8	7 — 0	hex
00	Command Code	Main Status	00
01	Extended Status		02
02	Max # of Retries	Actual # of retries	04
03	Dma Count	Command Options	06
04	Buffer Address Most Significant Word		08
05	Buffer Address Least Significant Word		0A
06	Buffer Length Requested		0C
07	# of Bytes Transferred		0E
08	Cylinder Number		10
09	Head Number	Sector Number	12
10	Current Cylinder Position		14
11	PRP Command Control Word		16
12	SCWT Most Significant Word		18
13	SCWT Least Significant Word		1A
14	Scan Terminator	RFU	1C
15	Maximum Record Length -1		1E
16	N0 Pre Index Gap	N1 Post Index Gap	20
17	N2 Sync Byte Count	N3 Post ID Gap	22
18	N4 Post Data Gap	N5 Add.Mark Count	24
19	R.F.U.	R.F.U.	26
20-22	E.C.C. Mask 3 Words		28 2A 2C

FIG. F.2 - ECA BLOCK FORMAT

23	Motor On Delay	# of Heads	2E
24	End Sector Number	Stepping Rate	30
25	Head Settling Time	Head Load Time	32
26	Seek Type	Phase Count	34
27	Low Write Current Boundary Track		36
28	Precompensation Boundary Track		38
29-31	E.C.C. Remainder 3 Words		3A/3C/3E
32	Maximum Number Of Cylinder Per Surface		40
33	Sector Length	Flag Byte	42
34-35	B Tree Pointer (2 Words)		44/46/42
36-45	IMDC Working Area 10 Words		48

FIG. F.2 - ECA BLOCK FORMAT (CONT.)

ATTENTION:

Sector length field in ECA table allows also with the meaning of the most significant bit (bit 7),to inform the IMDC that the track starts with either Sector 0 or Sector 1.

- FORMAT TABLES

Since the track formatting is user programmable, the host system can choose to format the track in any manner compatible with its requirements by means of some particular tables named format tables.
Here below fig. F.3 shows a summary of floppy and disk parameters and fig. F.4 shows the format tables for floppy and disk drive.

FLOPPY PARAMETERS				DISK PARAMETERS			
Descrip.	IMDC-uses	Hex Data Value	MFM-Cnt	Descrip.	IMDC-uses	Hex Data Value	MFM-Cnt
pre idx gap	N0	4E	80	idx gap	N1	4E	22
sync field	N2	00	12	sync fld	N2	00	13
idx mark	N5	C2	03	id add mrk	N5	A1	01
idx flag		FC	01	id add flg		FE	01
idx gap	N1	4E	50	cyl.val.	DMA	xx	01/02
sync fld	N2	00	12	head val.	DMA	xx	01
id add mrk		A1	03	sector	DMA	xx	01
id add flg		FE	01	crc		xx	02
cyl val	DMA	xx	01	id gap	N3	00	03
side	DMA	xx	01	sync fld	N2	00	13
sector	DMA	xx	01	data mrk	N5	A1	01
rn length	DMA	xx	01	data flg		FB	01
crc ccitt		xx	02	data byte		fill byte	512
id gap	N3	4E	22	crc		xx	02
sync field	N2	00	12	data gap	N4	00	03
data mark	N5	A1	03	sect gap		4E	15
data flag		FB	01	last gap		4E	346
data byte		fill byte	512				
crc ccitt		xx	02				
data gap	N4	4E	50				
last gap		4E	216				

Fig. F.3 - DISK AND FLOPPY PARAMETERS

DKT FORMAT TABLE			DISK FORMAT TABLE			
00	*	02	00	*	00	high
A1	*	FE	A1	*	FE	low
80	*	N3-3	00	*	N3-3	high
4E	*	sector length	00	*	sector length	low
00	*	02	00	*	N5-1	high
A1	*	FB	A1	*	FB	low
Filler = 6D	*	00	Filler = 6D	*	00	high
4E	*	N4-5	00	*	N4-2	low
4E	*	00	4E	*	00	high
00	*	N5-1	4E	*	00	low
C2	*	FC	00	*	00	high
4E	*	00	00	*	00	low

Fig F.4 - DISK AND FLOPPY FORMAT TABLES

NOTE

Format tables must be aligned to an even address boundary

F.5 MASTER-SLAVE INTERRUPT FEATURES

The IMDC can be either Master or Slave on the System Bus depending by the functionality executed.
The IMDC will be Slave when the Host System needs to read or write an internal register.
The IMDC will be Master when a data transfer will occour between the IMDC and the Main Memory or viceversa.
The IMDC becomes Master when needs to exchange data with the Main Memory, asserting the Bus Request line to the Bus Arbiter.
Since the disk controller is the highest priority processor, next bus cycle will be assigned to it. Disk controller will be Master on bus for the duration of data transfer that could be more than one single word.
In fact the IMDC can be programmed to obtain a BURST DATA TRANSFER. In this way the IMDC will be Master on bus for the time needed to exchange the total BURST length.
The IMDC is also an Interrupter Processor.
This means that when an operation is completed the IMDC notifies the host system asserting an Interrupt Request on the bus interrupter module (B.I.M.).
The B.I.M. asserts on bus to the interrupt handler all the interrupts of the board (IMDC, TAPE). The BIM is programmed in trasparent mode, because the IMDC is autovectorised, but for the tape it is enable for the internal interrupt vector.

F.6 THEORY OF OPERATIONS

The IMDC is capable to execute a set of commands by means of a Command Code byte in ECA table.
Two types of Commands can be distinguished:

-) OFF LINE COMMANDS
-) DATA TRANSFER COMMANDS

F.6.1 OFF LINE COMMANDS

Three function codes are included in this kind of commands:

- RECALIBRATE TO TRACK ZERO
- SEEK CYLINDER
- CHIP DIAGNOSTIC

F.6.1.1 RECALIBRATE COMMAND

Recalibrate command acts a retraction of the head carriage to track zero. The IMDC issues steps until the track zero signal goes active on interface. A check is made on the maximum number of steps to be supplied and a check also is made on seek completed time out.
This function can be initiated automatically by disk controller when special condition occurs as encountered error in cylinder number by reading or writing.

PROGRAMMING

To obtain the execution of a Recalibrate command the Host System must compile the following E.C.A. field in the appropriate E.C.A. table:
-Command Code 41 Hex Add.00 of ECA table
-Cylinder Number 00 Hex Add. 08 of ECA table
When the Interrupt is received the host system must check the Main Status to verify the correct execution of the command. In fact Main Status equal to 00 Hex means correct execution without error. A different value in Main Status means that an error has occourred. The host system, in case of error, must check also the Extended Status Bit 7 and BIT 13 of the Extended Status may be significant. In this case SW retries should occour. The host system should verify also that the Current Cylinder in ECA table Add.10 should be equal to zero.

F.6.1.2 SEEK COMMAND

Seek Command acts a positioning of the head carriage to the desired Cylinder number and selects the desired Head number needed by the user. Since the IMDC does not include a specific function code for Seek Command, the functionality can be obtained implementing a Read/Write command with data transfer length equal to zero.

PROGRAMMING

To obtain the execution of a Seek Command the host system must compile the following ECA field in the appropriate ECA table:

-Command Code =11 Add. 00 of the ECA table
-Cylinder number = loaded with the value of cylinder required Add. 08 of the ECA table
-Head number = loaded with the value of head required Add. 09 of the ECA table

When the Interrupt is received the host system must check the Main Status to verify the correct execution of the command. In fact Main Status equal to 00 Hex means correct execution without error. A different value in Main Status means that an error has occourred. The host system, in case of error, must check also the Extended Status. Bit 7 and Bit 13 of Extended Status may be significant. In this case SW retries should occur.

F.6.1.3 CHIP DIAGNOSTIC COMMAND

Chip Diagnostic command exercises some internal hardware register features. This command will never be activated by the host system.
Only Internal Diagnostic Routine of the system can activate this command.

PROGRAMMING

-Command Code = 80 Add. 00 of ECA table device 0

When the Interrupt is received the Internal Diagnostic Routine should check the Main Status in order to verify the correct execution of the command. Main Status equal 00 means that Internal Hardware special Register runs correctly, while XX value in Main Status means an internal failure detected.

F.6.2 DATA TRANSFER COMMANDS

Nine function codes are included in this kind of commands:

- FORMAT
- WRITE MULTIPLE SECTORS
- WRITE WITH DELETED DATA FLAG
- READ MULTIPLE SECTORS
- VERIFY
- READ IDENTIFIER
- TRANSPARENT SECTOR READ
- CORRECT DATA
- PROGRAMMABLE RECORD PROCESSING

F.7 SW GUIDELINES

F.7.1 ECA TABLE PROGRAMMING

The correct value to be applied at any fields of ECA table are specified below:

MAX NUMBER OF RETRIES = 10 hex

DMACOUNT = 0F hex

COMMAND OPTIONS = 21 hex (Floppy disk)
2B hex (Hard disk)

TRACK FORMAT DEFINITIONS

N0 = 50 hex
N1 = 32 hex
N2 = 0C hex for Floppy
N3 = 16 hex
N4 = 36 hex
N5 = 03 hex

N0 = 16 hex
N1 = 16 hex
N2 = 0D hex for Hard
N3 = 03 hex
N4 = 0D hex
N5 = 01 hex

MOTOR ON DELAY = 50 hex (Floppy disk)
00 hex (hard disk)

OF HEADS = 02 (Floppy disk)
09 (Wren2 94155-86)
05 (Wrenl 9415-5-36)
04 (NEC 5126)
04 (Miniscribe 3425)

ENDING SECTOR = this field is function of the format size
For example if the track ends with sector number 9
ending sector will contain 9.

STEPPING RATE = 0F (Floppy)
00 (Hard)

HEAD SETTLING TIME = 1E hex (Floppy)
00 (Hard)

HEAD LOAD TIME = 46 hex (Floppy)
00 (Hard)

SEEK TYPE = 0 (Floppy)
2 (Hard)

LOW WRITE CURRENT BOUNDARY TRACK = 2B hex (Floppy)
80 hex (Hard)

PRECOMPENSATION BOUNDARY TRACK = 2B hex (Floppy)
80 hex (Hard)

MAXIMUM NUMBER OF CYL PER SURFACE = 39D hex (Wren2 94155-86)
2B9 hex (Wrenl 9415-5-36)
264 hex (Miniscribe 3425)
267 hex (NEC 5126)
50 hex (Floppy)

ECC MASK 3 words
000000000000 value for floppy
0000140A0445 value for hard disk

F.7.2 OPERATIONS STRUCTURE

The first operation of the HOST is to initialize the IMDC , this phase consists of:

- transfer mode selection (byte/word mode)
- load interrupt vector register
- load ECA registers with the ECA pointers

After IMDC initialization the Host can issue a command request to the IMDC, by setting the busy bit (of DSCR) corresponding to the drive interesting.

The IMDC will accept the request and will put it pending, if it is doing others commands for the other drives.
The IMDC will start the execution of command as soon as possible.

Therefore the IMDC is able to accept different commands simultaneously one for every drive.

Each drive command must be sent indipendently of the commands for the other drives.

F.7.3 ERROR MANAGEMENT

During the I/O operations the controller will write in particular fields of ECA table (MAIN and EXTENDED STATUS) relative to the interested drive.
These fields qualify the I/O operation by means of proper bits.

We could have two possible situations:

- Corrected I/O operation (Main Status=0)
- Uncorrected I/O operation

We could divide the possible error messages into three types:

- SW ERRORS
- HW ERRORS
- MEDIA ERRORS

F.7.3.1 SW ERRORS

The IMDC signals these errors and doesn't execute retries, indipendently from the MAX # OF RETRIES value in ECA.

- COMMAND REJECTED (Main Status = 6)

  This bit is set if the command code value compiled by the HOST is unknown.

- DRIVE NOT READY (Main Status = 2)

  This bit will arise if the command operation is sent to a specific not ready drive.

- COMMAND ABORT (Main Status = 10)

  This bit is set by IMDC after the busy bit reset by HOST.

PRP OPERATION UNSUCCESSFULL (Main Status = 3)

... TBD ...

IRRECOVERABLE ERRORS (Main Status = 1)

WRITE PROTECT DISKETTE (bit6 EXTENDED STATUS)

This bit will arise if the Host attempts a write operation on a write protected diskette.

POSITIONING ERROR (bit7.EXTENDED STATUS)

This bit is set if the HOST sends an operation with CYLINDER NUMBER greater then MAX CYL per SURFACE of ECA table.

DELETED ADDRESS MARK (bit5.EXTENDED STATUS)

This error message arises if the HOST allows the deleted sector handling specified in COMMAND OPTION byte of ECA.

F.7.3.2 HW ERRORS

After these errors recognition, the IMDC attempts a number of retries equal the MAXIMUM NUMBER of RETRIES specified in ECA table.

IRRECOVERABLE ERRORS (Main Status = 1)

WRITE FAULT (bit0 EXTENDED STATUS)

It's occurs for particular error conditions of device.

FIFO OVERRUN/UNDERRUN (bit2 EXTENDED STATUS)

It's set if the system bus is lost for a long period of time.

DATA PART TIMEOUT (bit8 EXTENDED STATUS)

It's set if the IMDC lost the syncronism with the track before the data field.

POSITIONING TIMEOUT (bit13 EXTENDED STATUS)

It's set if the IMDC can't reach the final position of the carriage according with CYLINDER NUMBER of ECA.

BUS ERROR (bit15 EXTENDED STATUS)

System bus error signal.

F.7.3.3 MEDIA ERRORS

After these errors the IMDC attempts a number of retry equal to the MAXIMUM NUMBER of RETRIES specified in ECA table.

IRRECOVERABLE ERRORS (Main Status = 1)

CRC/ECC ERROR ON DATA (bitl EXTENDED STATUS)

This error occurs if a CRC error or ECC is detected on DATA field.

CRC ERROR ON ID (bit4 EXTENDED STATUS)

This error occurs if a CRC error is detected on ID field.

UNCORRECTABLE DATA ERROR (bit 10 exetended status)
This error occurs after a correct command, when there is an ECC uncorrectable error.

F.7.3.4 ECC HANDLING

When a ECC error arises, the IMDC executes n-retry of the operation: if the error message is removed, the IMDC continues the operation normally, otherwise it stops the operation with ECC error message in Extended status.
In this second situation the HOST should try to correct the error, by sending the CORRECT command.
The IMDC will verify the correctness of the ECC and will informs the HOST if there's an uncorrectable or correctable error.
In the first case it should set the bit10 of Extended status.
If there's a correctable error it supplies the relative informations in order to obtain the correction of the DATA field in Main Memory Buffer (24 bit correction mask and offset count)
The HOST will execute the correction in Main Memory buffer.

F.8 FORMAT UTILITY

Format utility is used to initialize all the tracks on the media,Hard or Floppy disk drive,and establish the sector length of the data field.
Each format command works track by track basis.
The user can establish the format of the track in term of data length and number of sectors per track by means of particular field in E.C.A. table and by means of particular field in Format table.
In this way a complete programmability is given to the user.
See SCN68454 eps for further details.
Format procedure will handle also media defect according with safety and performances requirements.
The major statements in order to have the above requirements could be:

- Save Vendor media certification (if possible)
- Save the information between Format procedure
- Check the media with critical pattern
  suggested pattern for MFM encoding method Hex 6DB6DB6B
- Verify ID and DATA fields
- Realloc Vendor Bad Sectors (read in automatic way or introduced in the system by operator)
- Realloc New Bad Sector discovered during verification
- Do not apply retries or error correction in order to have a good certification of bad spots

F.8.1 BAD SECTORS HANDLING

The IMDC provides an automatic handling for bad and alternative sectors.
In fact during normal Read/Write operations the IMDC is capable to recognize a bad sector,to reach the alternative sector and then to operate it.
Will be the format procedure that will prepare the media in order to assign at each defective sector an alternative sector following certain rules.
Here below is a proposal method to handle correctly bad and replacement sectors.
Each bad sector is flagged by means of Bit 7 in head byte in Identifier field.
In a track with a bad sector the format buffer will be prepared as follows:

e.g. BAD SECTOR (sector n.1)

Format Buffer

+ cyl +	cyl +	head +	sect +	
+xxxxxxxx	xxxxxxxx	xxxxxxxx	00000000+	good sector
+xxxxxxxx	xxxxxxxx	1xxxxxxx	00000001+	bad sector
+xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx+	alternative sector
+xxxxxxxx	xxxxxxxx	xxxxxxxx	00000002+	good sector

The IMDC will provide automatically to format the track including the informations related to the bad and alternative sector.For each defective sector the host will prepare a buffer that will contain the informations related to bad sector (8 bytes) repeated 20 times when using a data field 512 bytes long.Of course the length of operation will be compatible with Format Buffer Length.

F.8.2 ERROR RECOVERY PROCEDURE

In order to have a better certification a write of single data field should be necessary.
In this way a critical pattern should be used and during the verification phase all the bad spots could be discovered.
Note that during verification no retries will be attempted except one in order to avoid soft error due to electrical problems.No ECC correction will be attempted because also a sector with a correctable error should be declared defective.
The errors involved in the bad sector declaration wiil be:

1) CRC error on ID field
2) ECC error (correctable type on data field)
3) ECC error (uncorrectable error on data field)
4) Data part time out on data field

F.8.3 STRUCTURE of FORMAT UTILITY

In order to have a correct handling of bad spots will be necessary some special records called Bad Block Map and a complete cilynder to contain all the alternative sectors.
Bad block map will be located in an error free area on the media (e.g. cilynder 0 track 0 sectors 1.2.3) and the cilynder for the alternative sectors could be the inner most cilynder.
Here below is a proposal structure for format utililty:

A) Check if bad block map is present in order to establish if the media it's been already formatted.
   If bad block map present store it and go to step C
   Else
B) Take bad sector from input message
   Compose bad block map
   go to step C
C) Execute Format operation track by track basis formatting bad sector if present according to the bad block map
D) Format innnermost track
E) Start verify including write data field with critical pattern and execute track verification (read with skip)
   If verification runs OK repeat step E until end of volume is reached and then go to step F
   Else (error occurs)
   Search the first alternative free sector in bad block map
   reformat track with bad and alternative sector
   verify the track
   update bad block map and repeat step E
F) Write bad block map on disk
   END

F.8.4 STRUCTURE OF BAD BLOCK MAP

- cyl - cyl - head - sect. -	bad sector n.1
- cyl - cyl - head - sect. -	altern.sector n.1
- cyl - cyl - head - sect. -	bad sector n.2
- cyl - cyl - head - sect. -	altern.sector n.2
- FF FF - FF - FF -	next bad sector
- FF FF - FF - FF -	next free alt. sect.

In this way each sector that compose the bad block map can contain up to 64 bad sectors.

F.8.5 VENDOR INFORMATIONS COMPUTING

Here below is an eplanatin about the method to compute the Bad Sector related to the sector length starting from the informations supplied by vendor.
Vendor supplies a label that contains all the informations in order to compute the bad sector.
The label is composed in the following way:
1)Cilynder number...number of cilynder where to find bad spot
2)Head number.....number of head where to find bad spot
3)BCAI.number of bytes starting from index to reach bad spot
The format utility will take these informations and with the following formula can estabrish the sector to declare bad.

$$Sn = \frac{BCAI-23}{573}$$

This is meaningfull for data length equal to 512 bytes.
It should be noted that:
BCAI<= 22 means no error in any sector
BCAI>9190 means error out of the user sectors.
If the sector length is not 512 byte per sector this formula is change in this mode:

$$Sn = \frac{BCAI - 22}{61+(SLx256)}$$

SL= 1 256byte/sector
SL= 2 512byte/sector
SL= 4 1024byte/sector

$$LIMIT = \{[61+(SLx256)]x\ Nsector/track\}+22$$

F.9 DENSITY MODE

The D.C.S. controller can connect 1MByte or 1.6MByte floppy disk drive.
The functionality is the same for both modes. In order to use single or double density mode there are two registers, SDM and DDM: the host will set SDM for single density mode or DDM for double density mode.
The capacity in single density mode is 16 sector of 256 Byte each or equivalent format type, in double density mode is 16 sector of 512 Byte each or equivalent format.

SECTION G

MEMORY SUB-SYSTEM

PRODUCT DESIGN DESCRIPTION

PREPARED BY: A. QUADRARUOPOLO

REVIEWED BY: C. MANTELLINA

INDEX OF SECTION G

		PAGE
G.1	MEMORY SUB-SYSTEM GENERAL DESCRIPTION	314
G.1.1	MEMORY BLOCK	314
G.1.2	STORAGE BLOCK	316
G.1.3	ADDRESS BLOCK	316
G.1.4	DATA BLOCK	318
G.1.5	REFRESH	319
G.1.6	MEMORY MODULE GENERATION	320
G.1.6.1	WRITE CYCLE FROM VME	320

G.1 GENERAL DESCRIPTION

In SGM2 System the dual-port memory sub-system performs data storage (write mode), retrieval (read mode), and retention (memory refreshing) operations on both the path, IDP and VME bus.

The storage block has a minimum storage capacity of 1 MB and it is expandable by 1, 2 or 4 MB increments up to 12 MB.

Three slots are provided for memory sub-system, with possibility to mix in any combination the boards, complying with the only constraint of fitting them in a contiguous fashion starting from the first slot, when less than three boards are used.

SE1	1MBytes
SE2	2MBytes
SE4	4MBytes

Max capacity of 12 MBytes is achievable by using only 4 MBytes modules (SE4).

Each memory board connects to the SGM2 system by means of:

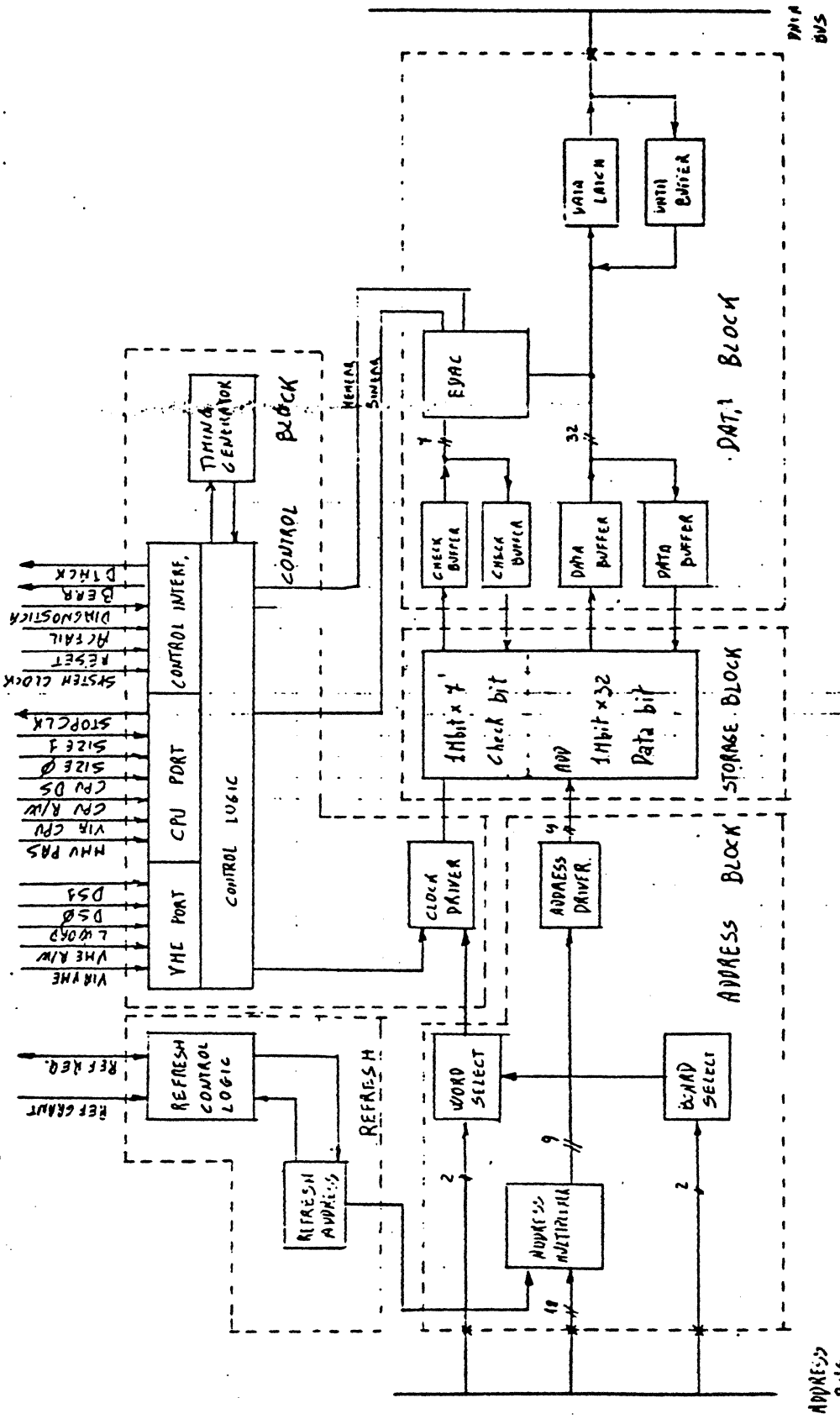
a 48-pin connector, located on the rear of the board, that provides supply voltages and some control signal that encode to the board the total memory capacity of previous boards chained.

three 50-pin connectors, located on the front of the board, to connect the board to CPO by means of a flat-wired memory bus.

G.1.1 MEMORY BLOCK

Each memory PWA, as depicted in fig. G.1, consists of the following functional blocks:

Memory block
Address block
Data block
Command and control block
Refresh block


DATA BUS
DATA BLOCK
DATA LATCH
DATA BUFFER
EDAC
32
7
CHECK BUFFER
CHECK BUFFER
DATA BUFFER
DATA BUFFER
STORAGE BLOCK
1Mbit x 7' Check bit
1Mbit x 32 Data bit
ADD
ADDRESS BLOCK
CLOCK DRIVER
ADDRESS DRIVER
41
CONTROL BLOCK
TIMING GENERATOR
MEMEN
SWERR
CONTROL INTERF.
CPU PORT
CONTROL LOGIC
VHE PORT
SYSTEM CLOCK
RESET
ALTAIL
DIAGNOSTICA
BERR
DTACK
STOPCLK
SIZE 1
SIZE 0
CPU DS
CPU R/W
VIA CPU
MMU PAS
VIA VME
VME R/W
LWORD
DS0
DS1
REFRESH CONTROL LOGIC
REFRESH
REFRESH ADDRESS
WORD SELECT
BOARD SELECT
ADDRESS MULTIPLEX
9
2
2
10
ADDRESS BUS

G.1.2 STORAGE BLOCK

Memory storage consists of a 156-device array that is arranged o the module in four row of 39 (262.144 x 1 bit) dynamic RAM's.

This RAM array will accept 1.048.576 32-bit long word and seve error correction bit (39 bits per row total).

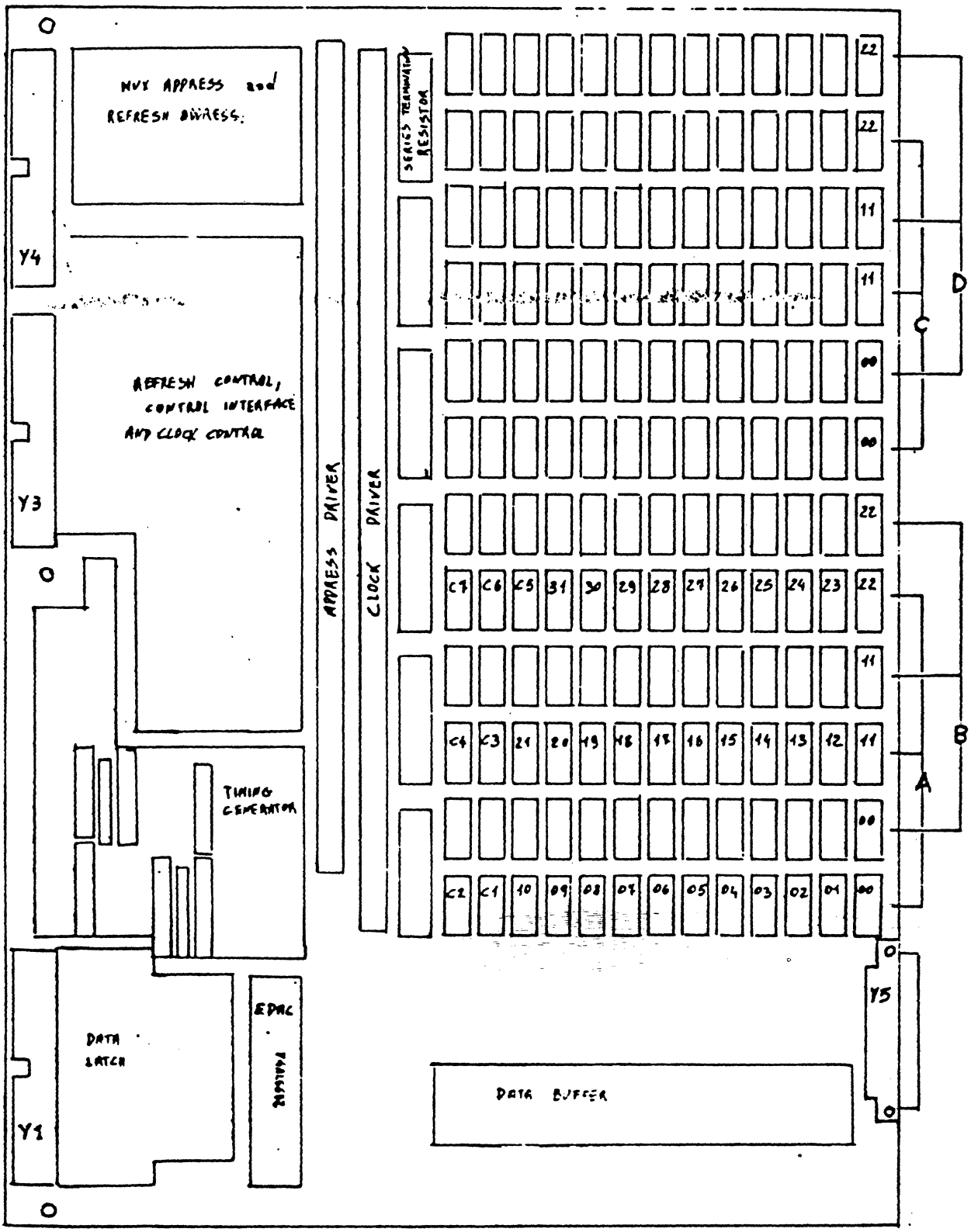
If the PWA is selected, the address MADD20 and MADD21 are used t enable the correct row.

The Rows are physically arranged on the PWA as depicted in fig. G.2.

G.1.3 ADDRESS BLOCK

The address lines are connected to the module through Y04 connector. Address lines MADD20-MADD25 and PADD28-PADD31 are routed to selection logic, address lines A02-A19 are routed to the address multiplexer and lower address MADD00-MADD01 for IDP and VMAD01 for VME bus are routed to the byte enable.

Address lines MADD02-MADD19 determine memory location access in the 256K RAM.


MUX ADDRESS and REFRESH ADDRESS
REFRESH CONTROL, CONTROL INTERFACE AND CLOCK CONTROL
TIMING GENERATOR
ADDRESS DRIVER
CLOCK DRIVER
SERIES TERMINATION RESISTOR
DATA LATCH
EDAC
DATA BUFFER
Y1
Y3
Y4
Y5
D
C
B
A
22
11
00
C7 C6 C5 31 30 29 28 27 26 25 24 23
C4 C3 21 20 19 18 17 16 15 14 13 12
C2 C1 10 09 08 07 06 05 04 03 02 01

G.1.4 DATA BLOCK

The bidirectional Data lines are connected to the module through Y01 connectors and data latches.

The data latches are activated/inactivated by internal and external signals.

Incoming data is transferred to the PWA whenever the BEDBY0-BEDBY3 signals are activated, (Bus Enable Data Byte 0 - 3). Data to be read from the PWA is gated by the internally generated signals LEREAD and OEREAD.

Whenever data is written to the memory module, seven check bit are generated internally in the EDAC.

These seven check bits are stored in memory, along with the original 32-bit data word.

During a memory read cycle, the 39-bit long word from memory is processed by the EDAC's to determine if errors have occurred.

Single bit error in the long word is corrected and generates STOPCK signal.

Double bit errors will cause the assertion of both STOPCK and BERR, which are the interrupt indication for the CPU.

The internal bidirectional data bus is separated from memory bus by means of data buffers, to allow Read-Modify-Write cycles.

G.1.5 REFRESH

Memory refresh is performed continuously to ensure that informations are not loss.

The system clock frequence 16.6 MHz is divided by 2^{12} and so the memory sub-system gives a REFREQ signal every 245.76 uS (control circuits on CPO provide arbitration between refresh and memory cycles). This signal will prevent any new memory access from the IDP or VME and will start to refresh in 16 consecutive refresh cycles (300 ns each), as soon as memory become available: thus the CPU is prevented from accessing the memory for 4.8 uS.

G.1.6 MEMORY MODULE GENERATION

The SEi operates in the following modes :

a. Access from VME

- Long-word read and write (aligned with a long-word boundary), word read and write (aligned with a word boundary), byte read and write.

b. Access from IDP

- Long-word, word and byte read and write, without contraints of alignement.

c. Refresh cycles

G.1.6.1 WRITE CYCLE FROM VME

The write cycle sequence, is performed by the memory module in order to store data into memory. A write cycle is initiated whenever WRITE, followed by START and VME or IDP signals, become active on the control lines.

When this occurs, the address currently on address lines MADD02-09 is latched into RAM with RAS.

The DS0 and DS1 signals from the VME bus are then used by the module to select the lower byte, upper byte, or both (word mode only); for long word cycle LWORD signal will activated.

The SIZE0, SIZ1, MADD00 and MADD01 signals from CPU are then used by the module to select the byte or word long word.

Data on the multiplexed bus is then transferred to the module. Upon receipt of the data, the module activates DTACK to indicate that the write cycle has been completed.

F.8.4 STRUCTURE OF BAD BLOCK MAP

cyl	cyl	head	sect.	bad sector n.1
cyl	cyl	head	sect.	altern.sector n.1
cyl	cyl	head	sect.	bad sector n.2
cyl	cyl	head	sect.	altern.sector n.2
FF	FF	FF	FF	next bad sector
FF	FF	FF	FF	next free alt. sect.

In this way each sector that compose the bad block map can contain up to 64 bad sectors.

F.8.5 VENDOR INFORMATIONS COMPUTING

Here below is an eplanatin about the method to compute the Bad Sector related to the sector length starting from the informations supplied by vendor.
Vendor supplies a label that contains all the informations in order to compute the bad sector.
The label is composed in the following way:
1)Cilynder number...number of cilynder where to find bad spot
2)Head number.....number of head where to find bad spot
3)BCAI.number of bytes starting from index to reach bad spot
The format utility will take these informations and with the following formula can estabrish the sector to declare bad.

$$Sn = \frac{BCAI-23}{573}$$

This is meaningfull for data length equal to 512 bytes.
It should be noted that:
BCAI<= 22 means no error in any sector
BCAI>9190 means error out of the user sectors.
If the sector length is not 512 byte per sector this formula is change in this mode:

$$Sn = \frac{BCAI - 22}{61+(SLx256)}$$

SL= 1 256byte/sector
SL= 2 512byte/sector
SL= 4 1024byte/sector

$$LIMIT = \{[61+(SLx256)]x\ Nsector/track\}+22$$

F.9 DENSITY MODE

The D.C.S. controller can connect 1MByte or 1.6MByte floppy disk drive.
The functionality is the same for both modes. In order to use single or double density mode there are two registers, SDM and DDM: the host will set SDM for single density mode or DDM for double density mode.
The capacity in single density mode is 16 sector of 256 Byte each or equivalent format type, in double density mode is 16 sector of 512 Byte each or equivalent format.

SECTION G

MEMORY SUB-SYSTEM

PRODUCT DESIGN DESCRIPTION

PREPARED BY: A. QUADRARUOPOLO

REVIEWED BY: C. MANTELLINA

INDEX OF SECTION G

		PAGE
G.1	MEMORY SUB-SYSTEM GENERAL DESCRIPTION	314
G.1.1	MEMORY BLOCK	314
G.1.2	STORAGE BLOCK	316
G.1.3	ADDRESS BLOCK	316
G.1.4	DATA BLOCK	318
G.1.5	REFRESH	319
G.1.6	MEMORY MODULE GENERATION	320
G.1.6.1	WRITE CYCLE FROM VME	320

G.1 GENERAL DESCRIPTION

In SGM2 System the dual-port memory sub-system performs data storage (write mode), retrieval (read mode), and retention (memory refreshing) operations on both the path, IDP and VME bus.

The storage block has a minimum storage capacity of 1 MB and it is expandable by 1, 2 or 4 MB increments up to 12 MB.

Three slots are provided for memory sub-system, with possibility to mix in any combination the boards, complying with the only constraint of fitting them in a contiguous fashion starting from the first slot, when less than three boards are used.

SE1	1MBytes
SE2	2MBytes
SE4	4MBytes

Max capacity of 12 MBytes is achievable by using only 4 MBytes modules (SE4).

Each memory board connects to the SGM2 system by means of:

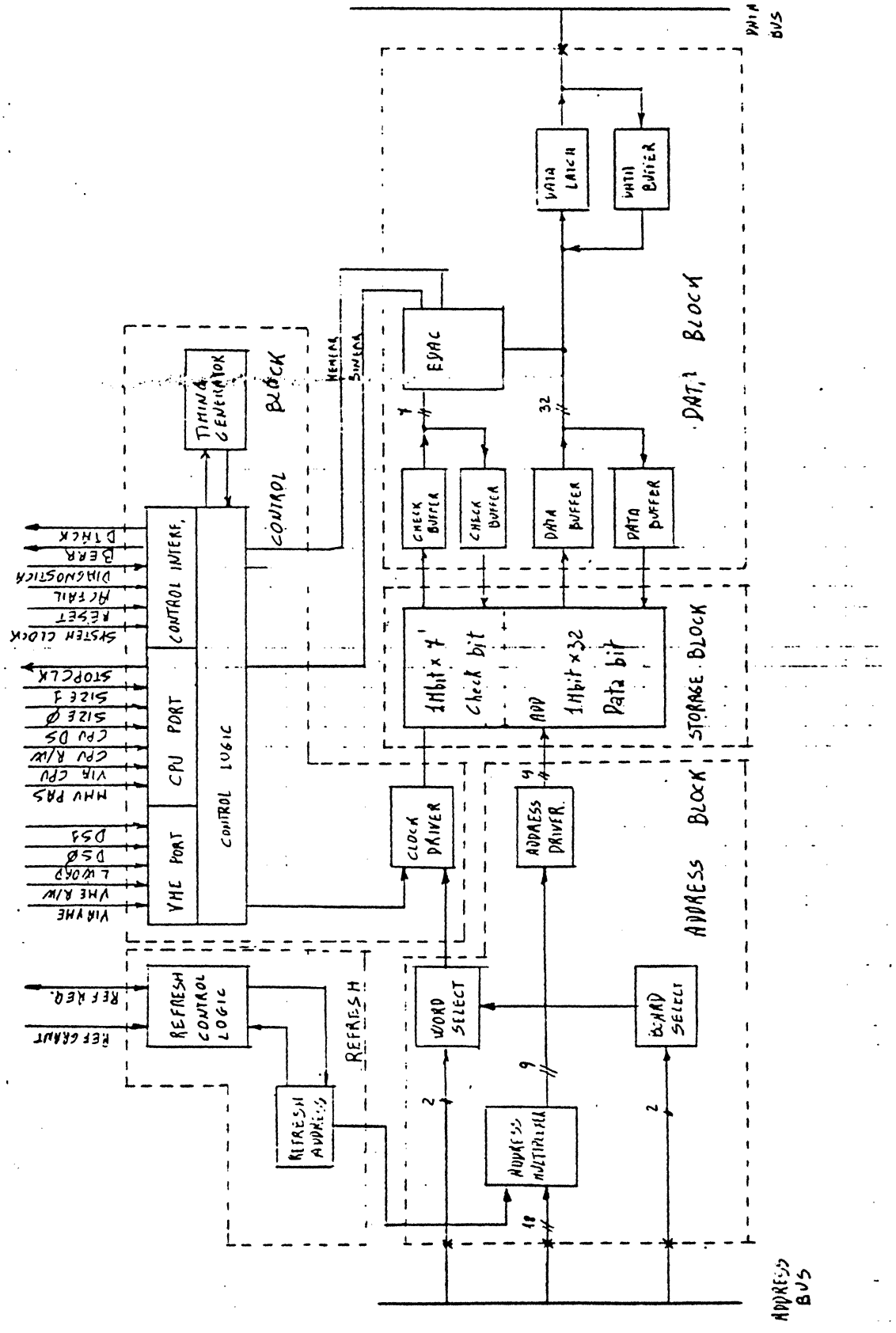
a 48-pin connector, located on the rear of the board, that provides supply voltages and some control signal that encode to the board the total memory capacity of previous boards chained.

three 50-pin connectors, located on the front of the board, to connect the board to CP0 by means of a flat-wired memory bus.

G.1.1 MEMORY BLOCK

Each memory PWA, as depicted in fig. G.1, consists of the following functional blocks:

Memory block
Address block
Data block
Command and control block
Refresh block


DATA BUS
DATA LATCH
DATA BUFFER
DATA BLOCK
EDAC
TIMING GENERATOR
CONTROL BLOCK
CHECK BUFFER
CHECK BUFFER
DATA BUFFER
DATA BUFFER
1Mbit x 7 Check bit
1Mbit x 32 Data bit
STORAGE BLOCK
CONTROL INTERF.
CPU PORT
CONTROL LOGIC
VME PORT
CLOCK DRIVER
ADDRESS DRIVER
ADDRESS BLOCK
REFRESH CONTROL LOGIC
WORD SELECT
BOARD SELECT
REFRESH ADDRESS
REFRESH
ADDRESS MULTIPLEXER
ADDRESS BUS
SYSTEM CLOCK
RESET
MF FAIL
DIAGNOSTICK
BERR
DTACK
STOPCLK
SIZE 1
SIZE 0
CPU DS
CPU R/W
VIA CPU
MMU PAS
VIA VME
VME R/W
LWORD
DS0
DS1
REF REQ.
REF GRANT

G.1.2 STORAGE BLOCK

Memory storage consists of a 156-device array that is arranged o the module in four row of 39 (262.144 x 1 bit) dynamic RAM's.

This RAM array will accept 1.048.576 32-bit long word and seve error correction bit (39 bits per row total).

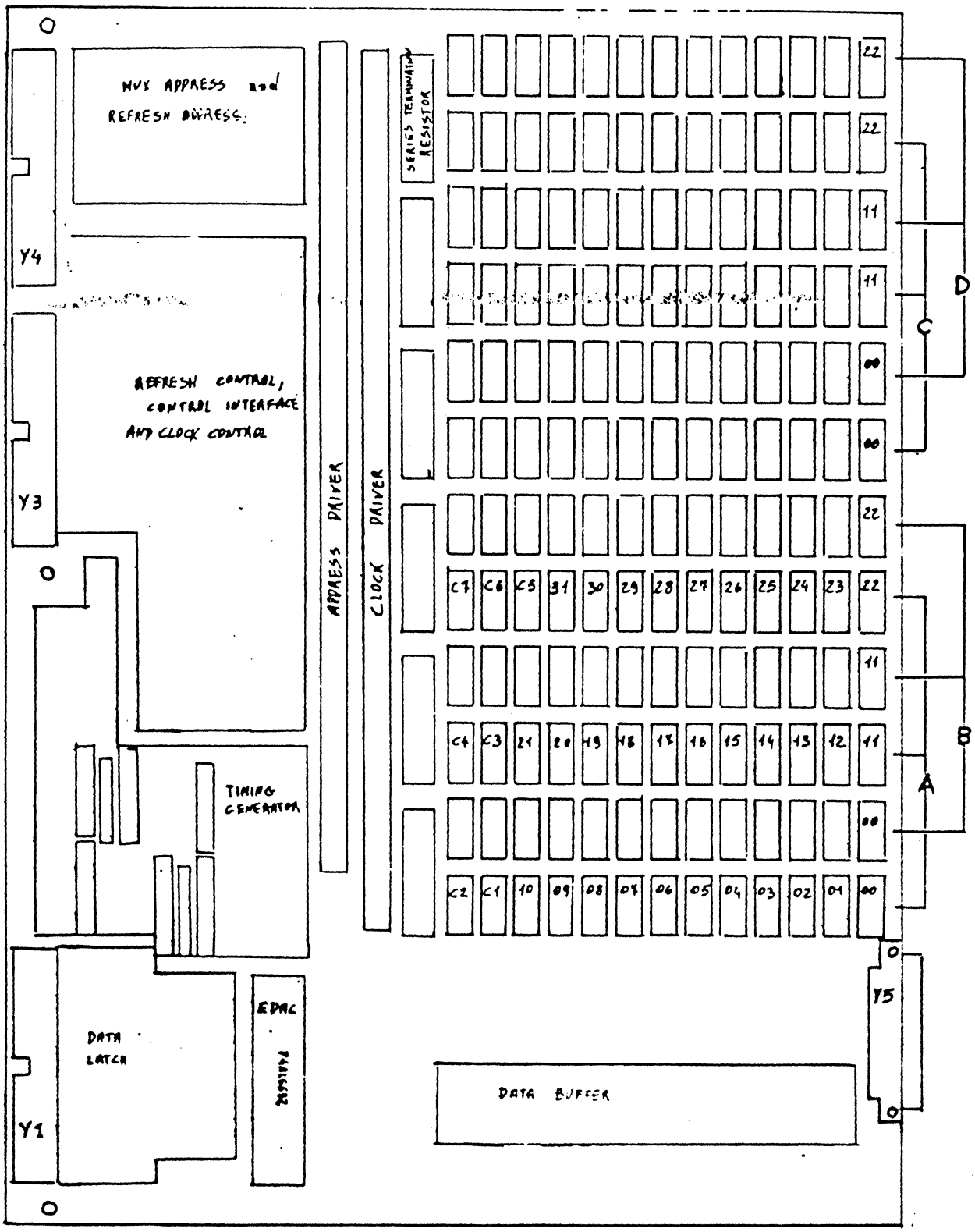
If the PWA is selected, the address MADD20 and MADD21 are used t enable the correct row.

The Rows are physically arranged on the PWA as depicted in fig. G.2.

G.1.3 ADDRESS BLOCK

The address lines are connected to the module through Y04 connector. Address lines MADD20-MADD25 and PADD28-PADD31 are routed to selection logic, address lines A02-A19 are routed to the address multiplexer and lower address MADD00-MADD01 for IDP and VMAD01 for VME bus are routed to the byte enable.

Address lines MADD02-MADD19 determine memory location access in the 256K RAM.


MUX ADDRESS and REFRESH ADDRESS
REFRESH CONTROL, CONTROL INTERFACE AND CLOCK CONTROL
TIMING GENERATOR
ADDRESS DRIVER
CLOCK DRIVER
SERIES TERMINATION RESISTOR
Y4
Y3
Y1
Y5
DATA LATCH
EDAC
DATA BUFFER
A
B
C
D

G.1.4 DATA BLOCK

The bidirectional Data lines are connected to the module through Y01 connectors and data latches.

The data latches are activated/inactivated by internal and external signals.

Incoming data is transferred to the PWA whenever the BEDBY0-BEDBY3 signals are activated, (Bus Enable Data Byte 0 - 3). Data to be read from the PWA is gated by the internally generated signals LEREAD and OEREAD.

Whenever data is written to the memory module, seven check bit are generated internally in the EDAC.

These seven check bits are stored in memory, along with the original 32-bit data word.

During a memory read cycle, the 39-bit long word from memory is processed by the EDAC's to determine if errors have occurred.

Single bit error in the long word is corrected and generates STOPCK signal.

Double bit errors will cause the assertion of both STOPCK and BERR, which are the interrupt indication for the CPU.

The internal bidirectional data bus is separated from memory bus by means of data buffers, to allow Read-Modify-Write cycles.

G.1.5 REFRESH

Memory refresh is performed continuously to ensure that informations are not loss.

The system clock frequence 16.6 MHz is divided by 2^{12} and so the memory sub-system gives a REFREQ signal every 245.76 uS (control circuits on CPO provide arbitration between refresh and memory cycles). This signal will prevent any new memory access from the IDP or VME and will start to refresh in 16 consecutive refresh cycles (300 ns each), as soon as memory become available: thus the CPU is prevented from accessing the memory for 4.8 uS.

G.1.6 MEMORY MODULE GENERATION

The SEi operates in the following modes :

a. Access from VME

- Long-word read and write (aligned with a long-word boundary), word read and write (aligned with a word boundary), byte read and write.

b. Access from IDP

- Long-word, word and byte read and write, without contraints of alignement.

c. Refresh cycles

G.1.6.1 WRITE CYCLE FROM VME

The write cycle sequence, is performed by the memory module in order to store data into memory. A write cycle is initiated whenever WRITE, followed by START and VME or IDP signals, become active on the control lines.

When this occurs, the address currently on address lines MADD02-09 is latched into RAM with RAS.

The DS0 and DS1 signals from the VME bus are then used by the module to select the lower byte, upper byte, or both (word mode only); for long word cycle LWORD signal will activated.

The SIZE0, SIZ1, MADD00 and MADD01 signals from CPU are then used by the module to select the byte or word long word.

Data on the multiplexed bus is then transferred to the module. Upon receipt of the data, the module activates DTACK to indicate that the write cycle has been completed.

SECTION H

CUSTOM MEMORY MANAGEMENT UNIT (CM1)

PRODUCT DESIGN DESCRIPTION

PREPARED BY: C. BAGNOLI

REVIEWED BY: C. MUNERO

INDEX OF SECTION H

		PAGE
H.1	OVERVIEW	324
H.2	CM1 GENERAL DESCRIPTION	325
H.2.1	PAGING SIZE AND GLOBAL STUCTURE	325
H.2.2	CONTEXTS	328
H.2.3	LOGICAL ADDRESS SPACE	328
H.2.3.1	MAPPED SPACE	330
H.2.3.2	UNMAPPED SPACE	331
H.2.3.3	PHYSICAL MEMORY ACCESS	332
H.2.4	SEGMENT TABLE	332
H.2.5	VIRTUAL ADDRESS	333
H.2.6	PAGE TABLE	333
H.2.7	PAGE STATUS TABLE	334
H.2.8	TRANSLATION LOOK-ASIDE BUFFER	336
H.2.9	REGISTER SPACE	337
H.2.9.1	CONTEXT REGISTERS	338
H.2.9.2	ERROR REGISTER	338
H.2.9.3	CONTROL/STATUS REGISTER	340

LIST OF ILLUSTRATIONS

		PAGE
FIG. H.1	ADDRESS TRANSLATION SCHEME	326
FIG. H.2	CM1 BLOCK DIAGRAM	327
FIG. H.3	LOGICAL ADDRESS	328
FIG. H.4	LOGICAL SPACE	329
FIG. H.5	TRANSLATED PHYSICAL ADDRESS	331
FIG. H.6	PAGE STATUS TABLE ADDRESS	335
FIG. H.7	MMU REGISTERS FORMAT	339

LIST OF TABLES

		PAGE
TAB. H.1	LOGICAL SPACE MAP	330
TAB. H.2	SEGMENT PROTECTION CODES	333
TAB H.3	DAEMON AND ACCESSED BITS CODING	335
TAB. H.4	PAGE STATUS TABLE UPDATING	336
TAB. H.5	REGISTER SPACE MAP	337

H.1 OVERVIEW

This document provides a major design description of the CUST MEMORY MANAGEMENT UNIT (MMU) board of the SGM2 computer, called CM1 (P GF4CM1). The board was designed according to the Demand Paging M E.P.S. (Document N. A78138743).

The CM1 Demand Paging Memory Management Unit is an enhancement the SUN (Stanford University Networking) MMU approach to support Virtu Memory with Demand Paging.

This two level, Demand Paging MMU, is designed to suppo multi-tasking, multi-user, or multi-processing operating systems SGM2. It provides segment protection, page sharing, physical memo access, statistics and fast context switching. Its main function is t translate a page logical address into a page physical addres controlling memory access privilege levels.

The versatility of the paged MMU allows mapping of large, contiguou sections of memory.

Demand paging is powerfully supported by a table that mantains pag statistic informations at a physical level.

The MMU allows up to 16 Megabytes of mappable process space and th definition of address maps makes the design upgradable up to 3 Megabytes.

H.2 CM1 GENERAL DESCRIPTION

H.2.1 PAGING SIZE AND GLOBAL STRUCTURE

A major block diagram of the address translation mechanism and a detailed block diagram of the CM1 board are respectively depicted in figs. H.1 and H.2.

The Logical Address Space is divided in logical pages. Their size is 4 kBytes.

Physical memory is also managed and allocated in 4 kBytes blocks or physical pages.

The CM1 demand paging, two level MMU, is based mainly on:

two Context Registers:

- Supervisory Context Register
- User Context Register

two translation maps

- Segment Map
- Page Map

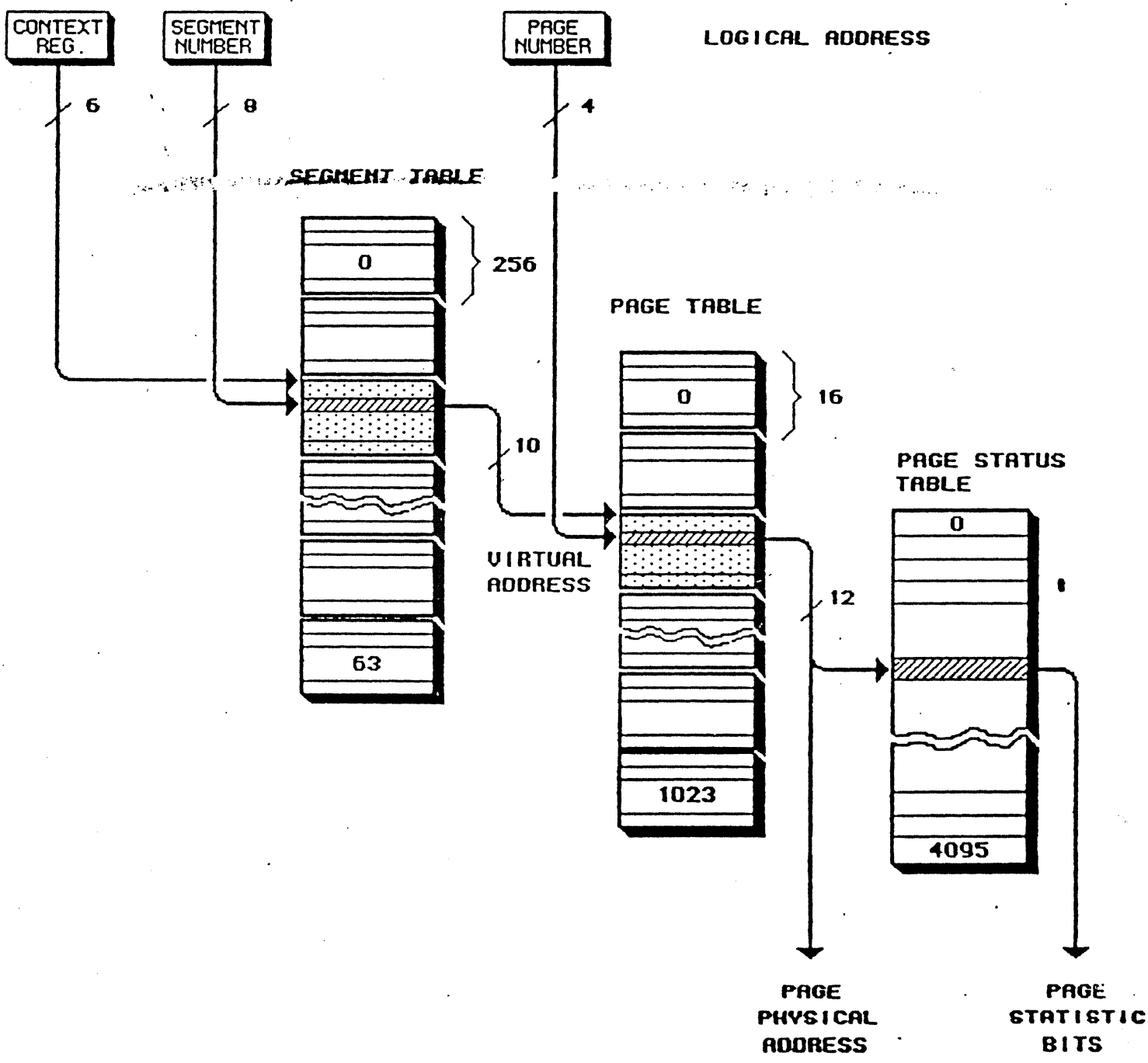
and a page statistic table

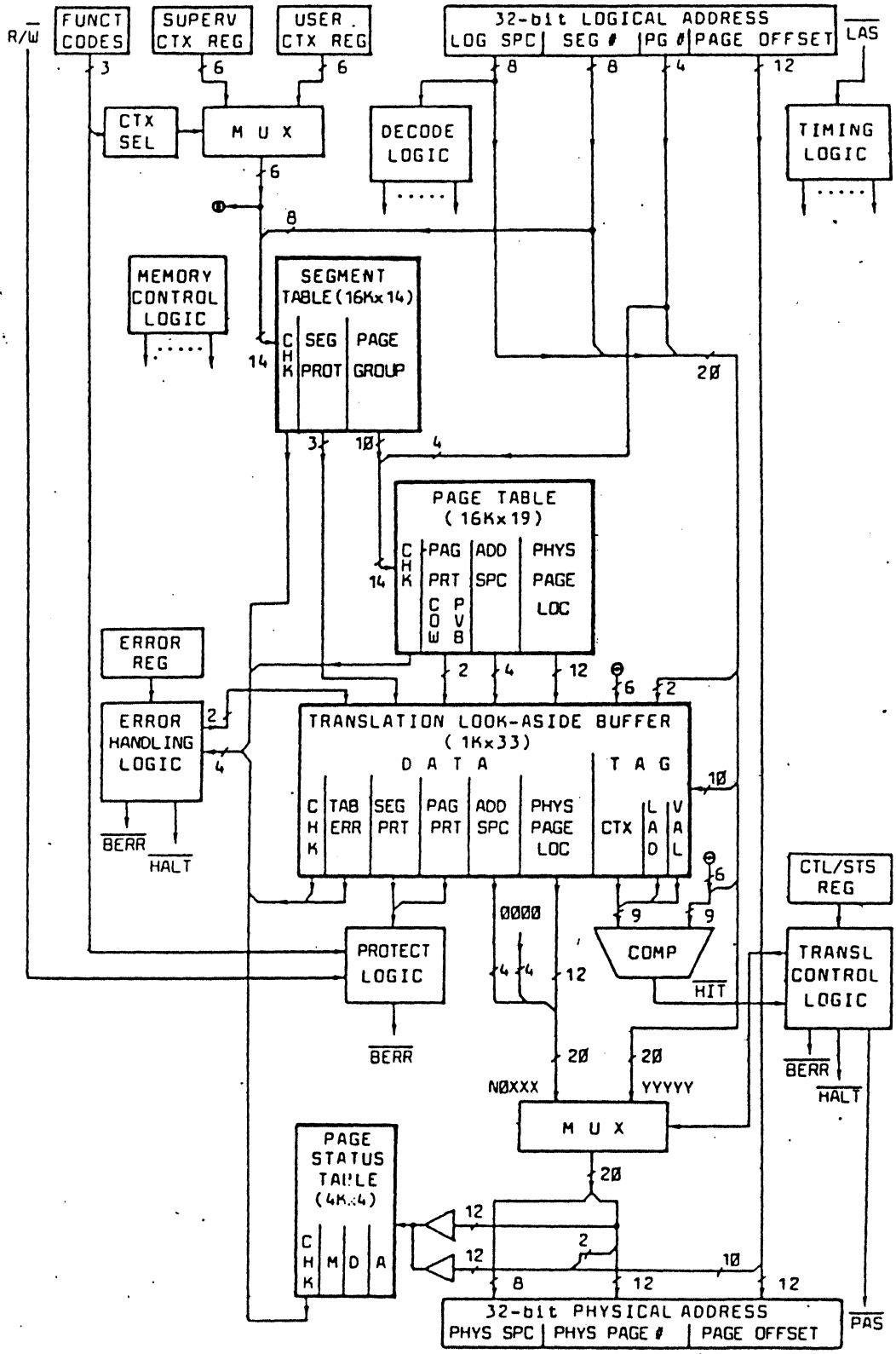
- Page Status Table

The Segment Map translates a logical address into a virtual address.

The Page Map translates a virtual address into a physical address.

The Page Status Table contains statistical informations about physical page accesses.


CONTEXT REG.
SEGMENT NUMBER
PAGE NUMBER
LOGICAL ADDRESS
6
8
4
SEGMENT TABLE
0
256
63
10
PAGE TABLE
0
16
1023
VIRTUAL ADDRESS
12
PAGE STATUS TABLE
0
4095
PAGE PHYSICAL ADDRESS
PAGE STATISTIC BITS


R/W
FUNCT CODES
SUPERV CTX REG
USER CTX REG
32-bit LOGICAL ADDRESS
LOG SPC
SEG #
PG #
PAGE OFFSET
LAS
CTX SEL
M U X
DECODE LOGIC
TIMING LOGIC
MEMORY CONTROL LOGIC
SEGMENT TABLE (16Kx14)
CHK
SEG PROT
PAGE GROUP
PAGE TABLE (16Kx19)
CHK
PAG PRT
ADD SPC
PHYS PAGE LOC
COW
PVB
ERROR REG
ERROR HANDLING LOGIC
BERR
HALT
TRANSLATION LOOK-ASIDE BUFFER (1Kx33)
DATA
TAG
CHK
TAB ERR
SEG PRT
PAG PRT
ADD SPC
PHYS PAGE LOC
CTX
LAD
VAL
CTL/STS REG
0000
COMP
HIT
PROTECT LOGIC
BERR
TRANSL CONTROL LOGIC
BERR
HALT
N0XXX
YYYYY
M U X
PAGE STATUS TABLE (4Kx4)
CHK
M
D
A
32-bit PHYSICAL ADDRESS
PHYS SPC
PHYS PAGE #
PAGE OFFSET
PAS

H.2.2 CONTEXTS

The virtual address space generated by the 68020 CPU is enhance with a 6 bit context number stored in one of two CM1 registers respectively reserved for Supervisor and User accesses (see H.2.9.1 Context Registers). In this way 64 separate processes can b simultaneously mapped on different context numbers, each having maximum virtual space of 16 MBytes.

Context Register are handled by the supervisor and can switc between 64 separate portions of the segment map. The processor ca switch between processes quickly without having to reload the segmen and page maps.

The selection of context registers depends on the Logical Space code and processor status.

When the Logical Address generated by the 68020 microprocessor selects a Mapped Space (i.e. the Logical Space is 00-01 hex, see H.2.3.1 Mapped Space), the Supervisor or User Context registers are selected according to the processor status (68020's "FC2" function code).

If the System Space selects the Segment Table (i.e. the Logical Space is 0A-0B hex, see H.2.4 Segment Table), the User Context Register is selected.

In all other cycles the Supervisor Context Register is selected.

H.2.3 LOGICAL ADDRESS SPACE

The logical address generated by the 68020 CPU can be considered divided into 4 parts (see Fig. H.3).

- Logical space
- Segment number
- Page number
- Offset

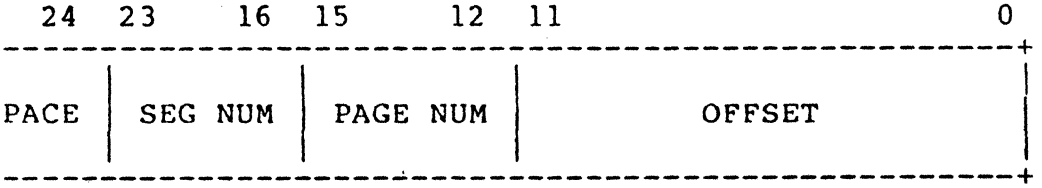


FIG. H.3: LOGICAL ADDRESS

The logical space is 8 bit wide, divided into 2 parts (see Fig H.4).

- Destination Space (bit 31-28)
- System Space (bit 27-24)

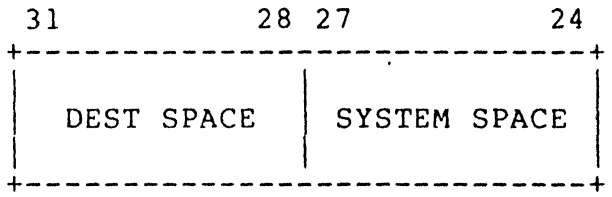


FIG. H.4: LOGICAL SPACE

The system is divided into 16 parts or spaces, selected by the Destination Space field. This field generally coincides with the "Name or Number" of the destination Processor and is used from the CM1 to access memory of external processors in a multi-IDP system (see H.2.3.1, Mapped Space).

The internal destination space, for all processors, is "zero" (hex code "0X").

"Internal" means to select a local unmapped space or, in mapped space, the Local MMU that can select his own or another destination space (see H.2.3.1, Mapped Space). When the Destination Space is internal, the System Space field selects the internal resource to be accessed. Table H.1 shows the Logical Space coding for SGM2 processors using the CM1 MMU board.

For processors not using this MMU the coding of System Space is machine depending.

HEX CODE	DEST. SPACE 31 30 29 28	SYST. SPACE 27 26 25 24	SYSTEM SPACE DENOMINATION	PERMISSION
00-01	0 0 0 0	0 0 0 X	MAPPED SPACE	SPV/U
02-03	0 0 0 0	0 0 1 X	RFU	SPV
04-07	0 0 0 0	0 1 X X	RFU	SPV
08-09	0 0 0 0	1 0 0 X	PAGE TABLE	SPV
0A-0B	0 0 0 0	1 0 1 X	SEGMENT TABLE	SPV
0C	0 0 0 0	1 1 0 0	PAGE STATUS TABLE	SPV
0D	0 0 0 0	1 1 0 1	REGISTER SPACE	SPV
0E	0 0 0 0	1 1 1 0	RFU (MACHINE DEP.)	SPV
0F	0 0 0 0	1 1 1 1	CPO INT RESOURCES	SPV
10-FF	- - - -	- - - -	EXTERNAL UNMAPPED SPACE	SPV

NOTE

X - Don't care
SPV - Supervisor mode
U - User mode

TABLE H.1: LOGICAL SPACE MAP

H.2.3.1 MAPPED SPACE

The Mapped Space is the space addressed by the Physical Address through the segment and page maps. It is the only logical address space accessable by the User. From this logical address window, internal and external memory can be accessed, in a mono or multi-IDP System.

A logical address generated by the 68020 CPU gets translated only if it belongs to the Mapped Space and if the MMU translation is enabled (see H.2.8.3, Control/Status Register).

The Physical Address generated with a translation (i.e. from Mapped Space access) is divided into four parts (see Fig. H.5).

- Destination Space
- 0000
- Page Address
- Offset

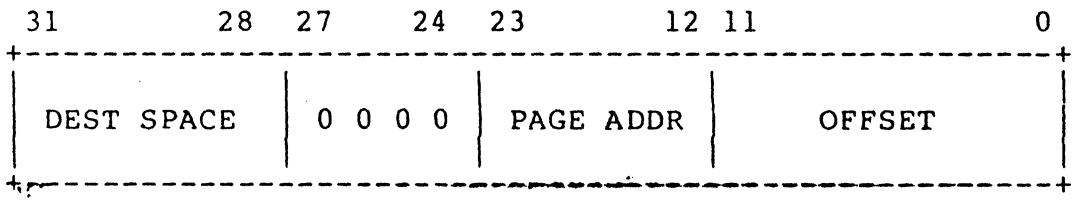


FIG. H.5: TRANSLATED PHYSICAL ADDRESS

H.2.3.2 UNMAPPED SPACE

Any Logical Space different from the Mapped Space is an Unmapped Space and the Physical address generated by the CM1 equals the Logical Address issued by the CPU. Access to Unmapped Space is only allowed to the Supervisor, all users cycle not involving mapped memory being trapped.

Two kinds of Bus Error exception can be generated in an Unmapped Space access:

- User Unmapped access (an unmapped space address was issued in user mode);
- Cycle Not Allowed (the function codes configuration issued by the 68020 CPU is not allowed).

For what concerns the second exception, the only function codes configurations allowed by the CM1 are the following:

FC2	FC1	FC0	
0	0	0	User Data
0	1	0	User Program
1	0	1	Supervisor Data
1	1	1	Supervisor Program
1	1	1	Cpu Space

Configuration of function codes different from these can be generated by the 68020 "MOVES" privileged instruction and are trapped from the CM1 board to avoid faulty operations. The check of the correctness of the function codes is performed also in Mapped Space accesses.

H.2.3.3 PHYSICAL MEMORY ACCESS

Memory can be accessed in physical mode only by the Supervisor.

Internal memory can be accessed at any time disabling the MMU translation (see H.2.8.3, Control/status Register).

External processor memory in a multi-IDP system can always be accessed in physical mode by the Supervisor issuing the proper physical address (in the form N0XXXXXX, where N, the Physical Destination Space, must be equal to the number of the requested processor).

H.2.4 SEGMENT TABLE

The diagram of a segment map entry is shown in Fig. H.7. The segment map is a table comprised of 16K entries. The map is indexed by the 6-bit Context Register and the eight bits of the Segment Number of the Logical Address (A23 to A16). Thus the segment map is divided into 64 contexts with each context having 256 segments, each segment having 16 pages, and each page having 4096 bytes. Each segment contains the ten high order Virtual Address bits and a three bits segment protection code defined in Table H.2. Note that an "r" indicates read allowed, "w" means write allowed, "x" means execute allowed and "-" indicates no access.

In addition the segment table contains an odd parity check bit to check the content of the table. This bit can be read but not written. In a table write operation this bit must be zero "MBZ". The generation of this bit is made directly via hardware.

The Protection Logic inhibis any not permitted memory access by setting the Bus Error signal. The protection is enforced as a function of the three segment protection bits, the two page protection bits (se H.2.6, Page Table), read or write operation and the processor status (68020's "FC0-FC2" function code). The Protection Logic also controls the correctness of the processor status field issued by the CPU (see H.2.3.2, Unmapped Space).

HEX CODE	BIT 14	BIT 13	BIT 13		DESCRIPTION
0	0	0	0	K---U---	No access
1	0	0	1	Kr-xU---	Supervisor Read, Execute
2	0	1	0	KrwxU---	Supervisor read, Write, Execute
3	0	1	1	KrwxUr-x	Supervisor Read, Write, Execute User Read, Execute
4	1	0	0	Krwxurwx	Supervisor and User Full access
5	1	0	1	K---U---	No access - RFU
6	1	1	0	K---U---	No access - RFU
7	1	1	1	K---U---	No access - RFU

NOTE RFU: Reserved for Future Use

TABLE H.2: SEGMENT PROTECTION CODES

Individual protection codes apply to each of segments of the logical address space. The pages of a segment can be shared with other segments, possibly in different contexts, or be kept private, depending on the protection and virtual address bits in the segment entry.

The Segment Table, in Supervisor mode, can be read or written (for further informations about Segment Table write accesses, see H.2.8, Translation Look-Aside Buffer).

H.2.5 VIRTUAL ADDRESS

The Virtual Address field of the segment map entries points to a block of 16 consecutive pages in the page map (called page groups or page frames). A Logical Segment can be as small as one page (4096 bytes) if undesidered pages are invalidated in the page map, or as large as 64 Kbytes (16 pages). If consecutive segment map entries are linked toghether a Logical Segment can be as large as 4096 pages (16 Mbytes), that is the full context logical size.

H.2.6 PAGE TABLE

The contents of a page map entry are depicted in Fig. 7. The page map is a table comprised of 16K entries. The map is divided into 1024 page groups, with each group having 16 pages, and each page having 4096 bytes. The map translates the page Virtual Address formed from ten bits of the segment map entry (Page Group) and the four bits of the Page Number of the Logical Address (A15-A12) into a twelve-bit Page Physical Address.

Each map entry also contains a four-bit Address Space selection field which indicates the Destination Space (see Destination Space) and two page protection bits:

- Copy On Write
- Page Valid Bit

The Copy on Write bit indicates that the page is a "shared page" with other processes. When this bit is set and a write operation tries to modify the contents of this page, the operation will be aborted with a Bus Error exception.

The Page Valid Bit, if not asserted, indicates a non-existent page or simply that the page access is not permitted. When this bit is set and the page is accessed, a Bus Error exception is generated.

In addition the page table, as the segment table, contains an odd parity check. For the specification, see the segment table description.

The Page Table, in supervisor mode, can be read or written (for further informations about Page table write accesses, see H.2.8, Translation Look-Aside Buffer).

The Destination Space field (see H.2.3) is used to select internal (i.e. local) or external processor memory.

At the present SGM2 supports two CP0 processor (say CP01 and CP02, with processor number 1 and 2), so this field must assume one of the following values:

0 for CP01 and CP02 to access own local memory

1 for CP02 to access local memory of CP01

2 for CP01 to access local memory of CP02

The bit field 24-27 is enforced to hex code "0" (Physical Memory Base address) to select memories of the local or the external processor.

The Page Address (bits 23-12) is the physical page address, and can address 4096 physical pages.

The Offset is a twelve-bit field (the same of the Logical Address), used to address the byte location inside the page.

H.2.7 PAGE STATUS TABLE

The Page Status Table contains the statistic bits used by the memory management algorithms and indicates whether the physical page has been modified and recently or not recently accessed. These bits are modified appropriately with each "internal" memory access, thus when the Destination Space field of the Physical Address is hex code "0".

The contents of a page map status entry are depicted in Fig. H.7. This table is comprised of 4096 entries and is addressed by the

The statistic bits are:

- Modified
- Daemon
- Accessed

The Modified bit is set at any write operation to indicate that the page content has been modified;

The Accessed and Daemon bits are set at each page access. These two bits indicate if the page has been recently accessed, not recently accessed, or not accessed (not accessed for a long time).

The coding is depicted in table H.3.

Daemon	Accessed	
0	0	Not accessed
0	1	Not recently accessed
1	0	(invalid code)
1	1	Recently accessed

TABLE H.3: DAEMON AND ACCESSED BITS CODING

In addition the page status table, as the segment table, contains an odd parity check. For the specification, see the segment table description. For information about Page Status check errors, see H.2.9.2, Error Register.

The Page Status Table, in supervisor mode, can be read and written. It can be directly addressed by the twelve-bit logical address A13-A2 and is therefore allocated at a long word boundary (fig. H.6).

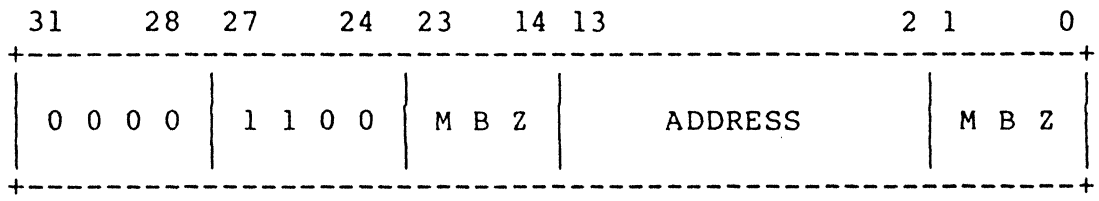


FIG. H.6: PAGE STATUS TABLE ADDRESS

Periodically (depending on the memory management algorithms), the page status table is updated. The following table shows the page status table updating algorithm:

Current Status			Update Status		
M	D	A	M	D	A
X	0	0	X	0	0
X	0	1	X	0	0
X	1	1	X	0	1

NOTE

X Not to be updated
M Modified bit
D Daemon bit
A Accessed bit

TABLE H.4: PAGE STATUS TABLE UPDATING

H.2.8 TRANSLATION LOOK-ASIDE BUFFER (TLB)

The CM1 board contains a Translation Look-Aside Buffer (TLB) (see fig. H.2) to speed up translation time for Mapped Space accesses. It is organized as 1K-entry direct mapped cache addressed by the low-order bits of the logical page address (LADD12-21).

TLB entries are 33 bits long and are formed of a data field and a tag field. THe tag field contains the high-order bits of the logical page number (LADD22-23), plus the context number and a valid bit, and is compared with the incoming logical page address to detect if the translation for the selected logical page is cached in the TLB. In case of a hit the cycle is terminated normally, issuing the physical address, checking the protection bits and eventually generating a Bus Error exception. In case of a miss the 68020 CPU is requested to retry the cycle (via the simultaneous assertion of the Berr and Halt lines), and during the end of the cycle the proper translation data are cached in the TLB from the segment and page maps.

The TLB is organized in such a way that any entry directly maps a logical page in a class of 256 (4 for the requested page address and the multiple of 4MB in the 16MB of mapped process space, times 64 for all context numbers). Considering the level of locality of page references and the very large size of TLB (the probability to have synonyms is very low), the hit rate of this kind of TLB is very high. To furhterly increase the hit rate, a mechanism is provided to selectively invalidate TLB entries on Segment and Page Table write accesses instead of fully flushing it. For this mechanism to work properly, software accesses in write to the segment table must be repeated more times presenting on the Page Number field of the logical address the numbers of the pages to be invalidated. This is owing to the fact that the page number field of the logical address is a don't care on Segment Table accesses, so the hardware must be specifically informed on what pages are to be invalidated. TLB selective invalidation on Page Table write accesses is performed automatically and does not need software control.

NOTE: At the present the feature of the selective invalidation of the TLB on Segment Table writes is not enabled and the coherency of the contents of the TLB is automatically guaranteed by Hardware for any software access to the MMU flushing the TLB at any Segment or Page Table write access.
To squeeze the higher performances out of the TLB, this feature

The Look-Aside Buffer contains an odd parity check bit transparent to the software, as the buffer itself, to ensure integrity of translation data. It can be checked together with the parity check logic entering the MMU Diagnostic Mode (see H.2.9.3, Control/status Register).

A parity error on the Look-Aside buffer when the MMU is performing a Mapped Space access gets the 68020 CPU to be halted. The presence of the Halt condition is also shown by a LED lamp on the CM1 board, for diagnostic purposes.

H.2.9 REGISTER SPACE

This Space is reserved to select the Hardware Registers of the MMU.

The lower logical address bits are used to select these registers as shown in the following table:

LOGICAL ADDRESS					
23-----5	4	3	2	1	0	
MBZ	0	0	0	MBZ	Supervisor Context Register
MBZ	0	0	1	MBZ	User Context Register
MBZ	0	1	0	MBZ	Error Register
MBZ	0	1	1	MBZ	RFU (Machine Depending)
MBZ	1	0	0	MBZ	Control/Status Register
MBZ	101-111			MBZ	RFU (Machine Depending)

NOTE

All Registers can be Read and Written except for the Error Register that can be Read but only Cleared in Write Operation (with all 32 Data bits equal to "0") and from the HR flag of the Control/Status Register.

TABLE H.5: REGISTER SPACE MAP

The layout of MMU Registers is depicted in fig. H.7, where accessable bits are showed.

Access to CM1 internal resources must always be performed with long word size to ensure proper operation.

As explained in H.2.3, Register Space is only accessable by the Supervisor.

H.2.9.1 CONTEXT REGISTERS

As illustrated in H.2.2, two Context Registers are provided. They are both six bit wide and can be read and written.

H.2.9.2 ERROR REGISTER

The Error Register is used to freze Parity Check error conditions occurred during MMU operations.

This register contains three separate error flags for the Segment Table, Page Table and Page Status Table and the logical OR of them to simplify software error checking. It must be cleared by the software after a System Reset sequence.

Error flags are set during an address translation operation (Mapped Space) or during a MMU table read operation (Segment, Page and Page Status Table read).

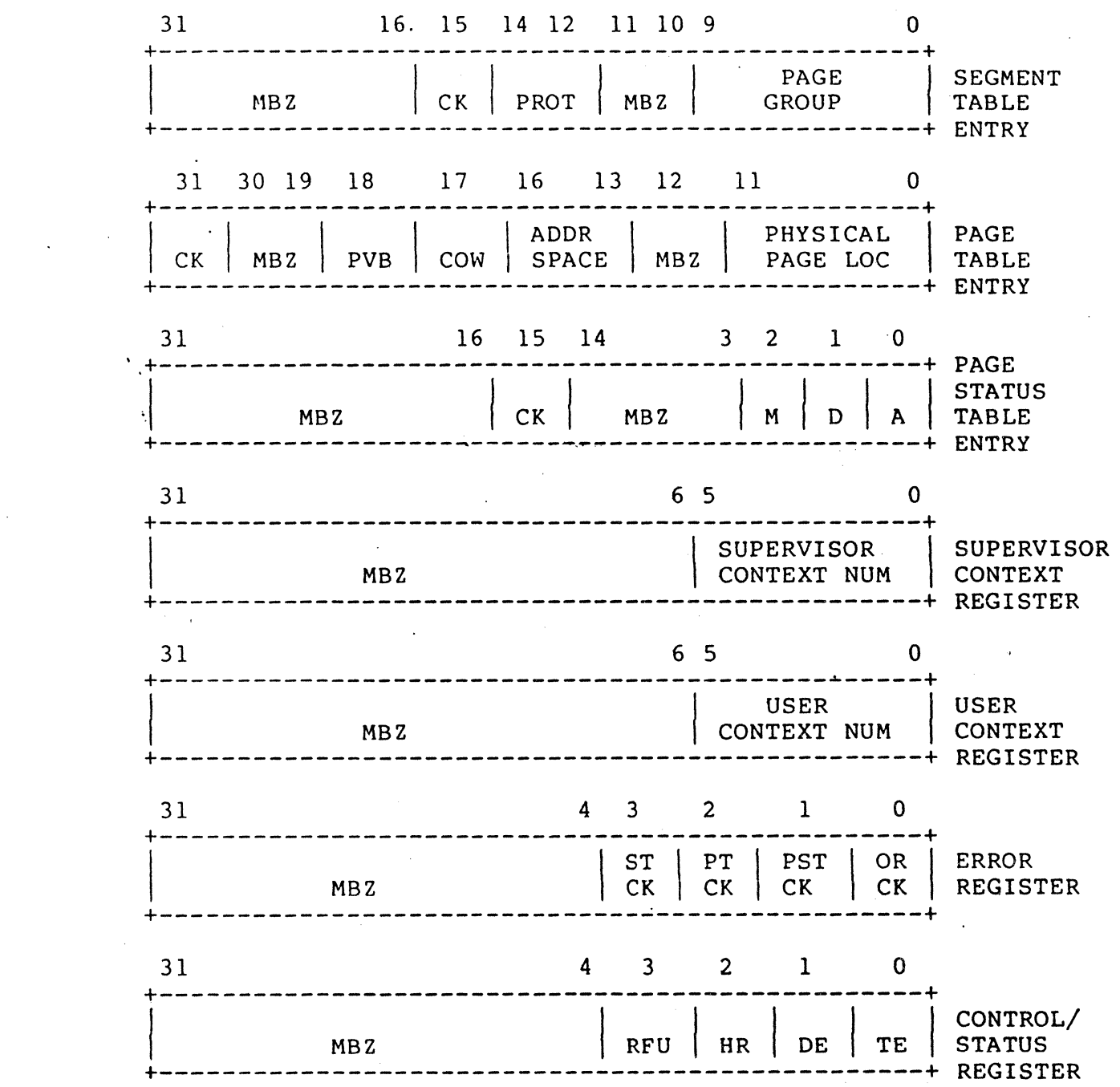
The Error Register can be set also during a Physical Unmapped Memory operation (see H.2.3.3, Physical Memory Access) if a Page Status Table parity check occurs.

A parity check error causes a Bus Error exception during an address translation operation or during a local unmapped memory operation, but not during a table read operation. Repeated check errors in both Segment and Page tables generate repeated Bus Errors (to be sure to get the CPU halted on a stuck error on these tables), while Page Status Table Check Bus Error is masked from the logical OR of error flags, so to be reported just once. This is owing to the fact that a system with a failed Page Status Table can still run in a downgraded fashion (i.e. all pages are intended to have been modified).

Owing to the fact that Page Status Table can generate a Bus Error, it must be setup by software (running EPROM code) before to access memory after a System Reset to avoid unexpected Bus Errors.

The presence of any check error (i.e. the logical OR active) is also notified by a LED lamp on the CM1 board, for diagnostic purposes.

The content of the Error Register, in supervisor mode, can be read but only cleared using a Write operation with all 32 Data bits equal to



NOTE:

- All MMU Resources can only be addressed with Long Word operations

- All Registers can be read and written except for the Error Register that can be read but CLEARED ONLY and for the HR flag of the Control/Status Register (see H.2.9.3)

- MBZ (RFU bits):
  In READ Don't Care
  In WRITE Must Be Zero

- CK (Parity Check bits):
  In READ Parity Check
  In WRITE Must Be Zero

H.2.9.3 CONTROL/STATUS REGISTER

A Control/Status Register is provided on the CM1 board to control operations and give MMU status information. This register contains three flags:

- TE Translation Enable
- DE Diagnostic Enable
- HR Halt Request

All flags can be read or written except for the HR flag that has a different function in Read and Write operations. The function of the flags are the following:

TE This flag, cleared after a System Reset, controls the address translation mechanism of the MMU and allows physical memory accesss without MMU intervent (e.g. at System Reset to run a program in memory before MMU tables are properly setup, or to run a diagnostic program in case of a failed MMU). Mapped Space accesses are actually translated only if this flag is set. Translation can be disabled at any time, allowing physical memory access also in the case of a failed MMU

DE This flag, when set, causes the MMU to enter a Diagnostic Mode used to verify the correctness of error checking logic. In diagnostic mode, check bits are inverted at generation on Segment and Page Table and on the Look-Aside Buffer, and stuck at one on the Page Status Table. Furthermore, the halt request generated on a Look-Aside check error is not issued to the CPU but only freezed in the HR flag of the Control/status Register. Properly operating on this flag, a system start-up program can verify that all error-control features of the CM1 board are working.

HR This flag is used to freeze a Halt Request condition generated on a Look-Aside Buffer check error. It has different meaning on Read and Write accesses. In read, it reports the state of the Halt Request condition and it is set whenever it is generated (e.g. on a diagnostic sequence of the Look-Aside Buffer, see DE flag). In write, it is used to reset a Halt Request generated in diagnostic mode before exiting diagnostic mode itself. The correct reset sequence is generated with a high-low pulse on this flag.