HIGH PERFORMANCE DIGITAL SIGNAL PROCESSOR FAMILY

1. GENERAL OUTLINE

The MSM699210 is a DSP in the same architecture as the MSM6992, but has doubled the internal memory size and reduced the number of pins by shared use of the external program and data buses. Program compatibility with the MSM6992 in the source level is maintained while newly added instructions and functional reinforcement have made it easier to use.

The MSM699210 is capable of high-speed execution of floating-point arithmetic operations (16-bit mantissa and 6-bit exponent part) and 16-bit fixed-point arithmetic operations. Devices will be available with 100nS machine cycle time.

The MSM699210 incorporates a 2K word × 32-bit ROM program memory and two 256-word × 22 bit data RAMs.

The program and data memories can both be expanded externally up to 64K words via dedicated data and address lines.

The MSM699210 has the processor mode and controller mode as external interface. In the processor mode, operation by the external program memory, operation by the on-chip program ROM, or operation by both the external program memory and on-chip program ROM is possible. Each of these is carried out only in the master mode. In the controller mode, the MSM699210 is operated only by the on-chip program ROM in both the master mode and slave mode in the same way as the MSM6992.

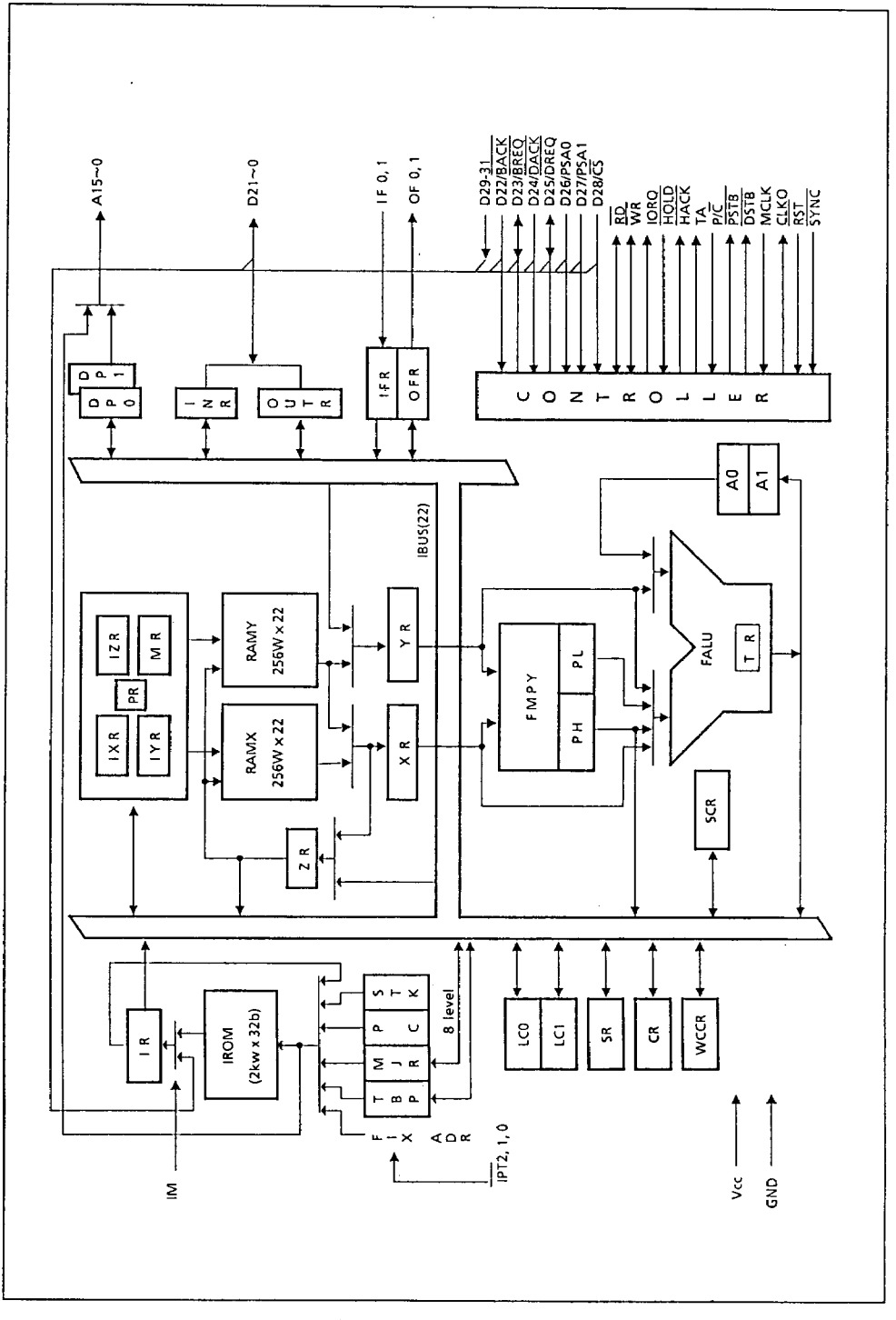
The abundant interface functions allow establishing a multiprocessor system or highly flexible system configuration.

Major 699210 applications include analysis for speech recognition and speech analysis/synthesis in speech processing equipment, high speed modems, codec, and echo cancelers in communication equipment. This device can also be effectively used for meter control, robotics and in audio equipment.

Key Features:

- Instruction cycle: 100ns
- Arithmetic formats:
  - Floating-point arithmetic 16E6
  - Fixed-point arithmetic 16-bit
  - Logical arithmetic 22-bit
- Built-in 2K-word × 32-bit instruction ROM (Also usable as data ROM)
- Built-in 512-word × 22-bit data RAM
- 32-bit wide horizontal microinstruction
- 32-bit wide horizontal microinstruction
- 64K-word program memory area
- 64K-word data memory area
- Multiprocessor interface
- DMA controller connection capability
- Slower memory interface
- Three external maskable interrupts
- Maximum 15-bit shift function (left or right)
- Double loop function
- Power down function
- 84-pin PLCLC/100-pin FLAT package
- 1.5μm silicon gate CMOS

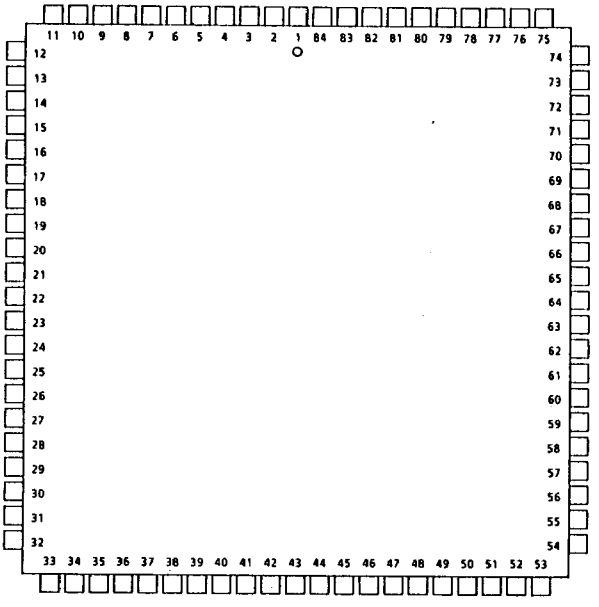
BLOCK DIAGRAM



SYMBOL	UNIT NAME	SIZE	SYMBOL	UNIT NAME	SIZE
IROM	Instruction ROM	2kW x 32b	FMPY	Floating-pont Multiplier	
IR	Instruction Register	32b	PH	Product Register High	22b
TBP	Table Pointer	16b	PL	Product Register Low	15b
MJR	Map Jump Register	16b	FALU	Floating-point ALU	
PC	Program Counter	16b	TR	Temporary Register	22b
STK	Program Couonter Stack	16b x 8 level	A0	Accumulator 0	22b
LC0	Loop Counter 0	12b	A1	Accumulator 1	22b
LC1	Loop Counter 1	4b	SCR	Shift Count Register	6b
CR	Control Register	12b	DP0	Data pointer 0	16b
SR	Status Register	10b	DP1	Data pointer 1	16b
WCCR	Wait Cycle Count Register	3b	INR	Input Register	22b
RAMX	Data RAM X	256w x 22b	OUTR	Output Register	22b
RAMY	Data RAM Y	256w x 22b	IFR	Input Flag Register	2b
XR	X Register	22b	OFR	Output Flag Register	2b
YR	Y Register	22b			
ZR	Z Register	22b	IBUS	Internal Data Bus	22b
IXR	Index Register X	8b			
IYR	Index Register Y	8b			
IZR	Index Register Z	8b			
MR	Mode Register	8b			
PR	Page Register	2b			

1.2 Pin Assignments

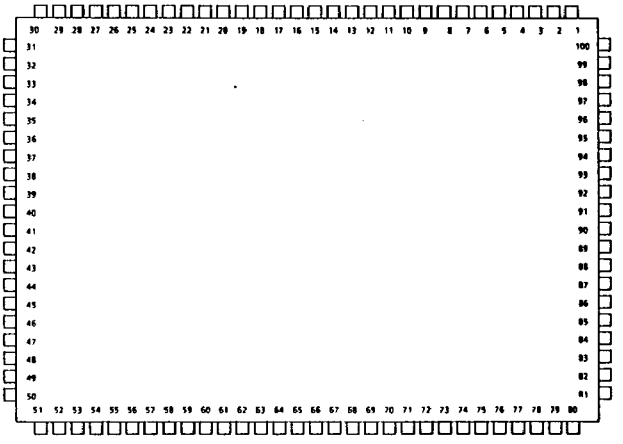
[84 pin PLCC]



Pin No.	Pin Name	I/O	Pin No.	Pin Name	I/O	Pin No.	Pin Name	I/O
1	GND		29	A15	O(3-state)	57	P/$\overline{C}$	I
2	D22/$\overline{BACK}$	I	30	A14	O(3-state)	58	IM	I
3	D21	I/O	31	A13	O(3-state)	59	$\overline{RST}$	I
4	D20	I/O	32	A12	O(3-state)	60	$\overline{SYNC}$	I
5	D19	I/O	33	A11	O(3-state)	61	☆	I
6	VCC		34	A10	O(3-state)	62	☆	I
7	D18	I/O	35	A9	O(3-state)	63	☆	I
8	D17	I/O	36	A8	O(3-state)	64	VCC	
9	D16	I/O	37	A7	O(3-state)	65	**NC**	O
10	D15	I/O	38	VCC		66	$\overline{DSTB}$	O(3-state)
11	D14	I/O	39	A6	O(3-state)	67	$\overline{WR}$	I/O
12	D13	I/O	40	A5	O(3-state)	68	$\overline{RD}$	I/O
13	GND		41	A4	O(3-state)	69	GND	
14	D12	I/O	42	A3	O(3-state)	70	MCLK	I
15	D11	I/O	43	GND		71	$\overline{PSTB}$	O(3-state)
16	D10	I/O	44	A2	O(3-state)	72	TA	O
17	D9	I/O	45	A1	O(3-state)	73	CLKO	O
18	D8	I/O	46	A0	O(3-state)	74	IORQ	O(3-state)
19	D7	I/O	47	$\overline{HACK}$	O	75	D31	I
20	D6	I/O	48	OF1	O	76	D30	I
21	VCC		49	OF0	O	77	VCC	
22	D5	I/O	50	IF1	I	78	D29	I
23	D4	I/O	51	IF0	I	79	D28/$\overline{CS}$	I
24	D3	I/O	52	$\overline{HOLD}$	I	80	D27/PSA1	I
25	D2	I/O	53	☆	I	81	D26/PSA0	I
26	D1	I/O	54	$\overline{IPT2}$	I	82	D21/$\overline{DREQ}$	I/O
27	GND		55	$\overline{IPT1}$	I	83	D24/$\overline{DACK}$	I
28	D0	I/O	56	$\overline{IPT0}$	I	84	D23/$\overline{BREQ}$	I/O

Note: Pins marked by an ☆ must be connected to ground

[100 pin QFP]



Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	A12	26	**NC**	51	IORQ	76	D14
2	**NC**	27	**NC**	52	**NC**	77	**NC**
3	**NC**	28	**NC**	53	**NC**	78	**NC**
4	**NC**	29	**NC**	54	**NC**	79	**NC**
5	A11	30	$\overline{IPT2}$	55	**NC**	80	D13
6	A10	31	$\overline{IPT1}$	56	D31	81	**NC**
7	A9	32	$\overline{IPT0}$	57	D30	82	GND
8	A8	33	$P/\overline{C}$	58	VCC	83	D12
9	A7	34	IM	59	D29	84	D11
10	VCC	35	$\overline{RST}$	60	D28/CS	85	D10
11	A6	36	$\overline{SYNC}$	61	D27/PSA1	86	D9
12	A5	37	☆	62	D26/PSA0	87	D8
13	A4	38	☆	63	D25/$\overline{DREQ}$	88	D7
14	A3	39	☆	64	D24/$\overline{DACK}$	89	D6
15	GND	40	VCC	65	D23/$\overline{BREQ}$	90	VCC
16	A2	41	**NC**	66	GND	91	D5
17	A1	42	**NC** (O)	67	D22/$\overline{BACK}$	92	D4
18	A0	43	$\overline{DSTB}$	68	D21	93	D3
19	$\overline{HACK}$	44	$\overline{WR}$	69	D20	94	D2
20	OF1	45	$\overline{RD}$	70	D19	95	D1
21	OF0	46	GND	71	VCC	96	GND
22	IF1	47	MCLK	72	D18	97	D0
23	IF0	48	$\overline{PSTB}$	73	D17	98	A15
24	$\overline{HOLD}$	49	TA	74	D16	99	A14
25	☆	50	CLKO	75	D15	100	A13

Note: Pins marked by an ☆ must be connected to ground.
See PLCC pin assignments for discription of I/O.

2. MSM699210 CONFIGURATION

2.1 Pin Function

Pin Symbol		I/O	Function
Processor	Controller		
A_{15-0}		O (3-state)	• Parallel address bus In the processor mode, address for program and address for data are sent based on time division. In the controller mode, specified as address bus only for data.
D_{21-0}		I/O (3-state)	• Parallel data bus In the processor mode, used as input andoutput pins for data transfer between external and host CPU, I/O device. In addition, used as program data input pin for the lower 22 bits of micro-instruction. In the controller mode, specified as data bus only for data.
D_{31-29}	—	I	• Program data input pins in the processor mode
D_{28}	$\overline{CS}$	I	• Program data input pin in the processor mode • Used as chip select pin in the controller mode
D_{27}	PSA_1	I	• Program data input pin in the processor mode • Used as port select address pin in the controller mode
D_{26}	PSA_0	I	• Program data input pin in the processor mode • Used as port select address pin in the controller mode
D_{25}	$\overline{DREQ}$	I/O	• Program data input pin in the processor mode • DMA request output pin in the controller mode
D_{24}	$\overline{DACK}$	I	• Program data input pin in the processor mode • DMA acknowledge input pin in the controller mode
D_{23}	$\overline{BREQ}$	I/O	• Program data input pin in the processor mode • Bus acknowledge input pin in the controller mode . Inputs the external data bus permission signal.

Pin Symbol		I/O	Function
Processor	Controller		
D_{22}	$\overline{BACK}$	I	• Program data input pin in the processor mode • Bus acknowledge input pin in the controller mode. Inputs the external data bus permission signal.
$\overline{RD}$		I/O	• Input/output pin for read control Outputs the read control signal for the external data memory and I/O device in the master mode. Input of the read control signal from the host CPU and other DSP in the slave mode
$\overline{WR}$		I/O	• Input/output pin for write control Outputs the write control signal for the external data memory and I/O device in the master mode. Input of the write control signal from the host CPU and other DSP in the slave mode
IORQ		O (3-state)	• I/O request output pin Indicates whether the write or read operation is with respect to external data memory or I/O device
TA		O	• Table data access indication output pin TA = "1" when reading the table in the program memory
$\overline{HOLD}$	———	I	• Hold request input pin Whenthis signal is accepted, DSP places A_{15-0}, D_{21-0} $\overline{RD}$, $\overline{WR}$, and IORQ signals in the "HZ" state.
$\overline{HACK}$	———	O	• Hold acknowledge output pin Indicates that the DSP is in the hold state

Pin Symbol		I/O	Function
Processor	Controller		
$IF_{1,0}$		I	• Universal input flag pins Inputs to set flag in IFR
$OF_{1,0}$		O	• Universal output flag pins Output of OFR
$\overline{IPT}_{2,1,0}$		I	• 3-level external maskable interupt pins Interrupts are accepted except for the "JMP" instruction Interrupts areaccepted if interrupt level is greater than interrrupt priority set in control register
$P/\overline{C}$		I	• Processor/controller mode select pin
IM	———	I	• Internal ROM select input pin Used to select between internal or external program memory in theprocessor mode Invalid in the controller mode
$\overline{PSTB}$	———	O (3-state)	• Program strob output pin Indicates the fetch cycle of the program data in the processor mode Invalid in the controller mode
$\overline{DSTB}$		O (3-state)	• Data strob output pin Indicates the access cycle of external data
$\overline{SYNC}$		I	• Synchronization input pin Allows the synchronization of some DSPs by switching this signal from the low state to the high state when the $\overline{RST}$ signal is in the low state.

Pin Symbol		I/O	Function
Processor	Controller		
$\overline{RST}$		I	• Reset input pin This signal initializes all internal state of DSP
MCLK		I	• Master clock input pin
CLKO		O	• Internal system clock output pin
VCC		–	• + 5V supply pins
GND		–	• Ground pins

2.2 Unique Features

The MSM699210 has some unique features that make this device easily adaptable to various applications. These features are high lighted below:

2.2.1 Data Structure

- 22-bit floating-point data format (16-bit mantissa and 6-bit exponent part)
- Double precision arithmetic operations on 16-bit fixed-point data possible
- Built-in high-throughout RAM : Capability to read to operands of data and write one operand of data within a single cycle is possible
- Simplified addressing of internal RAMs: Two types addressing M6992 and M699210 type. Four addressing modes including direct, index modification, and index updating.
- Data shift within memory : Data read from one location of the memory can be written into another location of memory immediately

2.2.2 Instruction Sequence Control

- Sequence control : Sequence, push, return, and loop control available in all arithmetic instructions plus the SFT and MOVE instructions
- Loop function : Double-loop can be enabled by use of 12-bit loop counter (LC0) and a 4-bit loop counter (LC1). LC0 is controlled by the sequence field in each instruction, and LC1 is controlled by the conditional jump instructions.
- Interrupt function : Three levels of interrupts are available
- Data ROM : The program ROM may also be used as a data ROM

2.2.3 External Interface

- Access to external data memory at the same time of operation is possible.

- Two external data RAM pointers (DP0 and DP1) are available to simplify handling of two types of data

- Many expansion functions

  Maximum of 64K words of external program memory

  Maximum of 64K words of external data memory

- Interface modes

  - Controller mode (on-chip ROM operation)
    - Slave mode : Use of host processor,8-bit or 16-bit microprocessoror another M699210 is possible
    - Master mode : Accessing of external memory and I/O device is possible
    - DMA mode : DMA transfer under external DMA control is possible

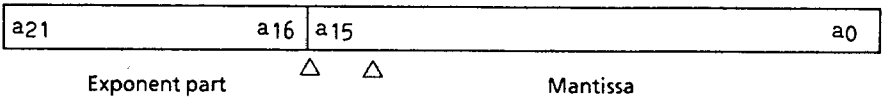
  - Processor mode (on-chip ROM and off-chip memory operation)
    - Master mode : Accessing of external memory and I/O device is possible Hold function is possible

  - Slower memory interface mode
    - Communication to slower off-chip memories and peripherals is possible

2.3 Data Format

The three data formats handled by M6992 are floating-point, fixed-point,and logical data format. The floating-point data mantissa and exponent parts plus the fixed-point data is represented by 2's complement. The △ symbol in these formats specifies the position of the decimal point.

(1) Floating-point data format



Mantissa $M=(-1)\times a^{15}+\sum_{i=1}^{15} a_{15-i}\times 2^{-i}$

Exponent part $E=(-32)\times a_{21}+\sum_{i=0}^{4} a_{i+16}\times 2^{i}$

Maximum positive value:

011111	0.11-----11

$(1-2^{15})\times 2^{31}\approx 2.15\times 10^{9}$

Minimum positive value:

100000	0.10-----00

$0.5\times 2^{-32}\approx 1.16\times 10^{-10}$

Zero representation:

100000	0.00-----00

Minimum negative value:

100000	1.01-----11

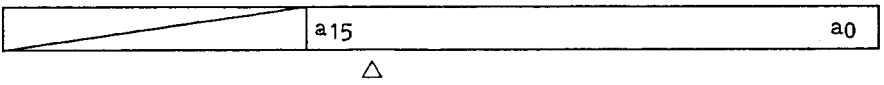
$(-0.5-2^{-15})\times 2^{-32}\approx -1.16\times 10^{-10}$

Maximum negative value:

011111	1.00-----00

$-1\times 2^{-31}\approx -2.15\times 10^{9}$

(2) Fixed-point data format

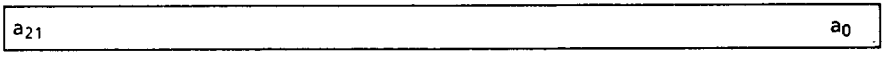


△

$$(-1) \times a_{15} + \sum_{i=1}^{15} a_{15-i} \times 2^{-i}$$

Maximum positive value:	0.11-----11	$1-2^{-15}$
Maximum negative value:	1.00-----00	-1
Zero representation:	0.00-----00	

(3) Logical data format



2.4 Functional Blocks

2.4.1 FMPY (Floating-point Multiplier)

(1) The floating-point multiplier consists of a 16 x 16-bit parallel multiplier and an exponential adder circuit. Contents of registers XR & YR are used as the input data, and the multiplications is executed in every cycle.

(2) Input data must be normalized when floating-point multiplications are executed. The resultant product of flating-point multiplications must also be normalized.

(3) Product format

Floating-point multiplication : 16E6 x 16E6 → 16E6

Fixed-point multiplication : 16-bit x 16-bit → 31-bit

The format of the data, floating-point or fixed-point, is specified by control register (CR).

(4) Rounding and clipping

- Rounding

  The execution of rounding/clipping is specified by control register (CR).

- Clipping

  If a positive or negative overflow occurs during a flating-point multiplication, the product is forced to a maximum positive or negative output value. And if an underflow occurs, the product is forced to a zero.

(5) The results of flating-point or fixed point multiplication are stored in registers PH and PL.

	PH	PL
Floating-point multiplication result	21 16 15 0 Exponent part 6 ibts | Mantissa 16 bits	
Fixed-point multiplication result	21 16 15 0 Higher order 16 bits	15 1 0 Lowr order 15 bits | 0

2.4.2 FALU (Floating-point Arithmetic Logic Unit)

(1) Floating-point and fixed-point additions and subtractions and logical operations between XR, YR, PH, PL, A0, and A1 are executed by the FALU.

(2) Pipeline operation: The FALU has a 2-stage pipeline structure where operations are completed in two machine cycles. The main processing in the first stage includes digit alignment and arithmetic operations, and in the second stage executes normalization,clipping, flag setting etc.

(3) FALU operation results are held in two 22-bit accumulators. These results can also be transferred directly to other registers via internal data bus.

(4) In floating-point data scaling and fixed-point data arithmetic shifts, the amount of shift can be specified by a 6-bit shift count register (SCR), or directly by the SFT instruction.

The types of operations possible are summarized below

Operation	Floating-point	Fixed-point	Logical operation
Addition and subtraction	○	○	
Through	○		
Sign inversion	○		
Absolute value operation	○		
Addition and subtraction with carry		○	
AND, OR, XOR, and 1's complement			○
Shift (left/right 0 ~ 15b)	○	○	
Data format conversion (Floating-point→fixed-point, Fixed-point→floating-point)	○		

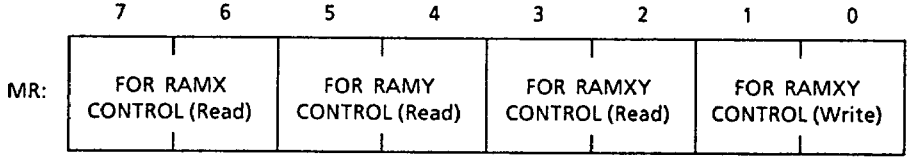
2.4.3 Address Control Functions

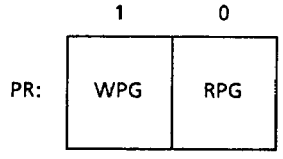
Address control types are divided into M699210 type and M6992 type. The M6992 type keeps compatibility with MSM6992, and when DSP is initialized by $\overline{RST}$ signal, the M6992 type is selected by control register (CR).

(1) IXR, IYR, IZR (Index Registers X, Y, and Z)

IXR is a 8-bit index register to modify XA fields, and IYR is an 8-bit index register to modify YA and XYA fields. Contents of these registers are used as read addresses for RAMX, RAMY, and RAMXY. IZR is an 8-bit index register to modify ZA fields. This register is used when the RAMXY is in write mode.

(2) MR (Mode Register) and PR (Page register)





RPG = "0" ..... RAMX Select
RPG = "1" ..... RAMY Select
WPG = "0" ..... RAMX Select
WPG = "1" ..... RAMY Select

The 8-bit mode register is used to specify the RAMX and RAMY addressing mode. The RAMX, RAMY, and RAMXY addresses are determined by arithmetic operations executed on index registers (IXR, IYR, and IZR) and instruction offset (XA, YA, XYA, and ZA) in accordance with the contents of this mode register.

The 2-bit PR specifies page of the internal memory. When data is written into the internal RAMXY or data is read from the internal RAMXY, they are accessed by means of specifying pages in the PR. When data is read from the RAMX or RAMY, or when addressing in the M6992 type, the PR cannot be used. The MR and PR are initialized to "0" by the $\overline{RST}$ signal.

RAMX and RAMY

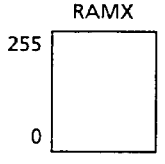
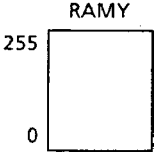
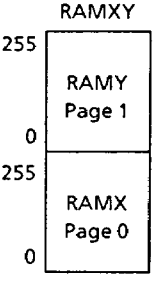
(1) RAMX and RAMY are both 256 x 22-bit data RAMs.

(2) These RAMs are capable of both a write and a read in the same cycle.

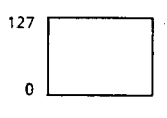
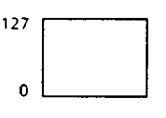
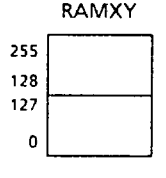
(3) RAMX and RAMY can be used independently or as two pages of memory in a read cycle. These two pages of memory is called RAMXY. But in a write cycle, these RAMs are always used as RAMXY.

(4) RAMX and RAMY outputs are held in XR and YR respectively. RAMXY output is held in XR. ZR is used as a data-write-buffer when writing to RAMX and RAMY. RAMX and RAMXY outputs can also be written to ZR and so, data within the memory can be shifted in one machine cycle.

The RAM address can be specified by the following. (M699210 type)

Field used in instruction	When XA field is used	When YA field is used	When XYA and ZA fields are used
Given adress			

The RAM address can be specified by the following. (M6992 type)

Field used in instruction	When XA field is used	When YA field is used	When XYA and ZA fields are used
Given adress			

- Read address when XA field is used

MR 7	MR 6	RAMX read address	IXR
0	0	0 0 XA	Hold
0	1	IXR + XA (no sign)	Hold
1	0	IXR + XA (with sign)	Hold
1	1	IXR + XA (with sign)	IXR update

- Read address when YA field is used

MR 5	MR 4	RAMY read address	IYR
0	0	0 0 YA	Hold
0	1	IYR + YA (no sign)	Hold
1	0	IYR + YA (with sign)	Hold
1	1	IYR + YA (with sign)	IYR update

- Read address when XYA field is used

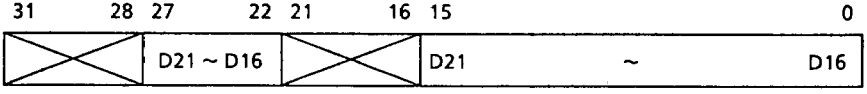
MR 3	MR 2	RAMXY read address	IYR
0	0	0 0 XYA	Hold
0	1	IYR + XYA (no sign)	Hold
1	0	IYR + XYA (with sign)	Hold
1	1	IYR + XYA (with sign)	IYR update

- Write address when ZA field is used

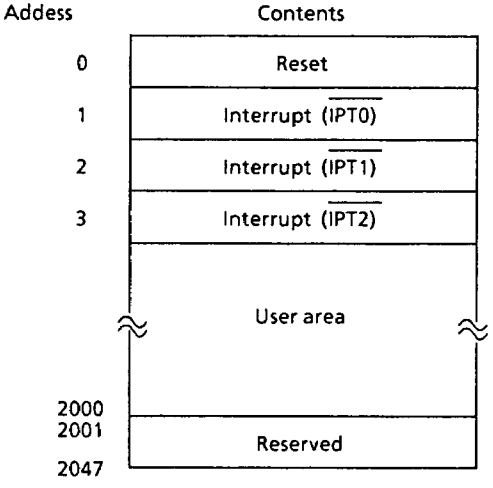
MR 1	MR 0	RAMXY write address	IZR
0	0	0 0 ZA	Hold
0	1	IZR + ZA (no sign)	Hold
1	0	IZR + ZA (with sign)	Hold
1	1	IZR + ZA (with sign)	IZR update

2.4.5 IROM (Instruction ROM)

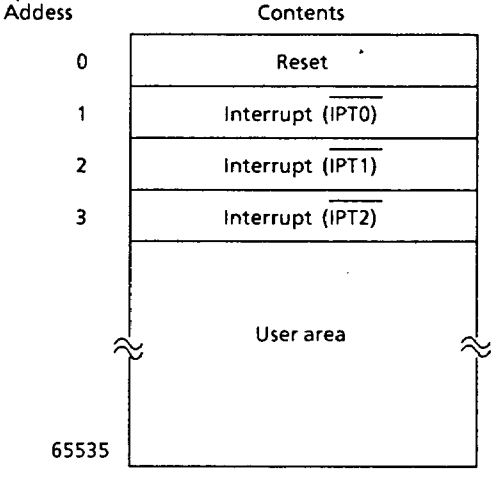
Programs and data are stored in this 2K-word × 32-bit mask ROM. Data must be stored in the following bit positions.



The address map is outlined below.



The external instruction memory address map is as follows.



2.4.6 Micro-sequence Control Functions

(1) The following functions are used in generation of micro-addresses.

Function	Address of enxt insruction
Reset	0
Interrupt	1 ~ 3
Sequence	Own address + 1
Return	Contents of last written stack
Jump	JMP instruction JA field, or own address + 1
MAP jump	MJR contents

(2) PC (Program Counter)

The program counter is a 16-bit register which holds the program addresses. The value obtained by adding 1 to the address of the instruction being executed currently (N) is set in the PC contents. i.e. PC = N + 1

(3) STK (Program Counter Stack)

The program counter stack is a 16-bit × 8-level register which saves PC contents when an interrupt, subroutine call, or push is executed. When the return instruction is executed, the last stored contents are read and become the program memory address.

(4) MJR (Map Jump Register)

The Map Jump Register is a 16-bit register used to specify jump destination addresses when a program start address is given.

(5) TBP (Table Pointer)

The table pointer is a 16-bit up-counter used to specify data addresses when data is read from a program area.

(6) IR (Instruction Register)

This 32-bit register is used to hold an instruction currently being executed. Table data and literal data can be transferred to other registers via the internal data bus, IBUS.

(7) LC0 and LC1 (Loop Counters 0 and 1)

These 12-bit (LC0) and 4-bit (LC1) down-counters are used to specify program loop counts. Double loop operations are possible.

2.4.7 Status Register (SR)

The status register can specify up to ten different status types.

9	8	7	6	5	4	3	2	1	0
IR	OR	MV	EV	EU	N	Z	Z22	V	C

IR - Input Ready

Indicates that data has been set in INR, input register.

OR - Output Ready

Indicates that data has been set in OUTR, output register.

MV - Mantissa Overflow Flag

Indicates that an over flow has been generated in a fixed-point arithmetic operation. If generated in a multiplier, however, the status is set after a delay of one machine cycle.

EV - Exponent Overflow Flag

Indicates that an exponent over flow has occurred during a flowing-point arithmetic operation.

EU - Exponent Underflow Flag

Indicates that an exponent underflow occurred during a floating-point arithmetic operation.

N - Negative Flag

Indicates that result of FALU operation is negative.
(Negative data when N = "1")

Z - Zero Flag (16-Bit)

"1" is set in this flag when all 16 bits of the mantissa are 0 as a result of a FALU operation.

Z22 - 22-Bit All Zero Flag

"1" is set in this flag when all 22 bits are 0 as a result of a FALU operation.

V - Overflow Flag

Indicates that a overflow generated by the 16-bit ALU in the FALU has been set. If an overflow occurs in the mantissa of a floating-point arithmetic operation, the result is scaled to half of the value so that no apparent overflow occurs.

C - Carry Flag

Indicates that a carry generated by the 16-bit ALU in the FALU has been set.

2.4.8 Control Register (CR)

The control register is used to specify different operation modes.

11	10	9	8	7	6	5	4	3	2	1	0
AT	HE	MM	MREL	MREX	ARE	CLE	DMAM	IOM1	IOM0	IP1	IP0

AT - Adderssing type

AT = "0" ---------- M6992 type addressing
AT = "1" ---------- M699210 type addressing

HE - Hold Enable

Enable the receive of HOLD signal

MM - Multiplication Mode

Designation of multiplier mode
MM = "0" ---------- floating-point operation mode
MM = "1" ---------- fixed-point operation mode

MREL - Multiplication Round Enable on Floating-Point Data

MREX - Multiplication round Enable on Fixed-Point Data

ARE - ALU Round Enable

Enable floating-point addition/subtraction rounding.

CLE - Clip Enable

Enable clipping in FMPY or FALU.

DMAM - DMA mode Select

IOM1 and IOM0 - I/O modes 1 and 0

Specifies the type of external processor communicating with MSM6992.

IOM1	IOM0	Type of processor
0	0	MSM6992
1	0	8-bit microprocessor
1	1	16-bit microprocessor

IP1 and IP0 - Interrupt Priority

Designation of level external interrupts which can be received.

IP1	IP0	Interrupt mask status
0	0	$\overline{\text{IPT0}}$, $\overline{\text{IPT1}}$, $\overline{\text{IPT2}}$ interrupts enabled
0	1	$\overline{\text{IPT1}}$ and $\overline{\text{IPT2}}$ interrupts enabled
1	0	$\overline{\text{IPT2}}$ interrupts enabled
1	1	External interupts disabled

2.4.9 Input/Output Control Functions

(1) DP1 and DP0 - Data Pointers 1 and 0

These two 16-bit registers indicate external data memory and I/O addresses. Selection and updating (+ 1, 0, -1) of DP0 & DP1 can be specified by an instruction.

(2) INR - Input Register

The input register is a 22-bit register which buffers input data from parallel data bus (D21~D0).

(3) OUTR - Output Register

The output register is a 22-bit register which buffers the DSP data being output to the system bus.

(4) IFR - Input Flag Register

The input flag register is a 2-bit register set from the universal input flag pins IF1 and IF0.

(5) OFR - Output Flag Register

The output flag register is a 2-bit register which holds data to be applied to the universal output flag pins OF1 and OF0.

2.4.10 Interface

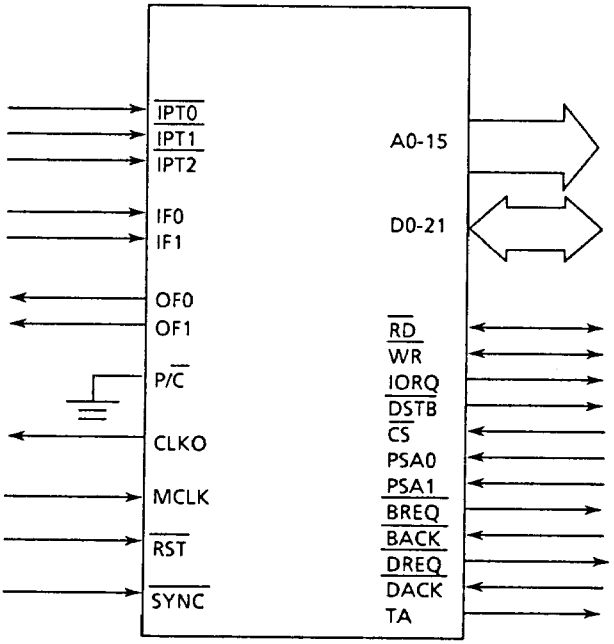
(1) Controller Mode

Only the on-chip ROM is available for the program memory. In this case, the program is requested to be made mask ROM. The address bus (A_{15-0}) is used only for the address of external data and the data bus (D_{21-0}) is used only for the external data.

The master and slave mode are used as external interface mode.

In the master mode, DSP can access to the external memory of 64 K words and I/O devices at full speed or low speed.

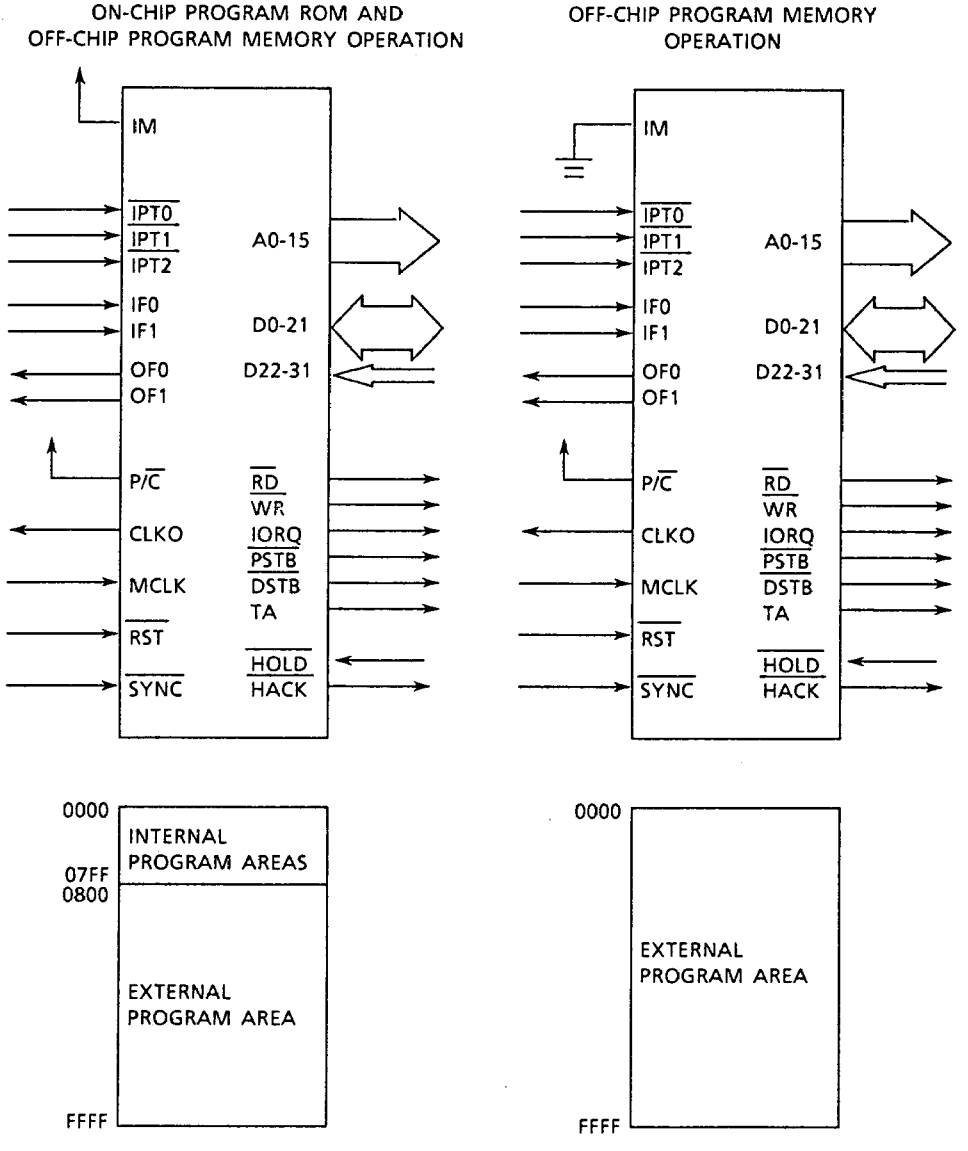
In the slave mode, communication between the 8-bit processor, 16-bit processor and other MSM699210 can be provided.



(2) Processor Mode

The on-chip ROM, off-chip memory, both on-chip ROM and off-chip memory can be used as program memory. Since the address bus and data bus are used for program and data on the basis of time division, 2 + α cycles are necessary for execution of the external data access instruction. α indicates the number of wait cycles at the interface of slower memory and peripheral, and is determined by the WCCR.
(α = 0~7)

The interface is provided only in the master mode, and it is not provided in the slave mode. However, using the hold function can request the DSP for permission of using the external bus. When the $\overline{\text{HOLD}}$ signal is made active and it can be accepted, the DSP makes the $\overline{\text{HACK}}$ signal active, places A15-0, D21-0, $\overline{\text{RD}}$, $\overline{\text{WR}}$ and IORQ signals in the high-impedance state, and stops internal processing. When the $\overline{\text{HOLD}}$ signal is released, the DSP restarts the processing. Whether to receive the $\overline{\text{HOLD}}$ signal is selected by the control register (CR).



PROGRAM MEMORY MAP

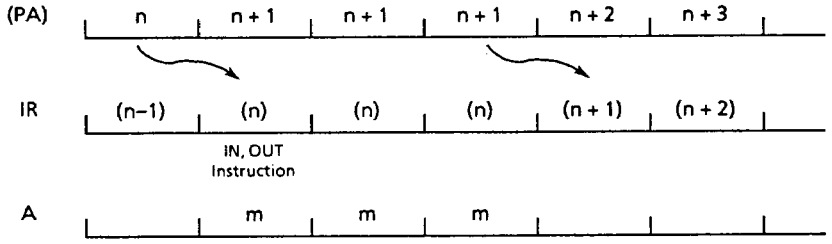
When the program address exceeds 7FF(H) in IM = "1", the program area is automatically switched from the internal area to the external area. therefore, both the internal ROM and external memory can be used as the program memory.

2.4.11 Slower Memory and Peripheral Interface

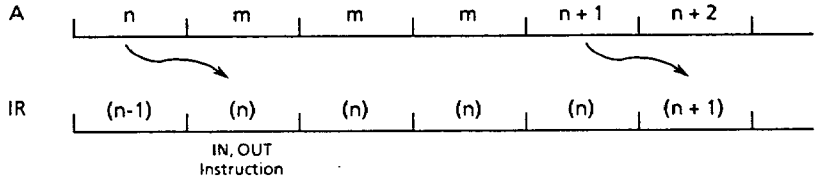
The WCCR (Wait cycle count register) is a 3-bit register for slower memory interface. The access speed of the external memory and I/O device can be changed according to the contents of WCCR.

WCCR	Number of wait cycles
000	0
001	1
010	2
.	
.	
.	
111	7

- Controllor Mode (WCCR = 02H)



- Processor Mode (WCCR = 02H)



(PA)	:	Internal program address
IR	:	Instruction
A	:	Address ($A_{15\text{-}0}$)
m	:	External memory adderss
n	:	Program address

The WCCR is initialized to "0" by the $\overline{RST}$ signal.

2.4.12 Interrupt

3-external maskable interrupts are possible.

level-0	< level-1	< level-2
($\overline{IPT_0}$)	($\overline{IPT_1}$)	($\overline{IPT_2}$)

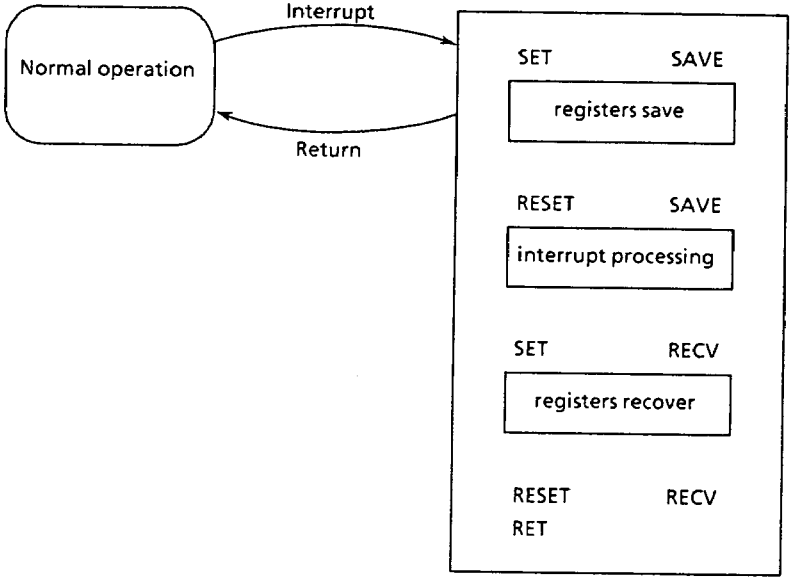
(1) Level 2 interrupt

When the level-2 interrupt is accepted, the PH, PL, SR, TR, A_0, A_1, XR and YR are placed in the write inhibition state, and their contents are held during the interrupt. After the interrupt is replaced by RET instruction, the arithmetic operation can be restarted. The leve-2 interrupt is used to transfer the data between the DSP and external memory mainly.

(2) Level 0, 1 interrupt

When the level-0, 1 interrupt is accepted, using the SAVE, RECV instruction, contents of each register excluded TR, PH, PL can be saved or recovered.

The basic sequence of interrupt at level 0 and 1 is as follows:



2.4.13 Power down Function

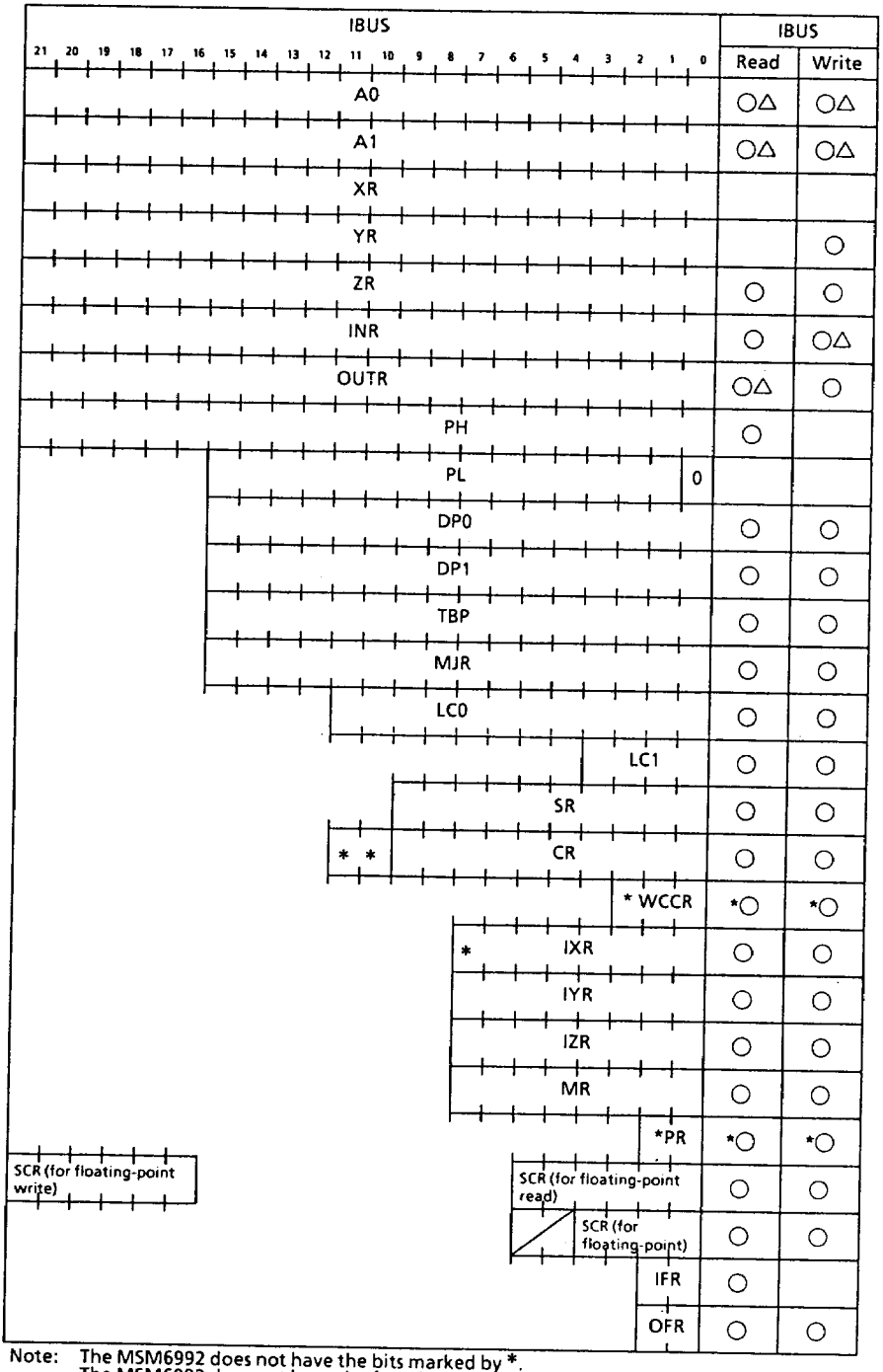
The power down mode can be set by the RST signal input ($\overline{RST}$ = "0") or by execution of the power down instruction. The power down mode can be released when the RST signal is released ($\overline{RST}$ = "1") or when the interrupt is accepted. Therefore, when the power down mode is released by means of interrupt, it is necessary to enable the $IP_{0,1}$ flag in the control register for the interrupt.

As the contents of registers are held at the time of power down, the instruction can be restarted by releasing the power down mode.

At the time of power down, only the following functions are executed, and the DSP is placed in the stop state.

- Internal clock generation circuit
  (The CLKO signal is fixed to "0")
- $\overline{RST}$ input circuit
- Power down control circuit
- Interrupt circuit
- $IF_{1,0}$ input circuit

2.4.14 Register Reat/Write List



Note: The MSM6992 does not have the bits marked by *.
The MSM6992 does not have the functions marked by * and △.

3. INSTRUCTION FORMATS

The MSM6992 instruction formats are outlined in this chapter.

The 1-word 32-bit horizontal instruction format has been adopted for the MSM6992 to achieve high processing performance. The 11 types of instructions are listed below.

A-type	I-type	IFLA	.....	Floating-point Arithmetic (Internal)
		IFXA	.....	Fixed-point Arithmetic (Internal)
		ILO	.....	Logical Operation (Internal)
	E-type	EFLA	.....	Floating-point Arithmetic (External)
		EFXA	.....	Fixed-point Arithmetic (External)
		ELO	.....	Logical Operation (External)
		SFT	.....	Shift Operation
		MOVE	.....	Move Operation
		LIT	.....	Literal Operation
		CTL	.....	Control Operation
		JMP	.....	Jump Operation

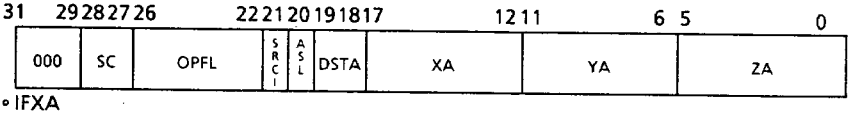
The internal arithmetic and logical operation instruction (IFLA, IFXA, ILO) can simultaneously specify three operands in the internal data memory, and the external type (EFLA, EFXA, ELO) of instructions can simultaneously specify two operands in the internal data memory while concurrently accessing an external data memory.

The I, E, and SFT instructions types are also called "A-type" (arithmetic type) instructions.

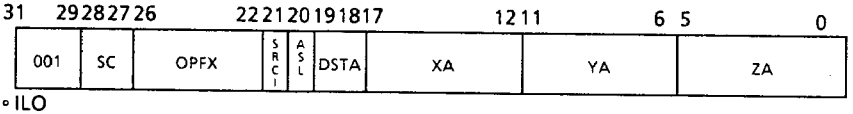
Each operation field in each instruction type is described in this chapter.

Instruction Formats

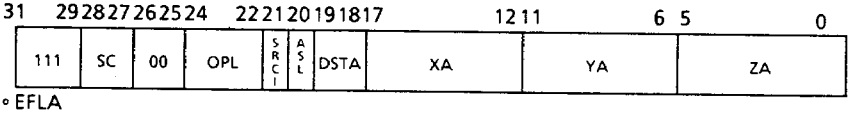
- IFLA



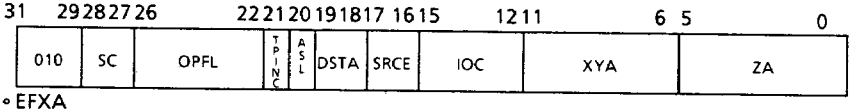
- IFXA



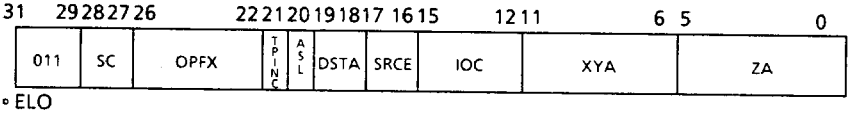
- ILO



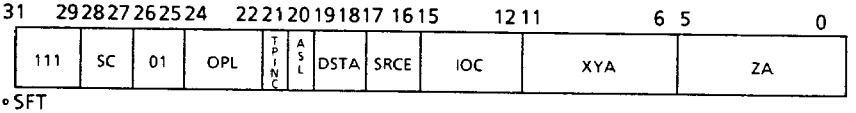
- EFLA



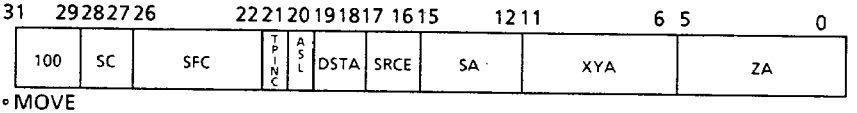
- EFXA



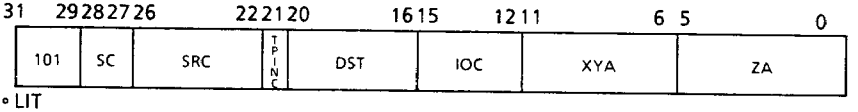
- ELO



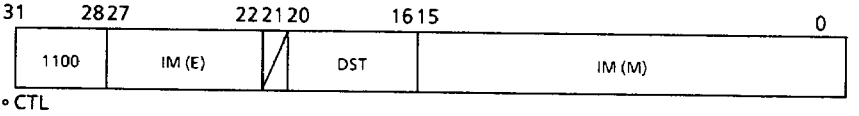
- SFT



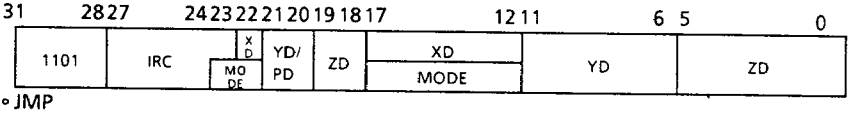
- MOVE



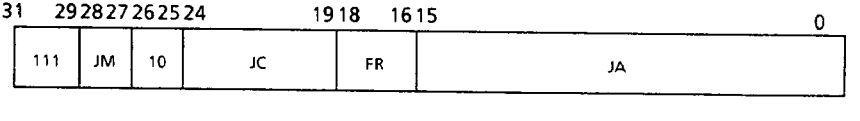
- LIT



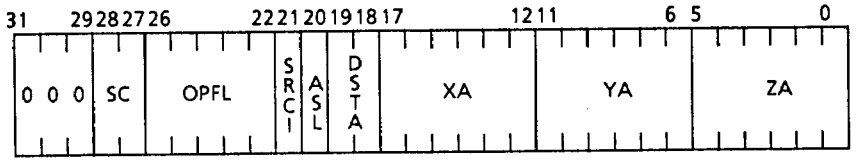
- CTL



- JMP



IFLA Instruction



SC field (Sequence control)

MNEMONIC	D28	D27	OPERATION
	0	0	Cycle: PC + 1 → PC
PUSH	0	1	Push Next Address & Sequence
RET	1	0	Return
LOOP	1	1	Loop

OPFL field (Operation of Floating-point Arithmetic)

MNEMONIC		D26	D25	D24	D23	D22	OPERATION
FLD	PH	0	0	0	0	0	PH through
FCHS	PH	0	0	0	0	1	– PH
FSUB	PH, A0	0	0	0	1	0	PH – A0
FSUB	PH, A1	0	0	0	1	1	PH – A1
FADD	PH, A0	0	0	1	0	0	PH + A0
FADD	PH, A1	0	0	1	0	1	PH + A1
FSUB	A0, PH	0	0	1	1	0	A0 – PH
FSUB	A1, PH	0	0	1	1	1	A1 – PH
FLD	A0	0	1	0	0	0	A0 through
FLD	A1	0	1	0	0	1	A1 through
FCHS	A0	0	1	0	1	0	– A0
FCHS	A1	0	1	0	1	1	– A1
FADD	YR, A0	0	1	1	0	0	YR + A0
FADD	YR, A1	0	1	1	0	1	YR + A1
FSUB	YR, A0	0	1	1	1	0	YR – A0
FSUB	YR, A1	0	1	1	1	1	YR – A1
FLD	XR	1	0	0	0	0	XR through
FCHS	XR	1	0	0	0	1	– XR
FABS	XR	1	0	0	1	0	|XR|
TRNSXL		1	0	0	1	1	Fix→Float
FADD	XR, A0	1	0	1	0	0	XR + A0
FADD	XR, A1	1	0	1	0	1	XR + A1
FSUB	YR, XR	1	0	1	1	0	YR – XR
FSUB	XR, YR	1	0	1	1	1	XR – YR
FSUB	XR, A0	1	1	0	0	0	XR – A0
FSUB	XR, A1	1	1	0	0	1	XR – A1
FADD	XR, YR	1	1	0	1	0	XR + YR
TRNSLX		1	1	0	1	1	Float→Fix
FABS	A0	1	1	1	0	0	|A0|
FABS	A1	1	1	1	0	1	|A1|
FLD	YR	1	1	1	1	0	YR through
FCHS	YR	1	1	1	1	1	– YR

SRCI field (Source IBUS for I type)

MNEMONIC	D21	SELECT REGISTER
PH	0	PRODUCT REGISTER HIGH
ALU	1	FALU OUTPUT (PSEUDO REGISTER)

ASL field (Accumulator Select)

MNEMONIC	D20	SELECT ACCUMULATOR
A0	0	ACCUMULATOR 0
A1	1	ACCUMULATOR 1

DSTA field (Destination IBUS for A type)

MNEMONIC	D20	D20	SELECT REGISTER
	0	0	NON SELECT
ZR	0	1	Z REGISTER
YR	1	0	Y REGISTER
OUTR	1	1	OUTPUT REGISTER

XA field (RAMX Read Address)

D17	D16	D15	D14	D13	D12	HEX
0	0	0	0	0	0	00
			⎛			⎛
			⎝			⎝
1	1	1	1	1	1	3F

YA field (RAMY Read Address)

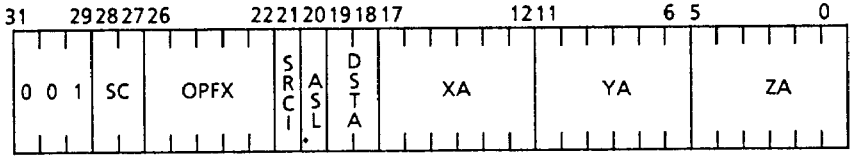
D11	D10	D09	D08	D07	D06	HEX
0	0	0	0	0	0	00
			⎛			⎛
			⎝			⎝
1	1	1	1	1	1	3F

ZA field (RAMXY Write Address)

D05	D04	D03	D02	D01	D00	HEX
0	0	0	0	0	0	00
1	1	1	1	1	1	3F

Writing to RAMXY is disabled if field value is 3F.

IFLX Instruction

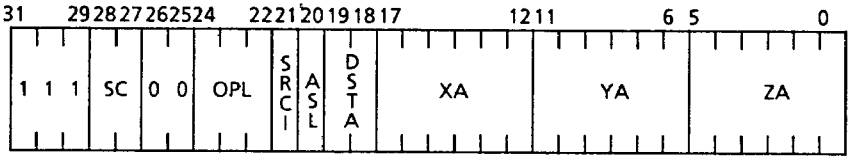


OPFX field (Operation of Fixed-point Arithmetic)

MNEMONIC	D26	D25	D24	D23	D22	OPERATION
SUB A0, PH	0	0	0	0	0	A0 – PH
SUB A1, PH	0	0	0	0	1	A1 – PH
ADD A0, PH	0	0	0	1	0	A0 + PH
ADD A1, PH	0	0	0	1	1	A1 + PH
SUBC A0, PH	0	0	1	0	0	A0 – PH – c
SUBC A1, PH	0	0	1	0	1	A1 – PH – c
ADDC A0, PH	0	0	1	1	0	A0 + PH + c
ADDC A1, PH	0	0	1	1	1	A1 + PH + c
AND XR, YR	0	1	0	0	0	XR AND YR
OR XR, YR	0	1	0	0	1	XR OR YR
XOR XR, YR	0	1	0	1	0	XR XOR YR
NOT YR	0	1	0	1	1	NOT YR
SUB A0, PL	0	1	1	0	0	A0 – PL
SUB A1, PL	0	1	1	0	1	A1 – PL
ADD A0, PL	0	1	1	1	0	A0 + PL
ADD A1, PL	0	1	1	1	1	A1 + PL
SUB XR, A0	1	0	0	0	0	XR – A0
SUB XR, A1	1	0	0	0	1	XR – A1
ADD XR, A0	1	0	0	1	0	XR + A0
ADD XR, A1	1	0	0	1	1	XR + A1
SUBC XR, A0	1	0	1	0	0	XR – A0 – c
SUBC XR, A1	1	0	1	0	1	XR – A1 – c
ADDC XR, A0	1	0	1	1	0	XR + A0 + c
ADDC XR, A1	1	0	1	1	1	XR + A1 + c
SUB XR, YR	1	1	0	0	0	XR – YR
SUBC XR, YR	1	1	0	0	1	XR – YR – c
ADD XR, YR	1	1	0	1	0	XR + YR
ADDC XR, YR	1	1	0	1	1	XR + YR + c
SUB YR, A0	1	1	1	0	0	YR – A0
SUB YR, A1	1	1	1	0	1	YR – A1
ADD YR, A0	1	1	1	1	0	YR + A0
ADD YR, A1	1	1	1	1	1	YR + A1

C: Carry generated by previous operation

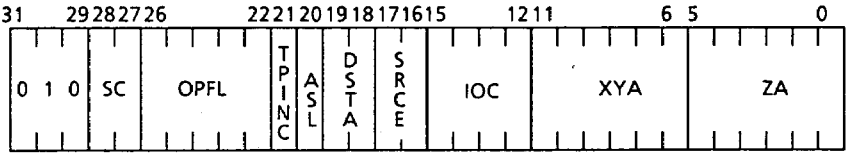
ILO Instruction



OPL field (Logical Operation)

MNEMONIC		D24	D23	D22	OPERATION
AND	XR, A0	0	0	0	XR AND A0
AND	YR, A0	0	0	1	YR AND A0
OR	XR, A0	0	1	0	XR OR A0
OR	YR, A0	0	1	1	YR OR A0
XOR	XR, A0	1	0	0	XR XOR A0
XOR	YR, A0	1	0	1	YR XOR A0
NOT	XR	1	1	0	NOT XR
NOT	A0	1	1	1	NOT A0

EFLA Instruction



TPINC field (Table Pointer Increment)

MNEMONIC	D21	OPERATION
	0	NO OPERATION
TBP + +	1	INCREMENT TBP

SRCE field (Source IBUS for Etype)

MNEMONIC	D17	D16	SELECT REGISTER
PH	0	0	PRODUCT REGISTER HIGH
ALU	0	1	FALU OUTPUT (PSEDO REGISTER)
INR	1	0	INPUT REGISTER
TBL	1	1	TABLE OUTPUT (PSEDO REGISTER)

IOC field (Input/Output Control)

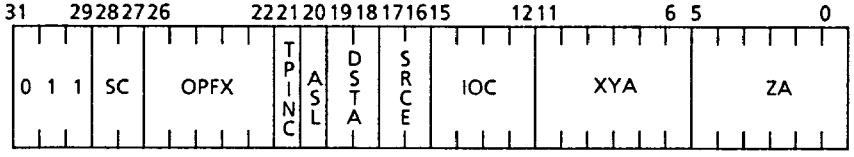
MNEMONIC	D15	D14	D13	D12	OPERATION
	0	0	0	0	NO OPERATION
	0	0	0	1	NO OPERATION
IN IO	0	0	1	0	I/O READ *
OUT IO	0	0	1	1	I/O WRITE *
IN [DP0]	0	1	0	0	((DP0)) → INR
OUT [DP0]	0	1	0	1	(OUTR) → (DP0)
IN [DP1]	0	1	1	0	((DP1)) → INR
OUT [DP1]	0	1	1	1	(OUTR) → (DP1)
IN [DP0 + +]	1	0	0	0	((DP0)) → INR, (DP0) + 1 → DP0
OUT [DP0 + +]	1	0	0	1	(OUTR) → (DP0), (DP0) + 1 → DP0
IN [DP0 – –]	1	0	1	0	((DP0)) → INR, (DP0) – 1 → DP0
OUT [DP0 – –]	1	0	1	1	(OUTR) → (DP0), (DP0) – 1 → DP0
IN [DP1 + +]	1	1	0	0	((DP1)) → INR, (DP1) + 1 → DP1
OUT [DP1 + +]	1	1	0	1	(OUTR) → (DP1), (DP1) + 1 → DP1
IN [DP1 – –]	1	1	1	0	((DP1)) → INR, (DP1) – 1 → DP1
OUT [DP1 – –]	1	1	1	1	(OUTR) → (DP1), (DP1) – 1 → DP1

* DP0 contents are passed to A_{15} - A_0. The IORQ signal must be active at this time.

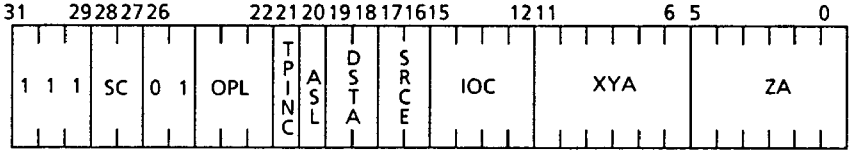
XYA field (RAMXY Read Address)

D11	D10	D09	D08	D07	D06	HEX
0	0	0	0	0	0	00
			≀			≀
1	1	1	1	1	1	3F

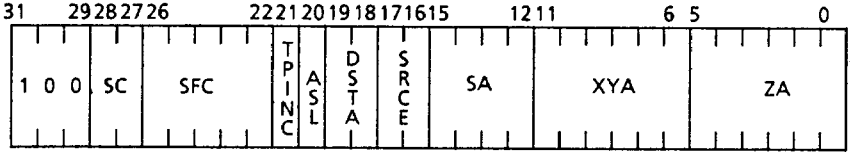
EFLX Instruction



ELO Instruction



SFT Instriction



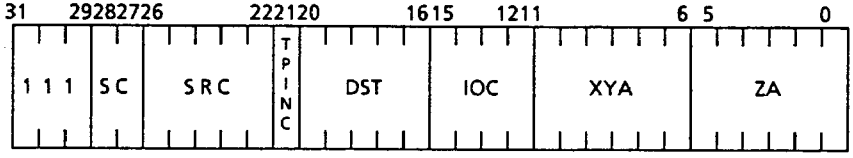
SFC field (Shift Control)

MNEMONIC	D26	D25	D24	D23	D22	OPERATION
SAL XR, imm	0	0	0	0	0	Left arithmetic shift (fixed literal value)
SAL YR, imm	0	0	0	0	1	
SAL A0, imm	0	0	0	1	0	
SAL A1, imm	0	0	0	1	1	
FSCALL XR, imm	0	0	1	0	0	Floating-point scale (left shift based on literal value)
FSCALL YR, imm	0	0	1	0	1	
FSCALL A0, imm	0	0	1	1	0	
FSCALL A1, imm	0	0	1	1	1	
SAL XR, SCR	0	1	0	0	0	Left arithmetic shift (fixed register value)
SAL YR, SCR	0	1	0	0	1	
SAL A0, SCR	0	1	0	1	0	
SAL A1, SCR	0	1	0	1	1	
FSCALL XR, SCR	0	1	1	0	0	Floating-point scale (left shift based on register value)
FSCALL YR, SCR	0	1	1	0	1	
FSCALL A0, SCR	0	1	1	1	0	
FSCALL A1, SCR	0	1	1	1	1	
SAR XR, imm	1	0	0	0	0	Right arithmetic shift (fixed literal value)
SAR YR, imm	1	0	0	0	1	
SAR A0, imm	1	0	0	1	0	
SAR A1, imm	1	0	0	1	1	
FSCALR XR, imm	1	0	1	0	0	Floating-point scale (right shift based on literal value)
FSCALR YR, imm	1	0	1	0	1	
FSCALR A0, imm	1	0	1	1	0	
FSCALR A1, imm	1	0	1	1	1	
SAR XR, SCR	1	1	0	0	0	Right arithmetic shift (fixed register value)
SAR YR, SCR	1	1	0	0	1	
SAR A0, SCR	1	1	0	1	0	
SAR A1, SCR	1	1	0	1	1	
FSCALR XR, SCR	1	1	1	0	0	Floating-point scale (right shift based on register value)
FSCALR YR, SCR	1	1	1	0	1	
FSCALR A0, SCR	1	1	1	1	0	
FSCALR A1, SCR	1	1	1	1	1	

SA field (Shift Amouont)

D15	D14	D13	D12	HEX
0	0	0	0	0
		⌇		⌇
1	1	1	1	F

MOVE Instruction



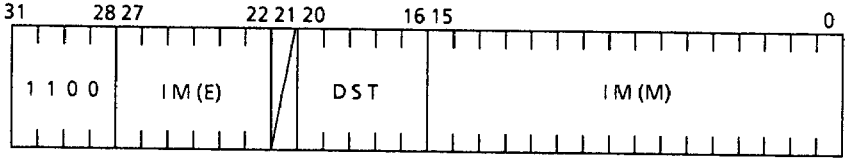
SRC field (Source IBUS)

MNEMONIC	D26	D25	D24	D23	D23	SELECT REGISTER
PH	0	0	0	0	0	PRODUCT REGISTER HIGH
ALU	0	0	0	0	1	FALU OUTPUT (PSEUDO REGISTER)
INR	0	0	0	1	0	INPUT REGISTER
TBL	0	0	0	1	1	TABLE OUTPUT (PSEUDO REGISTER)
IXR	0	0	1	0	0	INDEX REGISTER X
IYR	0	0	1	0	1	INDEX REGISTER Y
IZR	0	0	1	1	0	INDEX REGISTER Z
MR	0	0	1	1	1	MODE REGISTER
TBP	0	1	0	0	0	TABLE POINTER
DP0	0	1	0	0	1	DATA POINTER 0
DP1	0	1	0	1	0	DATA POINTER 1
LC0	0	1	0	1	1	LOOP COUNTER 0
SCR	0	1	1	0	0	SHIFT COUNT REGISTER
SR	0	1	1	0	1	STATUS REGISTER
CR	0	1	1	1	0	CONTROL REGISTER
ZR	0	1	1	1	1	Z REGISTER
MJR	1	0	0	0	0	MAP JUMP REGISTER
IFR	1	0	0	0	1	INPUT FLAG REGISTER
OFR	1	0	0	1	0	OUTPUT FLAG REGISTER
LC1	1	0	0	1	1	LOOP COUNTER 1
A0	1	0	1	0	0	ACCUMULATOR 0
A1	1	0	1	0	1	ACCUMULATOR 1
WCCR	1	0	1	1	0	WAIT CYCLE COUNT REGISTER
PR	1	0	1	1	1	PAGE REGISTER
OUTR	1	1	0	1	1	OUTPUT REGISTER

DST field (Destination IBUS)

MNEMONIC	D20	D19	D18	D17	D16	SELECT REGISTER
	0	0	0	0	0	NON SELECT
LC1	0	0	0	0	1	LOOP COUNTER 1
MJR	0	0	0	1	0	MAP JUMP REGISTER
OFR	0	0	0	1	1	OUTPUT REGISTER
IXR	0	0	1	0	0	INDEX REGISTER X
IYR	0	0	1	0	1	INDEX REGISTER Y
IZR	0	0	1	1	0	INDEX REGISTER Z
MR	0	0	1	1	1	MODE REGISTER
TBP	0	1	0	0	0	TABLE POINTER
DP0	0	1	0	0	1	DATA POINTER 0
DP1	0	1	0	1	0	DATA POINTER 1
LC0	0	1	0	1	1	LOOP COUNTER 0
SCR	0	1	1	0	0	SHIFT COUNT REGISTER (D3~D4)
SR	0	1	1	0	1	STATUS REGISTER
CR	0	1	1	1	0	CONTROL REGISTER
SCR6	0	1	1	1	1	SHIFT COUNT REGISTER (D21-D16)
INR	1	0	0	0	1	INPUT REGISTER
A0	1	0	0	1	0	ACCUMULATOR 0
ZR	1	0	1	0	0	Z REGISTER
A1	1	0	1	1	0	ACCUMULATOR 1
YR	1	1	0	0	0	Y REGISTER
WCCR	1	1	0	1	0	WAIT CYCLE COUNT REGISTER
OUTR	1	1	1	0	0	OUTPUT REGISTER
PR	1	1	1	1	0	PAGE REGISTER

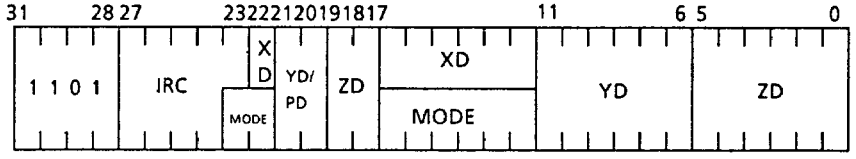
LIT Instruction



IM field (Immediate Data)

IM (EXPONENT)						IM (MANTISSA)														HEX
D 27	D 26	D 25	D 24	D 23	D 22	D 15	D 14	D 13	D 12	D 11	D 10	D 09	D 08	D 07	D 06	D 05	D 04	D 03	D 02	D 01	D 00	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000000
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFFF

CTL Instruction



IRC field (Index Registers Control)

MNEMONIC	D27	D26	D25	D24	D23	OPERATION
SET IXR, XD SET IYR, YD SET IZR, ZD	0	0	0	0	0	IXR ← XD IYR ← YD IZR ← ZD
SET IXR, XD	0	0	0	0	1	IXR ← XD
SET IYR, YD	0	0	0	1	0	IYR ← YD
SET IZR, ZD	0	0	0	1	1	IZR ← ZD
MODFY IXR, XD MODFY IYR, YD MODFY IZR, ZD	0	0	1	0	0	IXR ← IXR + XD IYR ← IYR + YD IZR ← IZR + ZD
MODFY IXR, XD	0	0	1	0	1	IXR ← IXR + XD
MODFY IYR, YD	0	0	1	1	0	IYR ← IYR + YD
MODFY IZR, ZD	0	0	1	1	1	IZR ← IZR + ZD
SET IYR, YD SET IZR, ZD	0	1	0	1	0	IYR ← YD IZR ← ZD
MODFY IYR, YD MODFY IZR, ZD	0	1	1	0	0	IYR ← IYR + YD IZR ← IZR + ZD
SET PWDN	0	1	1	1	1	Set power down mode
SET MR, MODE SET PR, PD	1	0	0	0		MR ← MODE PR ← PD
SET IYR, YD SET IZR, ZD SET MR, MODE	1	0	0	1		IYR ← YD IZR ← ZD MR ← MODE
MODFY IYR, YD MODFY IZR, ZD SET MR, MODE	1	0	1	0		IYR ← IYR + YD IZR ← IZR + ZD MR ← MODE
SET PR, PD	1	0	1	1		PR ← PD

(Cont'd)

MNEMONIC	D27	D26	D25	D24	D23	OPERATION
SET SAVE	1	1	0	1	0	Set register save mode
RESET SAVE	1	1	0	1	1	Reset register save mode
SET NINH	1	1	1	0	0	Set normalization inhibition mode
RESET NINH	1	1	1	0	1	Reset normalization inhibition mode
SET RECV	1	1	1	1	0	Set register recover mode
RESET RECV	1	1	1	1	1	Reset register recover mode

XD field (IXR Data)

D22	D17	D16	D15	D14	D13	D12	HEX
0	0	0	0	0	0	0	00
			≀				≀
1	1	1	1	1	1	1	7F

YD field (IYR Data)

D21	D20	D11	D10	D09	D08	D07	D06	HEX
0	0	0	0	0	0	0	0	00
				≀				≀
1	1	1	1	1	1	1	1	FF

ZD field (IZR Data)

D19	D18	D05	D04	D03	D02	D01	D00	HEX
0	0	0	0	0	0	0	0	00
			≀					≀
1	1	1	1	1	1	1	1	FF

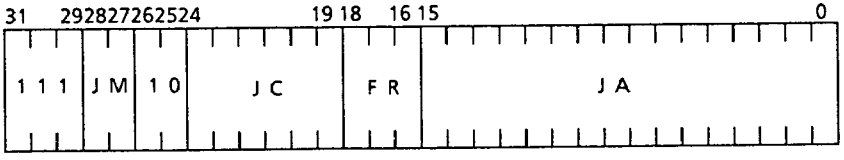
MODE field (MR Data)

D23	D22	D17	D16	D15	D14	D13	D12	HEX
0	0	0	0	0	0	0	0	00
			≀					≀
1	1	1	1	1	1	1	1	FF

PD field (PR Data)

D21	D20	HEX
0	0	00
≀		≀
1	1	03

JMP Instruction



JM field (Jump Mode Control)

MNEMONIC	D28	D27	OPERATION
Jcc	0	0	CONDITION JUMP
CALL	0	1	SUBROUTINE CALL
JSRcc			CONDITION SUBROUTINE CALL
Jcc [MJR]	1	0	CONDITION INDIRECT JUMP
CALL [MJR]	1	1	INDIRECT SUBROUTINE CALL
JSRcc [MJR]			CONDITION INDIRECT SUBROUTINE CALL

JC field (Jump Condition)

MNEMONIC	D24	D23	D22	D21	D20	D19	OPERATION
T	0	0	0	0	0	0	ALWAYS TRUE
F	1	0	0	0	0	0	ALWAYS FALSE
MI	0	0	0	0	0	1	MINUS
PL	1	0	0	0	0	1	PLUS
Z22	0	0	0	0	1	0	ZERO 22-BIT
NZ22	1	0	0	0	1	0	NOT ZERO 22-BIT
Z	0	0	0	0	1	1	ZERO
NZ	1	0	0	0	1	1	NOT ZERO
OV	0	0	0	1	0	0	OVERFLOW
NOV	1	0	0	1	0	0	NOT OVERFLOW
CS	0	0	0	1	0	1	CARRY SET
CC	1	0	0	1	0	1	CARRY CLEAR
MV	0	0	0	1	1	0	MANTISSA OVERFLOW
NMV	1	0	0	1	1	0	NOT MANTISSA OVERFLOW
EV	0	0	0	1	1	1	EXPONENT OVERFLOW
NEV	1	0	0	1	1	1	NOT EXPONENT OVERFLOW
EU	0	0	1	0	0	0	EXPONENT UNDERFLOW
NEU	1	0	1	0	0	0	NOT EXPONENT UNDERFLOW
IR	0	0	1	0	0	1	INPUT READY
NIR	1	0	1	0	0	1	NOT INPUT READY
OR	0	0	1	0	1	0	OUTPUT READY
NOR	1	0	1	0	1	0	NOT OUTPUT READY
EVU	0	0	1	0	1	1	EXPONENT OVERFLOW OR UNDERFLOW
NEVU	1	0	1	0	1	1	NOT EXPONENT OVERFLOW OR UNDERFLOW
IF0S	0	0	1	1	0	0	IF0 FLAG SET
IF0C	1	0	1	1	0	0	IF0 FLAG CLEAR
IF1S	0	0	1	1	0	1	IF1 FLAG SET
IF1C	1	0	1	1	0	1	IF1 FLAG CLEAR
C1Z	1	0	1	1	1	0	LOOP COUNTER 1 ZERO
C1NZ	0	0	1	1	1	0	LOOP COUNTER 1 NOT ZERO
BE	0	0	1	1	1	1	BUS ENABLE
NBE	1	0	1	1	1	1	BUS NOT ENABLE
GE	1	1	0	0	0	0	GREATER THAN OR EQUAL
LT	0	1	0	0	0	0	LESS THAN
XRPL	1	1	0	0	1	1	XR PLUS
XRMI	0	1	0	0	1	1	XR MINUS
YRPL	1	1	0	1	0	0	YR PLUS
YRMI	0	1	0	1	0	0	YR MINUS
A0PL	1	1	0	1	0	1	A0 PLUS
A0MI	0	1	0	1	0	1	A0 MINUS
A1PL	1	1	0	1	1	0	A1 PLUS
A1MI	0	1	0	1	1	0	A1 MINUS

FR FIELD (Flag Reseg)

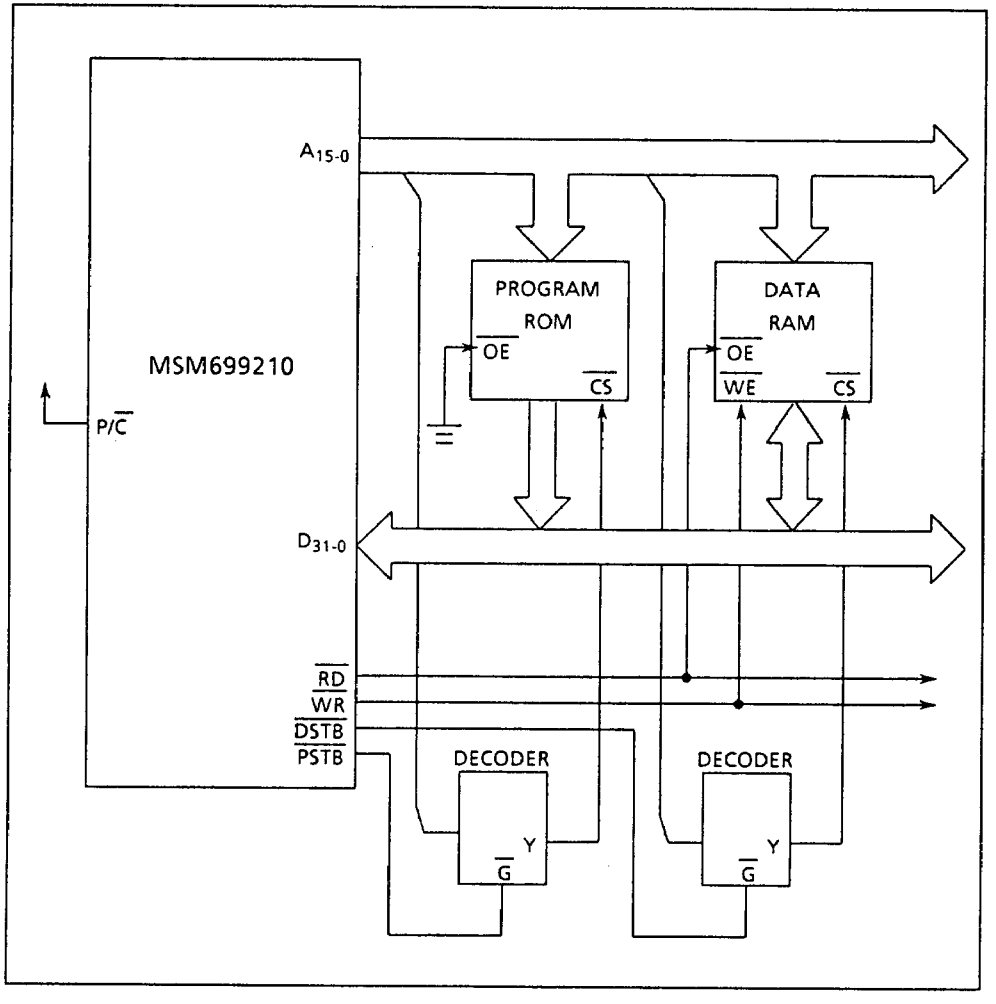
MNEMONIC	D18	D17	D16	OPERATION
–	0	0	0	NOP
FLGRST OVF	0	0	1	MV, EV, AND EU RESET
FLGRST MV	0	1	0	MV RESET
FLGRST EV	0	1	1	EV RESET
FLGRST EU	1	0	0	EU RESET
FLGRST EVU	1	0	1	EV AND EU RESET
FLGRST IF0	1	1	0	IF0 RESET
FLGRST IF1	1	1	1	IF1 RESET

A field (Jump Address)

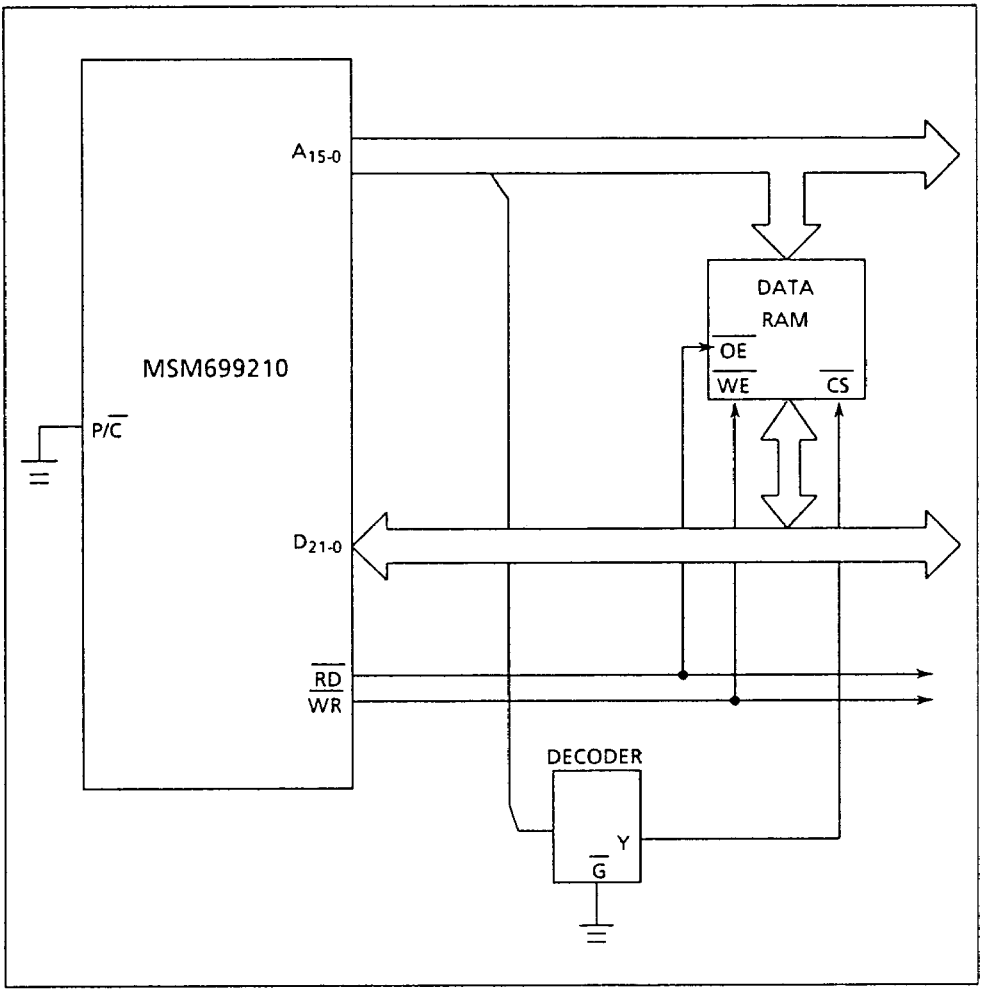
D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00	HEX
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
								⎛								⎛
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF

4. SYSTEM CONFIGURATION

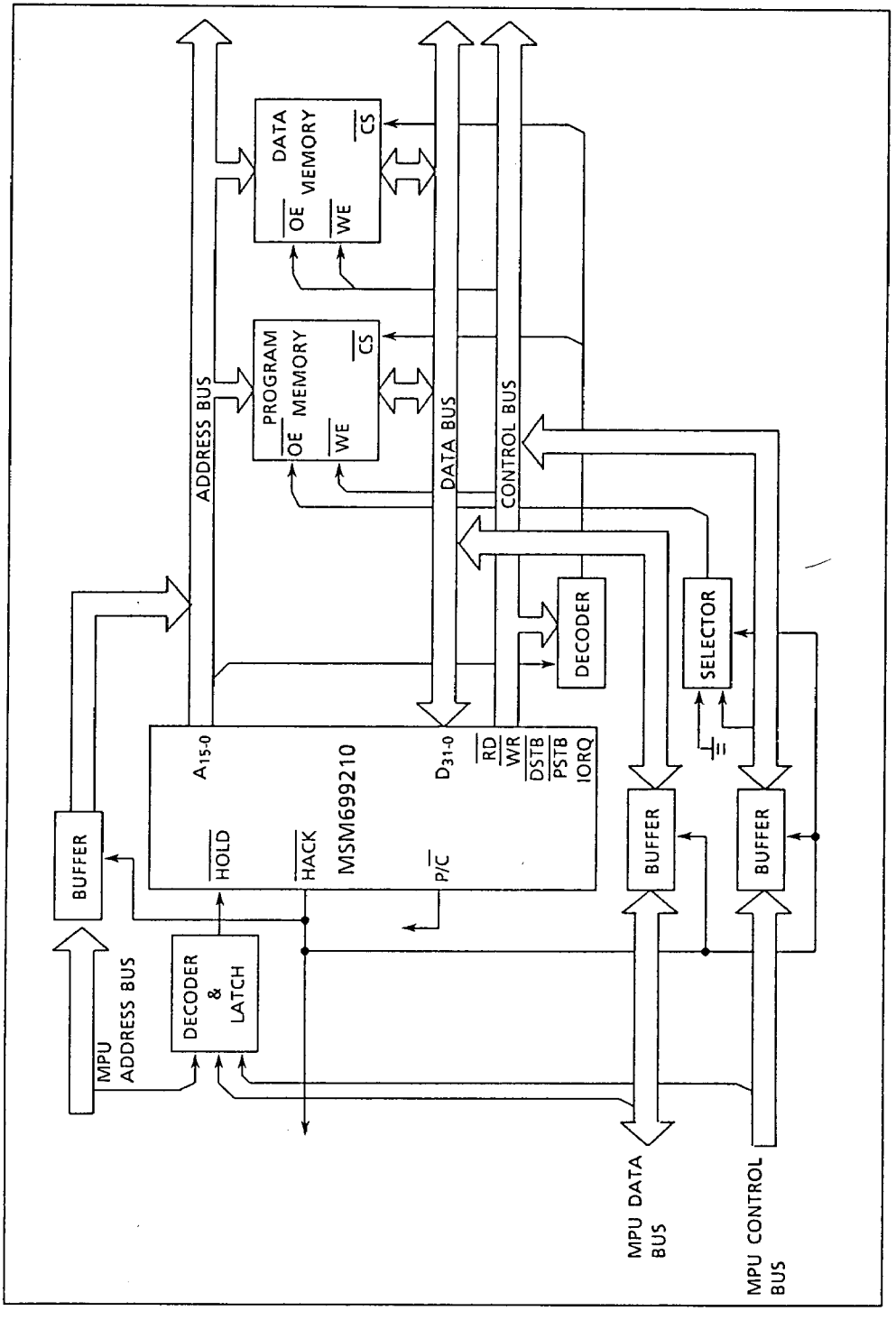
4.1 Memory Interface (Processor mode)



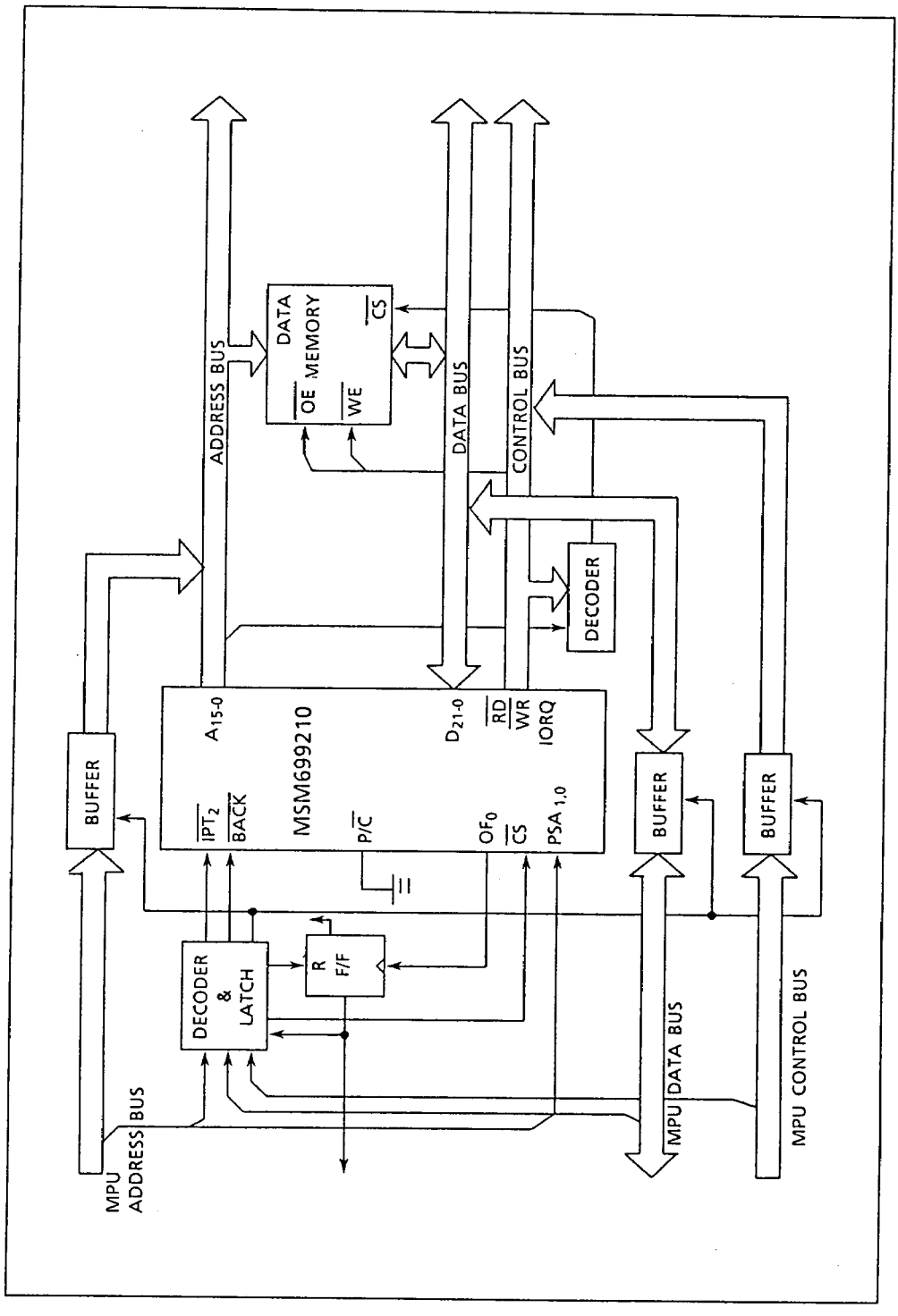
4.2 Memory Interface (Controller mode)



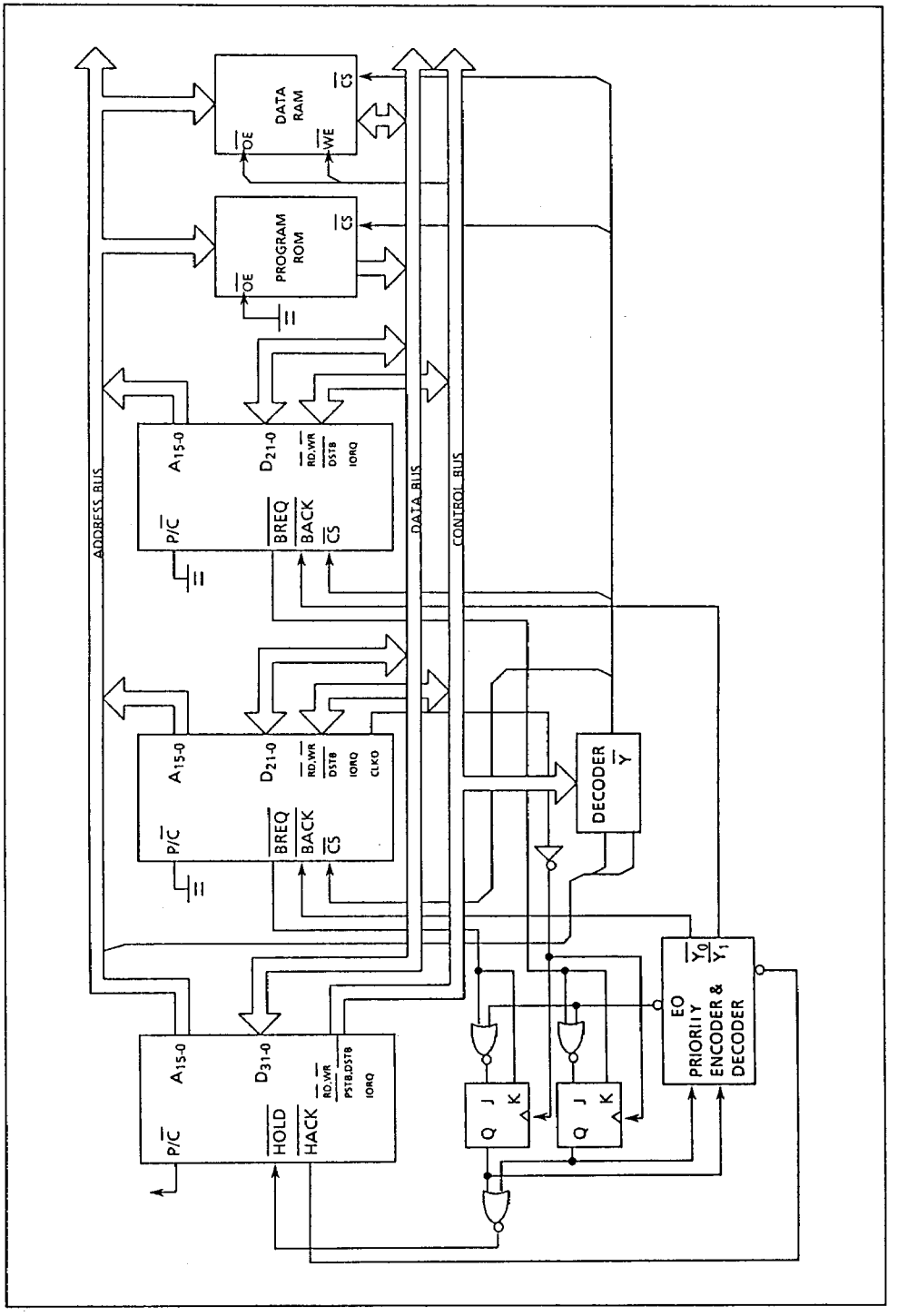
4.3 MPU Interface (Processor mode)



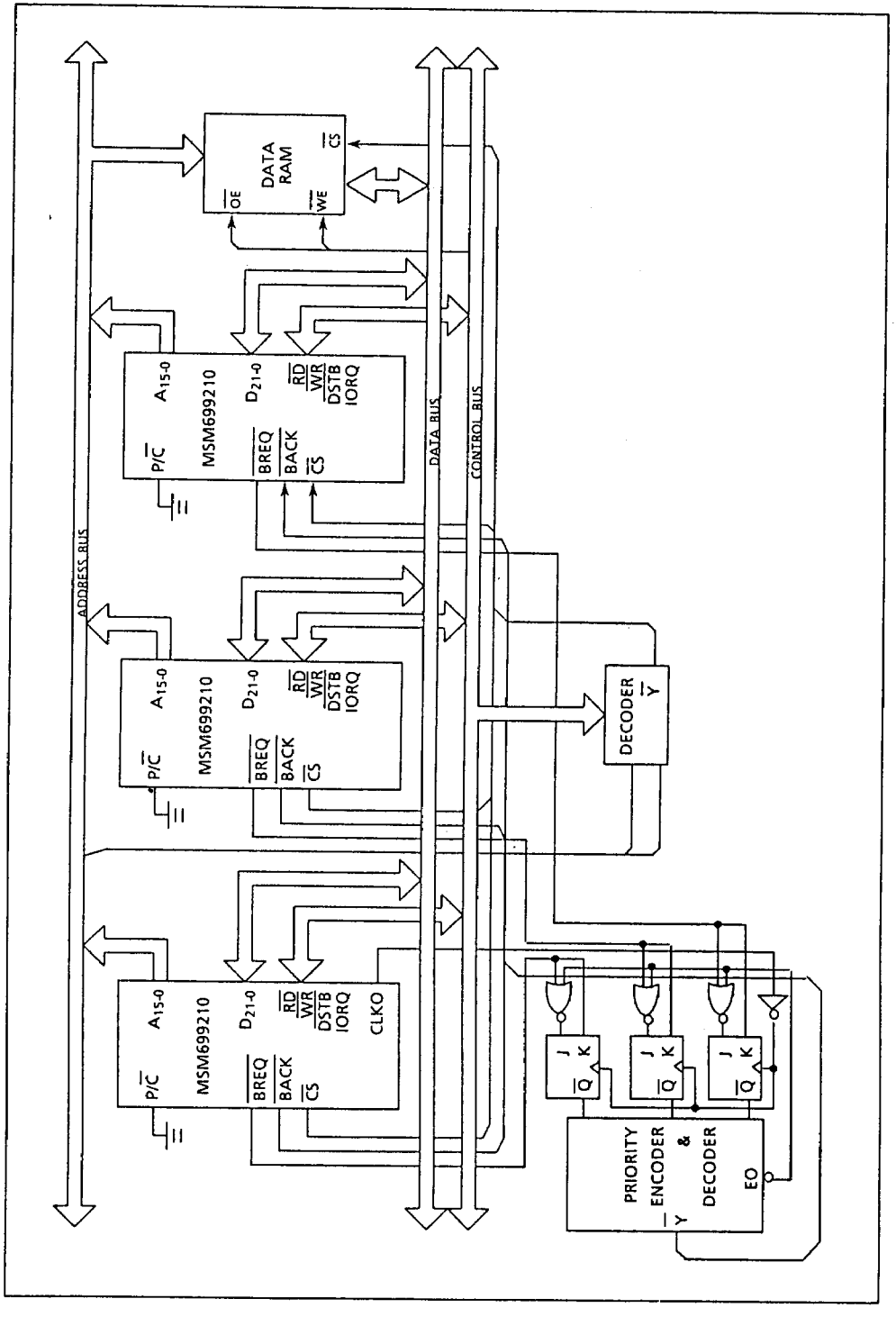
4.4 MPU Interface (Controller mode)



4.5 Multi-DSP Interface (Processor mode and controller mode)



4.6 Multi-DSP Interface (Controller mode)



ELECTRICAL CHARACTERISTICS

• Absolute Maximum Ratings

Parameter	Symbol	Condition	Limits	Unit
Supply Voltage	V_{CC}	GND Basis	− 0.5 ~ + 7.0	V
Input Voltage	V_{IN}		− 0.5 ~ V_{CC} + 0.5	V
Output Voltage	V_{OUT}		− 0.5 ~ V_{CC} + 0.5	V
Storage Temperature	Tstg		− 65 ~ + 150	°C
Power Dissipation	P_D	Ta = 25°C		W

• Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	4.75	5.00	5.25	V
Operating Temperature	Top	− 40	+ 25	+ 85(+ 70*)	°C
"H" Input Voltage	V_{IH}	2.2		V_{CC} + 0.3	V
"L" Input Voltage	V_{IL}	− 0.3		0.8	V

* Value in parentheses: 100 pin QFP

• DC Characteristics

(V_{CC} = 5V ± 5%, Ta = − 40 ~ + 85(+ 70*)°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Input Leak Current	I_{LI}	$0 \leqq V_{IN} \leqq V_{CC}$	− 10		10	μA
Output Leak Current	I_{LO}	$0 \leqq V_{OUT} \leqq V_{CC}$	− 10		10	μA
"H" Output Current	V_{OH}	I_{OH} = − 1.0mA	3.7			V
"L" Output Current	V_{OL}	I_{OL} = 4.2mA			0.4	V
Stand-by Supply Current at power down mode	I_{CCS}	$t_{\phi MC}$ = 25ns		4	10	mA
Operation Supply Current	I_{CCO}	$t_{\phi MC}$ = 25ns		80	110	mA

* Value in parentheses: 100 pin QFP

- Capacitance

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Input Capacitance	C_{IN}	f = 1 MHz			10	pF
Output Capacitance	C_{OUT}				20	pF

AC CHARACTERISTICS (V_{CC} = 5V ± 5%, Ta = −40~ +85(70*)°C)

* Value in parentheses: 100 pin QFP

- Clock Timing

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
$t_{\phi MC}$	MCLK cycle time		25		1000	ns
$t_{\phi MH}$	MCLK high level pulse width		10			ns
$t_{\phi ML}$	MCLK low level pulse width		10			ns
$t_{\phi Mr}$	MCLK rise time	Voltage at measurement point = 0.8V & 2.2V			5	ns
$t_{\phi Mf}$	MCLK fall time				5	ns
$t_{\phi C}$	CLKO cycle time		100		4000	ns
$t_{\phi H}$	CLKO high level pulse width		T-10			ns
$t_{\phi L}$	CLKO low level pulse width		3T-20			ns
$t_{\phi r}$	CLKO rise time	Voltage at measurement point = 0.8V & 2.2V			10	ns
$t_{\phi f}$	CLKO fall time				10	ns

Note: $T = t_{\phi c}/4$

- $\overline{\text{SYNC}}$ Timing

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{SS}	$\overline{\text{SYNC}}$ setup to MLCK ↑		10			ns
t_{SH}	$\overline{\text{SYNC}}$ hold after MCLK ↑		10			ns
t_{SSRS}	$\overline{\text{RST}}$ setup to $\overline{\text{SYNC}}$ ↑		4T			ns
t_{SHRS}	$\overline{\text{RST}}$ hold after $\overline{\text{SYNC}}$ ↑		3T			ns

• Reset Timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{DZRST}	Address & data disable delay ($\overline{RST}$ ↓)*1			60	ns
t_{ZDRST}	Address enable delay ($\overline{RST}$ ↑)*1	3T		8T	ns
t_{RSTW}	$\overline{RST}$ pulse width	6T			ns

Note: *1 "address" includes $A_{0\text{-}15}$, $\overline{PSTB}$, $\overline{DSTB}$, $\overline{RD}$, $\overline{WR}$ and IORQ.
"data" includes $D_{21\text{-}0}$.

• External Program Access Timing (Processor mode)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{APD}	$A_{15\text{-}0}$, TA valid to valid data in			4T-65	ns
t_{PSPD}	Trailing edge of $\overline{PSTB}$ to valid data in			4T-65	ns
t_{PDPS}	$D_{31\text{-}0}$ hold after $\overline{PSTB}$ ↑	0			ns
t_{PAHP}	$A_{15\text{-}0}$, TA hold after $\overline{PSTB}$ ↑	− 5			ns
t_{PAD}	$A_{15\text{-}0}$, TA output delay (CLKO ↓)			46	ns
t_{PAH}	$A_{15\text{-}0}$, TA hold after CLKO ↓	− 5			ns
t_{PSD}	$\overline{PSTB}$ output delay (CLKO ↓)			T + 15	ns
t_{PSH}	$\overline{PSTB}$ hold after CLKO ↓	− 5			ns

• External Data Access Timing (No wait cycle: processor mode & controller mode)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{DWS}	$D_{21\text{-}0}$, setup to $\overline{WR}$ ↑, $\overline{DSTB}$ ↑	2T-30			ns
t_{WDH}	$D_{21\text{-}0}$, hold after $\overline{WR}$ ↑	5			ns
t_{DAW}	$A_{15\text{-}0}$, IORQ setup to $\overline{WR}$ ↓	2T-40			ns
t_{WDA}	$A_{15\text{-}0}$, IORQ hold after $\overline{WR}$ ↑	0			ns
t_{DSW}	$\overline{DSTB}$ setup to $\overline{WR}$ ↑	3T-20			ns
t_{WW}	$\overline{WR}$ pulse width	2T-20			ns
t_{DAD}	$A_{15\text{-}0}$, IORQ output delay (CLKO ↓)			46	ns
t_{DAH}	$A_{15\text{-}0}$, IORQ hold after CLKO ↓	− 5			ns
t_{DSD}	$\overline{DSTB}$ output delay (CLKO ↓)			T + 15	ns
t_{DSH}	$\overline{DSTB}$ hold after CLKO ↓	− 5			ns
t_{DDA}	$A_{15\text{-}0}$, IORQ valid to valid data in			4T-65	ns

• External Data Access Timing (Continued)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{DDR}	Trailing edge of $\overline{RD}$ to valid data in			4T-65	ns
t_{DDS}	Trailing edge of $\overline{DSTB}$ to valid data in			4T-65	ns
t_{DRD}	D_{21-0} hold after $\overline{RD}$, $\overline{DSTB}$	0			ns
t_{RDA}	A_{15-0}, IORQ hold after $\overline{RD}$, $\overline{DSTB}$	– 5			ns
t_{RR}	$\overline{RD}$ pulse width	3T-25			ns
t_{RDD}	$\overline{RD}$ output delay (CLKO ↓)			T + 20	ns

• External Data Access Timing (Same wait cycle: processor mode & controller mode)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{DWS1}	D_{21-0} setup to $\overline{WR}$ ↑	T_W + 2T-30			ns
t_{WDH1}	D_{21-0} hold after $\overline{WR}$ ↑	0			ns
t_{DAW1}	A_{15-0}, IORQ set up to WR ↓	2T-35			ns
t_{WDA1}	A_{15-0}, IORQ hold after $\overline{WR}$ ↑	2T-20			ns
t_{DSW1}	$\overline{DSTB}$ setup to $\overline{WR}$ ↑	T_W + 3T-20			ns
t_{WW1}	$\overline{WR}$ pulse width	T_W-20			ns
t_{DDA1}	$\overline{A_{15-0}}$, IORQ valid to valid data in			T_W + 4T-65	ns
t_{DDR1}	Trailing edge of $\overline{RD}$ to valid data in			T_W + 4T-65	ns
t_{DDS1}	Trailing edge of $\overline{DSTB}$ to valid data in			T_W + 4T-65	ns
t_{RR1}	$\overline{RD}$ pulse width	T_W + 3T-25			ns

Note: T_W: Number of wait cycle x 4T

• Hold Timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{DHAL}	$\overline{HACK}$ low level output delay ($\overline{HOLD}$ ↓)	T			ns
t_{DHAH}	$\overline{HACK}$ high level output delay ($\overline{HOLD}$ ↑)	T		6T	ns
t_{HW}	$\overline{HOLD}$ pulse width	6T			ns
t_{DZHA}	Address disable delay (CLKO ↓)			30	ns
t_{ZDHA}	Address enable delay (CLKO ↓)			30	ns
t_{DHAC}	$\overline{HACK}$ output delay (CLKO ↓)			15	ns

• Read/Write Timing (Controller mode: Slave operation)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{AR}	$\overline{CS}$, $PSA_{0,1}$ set-up to $\overline{RD}$ ↓	0			ns
t_{RA}	$\overline{CS}$, $PSA_{0,1}$ hold after $\overline{RD}$ ↑	20			ns
t_{RR}	$\overline{RD}$ pulse width	50			ns
t_{RD}	D_{21-0} (data) access from $\overline{RD}$ ↓			50	ns
t_{DF}	D_{21-0} (data) float after $\overline{RD}$ ↑	0		50	ns
t_{AW}	$\overline{CS}$, $PSA_{0,1}$ setup to $\overline{WR}$ ↓	25			ns
t_{WA}	$\overline{CS}$, $PSA_{0,1}$ hold after $\overline{WR}$ ↑	20			ns
t_{WW}	$\overline{WR}$ pulse width	50			ns
t_{DW}	D_{21-0} (data) setup to $\overline{WR}$ ↓	20			ns
t_{WD}	D_{21-0} (data) hold after $\overline{WR}$ ↑	20			ns

• DMA Read/Write Timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{DRQ}	$\overline{DREQ}$ output delay (CLKO ↓)			30	ns
t_{AKQ}	$\overline{DREQ}$ output delay ($\overline{DACK}$ ↓)			8T + 30	ns
t_{AKC}	$\overline{DACK}$ set-up to $\overline{RD}/\overline{WR}$	0			ns
t_{CAK}	$\overline{DACK}$ hold after $\overline{RD}/\overline{WR}$	20			ns
t_{RW}	$\overline{RD}/\overline{WR}$ pulse width	5T			ns
t_{DC}	D_{21-0} (data) set-up to $\overline{WR}$ ↓	20			ns
t_{CD}	D_{21-0} (data) hold after $\overline{WR}$ ↑	20			ns
t_{RD}	D_{21-0} (data) access from $\overline{RD}$ ↓			50	ns
t_{DF}	D_{21-0} (data) float after $\overline{RD}$ ↑	0		50	ns
t_{DACKD}	$\overline{DACK}$ set-up to $\overline{DREQ}$ ↓	50			ns
t_{DACKW}	$\overline{DACK}$ pulse width	5T			ns

• $\overline{BREQ}$, $\overline{BACK}$ Timing

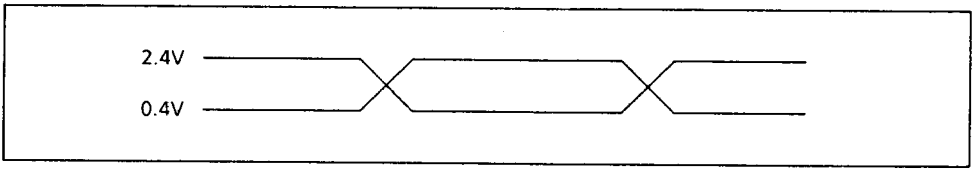
Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{BRQ}	$\overline{BREQ}$ output delay (CLKO ↓)			2T + 20	ns
t_{ZDA}	Address bus enable delay ($\overline{BACK}$ ↓)	T		5T + 30	ns
t_{ZDAC}	Address bus enable delay (CLKO ↓)			30	ns
t_{ZDB}	Data bus enable delay ($\overline{BACK}$ ↓)	T		5T + 30	ns
t_{ZDBC}	Data bus enable delay (CLKO ↓)			30	ns
t_{DZA}	Address and data bus disable delay ($\overline{BACK}$ ↑)	T		5T + 60	ns
t_{DZAC}	Address and data bus disable delay (CLKO ↓)			60	ns

Note: Address bus includes $A_{15\text{-}0}$, $\overline{RD}$, $\overline{WR}$, $\overline{DSTB}$ and IORQ

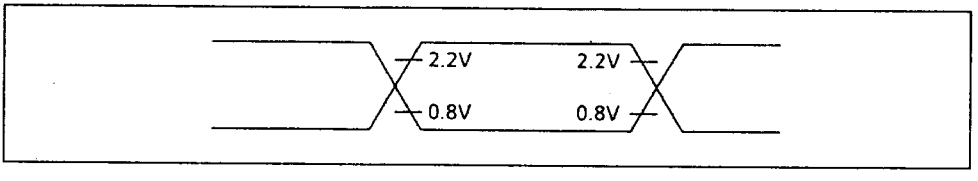
• Interrupt & Port Timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{IPW}	$\overline{IPT_{0,1,2}}$ pulse width	6T			ns
t_{IFW}	$IF_{0,1}$ high level pulse width	6T			ns
t_{OFD}	$OF_{0,1}$ output delay (CLKO ↓)			15	ns
t_{OFDR}	$OF_{0,1}$ output delay ($\overline{RST}$ ↓)	T – 10		5T + 10	ns

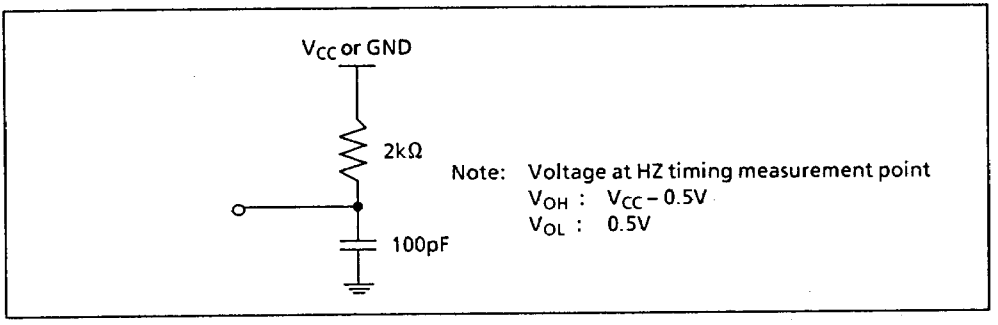
• AC Test Input Waveform



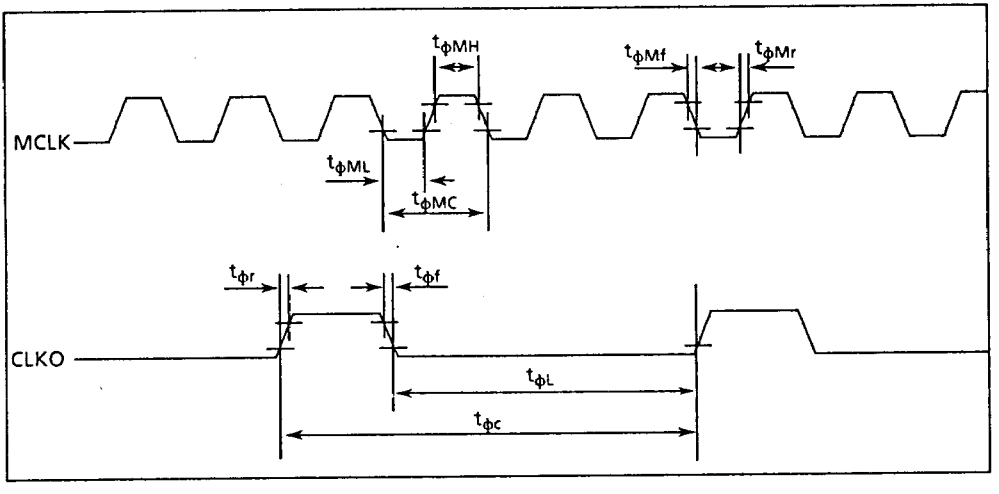
• Voltage at AC Timing Measurement Point



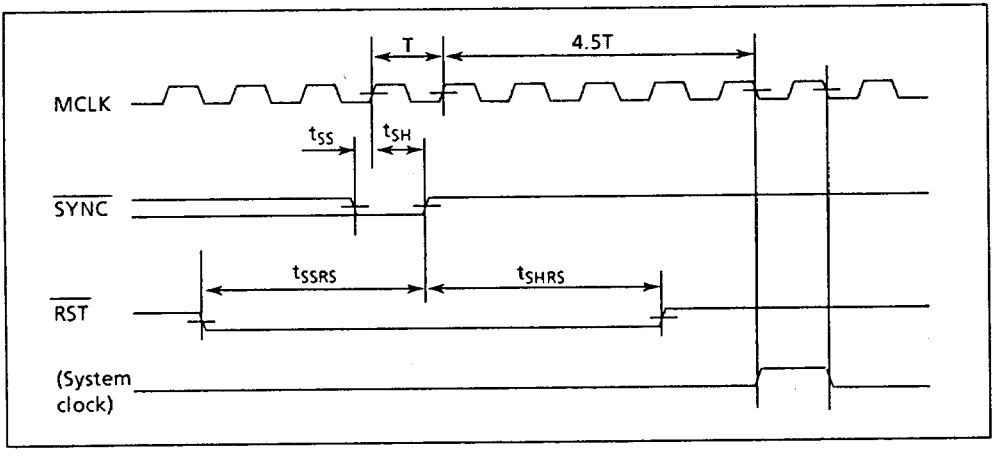
• Output HZ and DZ Test Circuit



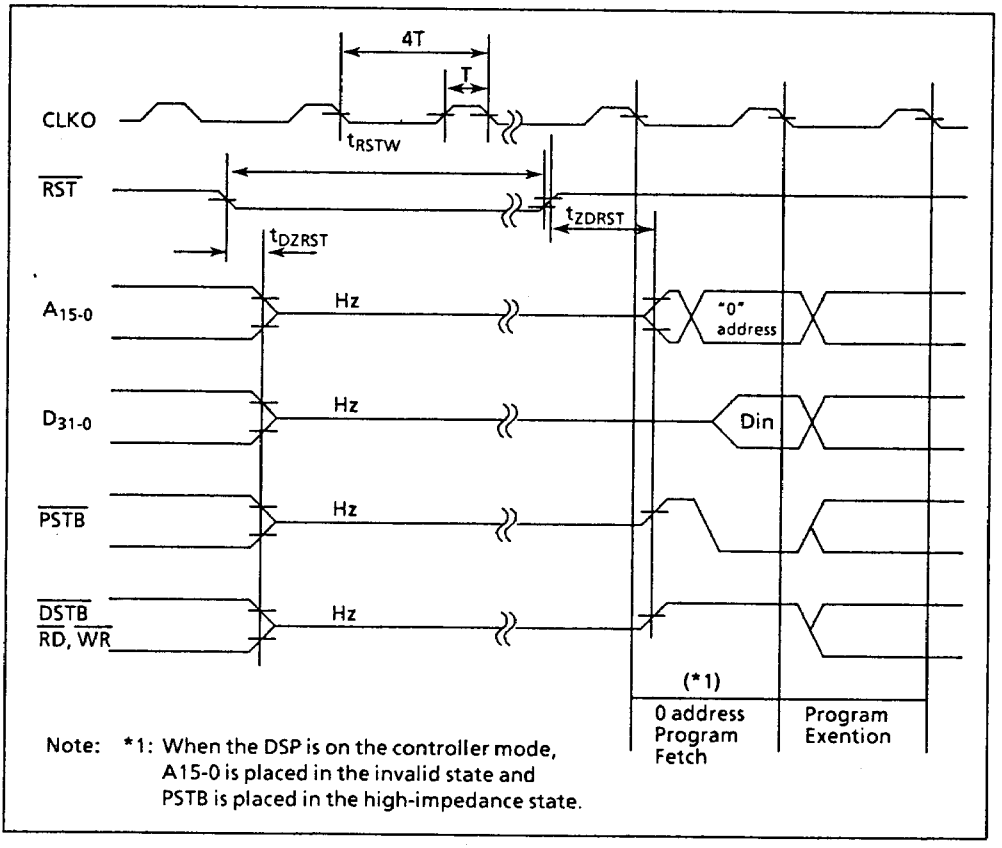
• Clock Timing



• SYNC Timing

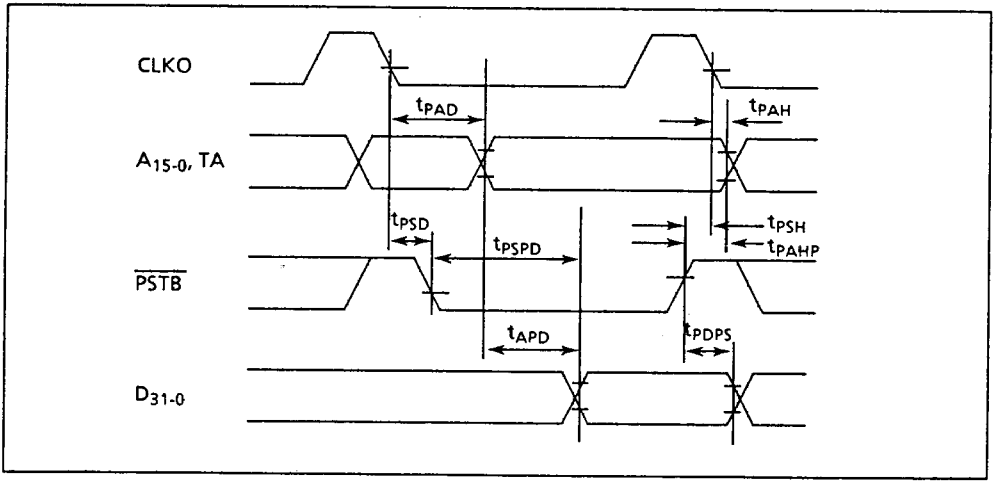


• Reset Timing

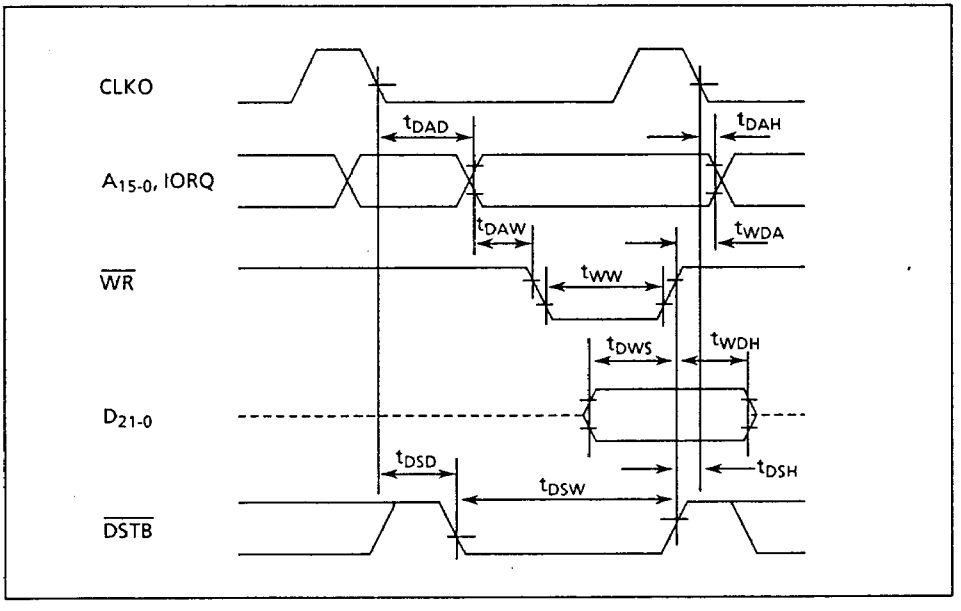


Note: *1: When the DSP is on the controller mode, A15-0 is placed in the invalid state and PSTB is placed in the high-impedance state.

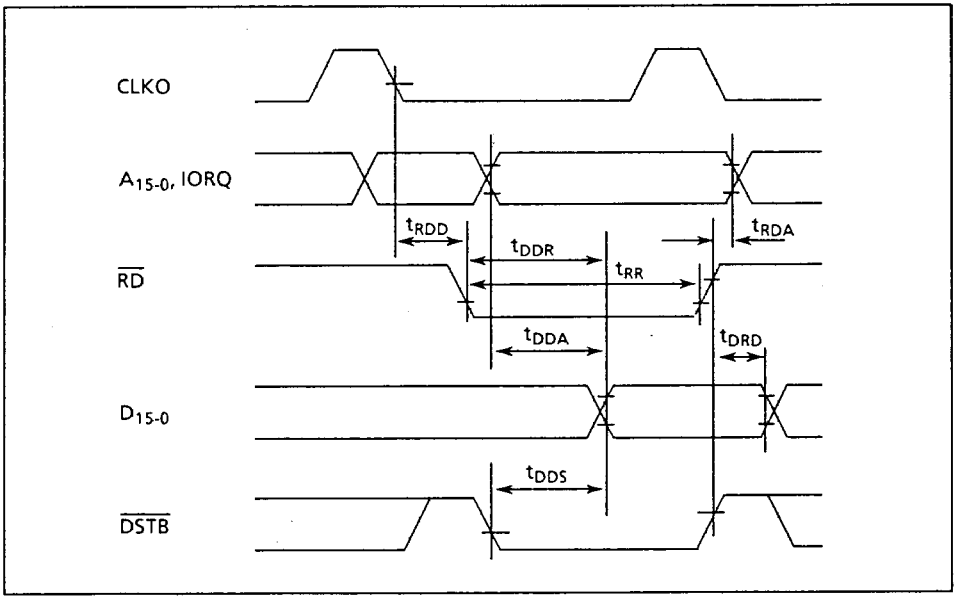
• External Program Access Timing (Processor mode)



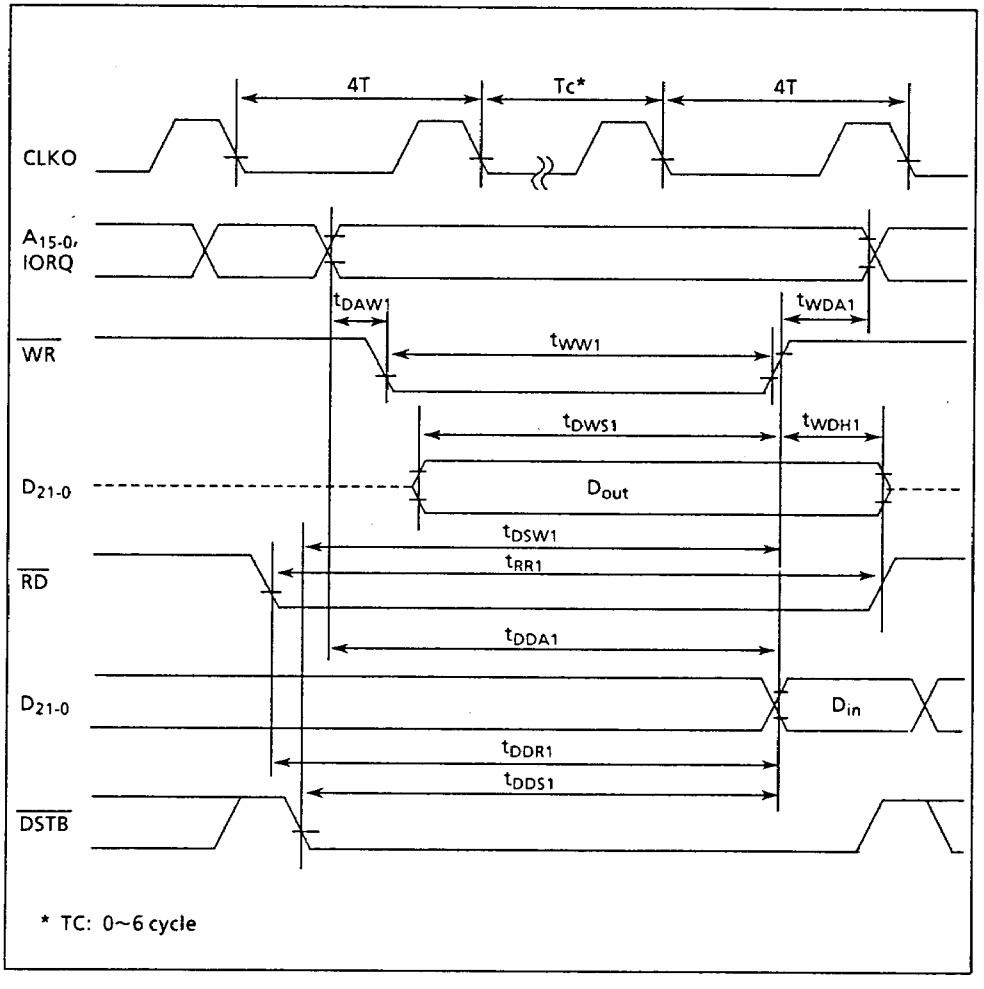
- **External Data Write Timing** (No wait cycle: processor mode & controller mode)



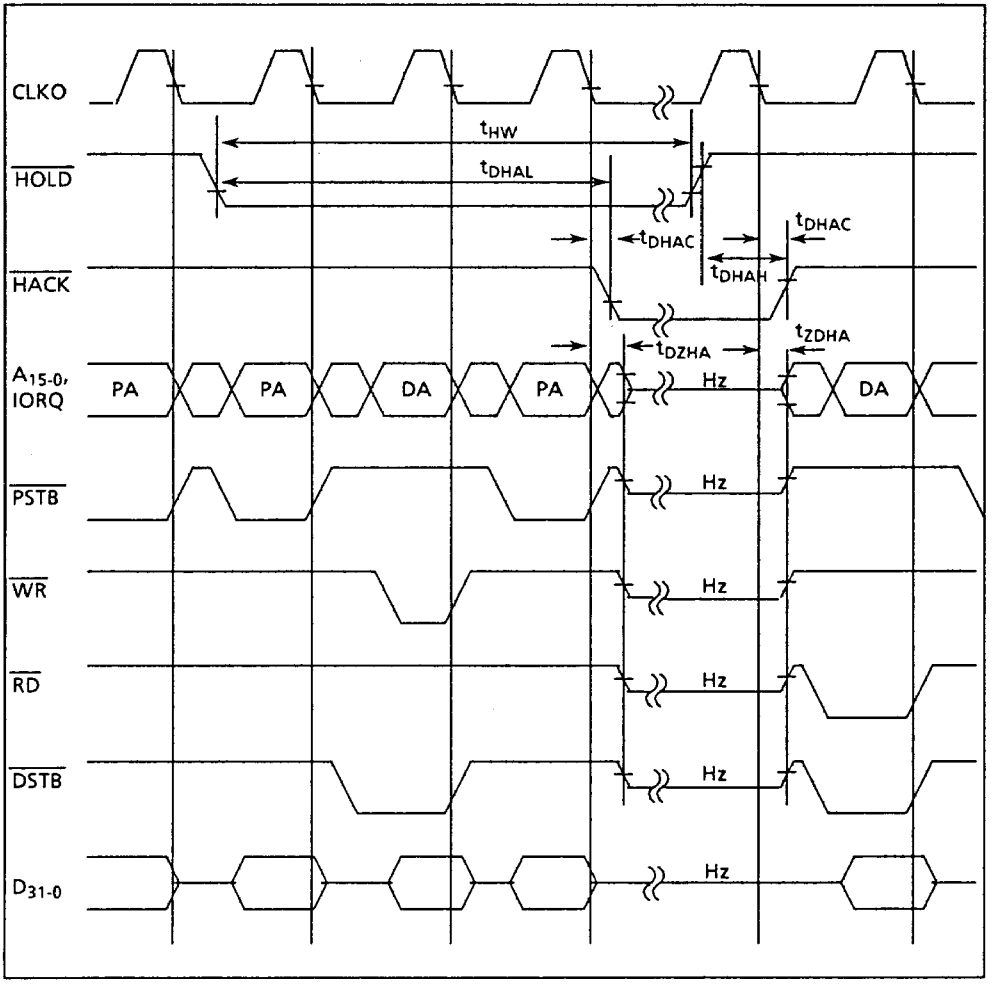
- **External Data Read Timing** (No wait cycle: processor mode & controller mode)



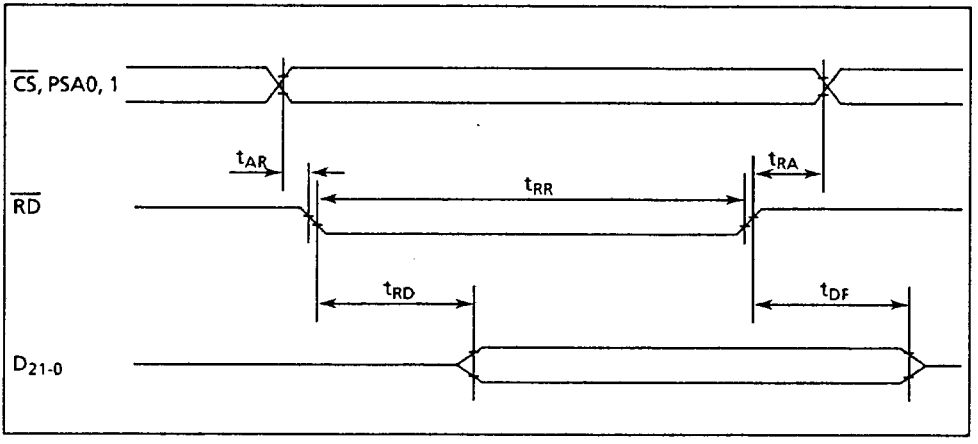
- **External Data Access Timing** (Some wait cycle: processor mode & controller mode)



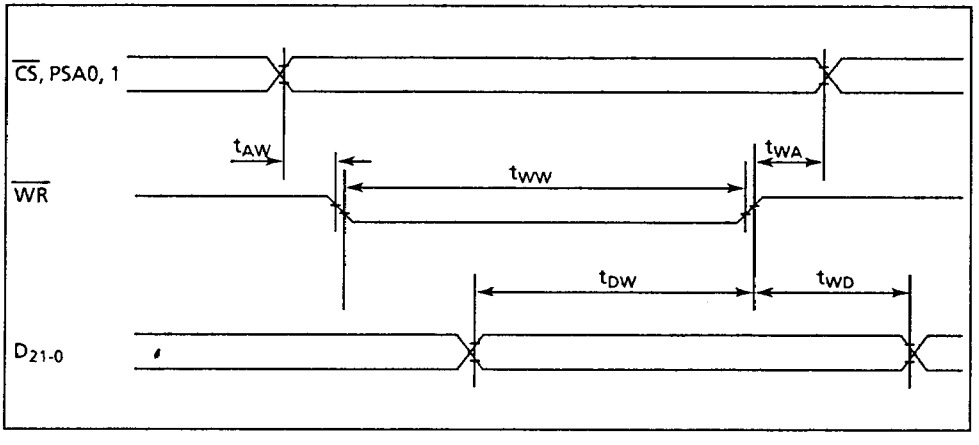
● HOLD Timing


CLKO
HOLD
HACK
A15-0, IORQ
PSTB
WR
RD
DSTB
D31-0
tHW
tDHAL
tDHAC
tDHAC
tDHAH
tDZHA
tZDHA
PA
PA
DA
PA
Hz
DA
Hz
Hz
Hz
Hz
Hz

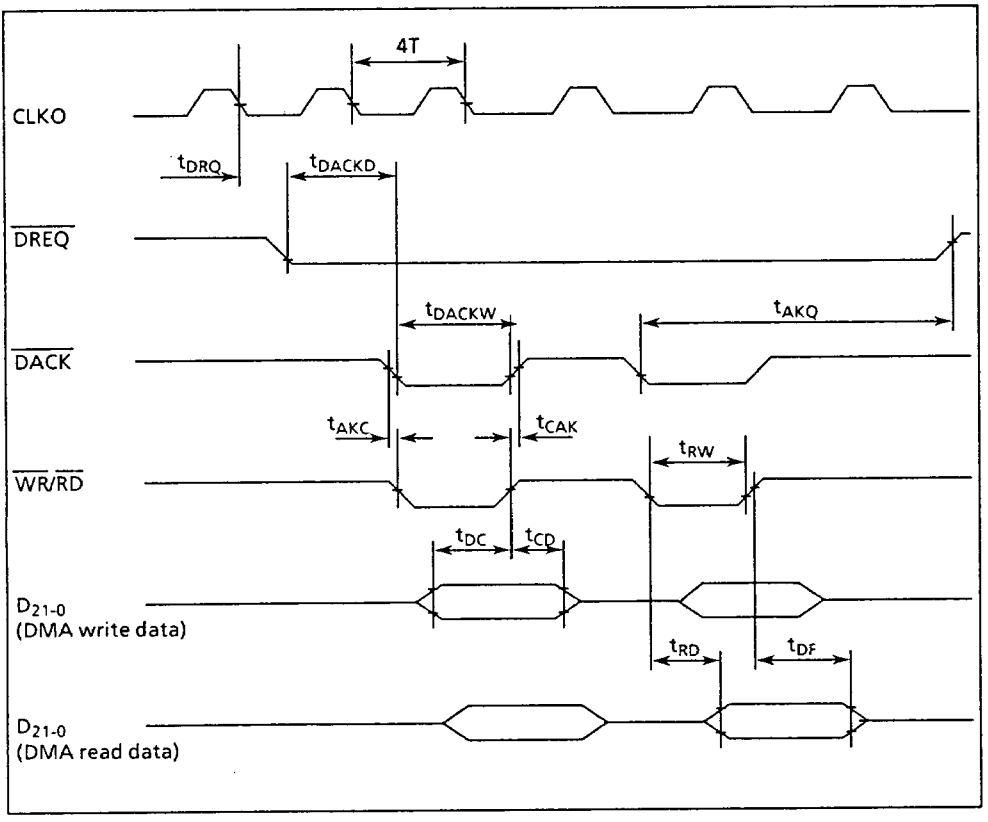
- **Read Timing** (Controller mode: Slave operation)



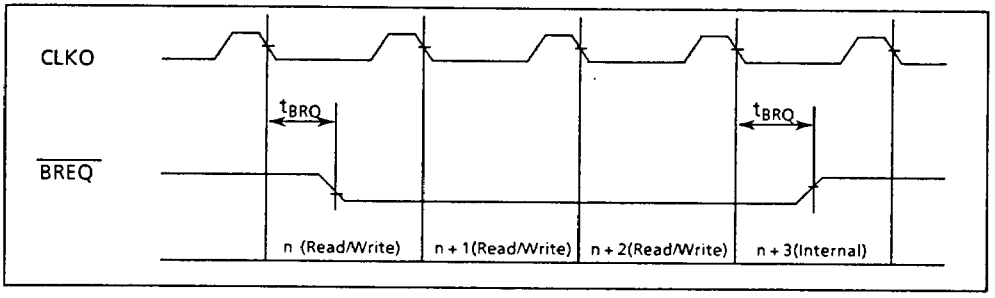
- **Write Timing** (Controller mode: Slave operation)



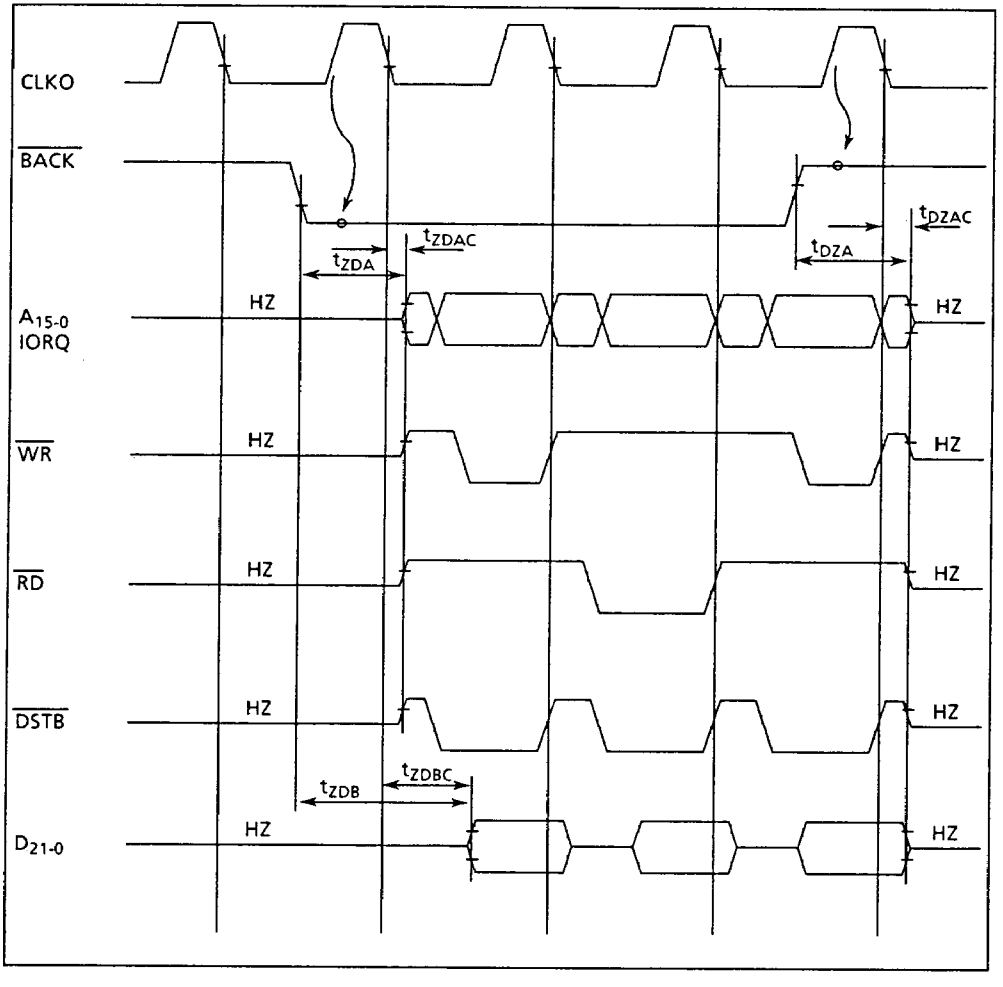
• DMA Read/Write Timing



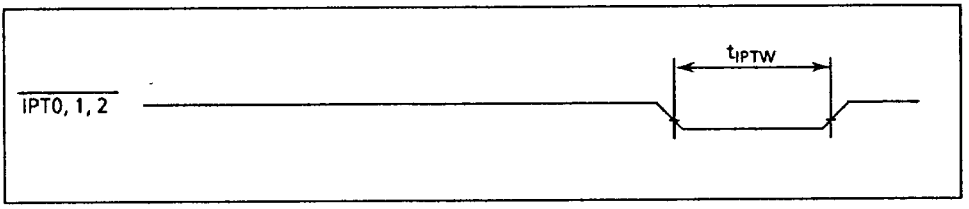
• $\overline{\text{BREQ}}$ Timing



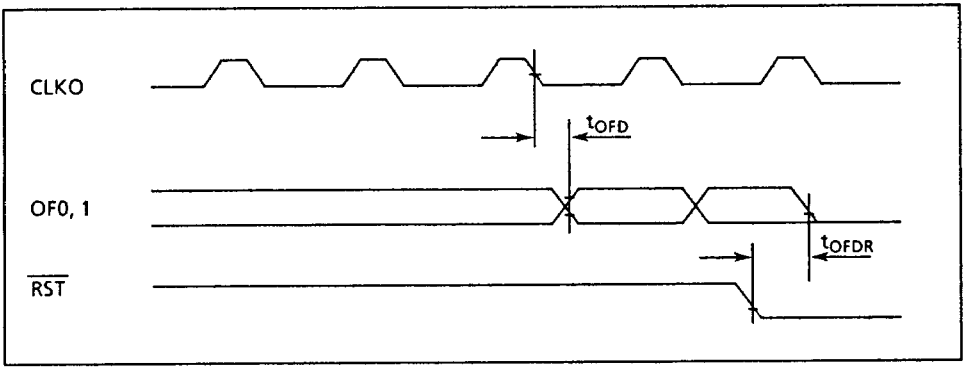
• $\overline{\text{BACK}}$ Timing



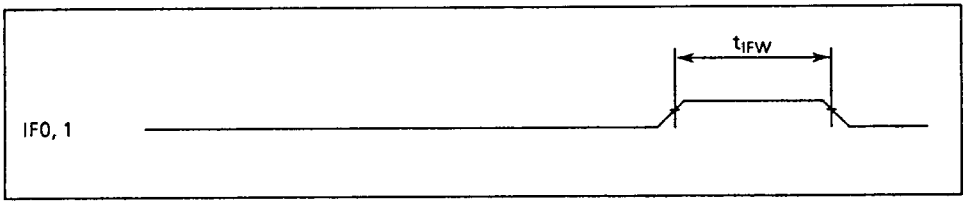
- Interrupt Timing



- Output Port Timing

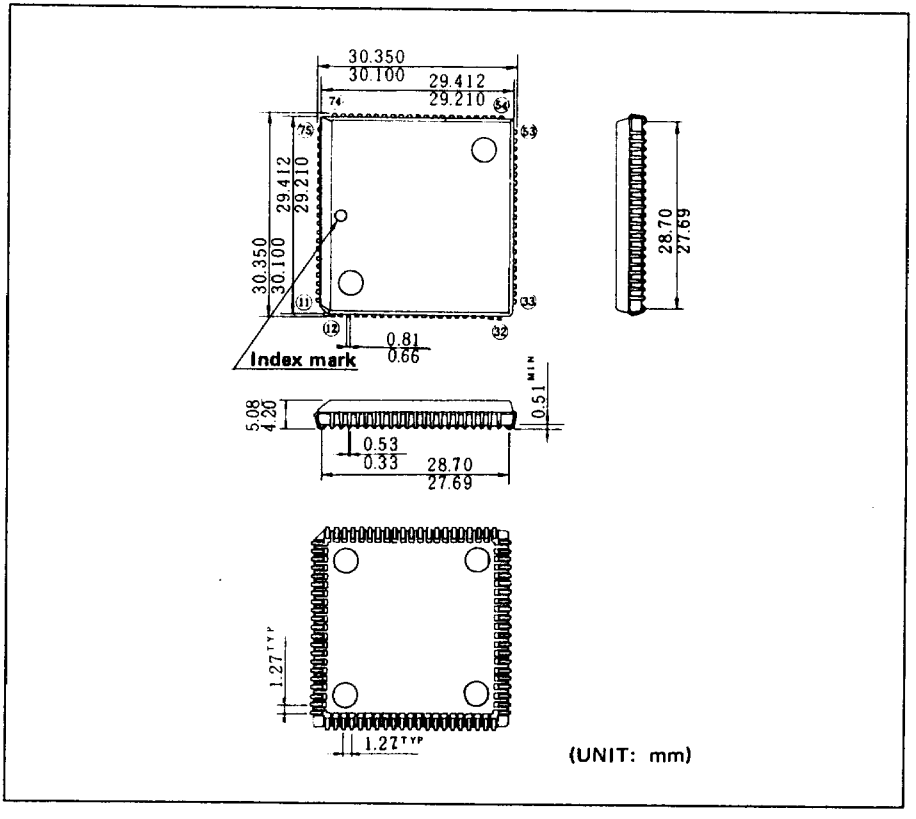


- Input Port Timing



7. PACKAGE DIMENSIONS

84 PIN PLCC



100 PIN FLAT