HD63310R

Smart Dual Port RAM (S-DPRAM)

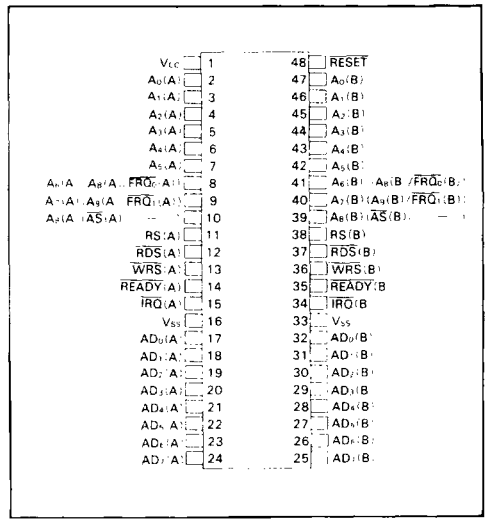
Description

Driven by the declining costs of VLSI CPUs, new multiprocessing and parallel-processing architectures are being explored to break the performance barrier of single CPU designs. A critical factor for multiprocessor performance is the choice of the communication mechanism between CPUs. A poorly designed communication scheme can destroy the potential for a multiprocessor design since message passing overhead can quickly offset the performance advantage of multiple CPUs.

An HD63310R S-DPRAM acts as a communication link between two CPUs. Thus, it replaces older, less flexible communication schemes such as parallel ports and FIFO ICs.

Programmable operating mode and bus interface tailor the S-DPRAM for a wide variety of multiprocessor system designs using industry standard MPUs.

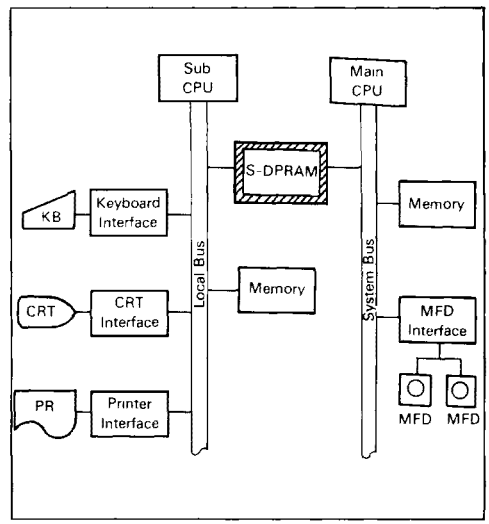
Pin Arrangement



Features

- 1024-byte data buffer with dual ports
  —Configurable as a RAM (DPRAM mode) or two FIFOs (FIFO mode)
- Asynchronous bus operation on each port
- Programmable bus interface
  —Each bus configurable as a multiplexed or non-multiplexed address and data bus.
- Built-in programmable registers
  —8 semaphore registers for multiprocessing applications, 32 user-definable registers
- Programmable outputs
  —Interrupt outputs
  —FIFO status (full, empty, etc) outputs
- High speed operation and low power dissipation 2-μm CMOS circuit
- Single + 5V power supply

System Block Diagram



Pin Description

Symbol	Pin Number	Name	I/O
V_{CC}	1	Power supply	
V_{SS}	16, 33	Ground	
$\overline{RESET}$	48	Reset	I
A_0-A_5(A) A_0-A_5(B)	2-7 47-42	Address bus	I
AD_0-AD_7(A) AD_0-AD_7(B)	17-24 32-25	Address/data bus	I/O
$A_6/A_8/\overline{FRQ_0}$(A) $A_6/A_8/\overline{FRQ_0}$(B)	8 41	Multipurpose line (1): Address bit 6/8, FIFO data request 0	I/O
$A_7/A_9/\overline{FRQ_1}$(A) $A_7/A_9/\overline{FRQ_1}$(B)	9 40	Multipurpose line (2): Address bit 7/9, FIFO data request 1	I/O
$A_8/\overline{AS}$ (A) $A_8/\overline{AS}$ (B)	10 39	Multipurpose line (3): Address bit 8, Address strobe	I
RS (A/B)	11, 38	Register select	I
$\overline{RDS}$ (A/B)	12, 37	Read strobe	I
$\overline{WRS}$ (A/B)	13, 36	Write strobe	I
$\overline{READY}$ (A/B)	14, 35	Ready	O
$\overline{IRQ}$(A/B)	15, 34	Interrupt request	O

5

Pin Function

This section describes the input and output signals. As shown in "Pin Arrangement", the S-DPRAM has two data ports (designated port A and port B). Identical interface signals are provided for each port except V_{SS}, V_{CC}, and $\overline{RESET}$ which are common to the chip. An "(A)" or a "(B)" after a signal name designates a port A or port B signal.

Power Supply (V_{CC})

V_{CC} is the +5 V ± 5% power supply.

Ground (V_{SS})

V_{SS} is the signal and power ground.

Reset ($\overline{RESET}$)

$\overline{RESET}$ places all the registers of the S-DPRAM in the initial state when asserted (refer to "Register Description" for details). This function is designated a hardware reset, as distinguished from a software reset which can be performed by a program. Active low.

Address Bus (A_0(A)−A_5(A), A_0(B)−A_5(B))

A_0-A_5 are the lower 6-bit address input pins when the I/O bus is configured as non-multiplexed. They should be connected to V_{SS} when the I/O bus is configured as multiplexed. (A CNFG register is selected by the address input 0)

During a RAM access, these lines, together with the A_6, A_7, A_8 address inputs and internal bank select bit, address one of the 1024 RAM bytes (in DPRAM mode).

Table 1. Address and Data Assignment for Each Operating Mode

Signal Name	Register Read/Write DPRAM Mode (Indirect Addressing)		DPRAM Mode (Direct Addressing)		FIFO Mode	
	Non-mult.	Mult.	Non-mult.	Mult.	Non-mult.	Mult.
Address bus A_0-A_5	A_0-A_5	Not used	A_0-A_5	Not used	A_0-A_5	Not used
Address/ data bus AD_0-AD_7	Data D_0-D_7	Address A_0-A_7 Data D_0-D_7	D_0-D_7	A_0-A_7 D_0-D_7	D_0-D_7	A_0-A_5 D_0-D_7
Multi-purpose line (1) $A_6/A_8/\overline{FRQ_0}$	Not used		A_6	A_8	$\overline{FRQ_0}$	
Multi-purpose line (2) $A_7/A_9/\overline{FRQ_1}$	Not used		A_7	A_9	$\overline{FRQ_1}$	
Multi-purpose line (3) $A_8/\overline{AS}$	Not used	$\overline{AS}$	A_8	$\overline{AS}$	Not used	$\overline{AS}$

Notes: 1. Non-mult. I/O bus is configured as non-multiplexed.
Mult. I/O bus is configured as multiplexed address and data bus.

2. A hardware reset places the S-DPRAM in DPRAM mode, and defines the I/O bus as non-multiplexed. The operation mode can be altered by setting the internal CNFG register as required.

During a register access, these lines select one of the 62 internal registers.

Address/Data Bus (AD_0(A)−AD_7(A), AD_0(B)−AD_7(B))

AD_0-AD_7 are the input/output pins for the data bus when the I/O bus is configured as non-multiplexed. They are the input pins for the lower 8-bit address, and input/output pins for data when the I/O bus is configured as multiplexed. Three-state.

Multipurpose Lines (A_6(A)/A_8(A)/$\overline{FRQ_0}$(A), A_6(B)/A_8(B)/$\overline{FRQ_0}$(B), A_7(A)/A_9(A)/$\overline{FRQ_1}$(A), A_7(B)/A_9(B)/$\overline{FRQ_1}$(B), A_8(A)/$\overline{AS}$(A), A_8(B)/$\overline{AS}$(B))

The function of the multipurpose lines varies according to the operation mode of the S-DPRAM (table 1). These functions are as follows.

A_6-A_8 or A_8, A_9 Address Bit (Upper): The A_6-A_8 upper address inputs are used together with A_0-A_5 for RAM addressing (in DPRAM mode), when the I/O bus is configured as non-multiplexed. Since only 512 bytes can be addressed by the address pins, the upper or lower 512 RAM bytes should be selected by the bank select bits (bit 0 or 1) of the CMD register.

When the I/O bus is configured as multiplexed, RAM addressing utilizes A_8 and A_9 as the input pins for the upper address bits, along with lower address bits A_0-A_7.

Address Strobe ($\overline{AS}$): $\overline{AS}$ latches the address A_0-A_7 on the address/data bus AD_0-AD_7 when the I/O bus is configured as a multiplexed bus. The address presented on AD_0-AD_7 pins is latched at the rising edge of $\overline{AS}$. Active low.

FIFO Data Request ($\overline{FRQ_0}$, $\overline{FRQ_1}$): The $\overline{FRQ_0}$/$\overline{FRQ_1}$ active low outputs represent the state of a programmable FIFO threshold: FIFO 0/1 full, empty, not full, or not empty. The operation of these signals are controlled by the internal FRC (FIFO request control) register. Active low.

Register Select (RS(A), RS(B))

RS indicates whether the address specified during the current read/write operation refers to a RAM byte or an internal register. If RS is asserted high, then the address refers to one of the internal registers. If RS is low the address refers to a RAM byte.

Read Strobe ($\overline{RDS}$(A), $\overline{RDS}$(B))

$\overline{RDS}$ selects the S-DPRAM for a read transaction. Whenever $\overline{RDS}$ is asserted low the data from the addressed internal register or RAM location is transferred to the data bus. In case of simultaneous transfer requests from both ports, the internal arbiter determines the order in which the requests are serviced at the falling edge of $\overline{RDS}$. Active low.

Write Strobe ($\overline{WRS}$(A), $\overline{WRS}$(B))

$\overline{WRS}$ selects the S-DPRAM for a write transaction. Whenever $\overline{WRS}$ is asserted low, a write access to an internal register or RAM is initiated. Write transactions are bufferred. The data is temporarily latched into the buffer register upon negation of $\overline{WRS}$. It is subsequently written into the specified internal register or RAM, using the internal clock timing. Active low.

In case of simultaneous transfer requests from both ports, the internal arbiter determines the order in which the requests are serviced. (Detailed explanations are shown in the paragraph "Arbitration (Hardware)".)

Ready ($\overline{READY}$(A), $\overline{READY}$(B))

$\overline{READY}$ reports the successful acceptance of the $\overline{RDS}$ (or $\overline{WRS}$) signal to the MPU. After its assertion, $\overline{RDS}$ (or $\overline{WRS}$) must be kept low for a minimum period specified in the bus timing characteristics in order to assure a proper completion of internal processing. Active low.

Interrupt Request ($\overline{IRQ}$ (A), $\overline{IRQ}$ (B))

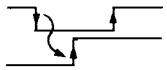
The $\overline{IRQ}$ open-drain output is asserted low when at least one bit of the AND of the corresponding bits of a port's ISRC and IEN register is set to 1. Active low, open drain.

5

Comparison between HD63310 and HD63310R

The HD63310R is an advanced R version of the HD63310, characterized by the following features: FIFO status operation error has been corrected (already reported by Hitachi Microcomputer Technical Update, TN-PSY-503A), some limitations have been eliminated, and noise resistance has been improved (noise caused by undefined read data has been removed). Table 2 shows a comparison between the HD63310 and HD63310R.

Table 2. HD63310R and HD63310

Item		HD63310RP20	HD63310P20
FCR register	In DPRAM mode	Read/write accessible	Write invalid, read as 0
	In FIFO mode	Read/write accessible	Read/write accessible
$\overline{FRQ}$ output timing characteristics	No. 32; $\overline{FRQ}$ negation delay time (at write access)	t_{FNLW} = 150 ns	t_{FNLW} = 300 ns
$\overline{FRQ}$ output timing specification ($\overline{FRQ}$ timing at write access in level mode) [type 1]		$\overline{FRQ}$ is negated at the rising edge of $\overline{WRS}$ 	$\overline{FRQ}$ is negated at the falling edge of $\overline{WRS}$ 
$\overline{FRQ}$ output timing item	$\overline{FRQ}$ timing at write access in pulse mode [type 1]	No. 35; $\overline{FRQ}$ negation delay time; t_{FNPW} = 250 ns	No. 32; $\overline{FRQ}$ negation delay time; t_{FNLW} = 300 ns
	$\overline{FRQ}$ timing at read access in pulse mode	No. 38; $\overline{FRQ}$ negation delay time; t_{FNPR} = 300 ns	No. 34; $\overline{FRQ}$ negation delay time; t_{FNLR} = 300 ns
Notes	limitations on ISRC register	Removed	Remain

Note: In FIFO operation, the HD63310R is not compatible with the HD63310 in order to improve the FIFO contention error (at enpty condition). Therefore some changes in the circuit are required around the $\overline{FRQ}$ signals.

Block Diagram

The S-DPRAM consists of 1024 bytes of static RAM and 32 bytes of register. See figure 1.

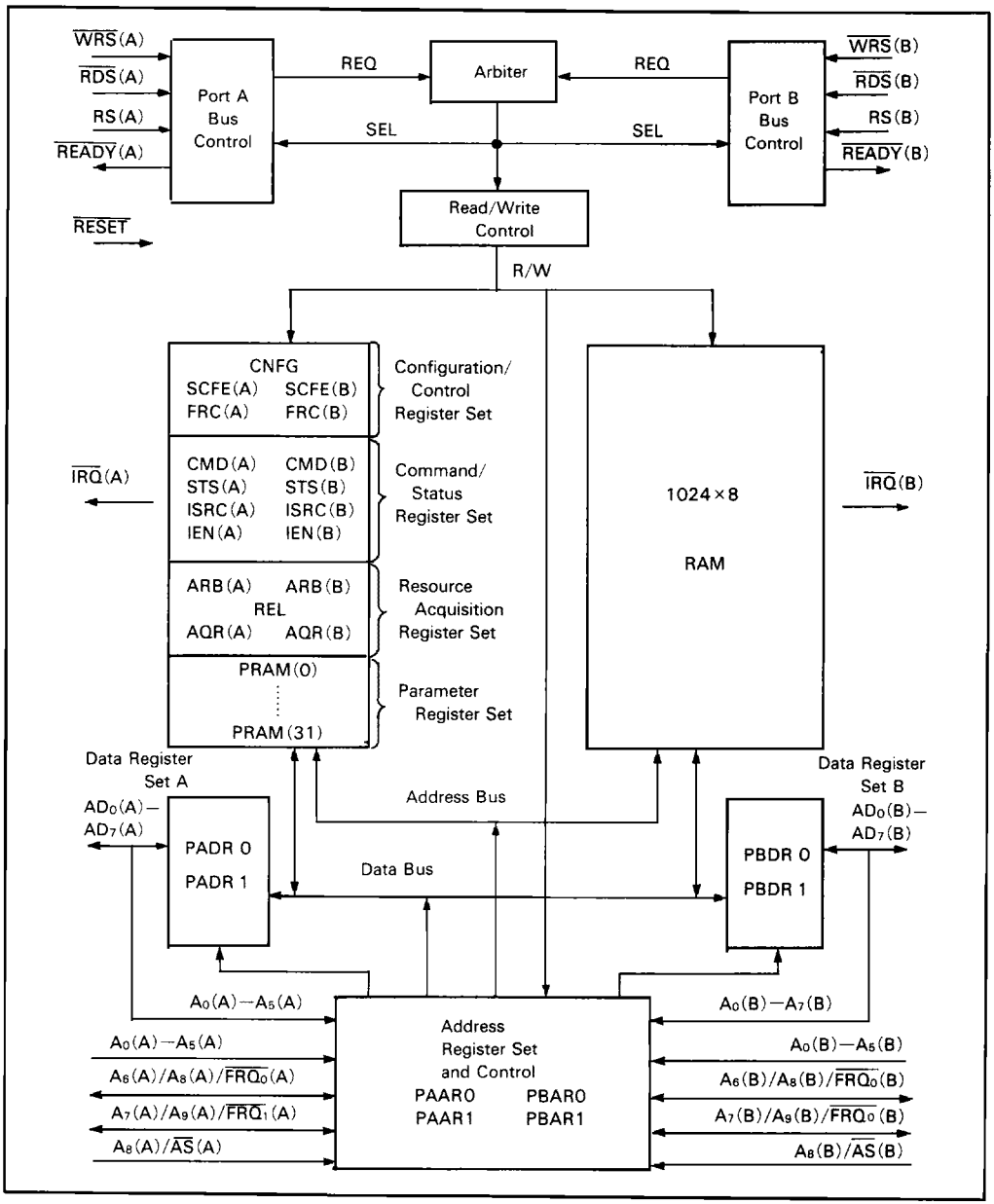


Figure 1. Block Diagram

5

Register Description

The operation of the S-DPRAM is controlled, as well as monitored, by a number of on-chip user-accessible registers. Some of these registers are shared by both ports, while others are for exclusive use of each port. The CNFG (configuration), REL (release/available), and PRAM (parameter) registers are shared by both ports, while the rest of them are provided for each port. An "(A)" or a "(B)" after a register name indicates a port A or port B register respectively. The internal register configuration is shown in figure 2, and their bit definition is in table 3.

Table 3. Internal Register: Bit Configuration and Operation

Add. #	Reg. Name	Bit Configuration 7	6	5	4	3	2	1	0	Reset	Read/Write Operation 7	6	5	4	3	2	1	0	Remarks
R00	CNFG	ABUS	BBUS	RCFG			0	F1D	F0D	00	R/W					0	R/W		Common SCR
R01	CMD	AP1	AP0	RST	0	0	0	F1FL / B1SL	F0FL / B0SL	00 (Note1)	R/W		RO /W	0	0	0	R/W		SCR
R02	IEN	ENA7	ENA6	ENA5	ENA4	ENA3	ENA2	ENA1	ENA0	00	R/W								SCR
R03	ISRC	ATT1	ATT0	RELE	FERR	F1F	F1E	F0F	F0E	00 (Note1)	R/W1C								SCR
R04	STS (A/B)	IRQ	FRQ1	FRQ0	FERR	F1F	F1E	F0F / B1EN	F0E / B0EN	01	R								
R05	ARB (A/B)	PID								00	R/WID								
R06	AQR (A/B)	OWN7	OWN6	OWN5	OWN4	OWN3	OWN2	OWN1	OWN0	00	R/W (Note 2)								
R07	REL	AVL7	AVL6	AVL5	AVL4	AVL3	AVL2	AVL1	AVL0	FF	R/W (Note 2)								Common
R08	PA/BDR0	D_7-D_0								un-defined	R/W								
R09	PA/BDR1	D_7-D_0								un-defined	R/W								
R0A	FRC (A/B)	0	FR1E	FR1T	FR1M	0	FR0E	FR0T	FR0M	00	0	R/W			0	R/W			SCR
R0B	SCFE (A/B)	CNT1	DEC1	CNT0	DEC0	F1O	F1U	F0O	F0U	00	R/W				R				SCR
P0C	PA/BAR0 (L)	A_7-A_0								00	R/W								
P0D	PA/BAR0 (H)	REG	0	0	0	0	0	A_9	A_8	00	R/W	0	0	0	0	0	R/W		
P0E	PA/BAR1 (L)	A_7-A_0								00	R/W								
P0F	PA/BAR1 (H)	REG	0	0	0	0	0	A_9	A_8	00	R/W	0	0	0	0	0	R/W		
R10 • • • R2F	PRAM	PRAM								un-defined	R/W								

Notes: 1. 00 at a hardware reset state.
At a software reset state, bits 7 and 6 are preserved, while the rest are cleared.
2. Semaphore bits to prevent access contention. OWN0 – OWN7 and AVL0 – AVL7 are related to each other.
3. R/W ——— Read/write
R ——— Read only
RO ——— Read as 0 only
SCR ——— Special Control Register
W1C ——— Write 1 to clear
WID ——— Write ID code to set if 0
0 ——— Not used; write invalid; read as 0 only
COMMON —— Shared register

When the S-DPRAM is in DPRAM mode, its indirect addressing capability allows both internal registers and the on-chip RAM to be accessed using address register PA/BAR0 or PA/BAR1 as an address pointer. The value stored in PA/BAR0 will be used as an indirect address whenever a read or write operation is performed through a port's data register PA/BDR0. Similarly, the value contained in PA/BAR1 will be used as an indirect address whenever a read or write operation is performed through a port's data register PA/BDR1.

Using indirect addressing, it is possible to access from port A the registers belonging to port B, and vice versa. The indirect address values 30-3F are reserved for accessing the opposite port's registers. Table 3 summarizes the direct and indirect address allocation for S-DPRAM's internal registers.

In FIFO mode, the indirect addressing is disabled because the address registers are utilized for FIFO control. The opposite port's registers, therefore, are not accessible in this mode.

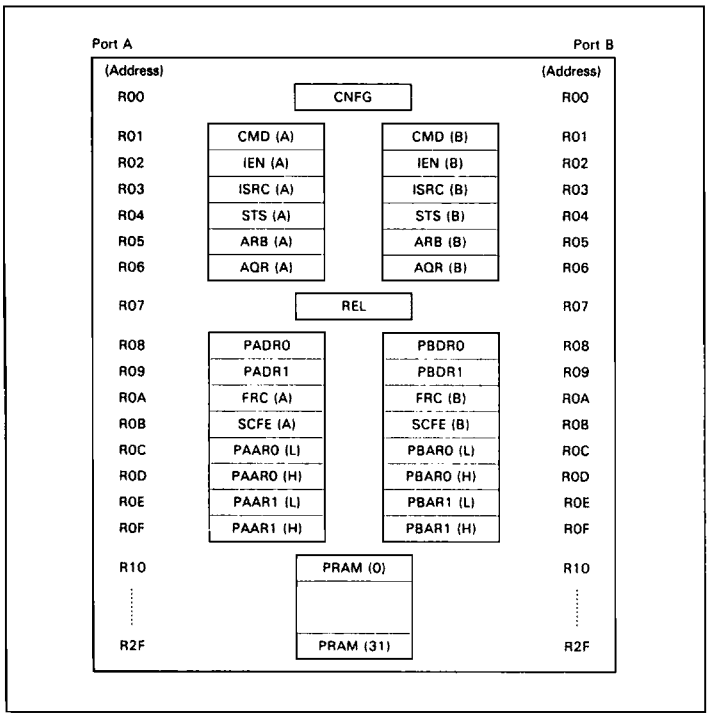


Figure 2. Internal Register Configuration

Table 4. Register Map and Address (In DPRAM mode)

Address	Port A	D	I	Port B	D	I
00 0F	Control Register (A)	○	○	Control Register (B)	○	○
10 2F	Parameter Register	○	○	Parameter Register	○	○
30 3F	Control Register (B)	×	○	Control Register (A)	×	○

Notes: 1. D: Direct addressing mode ○: Accessible
I: Indirect addressing mode ×: Not accessible
2. Address inputs $A_6 - A_9$ are not used for register access. Therefore addresses above 40 are the same as addresses 00–3F.

5

Configuration Register (CNFG)

The contents of the CNFG register (figure 3) should be defined after a reset, before any other register is accessed.

The ABUS and BBUS respectively define bus A and bus B to be either non-multiplexed or multiplexed address and data buses. The RCFG field defines the on-chip RAM either as DPRAM or as two FIFOs. The F1D/F0D define the data transfer direction between FIFO 1 and FIFO 0 and the data ports. F1D/F0D are not used (ineffective) when the S-DPRAM is set in DPRAM mode by the RCFG.

After a reset, both ports are set in DPRAM mode with the I/O bus configured as non-multiplexed (ABUS=0, BBUS=0, RCFG=000). CNFG is a shared register accessible from both ports.

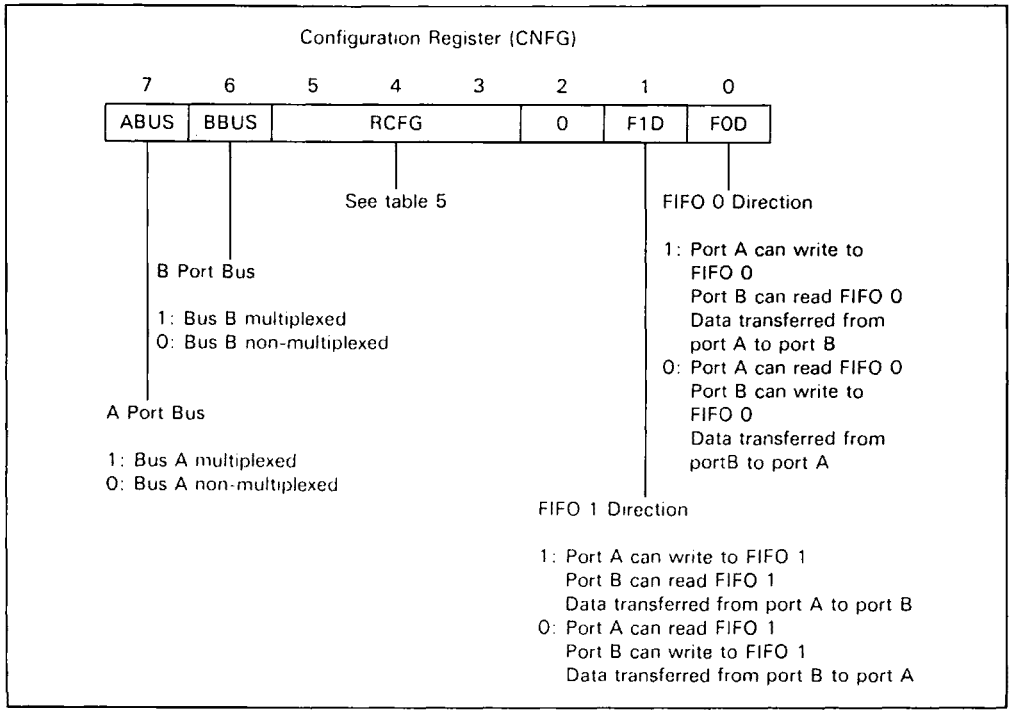


Figure 3. Configuration Register

Command Register (CMD(A), CMD(B))

Writing to a CMD register (figure 4) causes the commands corresponding to the bits set to 1 to be executed. The previous data in the CMD register does not affect the operation. For example, if the data in a bit is 1, writing a 1 to that bit still accesses the corresponding command.

The AP1 and AP0 bits perform interport communication. Writing a 1 to these bits sets bits ATT1 or ATT0 of the ISRC register of the opposite port, reporting a transaction request to that port.

The RST bit places the S-DPRAM in the software reset state. The software reset is slightly different from the hardware reset caused by asserting the RESET pin. Refer to "Reset Operation" for details.

The functions of bits 0 and 1 of the CMD register vary depending on the S-DPRAM

operating mode. In FIFO mode, they function as F1FL and F0FL, which initialize the FIFOs. Writing a 1 to F1FL/F0FL causes the following operations:

1. Sets F1E/F0E=1 and F1F/F0F=0 in the STS registers of both ports
2. Sets F1O/F0O=0 and F1U/F0U=0 in the SCFE registers of both ports
3. Sets F1E/F0E=1 in ISRC registers of both ports
4. Flushes FIFO 1/0 (initializes FIFO top and bottom pointers)

In DPRAM mode, they function as B1SL and B0SL, which select the upper or lower 512 bytes of RAM. Setting B1SL to 1 selects the upper 512-byte bank (bank 1). This sets B1EN =1 in the STS register. Setting B0SL to 1 selects the lower 512-byte bank (bank 0). This sets B0EN= 1 in the STS register. If an attempt is made to set both B1SL and B0SL to 1, B1SL will be set to 0 and B0SL will be set to 1, selecting the bank 0. These bits are only used for direct addressing with a non-multiplexed I/O bus. In the multiplexed mode, the A_9 signal selects the bank instead.

Writing a 0 to any bit of the CMD register clears that bit. Reading the CMD register returns the data that was most recently written into the register.

Table 5. 1024-Byte RAM Operation Mode and FIFO Size

RCFG				FIFO Size	
bit5	bit4	bit3	RAM Op Mode	FIFO 1	FIFO 0
0	0	0	DPRAM	—	—
0	0	1	FIFO	0	1024
0	1	0	FIFO	1024	0
0	1	1	FIFO	512	512
1	0	x	FIFO	256	768
1	1	x	FIFO	768	256

Notes: x: Don't care
Unit: bytes

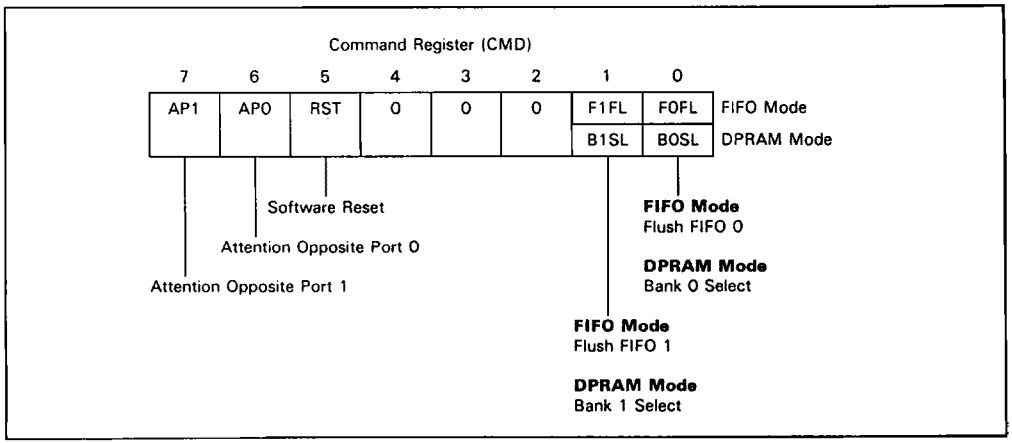


Figure 4. Command Register

Status Register (STS(A), STS(B))

The read-only STS register (figure 5) reports the operational status of the S-DPRAM. The IRQ bit displays the condition of the IRQ output terminal. FRQ1 and FRQ0 display internal signal conditions.

F1E and F0E (FIFO 1/0 empty) are set to 1 at FIFO 1/0 initialization.

FERR (FIFO error) is set when one of F1O, F1U, F0O, or F0U is set in the SCFE register:

FERR = F1O + F1U + F0O + F0U

5

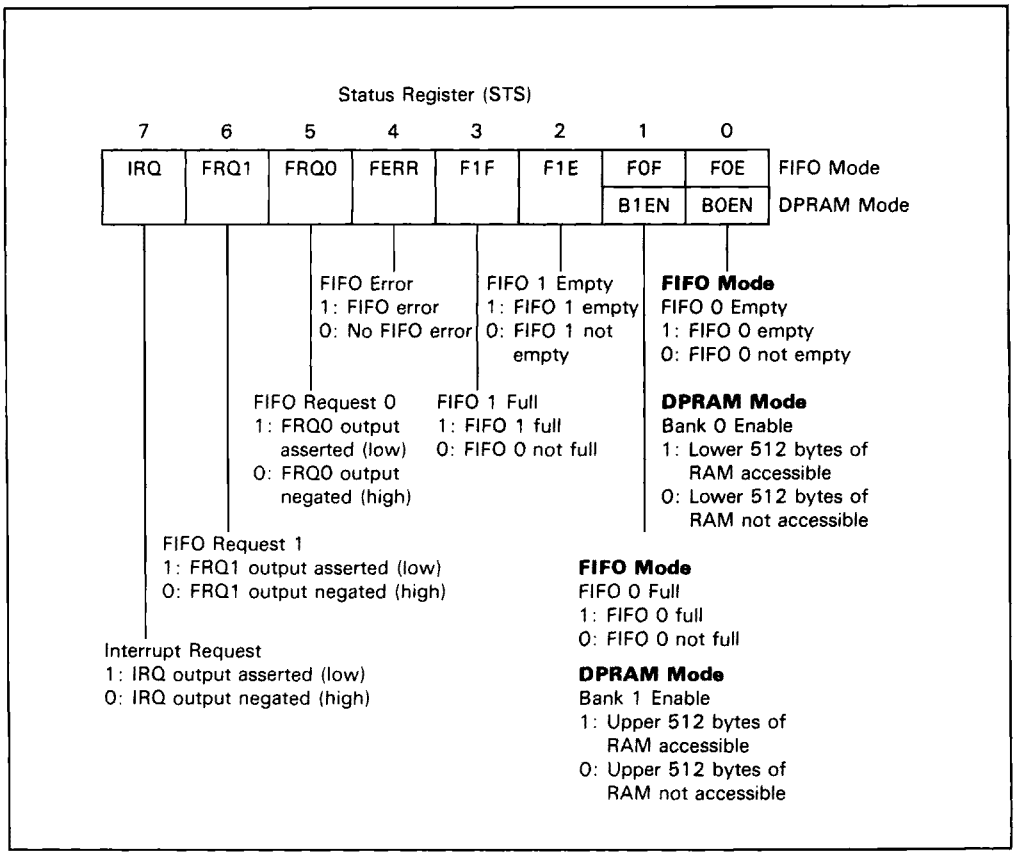


Figure 5. Status Register

Interrupt Source Register (ISRC(A), ISRC (B))

The ISRC register (figure 6) stores interrupt source information. The stored information is preserved regardless of FIFO, ATT, and RELE status change until the ISRC register is cleared. The interrupt request output IRQ is asserted when at least one bit of the AND of corresponding bits of IRSC and IEN registers is set to 1.

The ATT1 and ATT0 bits are set to 1 by a transaction request from the opposite port (AP1/AP0 of opposite port set).

RELE = 1 indicates that the shared resource semaphore of the opposite port has been released. This occurs when a bit of the REL register has changed from 0 to 1 due to the opposite port writing a 1 into the AVL bit of the REL register.

The FERR bit indicates a condition such as FIFO overflow or underflow. the F1F, F1E, F0F, and F0E bits, indicate FIFO full and empty conditions. They are set when the corresponding bit in the STS goes from 0 to 1. These bits in the ISRC register retain their state until the register is cleared. In the STS register, they show the current status when they are read.

Writing a 1 into an ISRC register bit clears that bit. Writing a 0 into a bit has no effect.

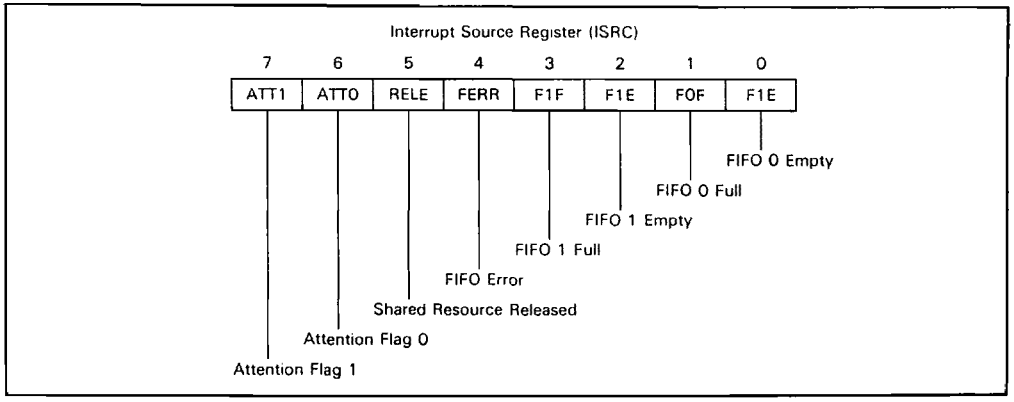


Figure 6. Interrupt Source Register

Interrupt Enable Register

The IEN register (figure 7) masks interrupt conditions in the ISRC register.

The IRQ signal is asserted when the AND of at least one pair of corresponding bits from IEN and ISRC is 1:

$$IRQ = ATT1 \cdot ENA7 + ATT0 \cdot ENA6 + RELE \cdot ENA5 + FERR \cdot ENA4 + F1F \cdot ENA3 + F1E \cdot ENA2 + F0F \cdot ENA1 + F0E \cdot ENA0$$

The IRQ bit of STS is also set when the IRQ signal is asserted.

Arbitration Register (ARB(A), ARB(B))

The ARB register (figure 8) provides a mechanism for multiple CPUs (or multiple processes on a single CPU) sharing a common port to arbitrate among themselves.

This register has a test-and-set hardware logic circuit, and allows system software to manage access to shared resources, including the S-DPRAM itself.

The ARB register access procedure is:

1. The requesting processor writes its ID code into the register.
2. The processor reads back the register. If it reads its own ID, its request has been accepted.
3. The processor makes the data transfer.
4. The processor releases the system resource by writing its ID into the register again.

Write Access: If the PID is 0, then ARB write access is allowed. if the PID is not 0:

1. Writing data equal to the PID clears the PID.
2. Writing data not equal to the PID does not change the PID.

Read Access: If the PID is equal to a processor's ID code, the system resource is owned by that processor. If the PID is not equal to a processor's ID code, the resource is owned by another processor. If the PID equals 0, the resource is free.

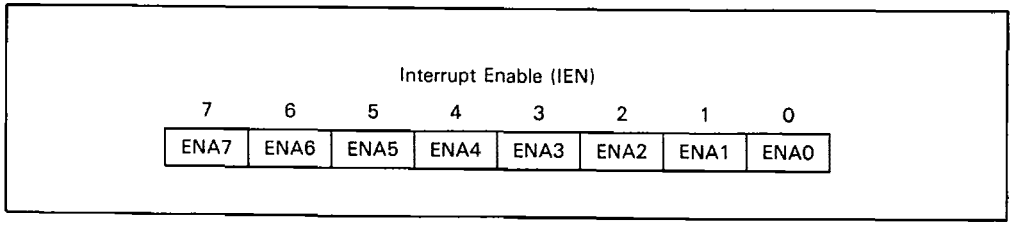


Figure 7. Interrupt Enable Register

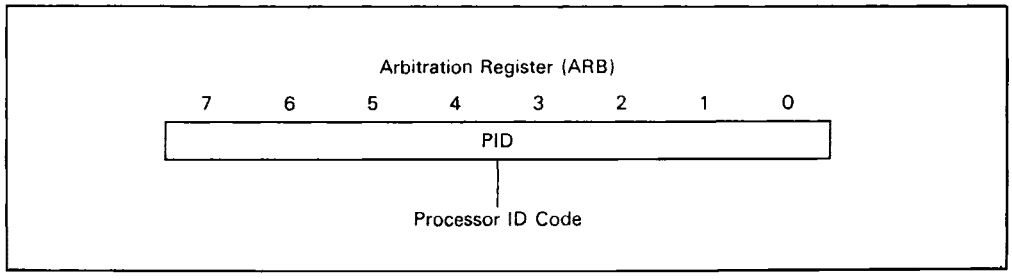


Figure 8. Arbitration Register

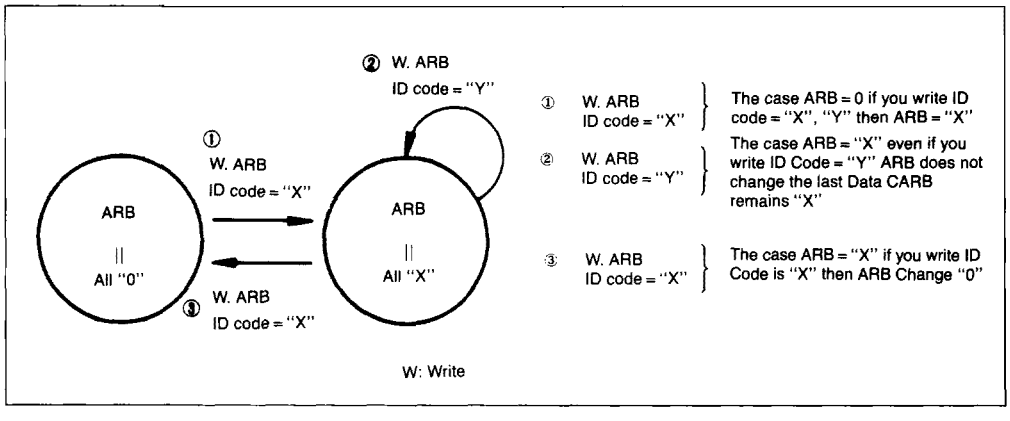


Figure 8A. ARB Status Chart

Acquire Ownership Register (AQR(A), AQR(B))

The AQR register (figure 9) provides 8 semaphores to arbitrate for exclusive ownership of system resources shared between ports A and B. It is used in conjunction with the REL register. Each of the 8 bits, OWN7-OWN0, can represent ownership of a separate system resource. An AQR register is provided for each port.

A resource is acquired as follows:

1. The requesting processor attempts to write a 1 into the bit corresponding to the desired shared resource.
2. The processor reads the bit back. A 1 indicates that the resource has been successfully acquired. A 0 indicatas that the resource has already been acquired by a device on the opposite port.
   Successfully writing a 1 to an AQR bit simultaneously clears the bit in the same position in the REL register.
3. The processor makes the data transfer.
4. The processor releases the resource by writing a 1 to the corresponding bit of the REL register. The OWN bit in the AQR register will be cleared and the shared resource will be released to the system.

Write Access: An unsuccessful attempt to write a 1 to an OWN bit indicates that the opposite port has acquired ownership of the corresponding shared resource. A successful attempt indicates that the resource has been acquired. On acquisition, a 0 is simultaneously written into the corresponding AVL bit in the REL register.

Read Access: A read access must be performed after an attempt to write a 1 into a bit. Writing a 1 does not guarantee that the bit will be set. When the bit is read as 1, it indicates that the device connected to this port has acquired ownership of the system resource corresponding to the bit.

Once successfully set, the bit remains set until the corresponding bit in the REL register is set or a reset occurs. When the bit is read as 0, it indicates that the request for ownership has failed.

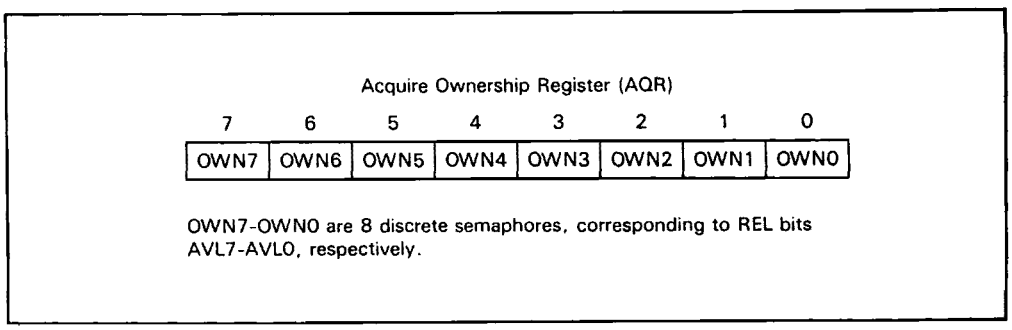


Figure 9. Acquire Ownership Register

Release Available Register (REL)

The REL register (figure 10), in conjunction with a port's AQR register, provides 8 semaphores for interport arbitration of system resources. The register is a bit-mapped summary of available system resources, and a mechanism for either port to release its acquired resources to the system. REL is a shared register accessible from both ports. AVL7-AVL0 correspond to OWN7-OWN0 of the AQR registers of each port.

When a port writes a 1 into a specific bit of the REL register, it releases the corresponding system resource, provided that it is owned by that port. A write access by the opposite port (which doesn't own the resource) causes no action. Successfully writing a 1 into an AVL bit simultaneously writes 0 the OWN bit of the AQR register of the port performing the write.

When the OWN bit is read as 1, it indicates that the corresponding system resource is available. Writing a 1 to an OWN bit of the AQR register simultaneously clears the corresponding AVL bit of the REL register.

The state transition diagram (figure 11) summarizes the relation between the REL register and the AQR registers of ports A and B.

Write Access: Writing a 1 into a bit releases the corresponding shared resource to the system. A port can set an AVL successfully only if the AQR register of that port has a 1 in the same bit position. A port successfully writing a 1 into an AVL bit clears the corresponding OWN bit in that port's AQR register. Writing a 0 has no effect. A transition from 1 to 0 in any bit of the REL register sets the RELE interrupt bit in the ISRC register.

Read Access: When a bit is read as 1, it indicates that the corresponding shared resource is available. The bit remains set until a 1 is written to the corresponding OWN bit of the AQR register. When a bit is read as 0, it indicates that the system resource has been acquired by either port A or port B.

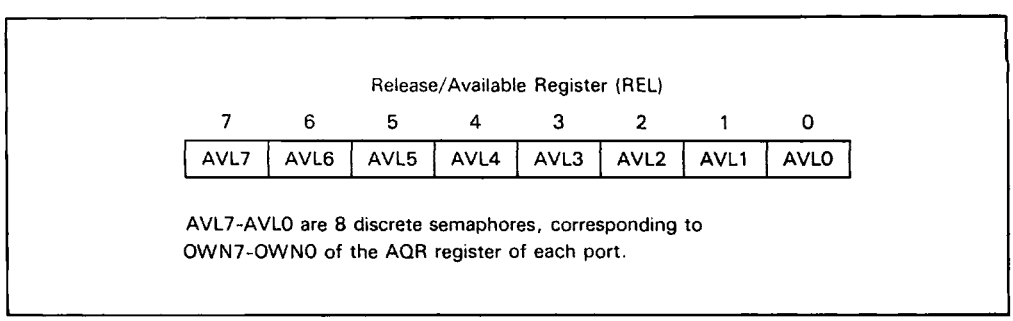


Figure 10. Release/Available Register

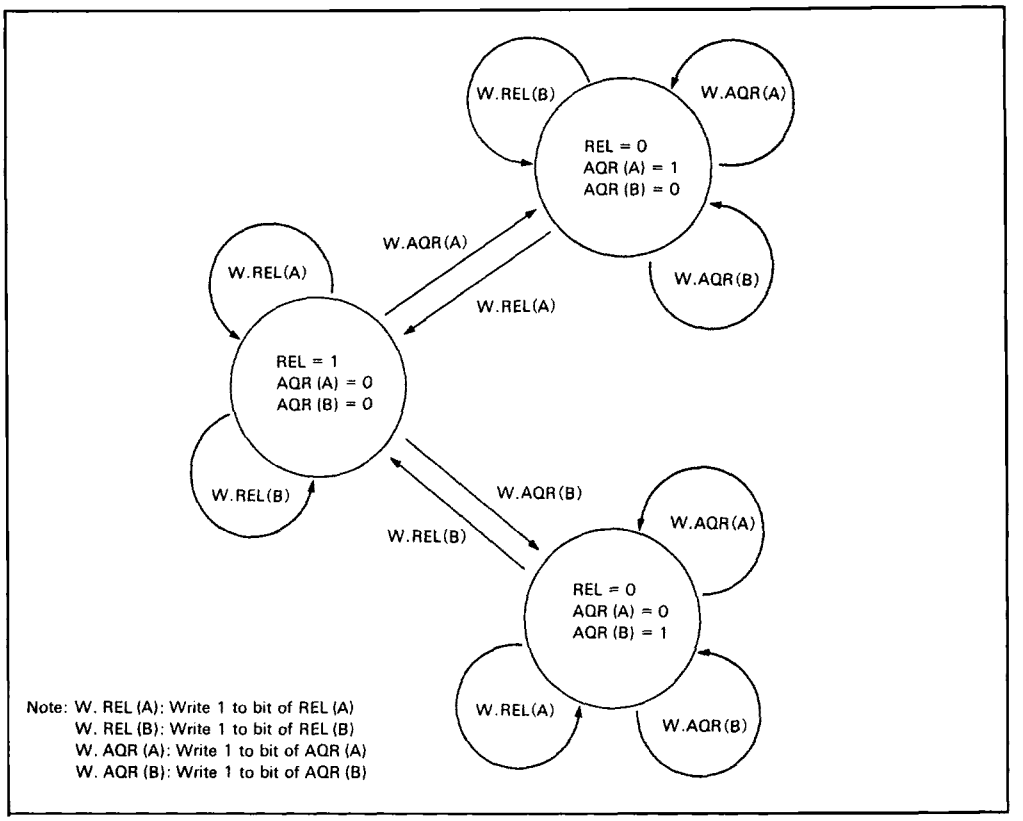


Figure 11. AQR, REL Register State Transition Diagram

Data Register (PADR0, PBDR0, PADR1, PBDR1)

Each port of the S-DPRAM has two data registers (figure 12). The data transferred through these registers depends on the S-DPRAM operating mode.

DPRAM Mode, Indirect Addressing Operation: An access to a port's data register in DPRAM mode results in an indirect access to the target register or RAM. If PADR0 or PBDR0 is accessed, the address will be supplied by PAAR0 or PBAR0. If PADR1 or PBDR1 is accessed, the address will be supplied by PAAR1 or PBAR1 (table 6).

The opposite port's registers are also accessible in indirect addressing mode. This facilitates system status checking and more efficient data transfer. Also, in this mode the SCFE register can be defined to allow the address register to increment by 1, decrement by 1, or remain unchanged after each indirect access.

FIFO Mode: During FIFO mode, accessing these registers transfers data to and from the two FIFOs. The data transfer direction is defined by bits F1D and F0D in the CNFG register.

PADR0 or PBDR0 transfer data between FIFO 0 and the corresponding port. Similarly PADR1 or PBDR1 transfer data between FIFO 1 and the corresponding port (table 7).

The RAM address for FIFO accesses is internally supplied from the port address registers PAAR0, PBAR0, PAAR1, and PBAR1 (table 8). The registers are automatically incremented after each transfer. These registers should not be accessed in FIFO mode.

Table 6. Address Register Used as Pointer

Port	Accessed Data Register	Pointer Address Register
Port A	PADR 0	PAAR 0
	PADR 1	PAAR 1
Port B	PBDR 0	PBAR 0
	PBDR 1	PBAR 1

Table 7. FIFO 0/1 and Data Registers

Port	FIFO	Accessed Data Register
Port A	FIFO 0	PADR 0
	FIFO 1	PADR 1
Port B	FIFO 0	PBDR 0
	FIFO 1	PBDR 1

Table 8. Address Control in FIFO Mode

FIFO	Pointer	Address Register
FIFO 0	Bottom (read)	PAAR 0
	Top (write)	PBAR 0
FIFO 1	Bottom (read)	PAAR 1
	Top (write)	PBAR 1

Note: Top = Top of FIFO, address for write transaction
Bottom = Bottom of FIFO, address for read transaction

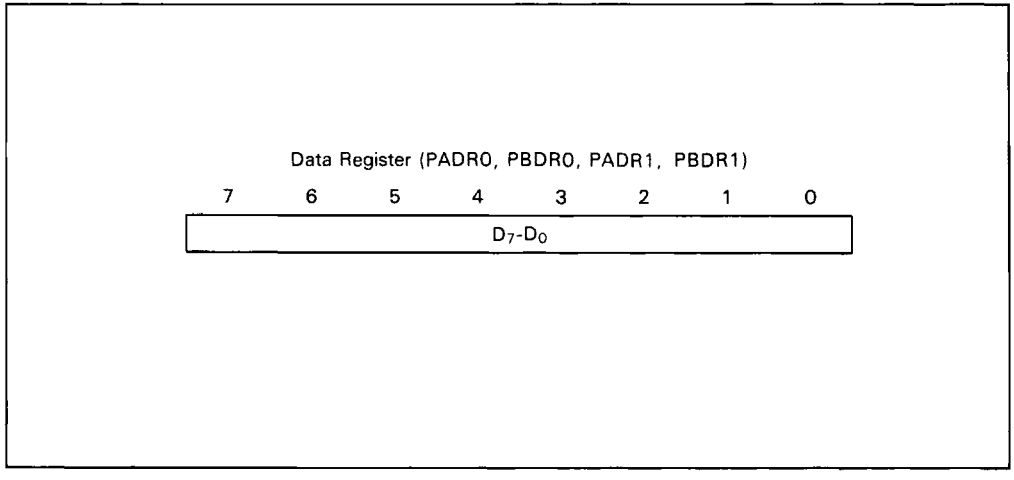


Figure 12. Data Register

5

FIFO Request Control Register (FRC(A), FRC(B))

The FRC registers (figure 13) control the FIFO request output signals $\overline{FRQ_1}$ and $\overline{FRQ_0}$ in FIFO mode.

The FR1E and FR0E bits enable $\overline{FRQ_1}$ and $\overline{FRQ_0}$ outputs, respectively. The FR1T and FR0T bits define the assertion conditions of these signals (tables 9, 10).

The FR1M and FR0M bits specify whether $\overline{FRQ_1}$ and $\overline{FRQ_0}$ are asserted as pulse or level outputs. As a pulse output, a $\overline{FRQ_1}$ or $\overline{FRQ_0}$ pulse is asserted each time a FIFO bus transaction occurs in the specified direction, and is held for as long as the asserting condition remains true. As a level output, $\overline{FRQ_1}$ or $\overline{FRQ_0}$ level is held asserted as long as the asserting condition remains true. (See timing chart for details.)

Table 9. $\overline{FRQ_1}$ Assertion Condition

F1D (CNFG Reg)	Data Access Direction	FR1T	Assertion Condition Port A	Assertion Condition Port B
0	A: Read	1	FIFO 1 full	FIFO 1 empty
	B: Write	0	FIFO 1 not empty	FIFO 1 not full
1	A: Write	1	FIFO 1 empty	FIFO 1 full
	B: Read	0	FIFO 1 not full	FIFO 1 not empty

Table 10. $\overline{FRQ_0}$ Assertion Condition

F0D (CNFG Reg)	Data Access Direction	FR0T	Assertion Condition Port A	Assertion Condition Port B
0	A: Read	1	FIFO 0 full	FIFO 0 empty
	B: Write	0	FIFO 0 not empty	FIFO 0 not full
1	A: Write	1	FIFO 0 empty	FIFO 0 full
	B: Read	0	FIFO 0 not full	FIFO 0 not empty

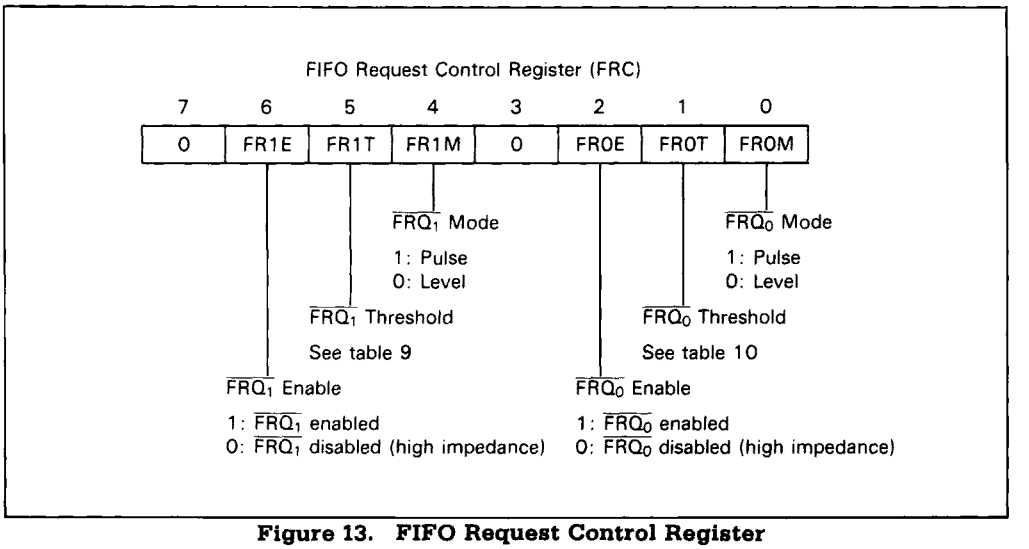


Figure 13. FIFO Request Control Register

Sequence Control and FIFO Error Register (SCFE(A), SCFE(B))

The SCFE register (figure 14) has 4 bits for address register control, and 4 bits for FIFO error indication.

The upper 4 bits (CNT1, DEC1, CNT0, DEC0) control the corresponding address register count. If CNT1/CNT0 = 0, the address register will not change after access to the corresponding data register. If CNT1/CNT0 = 1, the address register will decrement by when DEC1/DEC0 = 1 or increment by 1 when DEC1/DEC0 = 0 (table 11).

Table 11. Address Register Count Control (DPRAM-Indirect Addressing Mode)

Accessed Data Register	CNT0	Address Register DEC0 Counting			CNT1	DEC1
PADR1/PBDR1	PAAR1/PBAR1	0	x	x	x	+ 0
		1	0	x	x	+ 1
		1	1	x	x	− 1
PADR0/PBDR0	PAAR0/PBAR0	x	x	0	x	+ 0
		x	x	1	0	+ 1
		x	x	1	1	− 1

Note: x = Don't care

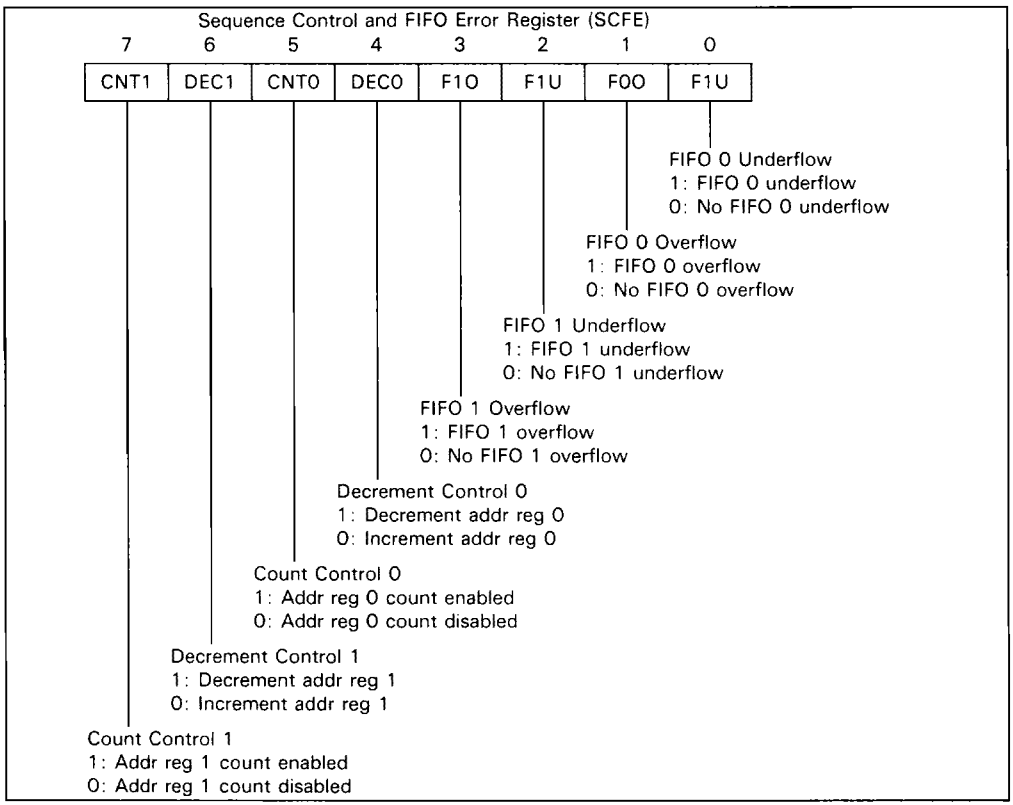


Figure 14. Sequence Control and FIFO Error Register

5

The lower 4 bits (F1O, F1U, F0O, F0U) indicate FIFO 1/0 overflow and underflow errors. After these errors are detected, the FIFO must be initialized using the F0FL and F1FL bits of the CMD register. These bits remain set until the FIFO is initialized until a reset occurs.

Address Register (PAAR0, PBAR0, PAAR1, PBAR1)

The S-PDRAM has two address registers (figure 15) for each port. These registers specify the RAM or register address during indirect addressing in DPRAM mode. The automatic address count is controlled by the CNT1, DEC1, CNT0, and DEC0 bits of the SCFE registers. In FIFO mode, these registers are used as the read/write pointers, and cannot be accessed externally.

Parameter Registers (PRAM31-PRAM0)

The 32-byte PRAM registers PRAM31-PRAM0 (figure 16) are user definable registers. They can be used for passing parameters between ports, or for extra storage space. These registers are shared by both ports.

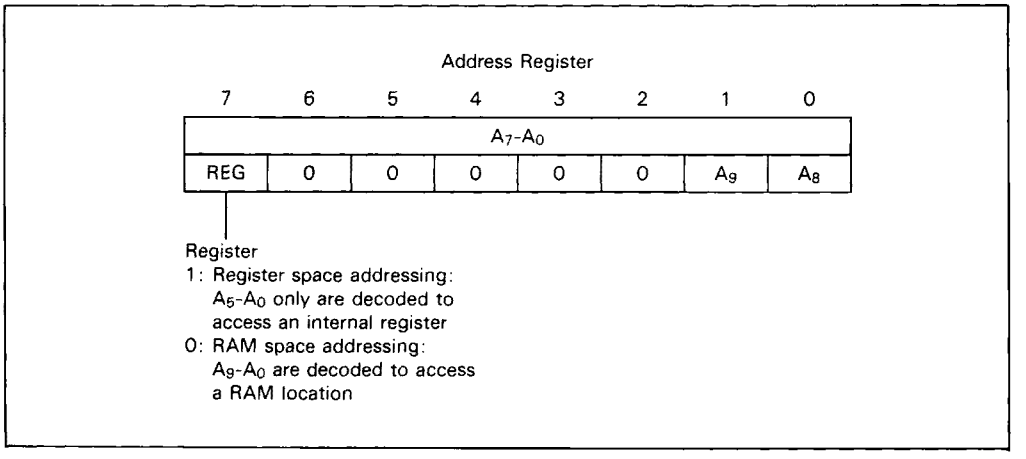


Figure 15. Address Register

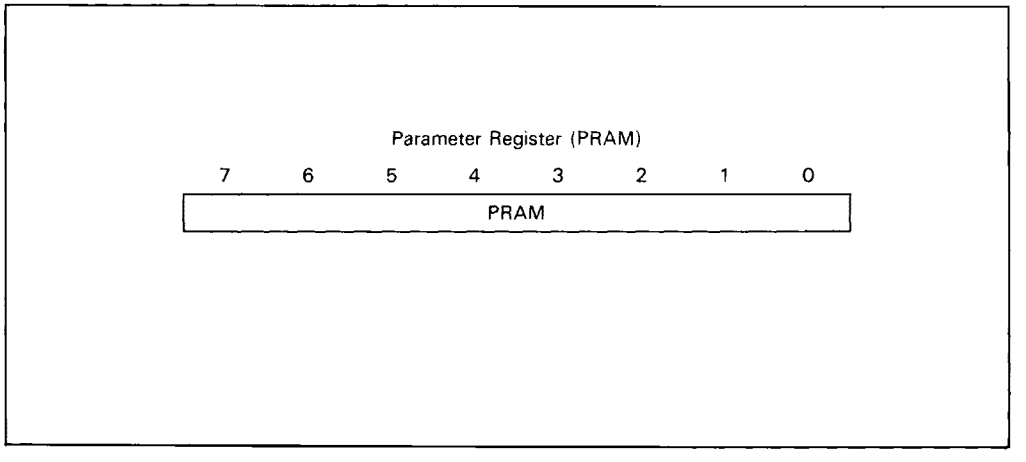


Figure 16. Parameter Register

Reset

Figures 17 and 18 show the procedure for driving the S-DPRAM after reset.

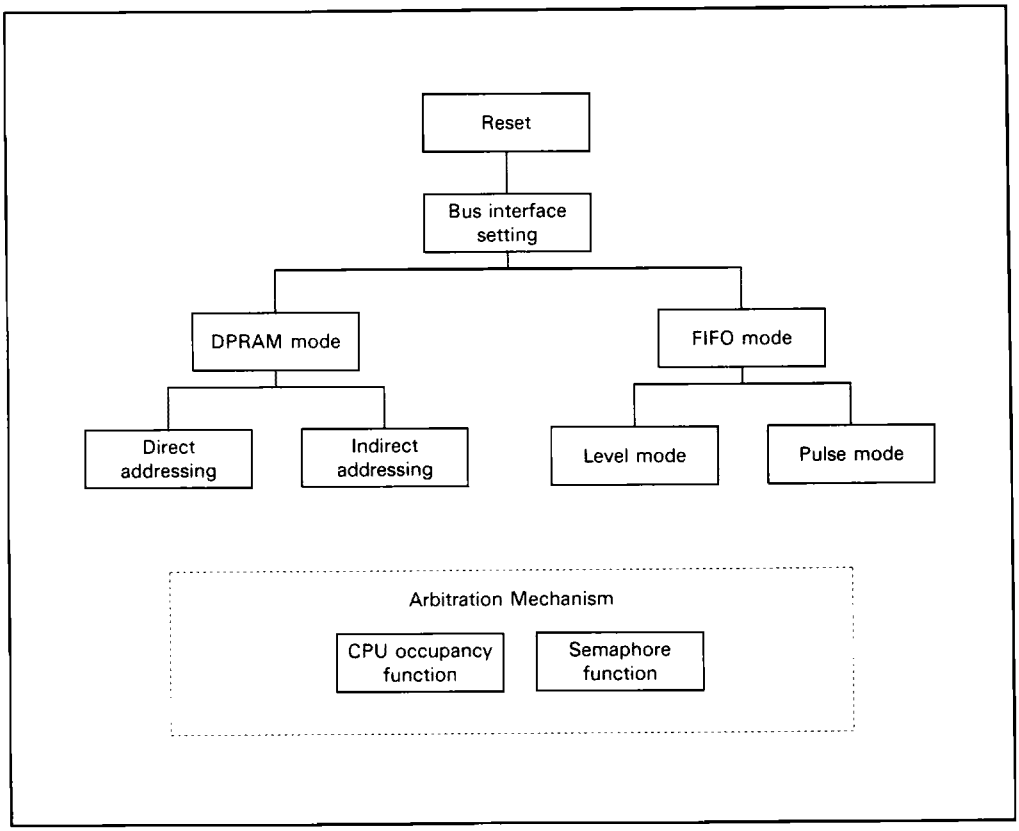


Figure 17. S-DPRAM Setup

5

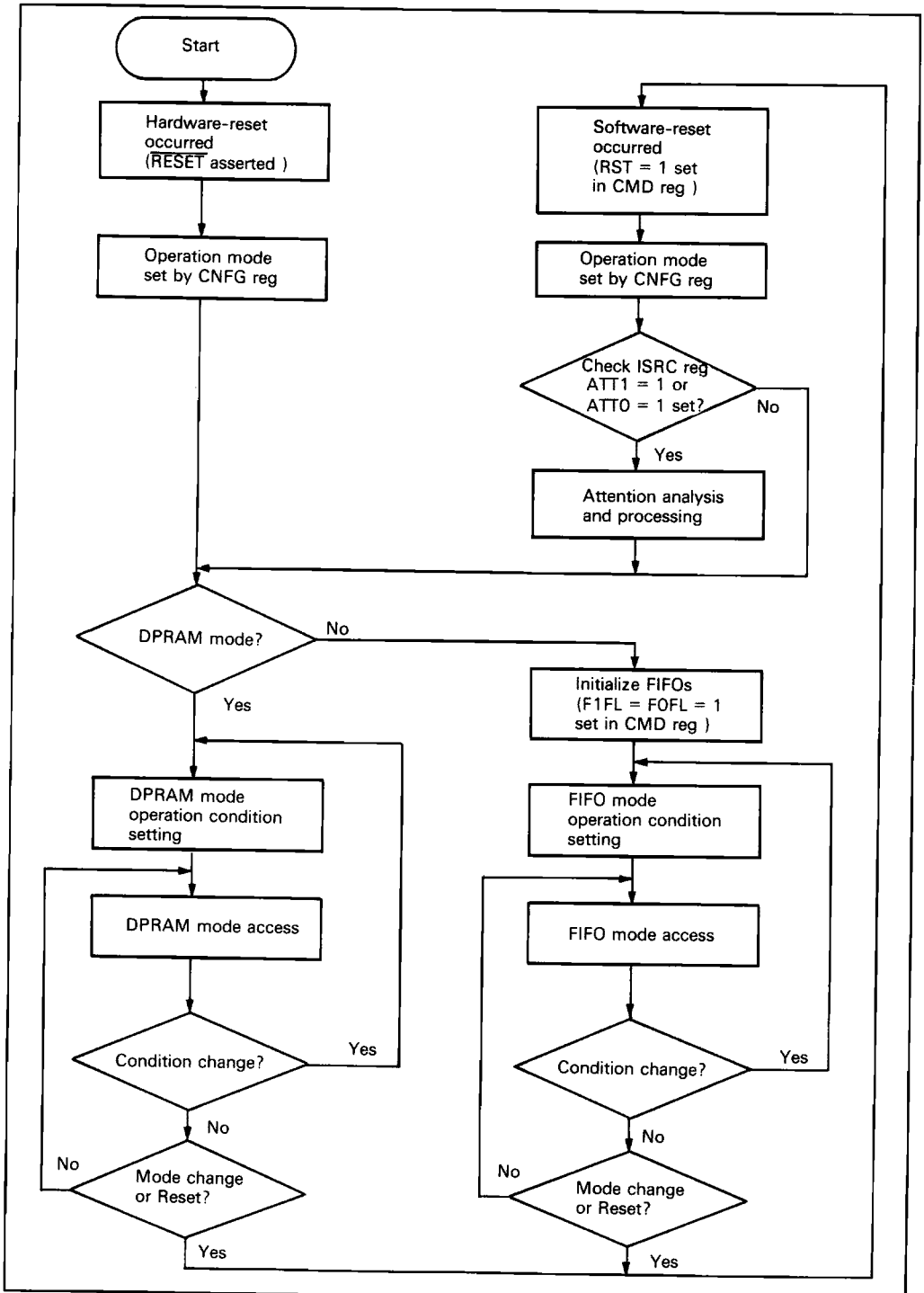


Figure 18. S-DPRAM Driving Procedure

The S-DPRAM performs two types of reset operations: hardware reset due to an external signal assertion and software reset due to a command setting (table 12).

- Hardware reset: Asserting the $\overline{\text{RESET}}$ input places the S-DPRAM into the initial state. Hardware reset is used for system initialization such as power-on reset, and must be performed after applying power.
- Software reset: Writing a 1 into the RST bit of the CMD register of either port initializes the device. Software reset does not affect the interrupt flags (AP1 and AP0 of CMD register or ATT1 and ATT0 of ISRC register).

  Owing to this architecture, the S-DPRAM can recognize an interrupt request from the opposite port even after reset.

The S-DPRAM is set to the status shown in table 12 after initialization, and is therefore operable immediately after reset in a normal interface system.

Bus Interface

Each port's I/O bus can be configured as a non-multiplexed or multiplexed address and data bus. Non-multiplexed mode is selected when ABUS/BBUS bit of the CNFG register is set to 0, and multiplexed mode when it is 1. In the bus connection example of figure 19, port A is configured as a non-multiplexed bus and port B as a multiplexed bus.

In multiplexed mode, the lower address is input via address data pins AD_0-AD_7. Therefore, address pins A_0-A_5 of A_0-A_7 are left unused. A_6 and A_7, which are multipurpose lines, are switched to A_8 and A_9 and used as input pins for the upper address. It should be noted that the unused A_0-A_5 must be connected to GND. Since the S-DPRAM is set to non-multiplexed mode after initialization, the CNFG register must be accessed first to have the I/O bus multiplexed. If A_0-A_5 are set to 0 by connecting them to GND, the CNFG register at register address 00 is automatically selected by asserting $\overline{\text{WRS}}$ or $\overline{\text{RDS}}$. Therefore, RS must be set high.

Table 12. S-DPRAM Status After Reset

Item	Status					
Internal register	Refer to table 3 "Internal Register: Bit Configuration and Operation"					
RAM	Not affected					
FIFO status flags		Empty	Full	Underflow	Overflow	
	FIFO1	0	0	0	0	Not Empty and
	FIFO0	0	0	0	0	Not Full
Terminal (both ports)	Multipurpose line (1): A_6, Multipurpose line (2): A_7, Multipurpose line (3): A_8, Data bus AD_0-AD_7: high impedance $\overline{\text{IRQ}}$: high impedance $\overline{\text{READY}}$: high level					
Operation mode	Mode: DPRAM mode Bus interface: non-multiplexed bus Access method: direct addressing method (bank = 0)					

5

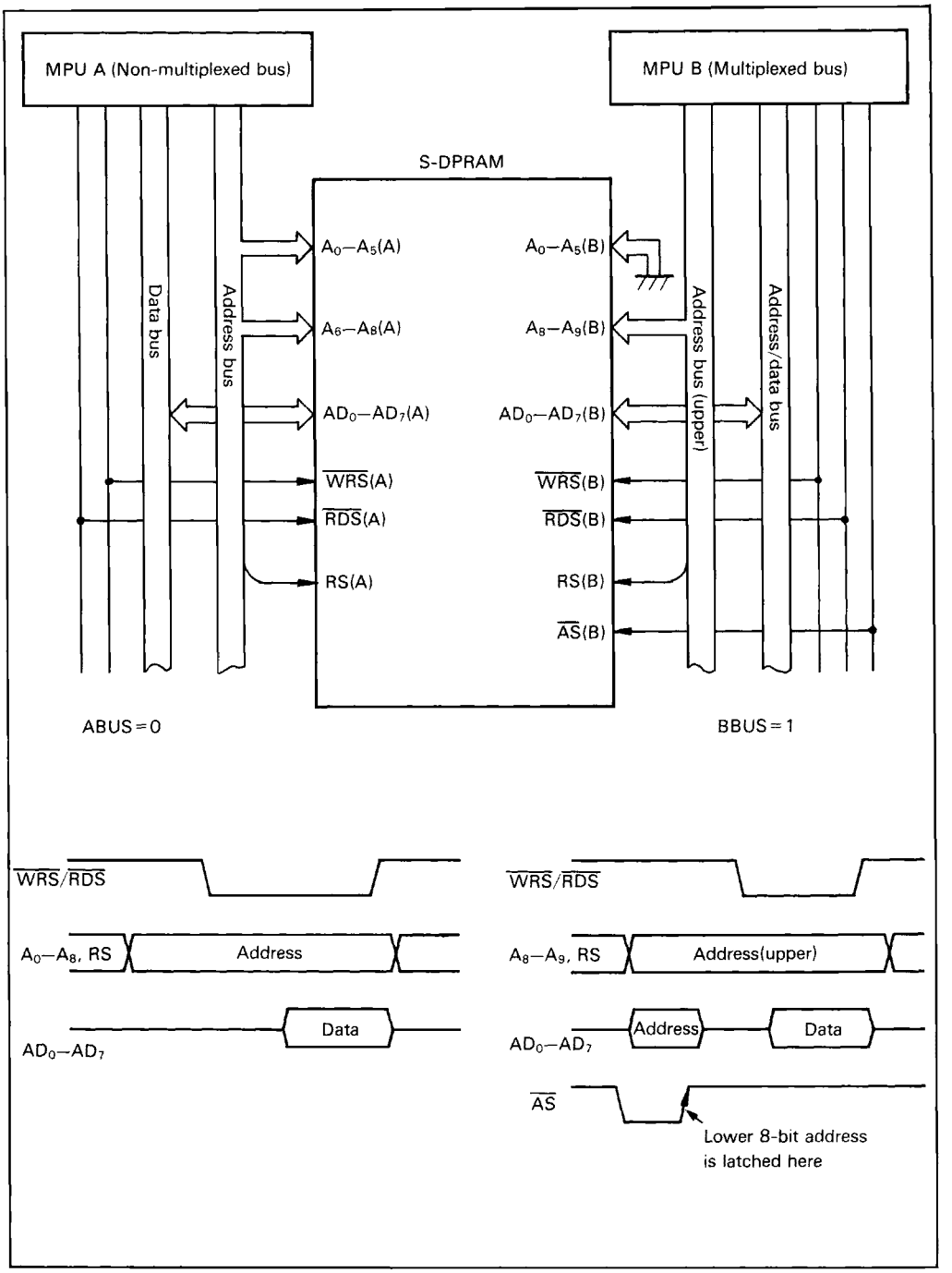


Figure 19. Bus Connection Example

Operation Mode

Two types of operation modes can be programmed in the S-DPRAM: DPRAM mode and FIFO mode. DPRAM mode is selected when the RCFG field of the CNFG register is 000, and FIFO mode when it is other than 000. (See table 5 "1024-Byte RAM Operation Mode and FIFO Size".) Please note that the registers may have various functions depending on the operation mode. Table 13 shows the bit configuration of registers in each mode.

Table 13. Bit Configuration of Internal Registers

DPRAM mode (RCFG = 000)

Address	Register name	Bit configuration 7	6	5	4	3	2	1	0
R00	CNFG	ABUS	BBUS	RCFG			0	—	—
R01	CMD	AP1	AP0	RST	0	0	0	—	—
								B1SL	B0SL
R02	IEN	ENA7	ENA6	ENA5	ENA4	ENA3	ENA2	ENA1	ENA0
R03	ISRC	ATT1	ATT0	RELE	—	—	—	—	—
R04	STS	IRQ	0	0	0	0	0	—	—
								B1EN	B0EN
R05	ARB	PID							
R06	AQR	OWN7	OWN6	OWN5	OWN4	OWN3	OWN2	OWN1	OWN0
R07	REL	AVL7	AVL6	AVL5	AVL4	AVL3	AVL2	AVL1	AVL0
R08	PA/B DR0	D_7-D_0 (Indirect addressing)							
R09	PA/B DR1	D_7-D_0 (Indirect addressing)							
R0A	FRC	0	—	—	—	0	—	—	—
R0B	SCFE	CNT1	DEC1	CNT0	DEC0	0	0	0	0
R0C	PA/B AR0(L)	A_7-A_0							
R0D	PA/B AR0(H)	REG	0	0	0	0	0	A_9	A_8
R0E	PA/B AR1(L)	A_7-A_0							
R0F	PA/B AR1(H)	REG	0	0	0	0	0	A_9	A_8
R10 : R2F	PRAM	PRAM							

5

FIFO mode (RCFG ≠ 000)

Address	Register name	Bit configuration 7	6	5	4	3	2	1	0
R00	CNFG	ABUS	BBUS	RCFG			0	F1D	F0D
R01	CMD	AP1	AP0	RST	0	0	0	F1FL	F0FL
								—	—
R02	IEN	ENA7	ENA6	ENA5	ENA4	ENA3	ENA2	ENA1	ENA0
R03	ISRC	ATT1	ATT0	—	FERR	F1F	F1E	F0F	F0E
R04	STS	IRQ	FRQ1	FRQ0	FERR	F1F	F1E	F0F	F0E
								—	—
R05	ARB	PID							
R06	AQR	OWN7	OWN6	OWN5	OWN4	OWN3	OWN2	OWN1	OWN0
R07	REL	AVL7	AVL6	AVL5	AVL4	AVL3	AVL2	AVL1	AVL0
R08	PA/B DR0	D_7-D_0							
R09	PA/B DR1	D_7-D_0							
R0A	FRC	0	FR1E	FR1T	FR1M	0	FR0E	FR0T	FR0M
R0B	SCFE	—	—	—	—	F1O	F1U	F0O	F0U
R0C	PA/B AR0(L)	FIFO 0: A=Read pointer (L), B=Write pointer (L)							
R0D	PA/B AR0(H)	FIFO 0: A=Read pointer (H), B=Write pointer (H)							
R0E	PA/B AR1(L)	FIFO 1: A=Read pointer (L), B=Write pointer (L)							
R0F	PA/B AR1(H)	FIFO 1: A=Read pointer (H), B=Write pointer (H)							
R10	PRAM	PRAM							
:									
R2F									

Note: 0: Write causes no action, and read as 0 only.
—: Invalid in the specified operation mode.

DPRAM Mode

In DPRAM mode, 1024-byte RAM can be used as shared memory accessible from either port A or B. The address space accessible at a time differs depending on the bus interface configuration: non-multiplexed or multiplexed (table 15).

- Non-multiplexed mode: Up to 512 bytes can be addressed via A_0-A_8. All addresses are accessible due to bank switching by B1SL (upper) and B0SL (lower) bits of the CMD register.
- Multiplexed mode: Up to 1024 bytes can be addressed via AD_0-AD_7, A_8, and A_9. All addresses are accessible.

To increase efficiency in software development, two types of addressing methods are implemented: direct addressing and indirect addressing.

Direct Addressing Mode: In direct addressing mode, the target address is specified by externally supplied address signals and internal bank select bits B1SL and B0SL. When the I/O bus is non-multiplexed, bank select bit information is reflected on A_9. Table 14 shows how the address space is specified by B1SL and B0SL and how its status is reflected on the B1EN and B0EN bits of the STS register.

Indirect Addressing Mode: In indirect addressing mode, the target address is obtained by accessing the data register. When the data register is accessed, the corresponding address register is referenced, and memory is accessed according to this information. Therefore, the target address must be written into the address register before accessing the data register.

Table 14. Direct Addressing Mode Address Space

CMD Register Write			STS Register Read	
B1SL	**B0SL**	**Address space**	**B1EN**	**B0EN**
0	0	Not affected	Not affected	
0	1	000–1FF (lower)	0	1
1	0	200–3FF (upper)	1	0
1	1	000–1FF (lower)	0	1

Table 15. DPRAM Mode Table (RCFG=000)

		Address to be Accessed			
Addressing	**Bus interface**	**RS**	**Address inputs**	**Register**	**Associated Register**
Direct Addressing	Non-multiplexed	0	A_0-A_5 A_6-A_8	STS reg: B1EN, B0EN (A9)	CMD reg: B1SL, B0SL (for bank selection)
	Multiplexed	0	AD_0-AD_7 A_8, A_9	—	—
Indirect Addressing	Non-multiplexed	1	A_0-A_5 P(A/B) DR0 P(A/B) DR1	—	SCFE reg: CNT1, DEC1, SCFE reg: CNT0, DEC0 (for address counting)
	Multiplexed	1	AD_0-AD_5 P(A/B) DR0 P(A/B) DR1	—	PA/BAR0, PA/BAR1 register (for address reference)

Note: Refer to table 6 in the data register section for the relation between data register and address register in indirect addressing mode.

In indirect addressing mode, the address can be automatically incremented or decremented after each read/write access of the data register. Either increment, decrement, or no count can be selected by setting the SCFE register. This automatic address count function facilitates sequential memory access. In addition, two memory groups can be separately accessed because there are two channels of address registers for each port.

FIFO Mode

In FIFO mode, memory is configured as a buffer with first-in/first-out facilities. This mode is effective for sequential data communications between transmitter and receiver having different data transfer speeds. FIFO mode is selected when the RCFG field of the CNFG register equals any bit pattern from 001 to 111. In this mode, RAM is configured as two FIFOs. The size of each FIFO is determined by the RCFG field code (See table 5).

The FIFO-associated registers are listed in table 16. They are implemented for both ports independently. FIFOs 0 and 1 are separately controlled by these registers. The RCFG field and FERR bit are shared by FIFO 0 and 1, while the others are respectively provided for each FIFO.

Indirect addressing mode is disabled in FIFO mode, since the address registers are used for FIFO read/write pointers. Therefore, the opposite port's registers are not accessible in this mode. Refer to the subsection "Special FIFO Mode Application" for access to the opposite port's registers.

The S-DPRAM $\overline{FRQ}$ output signal represents FIFO status. The multipurpose line functions as $\overline{FRQ}$ in FIFO mode. The FRC register controls the following with respect to $\overline{FRQ}$: output enabled or disabled, assertion condition, and output waveform mode. In particular, for output waveform mode, either level mode or pulse mode can be selected depending on the system used.

Table 16. FIFO-Associated Registers

	Associated Bit		
Register Name	**FIFO 0**	**FIFO 1**	**Function**
CNFG	RCFG		FIFO configuration (001 – 111)
	F0D	F1D	FIFO data transfer direction
CMD (A/B)	F0FL	F1FL	FIFO initialization
STS (A/B)	FRQ0	FRQ1	$\overline{FRQ_0}$ & $\overline{FRQ_1}$ outputs status
	FERR		FIFO error status
	F0F, F0E	F1F, F1E	FIFO full or empty
ISRC (A/B)	FERR		FIFO error status
	F0F, F0E	F1F, F1E	FIFO full or empty
FRC (A/B)	FR0E, FR0T, FR0M	FR1E, FR1T, FR1M	$\overline{FRQ_0}$, $\overline{FRQ_1}$ mode setting
SCFE (A/B)	F0O, F0U	F1O, F1U	FIFO overflow or underflow
Read/write register	PADR0 (Note) PBDR0 (Note)	PADR1 (Note) PBDR1 (Note)	Data register to be accessed
Read pointer write pointer	PAAR0 (Note) PBAR0 (Note)	PAAR1 (Note) PBAR1 (Note)	Address reg to be referenced

Note: Register name

Level Mode: Level mode is selected when the FR1M or FR0M bit of the FRC register is set to 0. In this mode, $\overline{FRQ}$ is asserted continuously as long as the assertion condition defined by FR1T or FR0T remains true. In level mode, data transfer is allowed if only the detected $\overline{FRQ}$ level is low. Example 1 shows a handshaking protocol realized using this mode.

Example 1:
FIFO partition: FIFO 1 (1024 bytes)
·····RCFG = 010
Data transfer direction: Port A to B
·····F1D = 1
Output enable: $\overline{FRQ1}$ output enabled
·····FR1E = 1
Waveform mode: Level mode·····FR1M = 0
Threshold: A = Empty, B = Full·····FR1T = 1

Port A
(1) $\overline{FRQ}$ is asserted and FIFO is judged to be empty.
(2) CPU writes up to 1024 bytes of data into FIFO
(3) Wait until the next assertion.

Port B
(1) $\overline{FRQ}$ is asserted and FIFO is judged to be full.
(2) CPU reads up to 1024 bytes of data from FIFO
(3) Wait until the next assertion.

Data transfer is carried out by repeating procedures (1) to (3). A different handshaking protocol uses different threshold conditions.

Example 2:
Threshold: A = Not full, B = Not empty
·····FR1T = 0
(Other conditions are the same as those in example 1)

Port A
(1) $\overline{FRQ}$ is asserted and FIFO is judged to be not full.
(2) Write access continues until $\overline{FRQ}$ is negated (until FIFO becomes full)

Port B
(1) $\overline{FRQ}$ is asserted and FIFO is judged to be not empty.
(2) Read access continues until $\overline{FRQ}$ is negated (until FIFO becomes empty)

Data transfer is carried out by repeating procedures (1) to (2).

Notes for HD63310R interface: The $\overline{FRQ}$ signal of the HD63310R is negated after it completes the $\overline{WRS}$ cycle. and therefore $\overline{FRQ}$ acknowledge timing must be specified after $\overline{WRS}$ cycle completion. In general, t_{FNLW}, the $\overline{FRQ}$ negation delay time at write access, is 150 ns. Taking the delay due to access contention into account, at least 400 ns of delay is required after $\overline{WRS}$ negation.

Pulse Mode: Pulse mode is selected when the FR1M or FR0M bit of the FRC register is set to 1. In this mode, $\overline{FRQ}$ is asserted and negated repeatedly each time a data transfer occurs as long as the assertion condition defined by FR1T or FR0T remains true. In pulse mode, handshaking takes place by detecting the falling edge of $\overline{FRQ}$ (figure 20).

Example:
FIFO partition: FIFO 0 (1024 bytes)
·····RCFG = 001
Data transfer direction: Port B to A
·····F0D = 0
Output enable: $\overline{FRQ0}$ output enabled
·····FR0E = 1

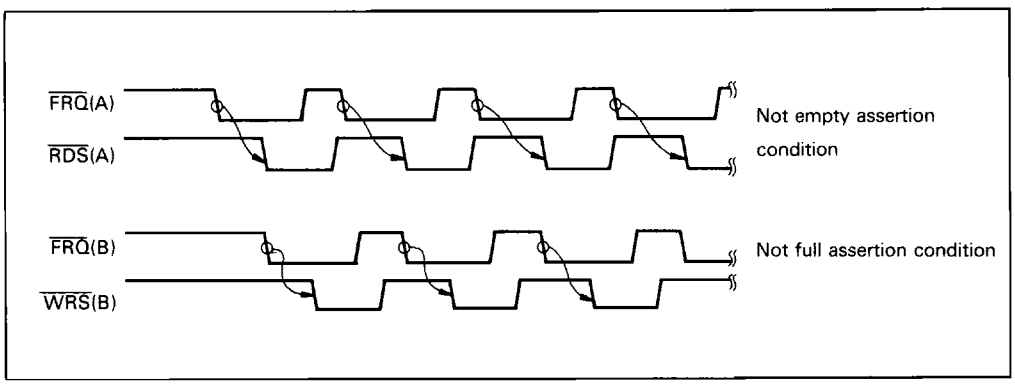


Figure 20. Pulse Mode Handshaking

Waveform mode: Pulse mode
..... FR0M = 1
Threshold: A = Not empty, B = Not full
..... FR0T = 0

Port A— Reads 1-byte of data from FIFO upon each assertion of $\overline{FRQ}$

Port B— Writes 1-byte of data into FIFO upon each assertion of $\overline{FRQ}$

In pulse mode, handshaking is performed by detecting the falling edge of $\overline{FRQ}$, and therefore special timing is not required. In addition, overrun (excessive write/read access) due to $\overline{FRQ}$ delay as in level mode cannot occur, since the falling pulse count agrees with the transfer byte count in this mode.

FIFO Operation: Figure 21 illustrates an example of FIFO operation. In FIFO mode, RAM is configure as two ring-shaped memories. Each memory has read/write pointers and thus can be accessed independently. The PBAR (0/1) and PAAR (0/1) registers, used in indirect addressing DPRAM mode, are assigned here as write pointer and a read pointer, respectively. These pointers are automatically incremented upon each access. During write access, a write pointer is incremented after writing data into the address indicated by the write pointer. In the same way, a read pointer is incremented after reading data from the address indicated by the read pointer. When a write pointer value equals a read pointer value, FIFO memory is considered to be empty. When a write pointer value exceeds a read pointer value by one, FIFO memory is considered to be full. These FIFO statuses are represented by the status bits of internal registers or are output via the $\overline{FRQ}$ line.

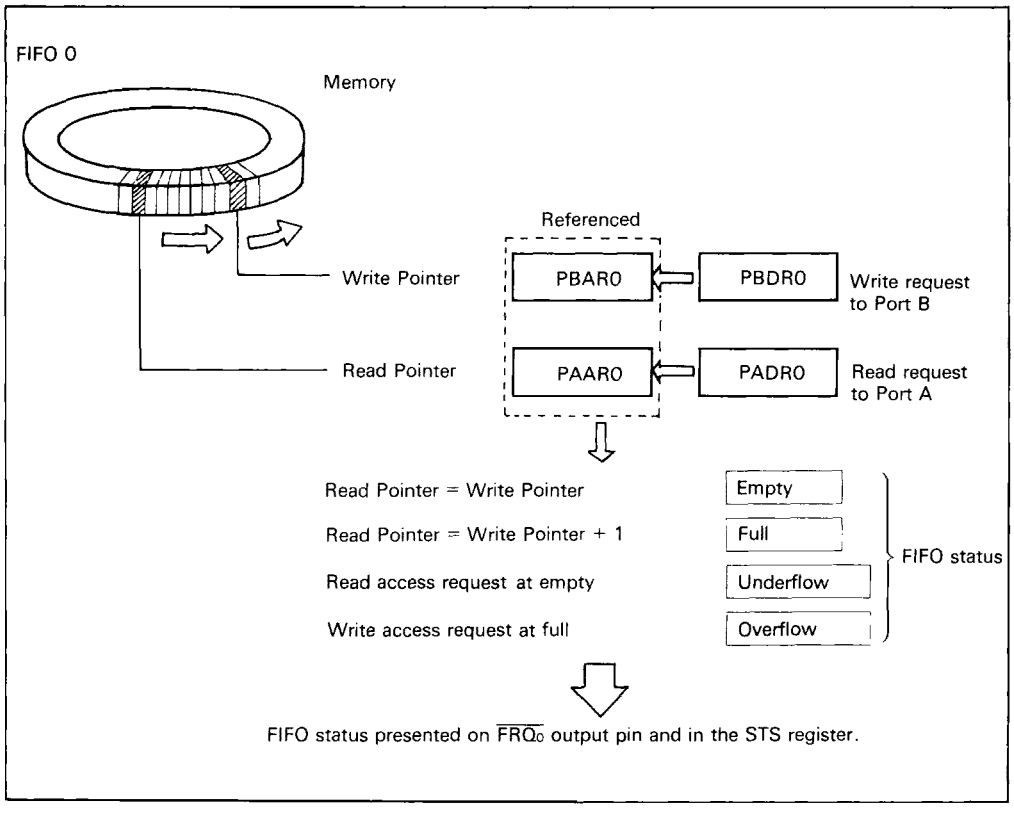


Figure 21. FIFO 0 Control Example (Transfer Direction: Port B to Port A)

Special FIFO Mode Application: The S-DPRAM has been basically designed as an interface between CPUs. Therefore, there are some limitations when it is used with non-intelligent devices. In particular, when the S-DPRAM is used in FIFO mode, some registers must be set and therefore the S-DPRAM cannot support an interface between non-intelligent devices.

Figure 22 shows an example of an interface between a CPU and non-intelligent device. In this example, the CPU reads sequential digital data that is converted from analog data through synchronous sampling with a clock; the digital data is then stored into FIFO. Moreover, the FIFO status is returned to the A/D converter via the FRQ line. The S-DPRAM performs handshaking by referring to this signal's status.

- Non-intelligent device setting: The address bus must be fixed to the address of the data register of the FIFO being used. $\overline{RS}$ and $\overline{RDS}$ must be set to high, and the sampling clock for A/D conversion must be input to $\overline{WRS}$. This clock circuit must be controlled using $\overline{FRQ}$.
- Register setting: Through the non-intelligent device's port, only the data register is accessible, and therefore each register must be set through the CPU's port. The opposite port's registers can be accessed using the indirect addressing method, but in FIFO mode this method cannot be used. Accordingly, the S-DPRAM must be placed in FIFO mode after the FIFO-associated registers of the opposite port are set using the indirect addressing method in DPRAM mode.

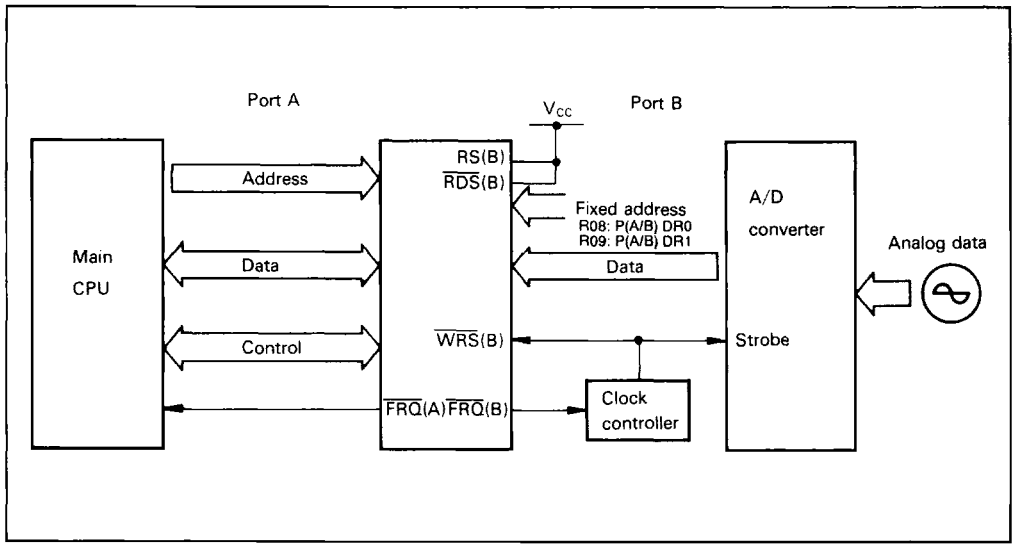


Figure 22. FIFO Application Example (CPU-A/D Converter Interface)

5

General Notes for FIFO Mode: Keep the following in mind when operating in FIFO mode.

1. Mode Transition: Switching from DPRAM or FIFO operation mode does not affect the contents of registers or RAM. The FIFO status flag is reset upon transition from FIFO mode to DPRAM mode. However, when the address that will result in empty or full status is stored in DPRAM mode, the FIFO status flag is not set even if the S-DPRAM is switched to FIFO mode.

2. Underflow and overflow: Attempting a read access after empty status causes an underflow and invalid data is read out. The read pointer is incremented and the empty flag is reset. In the same way, a write access after full status causes an overflow and data at the bottom of the stack is destroyed. The write pointer is incremented and the full flag is reset.

3. Data transfer direction: When data access against the transfer direction specified by CNFG is requested, flags and FIFO operation are not affected except that input and output are replaced by each other. That is, when A to B direction is specified, a write access to port B results in a successful data write to B, and a read access from port A results in a successful data read from A; internal operation is not affected. However, it should be noted that the $\overline{FRQ}$ output condition is reversed.

4. FIFO partitioning and addresses: Table 17 shows how memory addresses are allocated corresponding to FIFO partitions.

Arbitration

The S-DPRAM arbitration function supports multiprocessor systems. It facilitates effective data communications between multiple CPUs on one system bus or between two system buses. There are two arbitration methods:
1. Software arbitration by register flag control
2. Hardware arbitration by an internal arbiter

This section describes the software arbitration control method. Hardware arbitration will be described later in the section about internal operation.

The S-DPRAM incorporates the following registers for interprocessor arbitration: arbitration register (ARB), acquire ownership register (AQR), and release/available register (REL). These registers only act as flags to control interprocessor arbitration, and are not directly related to S-DPRAM operations. Accordingly, they are designated as a flag register group and clearly distinguished from the special control register (SCR) group and address/data register group.

These registers are used for two types of arbitration: CPU occupancy or semaphore function, depending on the architecture of the multiprocessor system (table 18).

Table 17. FIFO Memory Addresses

RCFG			Memory address	
b5	b4	b3	FIFO 1	FIFO 0
0	0	1	—	000–3FF
0	1	0	000–3FF	—
0	1	1	200–3FF	000–1FF
1	0	x	300–3FF	000–2FF
1	1	x	100–3FF	000–0FF

CPU Occupancy Function

In a system having several processors on one system bus (multimaster bus system), each processor can acquire ownership of the system bus, and thus there may be contention for control of a shared resource (I/O, memory, etc). To avoid this contention, the S-DPRAM uses the ARB register to indicate which processor currently controls the shared resource . Figure 23 shows an example of a multiprocessor system configuration. In this system, processor A, B, and C, on one system bus, are connected with processor D via the S-DPRAM, sharing the I/O devices. For one of processors A, B, and C to drive the I/O controller normally, one processor must occupy the I/O device for a specific period. The following procedure gives processor A control of the printer (figure 24).

1. Processor A writes its ID code into the ARB register to acquire bus ownership of port A. After acquiring the bus ownership, processor A starts to transfer data to processor D dia the S-DPRAM, and then processor D drives the printer.
2. Since processor A performs data output on a page basis, there is an interval between one page and another when no data transfer is executed. The system bus is released to the other processors so as not to decrease system throughput.
3. However, during this interval, if processor B attempts to write its ID code into ARB, it is rejected because ARB already has the ID code of processor A (figure 25). Thus, this effective arbitration mechanism prevents processor B from using the S-DPRAM.
4. Thereafter, the bus ownership is returned to processor A and data transfer is resumed. After finishing a series of processes, processor A clears ARB by writing its ID code into ARB again, and bus ownership is then released.

Table 18. Arbitration Function

	Related Register	Mode	Function	Application
CPU occupancy function	ARB (Arbitration ownership register)	DPRAM FIFO	Requesting processor writes its ID code into ARB. The processor can acquire the S-DPRAM if the request is accepted. If ARB contains another ID code, the request is rejected.	Determines which CPU among multiple CPUs on one system bus can acquire the shared resource S-DPRAM
Semaphore function	AQR, REL (Acquire/ ownership, release/ available registers)	DPRAM	Setting AQR register bit indicates that the S-DPRAM is currently acquired by the CPU of the requesting port, and that the opposite port request is to be rejected. Ownership is released by setting the corresponding bit of REL.	Determines which CPU on which system bus can acquire the shared resource S-DPRAM

5

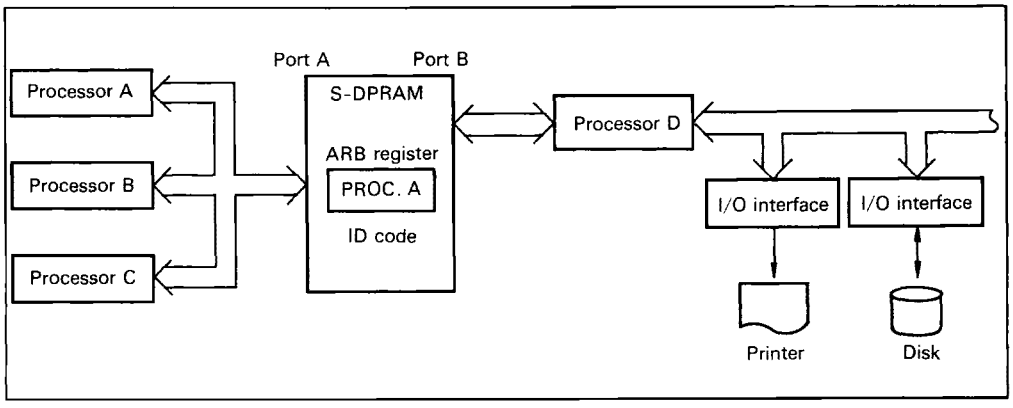


Figure 23. Multiprocessor System Configuration Example (CPU Occupancy Function)

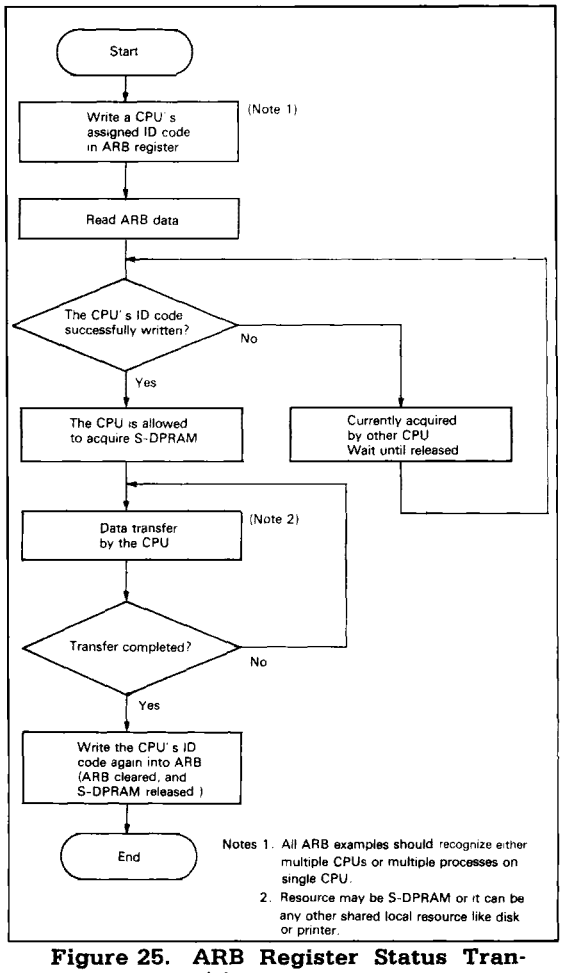


Figure 25. ARB Register Status Transition

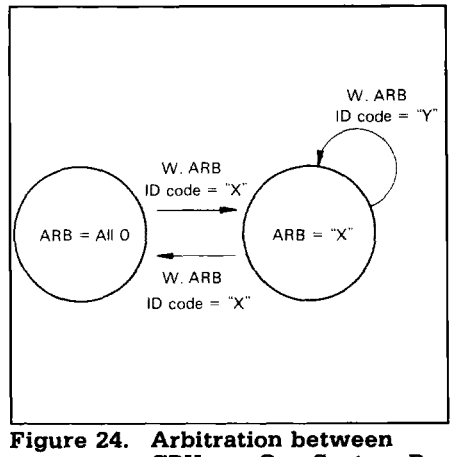


Figure 24. Arbitration between CPUs on One System Bus

Semaphore Function

For a system sharing a resource between two system buses, the S-DPRAM AQR (acquire ownership) and REL (release/available) registers arbitrate access contention in the case of simultaneous requests from both ports. These registers have the following functions and are used as a pair.

- AQR register: Represents ownership of one port against the opposite port
- REL register: Relinquishes the acquired ownership

The AQR register implements "test and set" logic circuit in hardware. "Test" logic checks the current ownership status of a resource, and "set" logic establishes the ownership of a resource before it can be acquired by another requestor.

Each register provides eight semaphores, which can discrete control eight shared resources. For instance, if 1024-byte memory is divided into eight areas, each area can be represented by a corresponding semaphore bit.

This mechanism can be applied not only to the memory of the S-DPRAM itself but also other I/O devices.

Figure 26 shows an example of a multiprocessor system configuration. In this system, processors A and B, which are respectively allocated to ports A and B, share two I/O devices and two blocks of memory in the S-DPRAM. These four resources are controlled by referring to the four corresponding semaphore bits (figure 27).

1. Processor A attempts to write a 1 into the corresponding semaphore bit AA3 to read out data from a disk. If it is successful, gate GA3 is opened and connected to the system bus of processor A. Data transfer can then be initiated between processor A and the disk.
2. At this time, if processor B attempts to write a 1 into AA3 in order to access the same disk, it is not allowed and processor B must wait until the disk is released from processor A (figure 28).
3. However, since shared resources other than this disk are free, processor B can acquire them by writing a 1 into AB0-AB2.
4. In the same way, processor A cannot access the shared resources currently acquired by processor B.
5. After each processor has completed its desired processing, the AQR register bit can be cleared by writing a 1 into the same bit position of the REL register; the corresponding shared resource is then released.

In this way, resources shared between two system buses can be effectively controlled via the semaphore registers. Note that this arbitration mechanism can also be applied to a single processor executing multiple tasks by means of time sharing.

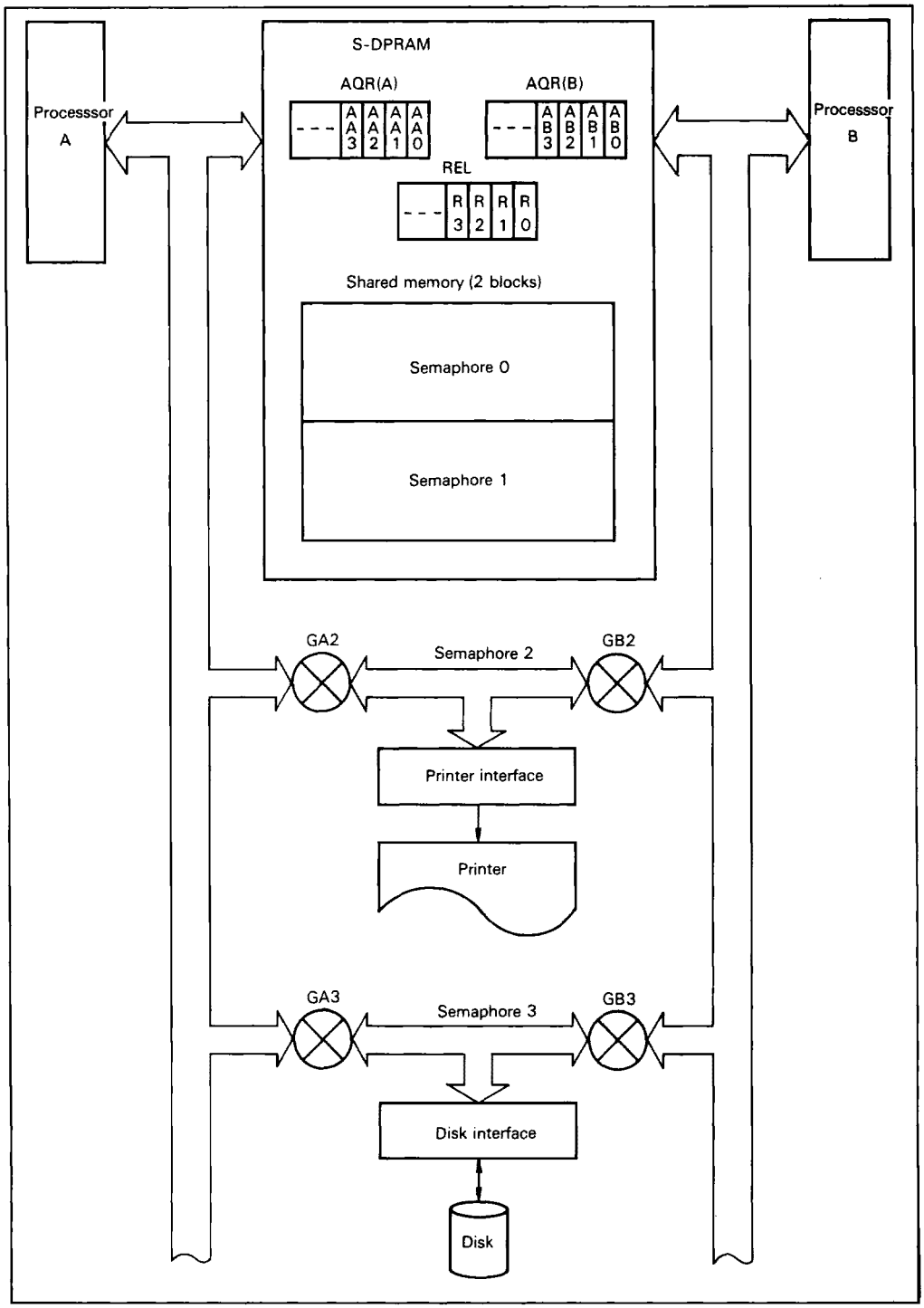


Figure 26. Multiprocessor Configuration Example (Semaphore Function)

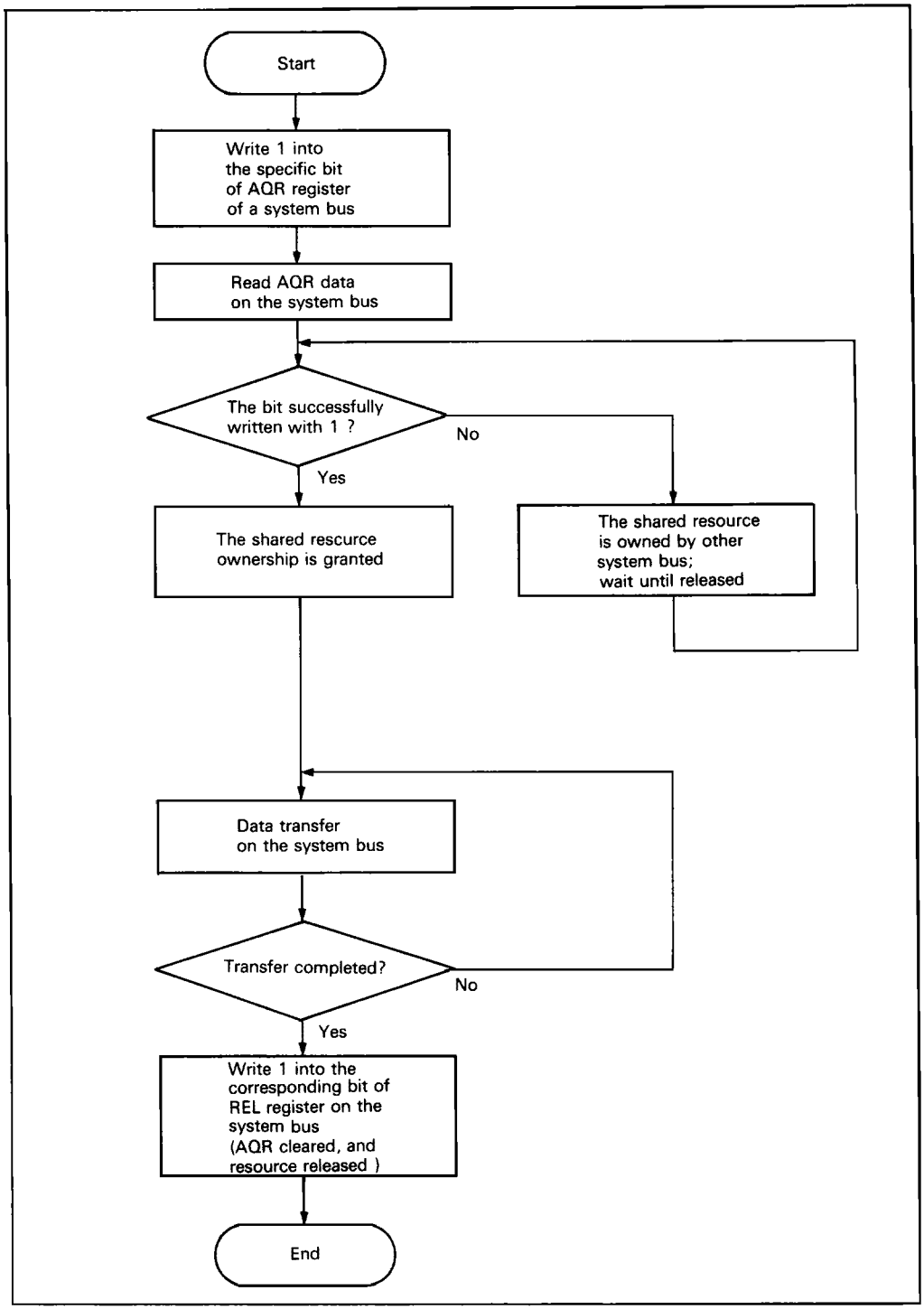
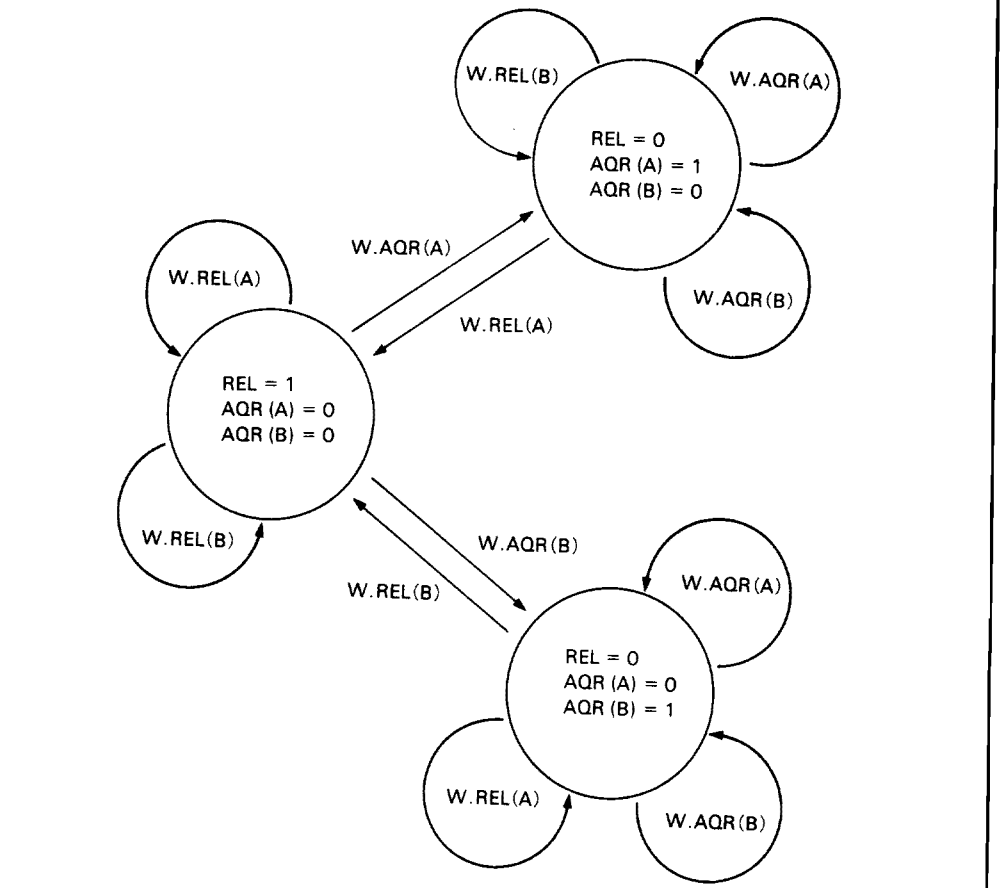


Figure 27. Arbitration Between Two System Buses

5



Note: W. REL (A): Write 1 to bit of REL (A)
W. REL (B): Write 1 to bit of REL (B)
W. AQR (A): Write 1 to bit of AQR (A)
W. AQR (B): Write 1 to bit of AQR (B)

Figure 28. Semaphore Register Status Transition

Internal Operation Description

Port A and port B can independently read and write internal registers and the 1024-byte RAM array of the S-DPRAM. Access requests from both ports are entirely asynchronous, and therefore access contention may occur. To avoid this contention, the internal arbiter determines the order in which the requests are serviced.

It should be noted that processing start timing may be somewhat delayed in the following cases:

1. Handling access contention between both ports
2. Waiting until busy state, which is caused by a buffered write operation

To handle this problem, the S-DPRAM $\overline{\text{READY}}$ signal indicates whether a posted request can be accepted or not. Using $\overline{\text{READY}}$, an effective interface circuit can be organized to properly control timing.

Write Operation

The S-DPRAM implements a buffered write operation. It is initiated by $\overline{\text{WRS}}$ assertion. When $\overline{\text{WRS}}$ is negated, write data is temporarily latched in a buffer register. Thereafter, the write operation can proceed according to the internal sequence. (SCR write and FIFO write are excluded.) Buffered write is characterized as follows:

1. Actual processing is executed during $\overline{\text{WRS}}$ high, and therefore $\overline{\text{READY}}$ is not delayed even when access contention occurs. In general, the negation period is longer than the assertion period in MPU access. Consequently, cycle times can be used effectively and system throughput can be improved.
2. For an interface between systems having different cycle times, one cycle of one port may last as long as several cycles of the other port. While $\overline{\text{WRS}}$ is asserted, an access contention cannot occur, and therefore system throughput is not reduced.

In the buffered write operation, actual processing starts while $\overline{\text{WRS}}$ is negated. Consequently, the next request cannot be serviced until the previous is finished. This state is called a busy state, and the busy flag is set accordingly. Figure 29 shows the S-DPRAM write operation sequence.

Read Operation

$\overline{\text{RDS}}$ assertion initiates read operations. In comparison with the write operation, read operation is entirely completed during $\overline{\text{RDS}}$ low. Consequently, busy states cannot occur. Figure 30 shows the S-DPRAM read operation sequence.

5

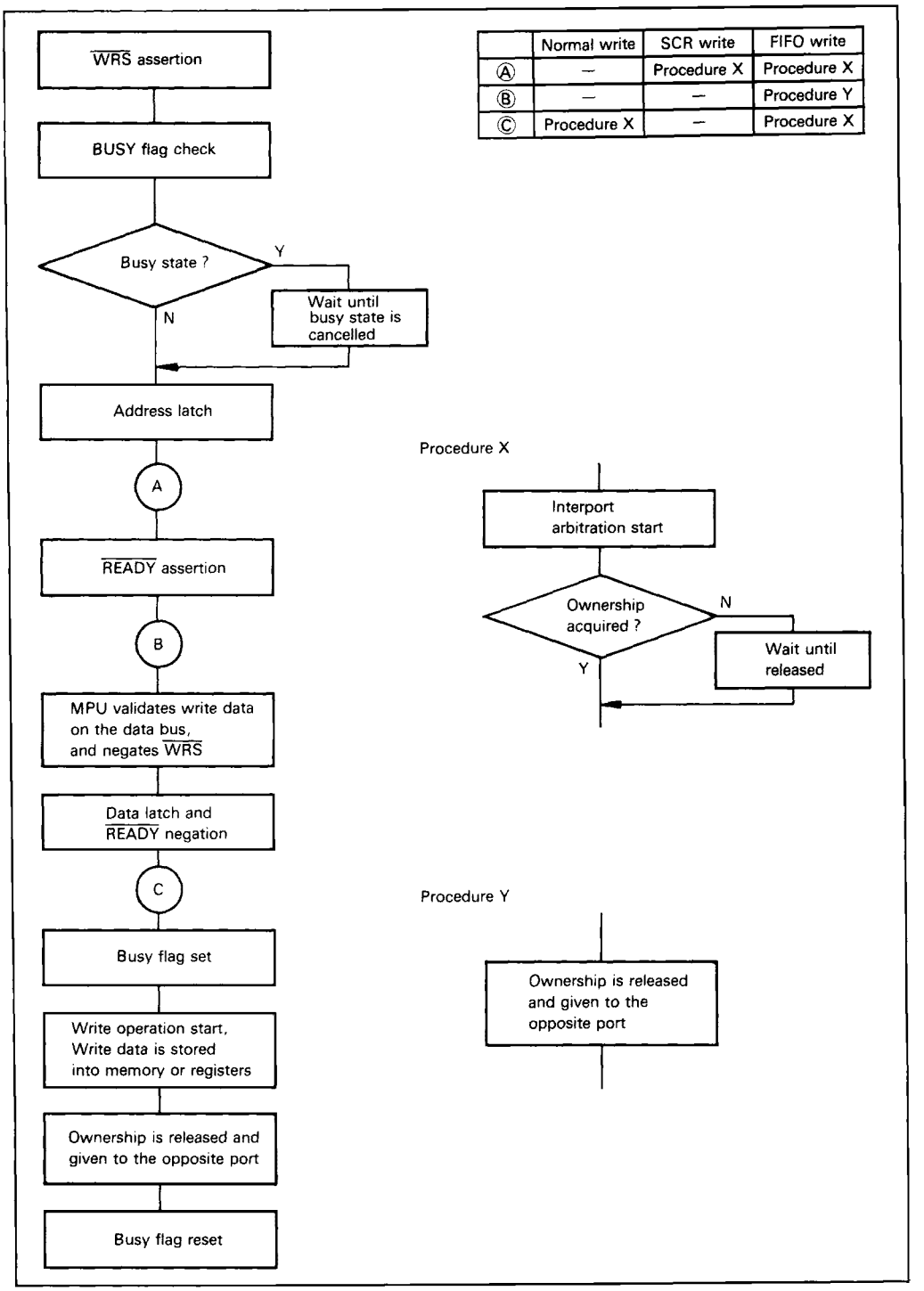


Figure 29. Write Transaction

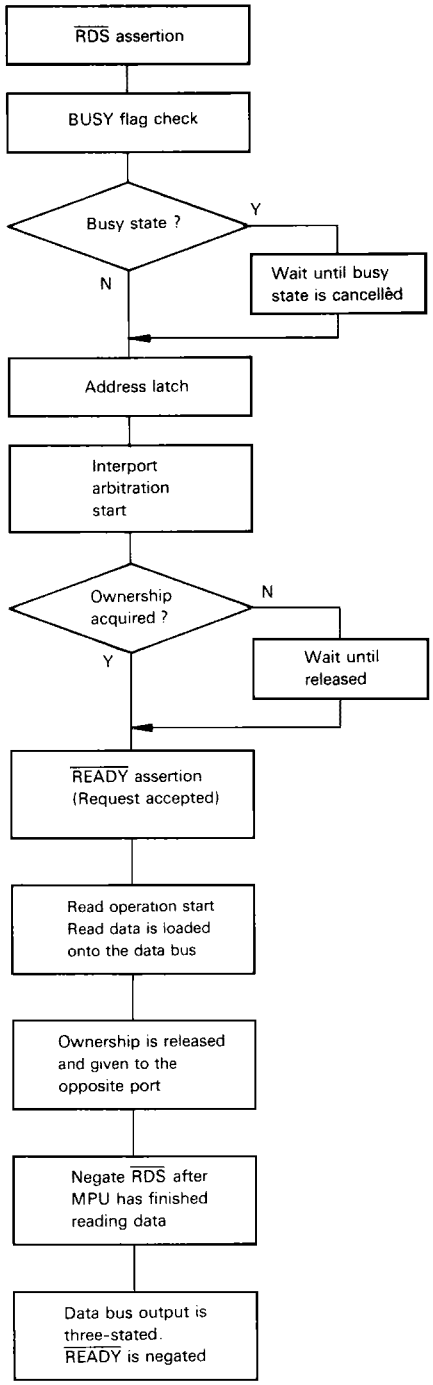


Figure 30. Read Transaction

Processing Mode

The S-DPRAM implements four types of processing modes: SCR white, FIFO write, normal write, and normal read. One of these modes is selected corresponding to memory or registers to be accessed. Figure 31 shows the timing chart for each mode.

SCR Write (Special Control Register Write)

Registers which directly control the S-DPRAM itself or are closely related to its operation are called special control registers (SCR). In SCR write mode, arbitration is performed at the falling edge of $\overline{WRS}$, and thereafter ownership is retained even during $\overline{WRS}$ or low. Consequetly, access requests from the opposite port cannot be accepted during $\overline{WRS}$ low. In other words, $\overline{READY}$ of the opposite port cannot be asserted until $\overline{WRS}$ is negated.

This mode is selected by directly accessing the registers in table 19. However, it should be noted that normal write mode is selected if they are indirectly accessed.

FIFO Write

The S-DPRAM is set to FIFO mode by setting a bit field from 001-111 into RCFG of the CNFG register. In FIFO mode, FIFOs can be read written to accessing data registers (table 20). In FIFO write mode, arbitration is performed at the falling edge of $\overline{WRS}$; ownership is then released after checking FIFO status during $\overline{WRS}$ low. Thereafter, arbitration starts again at the rising edge of $\overline{WRS}$, and data write is executed during $\overline{WRS}$ high.

Table 19. Special Control Registers

	R0	R1	R2	R3	RA	RB
Direct addressing (DPRAM mode, FIFO mode)	CNFG (COM)	CMD (A/B)	IEN (A/B)	ISRC (A/B)	FRC (A/B)	SCFE (A/B)

Table 20. FIFO Write Data Registers

	R8	R9
Indirect addressing (FIFO mode)	P(A/B)DR0	P(A/B)DR1

Normal Write

Write accesses other than SCR write and FIFO write are all designated as normal writes. That is, write accesses to registers other than SCR (by direct access) or to RAM, and all write accesses executed by indirect addressing belong to this mode (table 21). In this mode, arbitration is performed at the rising edge of $\overline{WRS}$ and data write is executed during $\overline{WRS}$ high (buffered write). Therefore, $\overline{READY}$ is not delayed even in the case of simultaneous access requests, except that processing from the previous cycle is continued.

Normal Read

All read accesses are made in normal read mode (table 22). Data read is completed during $\overline{RDS}$ low, and $\overline{READY}$ cannot be delayed due to busy states in the next cycle. In this mode, arbitration is performed at the falling edge of $\overline{RDS}$, and data read is executed in the order determined through arbitration. When the read data becomes valid, ownership is given to the opposite port after the data is latched into the output registers, and $\overline{READY}$ is then asserted. Therefore, the opposite port is not kept waiting by strobe conditions as for SCR.

Table 21. Normal Write Registers and RAM

	Register						RAM
Direct addressing (DPRAM mode, FIFO mode)	R4 STS (A/B)	R5 ARB (A/B)	R6 AQR (A/B)	R7 REL (COM)	RC-RF Address register (A/B)	R10-R2F Parameter register (COM)	M000 ⋮
Indirect addressing (DPRAM mode)	R0-RF: Requesting port's control register R10-R2F: Shared parameter register R30-R3F: Opposite port's control register						⋮ M3FF

Table 22. Normal Read Register and RAM

	Register	RAM
Direct addressing (DPRAM mode, FIFO mode)	R0-RF: Requesting port's control register R10-R2F: Shared parameter register R30-R3F: Opposite port's control register	M000 ⋮ M3FF
Indirect addressing (DPRAM mode, FIFO mode)		

5

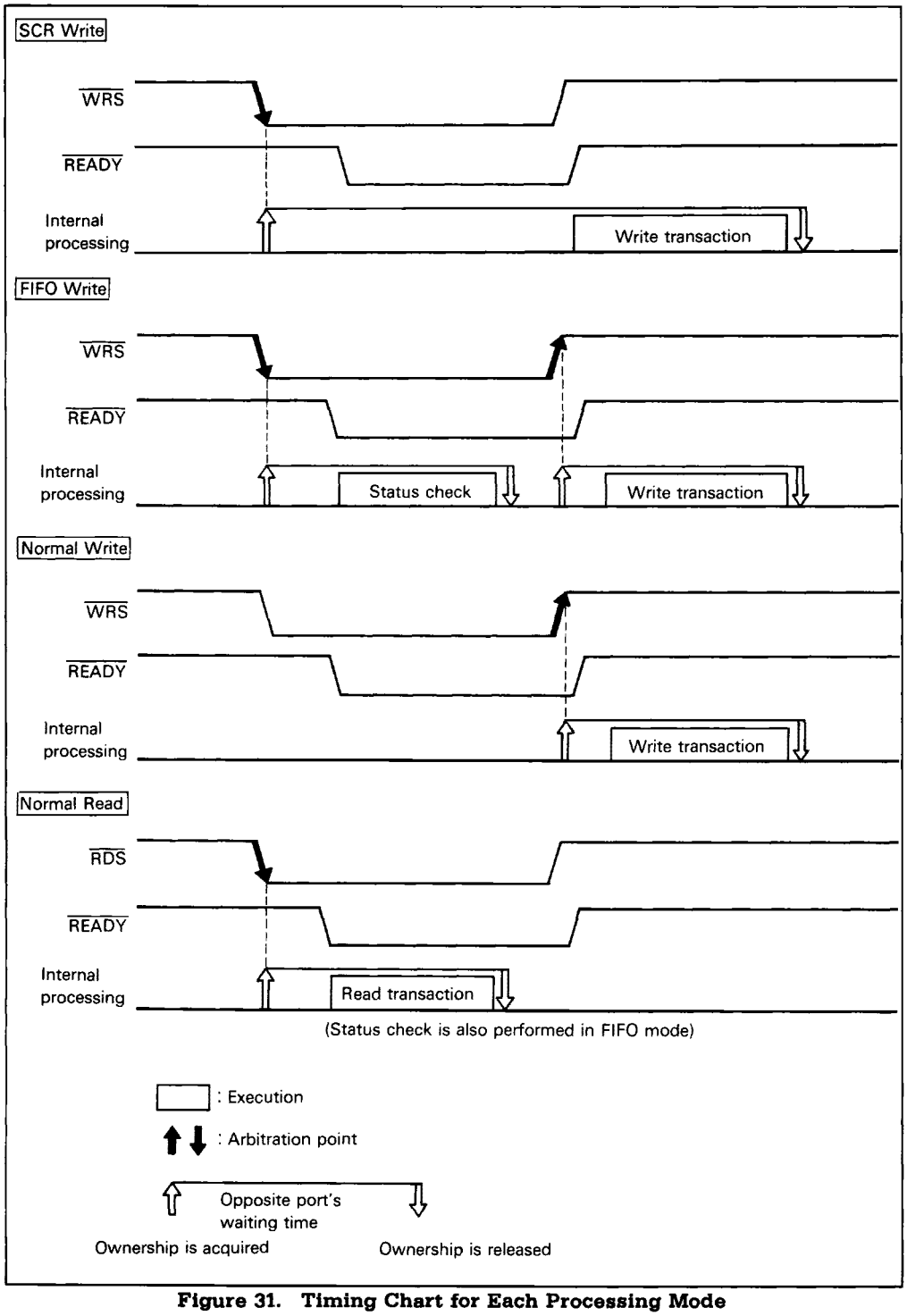


Figure 31. Timing Chart for Each Processing Mode

S-DPRAM Expansion

S-DPRAM Capacity Expansion

The S-DPRAM incorporates a 1024-byte RAM, but its capacity can be expanded if necessary. Figure 32 shows an example of a system with a 2-kbyte RAM.

S-DPRAM capacity can be expanded in the same way as other standard memories without any restrictions. In a RAM-expanded system, there must be a mechanism to control which S-DPRAM is accessed. This is done by combining CS (chip select), obtained by decoding the upper address, and $\overline{RDS}/\overline{WRS}$.

$\overline{IRQ}$ of each S-DPRAM can be processed independently, or processed jointly by connecting all $\overline{IRQ}$ lines together. $\overline{IRQ}$ is an open-drain output, and can easily be wired-ORed with a pull-up resistor.

Each S-DPRAM is assigned to a different address, and thus can be exclusively accessed. Therefore, negative logic OR is applied to all S-DPRAM $\overline{READY}$s to return $\overline{READY}$ to the MPU when at least one $\overline{READY}$ is asserted. That is, $\overline{READY}$ lines are connected by an AND gate.

S-DPRAM capacity can be expanded in DPRAM mode only. For the S-DPRAM in FIFO mode, its capacity cannot be expanded since the functions related to expansion of the $\overline{FRQ}$ signal, etc, are not supported.

S-DPRAM Bit Width Expansion

The S-DPRAM basically supports an 8-bit data bus, but its bit width can be expanded to 16 bits or 32 bits. Figure 32 shows an example of a system with a 16-bit data bus.

The bit width can also be expanded in the same way as with other standard memories. $\overline{RDS}$ and $\overline{WRS}$, generated by address control and other input signals, must be connected to each S-DPRAM in parallel.

In this system, data read/write must be executed after both upper byte and lower byte S-DPRAMs are ready for data transfer. Therefore, $\overline{READY}$s must be connected by an OR gate to return $\overline{READY}$ to the MPU when both $\overline{READY}$s are asserted.

Please note that each S-DPRAM can arbitrate independently using its own arbiter. Accordingly, port ownership may be confused between the upper and lower byte S-DPRAMs. For example, when the MPU accesses two S-DPRAM simultaneously, port A is selected for upper bytes and port B for lower bytes, or vice versa. In this way, two S-DPRAMs may respectively acquire different MPUs (figure 34).

5

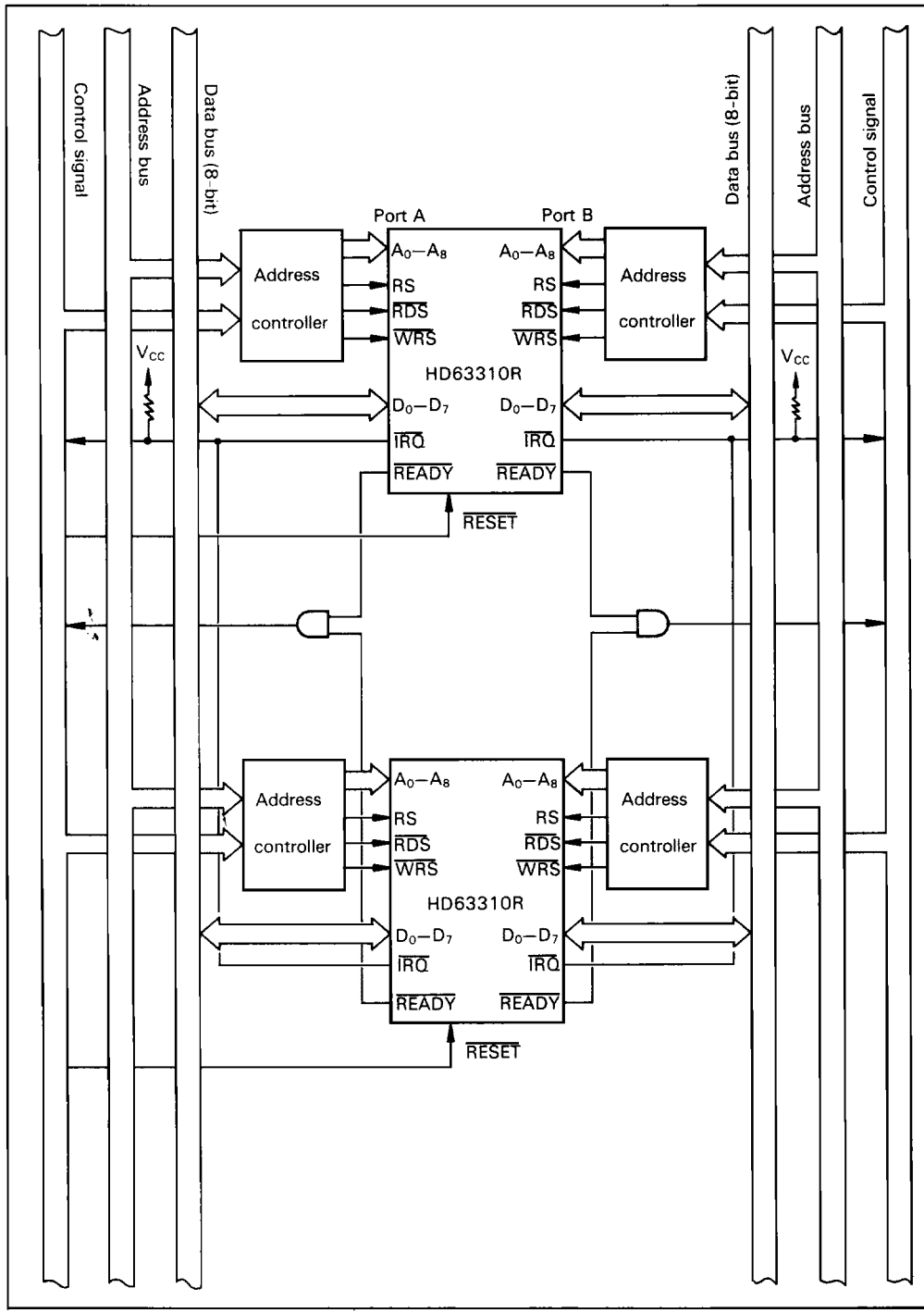


Figure 32. RAM-Expanded HD63310R (DPRAM Mode, 8-Bit Width, Address Expansion)

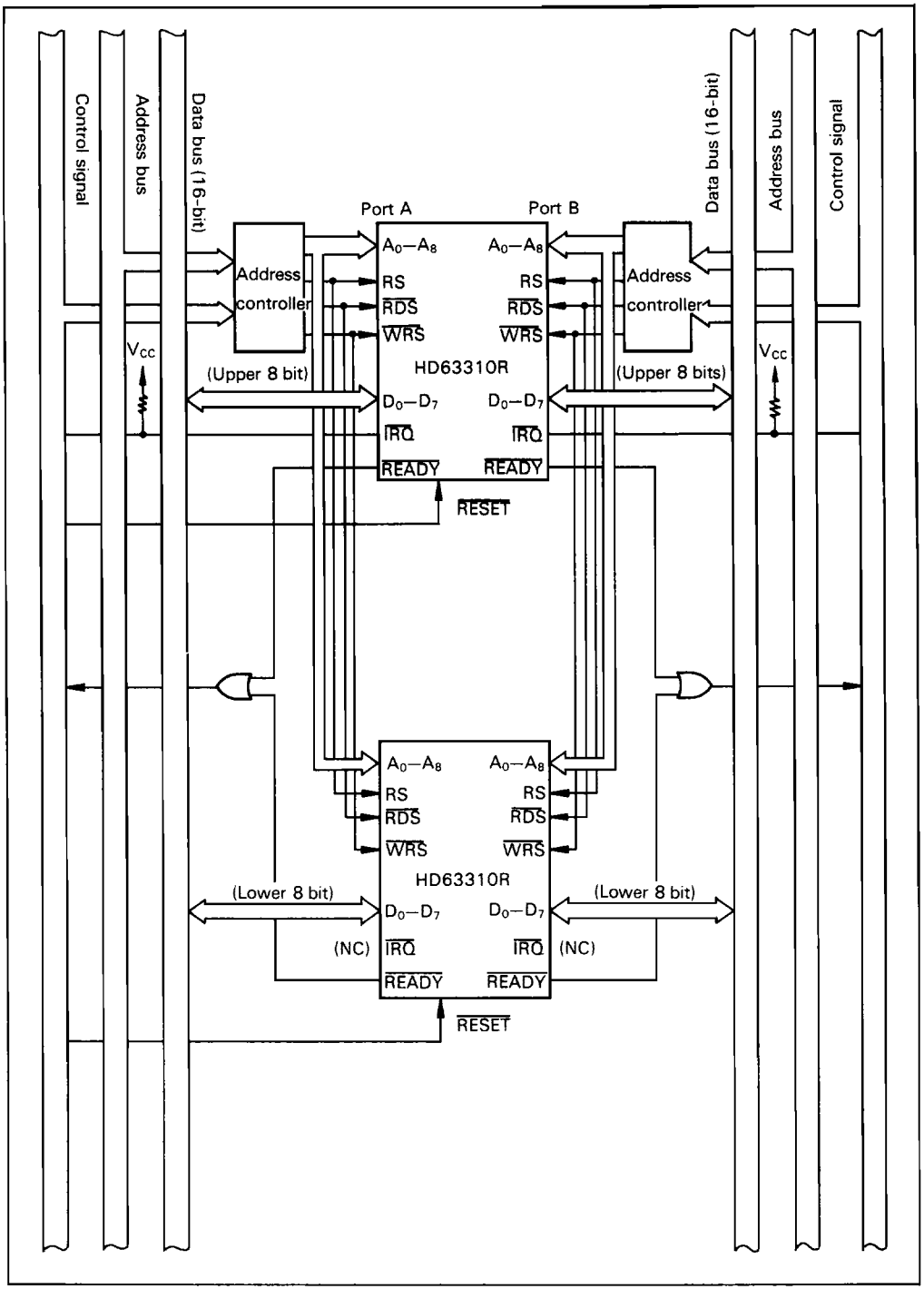


Figure 33. Bit-Width-Expanded HD63310R (DPRAM Mode/FIFO Mode, 16-Bit Width)

5

$\overline{\text{READY}}$ is asserted by the S-DPRAM which has acquired ownership, but it cannot be returned to the MPU until the other $\overline{\text{READY}}$ is asserted. In this case, if one of the S-DPRAMs is writing to an SCR (special control register) by direct addressing, the system may be deadlocked. As described before, if an access contention has occurred in an SCR write, $\overline{\text{READY}}$ of the opposite port is not asserted until SCR write is negated. Therefore, the MPU cannot accept $\overline{\text{READY}}$ and thus cannot negate $\overline{\text{WRS}}$. As a result, the system may be deadlocked.

Special attention must therefore be paid when designing a system performing word access involved in SCR write, as follows:

1. SCR write should be executed by indirect access to that normal write mode is selected.
2. The opposite port should not be accessed during SCR write.
3. SCR write should be executed on a byte basis.

One $\overline{\text{IRQ}}$ is enough to control the system; $\overline{\text{IRQ}}$ of the lower byte S-DPRAM can be disconnected.

Bit width expansion can be applied to the S-DPRAM in FIFO mode. In this case, $\overline{\text{FRQ}}$ signals must be connected by an OR gete, as for $\overline{\text{READY}}$.

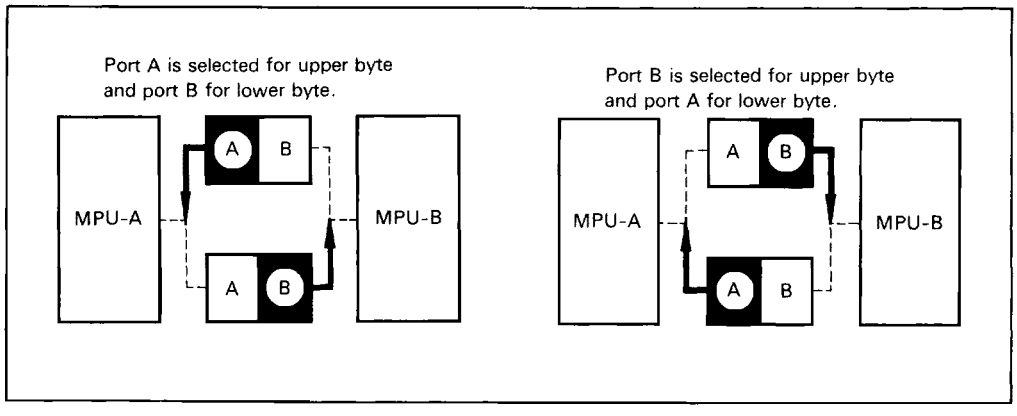


Figure 34. Port Ownership Confusion in Bit-Width-Expanded System

Notes for Use

Upgrading Power Supply Lines

Since the HD63310 is a high-speed CMOS LSI, a large peak current may flow at signal transitions. If the peak current is not adequately handled, a power supply drop or ground electric potential rise may occur. In general, to handle this peak current, a bypass capacitor should be connected to the device. The following lists some recommendations for this bypass capacitor.

1. A bypass capacitor which can handle high frequencies should be used, such as a titanium ceramic tantalum condenser.
2. The bypass capacitor must be located as near as possible to the power supply pins to ensure the smallest bus inductance.
3. Thick gauge wiring should be used to connect the power supply to the circuit board.

$\overline{\text{FRQ}}$ Signal Interface

The HD63310R negates the $\overline{\text{FRQ}}$ signal after completing the $\overline{\text{WRS}}$ cycle, and therefore $\overline{\text{FRQ}}$ acknowledge timing must be specified as after the point of $\overline{\text{WRS}}$ cycle completion.
In general, t_{FNLW} (the $\overline{\text{FRQ}}$ negation delay time at write access) is 150 ns. Taking the delay due to access contention into account, at least 400 ns of delay is required after $\overline{\text{WRS}}$ negation.

Absolute Maximum Ratings

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} (Note 2)	−0.3 to +7.0	V
Input Voltage	Vin (Note 2)	−0.3 to V_{CC} +0.3	V
Allowable Output Current	|Io| (Note 3)	5	mA
Total Allowable Output Current	|Σ Io| (Note 4)	60	mA
Operating Temperature	Topr	0 to +70	°C
Storage Temperature	Tstg	−55 to +150	°C

Notes: 1. Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect the reliability of the LSI.
2. With respect to V_{SS} (system GND)
3. The allowable output current is the maximum current that may be drawn from, or flow out to, one output terminal or one input/output common terminal.
4. The total allowable output current is the total sum of currents that may be drawn from, or flow out to, output terminals or input/output common terminals.

Recommended Operating Conditions

Item	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC} (Note)	4.75	5.0	5.25	V
Input Voltage	V_{IH} (Note)	2.2	–	V_{CC}	V
	V_{IL} (Note)	0	–	0.8	V
Operating Temperature	Topr	0	25	70	°C

Note: With respect to V_{SS} (system GND)

5

Electrical Charcteristics

DC Characteristics (V_{CC} = 5 V ± 5%, V_{SS} = 0 V, Ta = 0°C to +70°C, unless otherwise noted)

Item		Sym.	Min	Max	Unit	Test Condition
Input High Voltage		V_{IH}	2.2	V_{CC}	V	
Input Low Voltage		V_{IL}	V_{SS} −0.3	0.8	V	
Input Leakage Current	$A_0(A)-A_5(A)$, $A_0(B)-A_5(B)$, RS(A), $\overline{RS}$(B), $\overline{RDS}$(A), $\overline{RDS}$(B), $A_8(A)/\overline{AS}(A)$, $A_8(B)/\overline{AS}(B)$, $\overline{WRS}$(A), $\overline{WRS}$(B), $\overline{RESET}$	Iin	−2.5	2.5	μA	Vin = 0 to V_{CC}
Three-State (Off State) Leakage Current	$AD_0(A)-AD_7(A)$, $AD_0(B)-AD_7(B)$, $A_6(A)/A_8(A)\overline{FRQ_0}(A)$, $A_6(B)/A_8(B)/\overline{FRQ_0}(B)$, $A_7(A)/A_9(A)/\overline{FRQ_1}(A)$, $A_7(B)/A_9(B)/\overline{FRQ_1}(B)$	I_{TSI}	−10	10	μA	Vin = 0 to V_{CC}
Output Leakage Current	$\overline{IRQ}$(A), $\overline{IRQ}$(B)	I_{LOH}	−10	10	μA	V_{OH} = V_{CC}
Output High Voltage		V_{OH}	V_{CC} − 1.0	−	V	I_{OH} = −400 μA
			V_{CC} − 0.1	−	V	I_{OH} ≦ −10 μA
Output Low Voltage	$AD_0(A)-AD_7(A)$, $AD_0(B)-AD_7(B)$, $A_6(A)/A_8(A)/\overline{FRQ_0}(A)$, $A_6(B)/A_8(B)/\overline{FRQ_0}(B)$, $A_7(A)/A_9(A)/\overline{FRQ_1}(A)$, $A_7(B)/A_9(B)/\overline{FRQ_1}(B)$, $\overline{READY}$(A), $\overline{READY}$(B)	V_{OL}	−	0.5	V	I_{OL} = 2 mA
	$\overline{IRQ}$(A), $\overline{IRQ}$(B)		−	0.5	V	I_{OL} = 3.2 mA
Input Capacitance		Cin	−	20	pF	Vin = 0 V Ta = 25°C f = 1.0 MHz
Output Capacitance		Cout	−	20		
Current Dissipation	Operating Mode	I_{CC}	−	30	mA	V_{CC} = 5.25 V V_{IH} = 2.2 V V_{IL} = 0.8 V
	Standby Mode ($\overline{RDS}$, $\overline{WRS}$ = High)		−	1	mA	V_{CC} = 5.25 V V_{IH} = V_{CC} − 0.2 V V_{IL} = 0.2 V

Timing Measurement

The timing measurement point for the output level is defined at the following point throughout this specification:

V_{OL} = 0.8 V, V_{OH} = 2.0 V

The output level at stable condition (DC characteristics) is defined at following point:

V_{OL} (DC) = 0.5 V, V_{OH} (DC) = V_{CC} − 1.0 V

See figure 35.

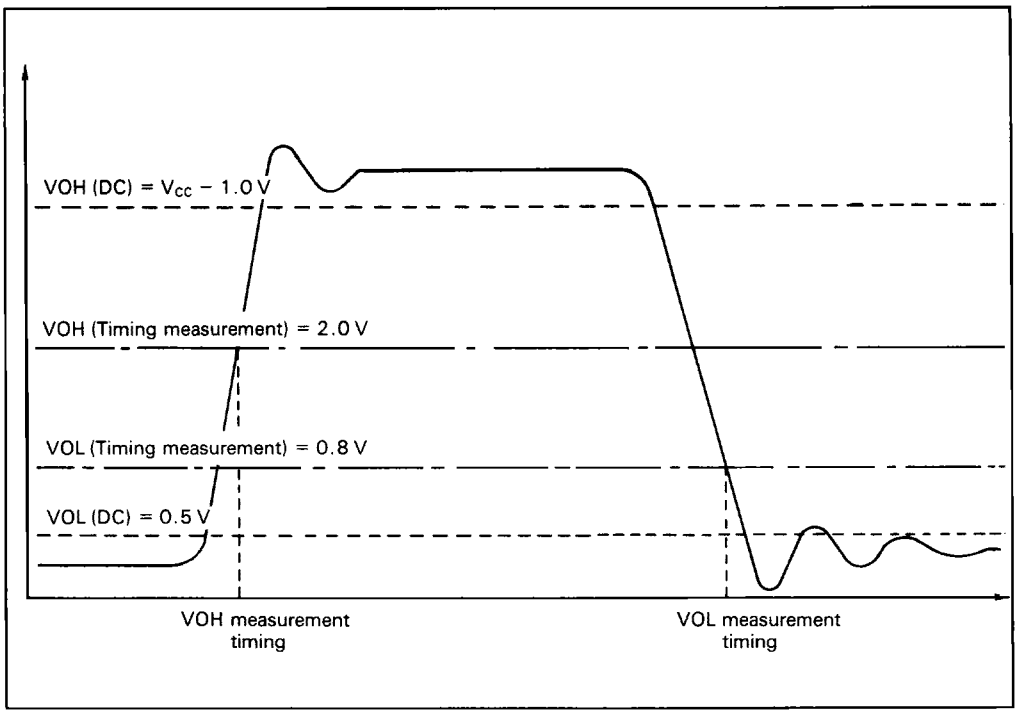


Figure 35. Test Measurement Points

5

AC Characteristics (V_{CC} = 5 V ± 5%, V_{SS} = 0 V, Ta = 0 °C to +70 °C, unless otherwise noted)

Bus Timing

No.	Item		Symbol	HD63310RP20 Min	HD63310RP20 Max	Unit	Figures
1	Address Setup Time (A_0-A_9)		t_{AS}	0		ns	Fig 36-39
2	Address Hold Time (A_0-A_9)		t_{AH}	0		ns	
3	Address Setup Time (RS)		t_{ASRS}	0		ns	
4	Address Hold Time (RS)		t_{AHRS}	15		ns	
5	Address Setup Time ($\overline{AS}$)		t_{ASAS}	40		ns	Fig 37,39
6	Address Hold Time ($\overline{AS}$)		t_{AHAS}	0		ns	
7	$\overline{AS}$ Low Level Width		t_{ASLW}	40		ns	
8	$\overline{AS}$ Setup time		t_{ASS}	0		ns	
9	$\overline{READY}$ Delay Time (Read)		t_{RYRD}		(note 1)	ns	Fig 36,37
10	$\overline{READY}$ Release Time (Read)		t_{RYRH}	0	60	ns	
11	Read Data Delay Time	Direct Addressing Access	t_{DDR}		120	ns	
		Indirect Addressing/ FIFO Access			170	ns	
11A	Read Data Access Time (Note 2)	Direct Addressing Access	t_{ACC}		200	ns	
		Indirect Addressing/ FIFO Access			250	ns	
12	Read Data Hold Time		t_{DHR}	10		ns	
13	Data Bus 3-State Off Time		t_{DTOF}		60	ns	
14	Data Bus 3-State On Time		t_{DTON}		60	ns	
15	$\overline{READY}$ Delay Time (Write)		t_{RYWD}		(note 1)	ns	Fig 38,39
16	$\overline{READY}$ Release Time (Write)		t_{RYWH}	0	60	ns	
17	Write Data Setup Time		t_{DSW}	60		ns	
18	Write Data Hold Time		t_{DHW}	0		ns	
19	$\overline{RDS}$ Hold Time	Direct Addressing Access	t_{RDSH}	120		ns	Fig 36,37
		Indirect Addressing/ FIFO Access		170		ns	
20	$\overline{WRS}$ Hold Time	Except FIFO Access	t_{WRSH}	120		ns	Fig 38,39
		FIFO Access		170		ns	
21	$\overline{RDS}$ Recovery Time		t_{RRC}	70		ns	Fig 36,37
22	$\overline{WRS}$ Recovery Time		t_{WRC}	70		ns	Fig 38,39
23	Recovery Time ($\overline{RDS}$ to $\overline{WRS}$)		t_{RWRC}	70		ns	Fig 44
24	Recovery Time ($\overline{WRS}$ to $\overline{RDS}$)		t_{WRRC}	70		ns	

$\overline{FRQ}$ Output Timing

No.		Item	Symbol	HD63310RP20 Min	HD63310RP20 Max	Unit	Figures
31	Level Mode	$\overline{FRQ}$ Assert Delay Time (Read)	t_{FALR}		300	ns	Fig 40
32		$\overline{FRQ}$ Negate Delay Time (Write)	t_{FNLW}		150	ns	
33		$\overline{FRQ}$ Assert Delay Time (Write)	t_{FALW}		150	ns	
34		$\overline{FRQ}$ Negate Delay Time (Read)	t_{FNLR}		300	ns	
35	Pulse Mode	$\overline{FRQ}$ Negate Delay Time (Write)	t_{FNPW}		250	ns	Fig 41
36		$\overline{FRQ}$ Assert Delay Time (Write)	t_{FAPW}		300	ns	
37		$\overline{FRQ}$ Pulse Width (Write) (Note 4)	t_{FPPW}	100		ns	
38		$\overline{FRQ}$ Negate Delay Time (Read)	t_{FNPR}		300	ns	
39		$\overline{FRQ}$ Assert Delay Time (Read)	t_{FAPR}		80	ns	
40		$\overline{FRQ}$ Pulse Width (Read) (Note 4)	t_{FPPR}	30		ns	
41	Register Write	$\overline{FRQ}$ assert Delay Time	t_{FARW}		150	ns	Fig 43
42		$\overline{FRQ}$ Negate Delay Time	t_{FNRW}		150	ns	

$\overline{IRQ}$, $\overline{RESET}$ Timing

No.	Item	Symbol	HD63310RP20 Min	HD63310RP20 Max	Unit	Figures
51	$\overline{IRQ}$ Response Time (Register Write)	t_{RS1}		300	ns	
52	$\overline{IRQ}$ Release Time	t_{IR}		600	ns	Fig. 47
53	$\overline{IRQ}$ Response Time (FIFO Access)	t_{RS2}		300	ns	
54	$\overline{RESET}$ Low Level Width	t_{RL}	200		ns	Fig. 48
55	$\overline{RDS}$ Wait Time from $\overline{RESET}$ Negate	t_{RESR}	70		ns	Fig. 45
56	$\overline{WRS}$ Wait Time from $\overline{RESET}$ Negate	t_{RESW}	70		ns	

$\overline{RDS}$, $\overline{WRS}$ Timing

No.	Item	Symbol	HD63310RP20 Min	HD63310RP20 Max	Unit	Figures
61	$\overline{RDS}$ Fall Time	t_{RF}		25	ns	Fig 46
62	$\overline{RDS}$ Rise Time	t_{RR}		25	ns	
63	$\overline{WRS}$ Fall Time	t_{WF}		25	ns	
64	$\overline{WRS}$ Rise Time	t_{WR}		25	ns	

Notes: 1. The $\overline{READY}$ signal is used to synchronize, and the delay time depends on the access condition of both ports. For example, when both ports request the data read simultaneously, one port goes to wait state and the $\overline{READY}$ signal delays. A detailed explanation is shown in "READY Timing".
2. Two port access with no contention timing.
3. No. 31-42 include no wait state.
4. No. 37 $\overline{FRQ}$ pulse width depends on No. 20 $\overline{WRS}$ hold time.
No. 40 $\overline{FRQ}$ pulse width depends on No. 19 $\overline{RDS}$ hold time.
5. No. 51-No. 54 include no wait state.

5

$\overline{\text{READY}}$ Delay Timing, No Access Contention (DPRAM/FIFO Mode)

No.	Item		Min	Max	Unit	Figure
R1	Read		—	110	ns	Fig 47-50
R2	Write	Normal	—	110	ns	Fig 47-54
		SCR	—	125	ns	
		FIFO	—	125	ns	

$\overline{\text{READY}}$ Delay Timing, Access Contention (DPRAM Mode)

No.	Item	Contention	Preceding Access	Subsequent Access	Min	Max	Unit	Figure
R3	Read	Read-read	Normal read (Indirect)	Normal read (Dir/Ind)	—	355	ns	Fig 47
R4		Write-read	Normal write (Indirect)	Normal read (Dir/Ind)	—	355	ns	
R5		Write (SCR)-read	SCR write	Normal read (Dir/Ind)	—	250	ns	
R6	Write	Read-write (SCR)	Normal read (Indirect)	SCR write	—	370	ns	Fig 48
R7		Write-write (SCR)	Normal write (Indirect)	SCR write	—	370	ns	
R8		Write (SCR)-write (SCR)	SCR write	SCR write	—	250	ns	

$\overline{\text{READY}}$ Delay Time, Port Busy (DPRAM Mode)

No.	Item	Contention	Preceding Access	Subsequent Access	Min	Max	Unit	Figure
R9	Read	Write-read Port busy	Normal write (Indirect)	Normal read (Dir/Ind)	—	285	ns	Fig 51
R10	Write	Write-write Port busy	Normal write (Indirect)	SCR write	—	285	ns	Fig 52

Notes: 1. $\overline{\text{READY}}$ delay time does not depend on whether the access mode is direct or indirect at nonaccess contention.
2. The $\overline{\text{WRS}}$ arbitration point does not affect the $\overline{\text{READY}}$ delay time in normal mode since it appears at the negative edge, which is at the end of the cycle. Therefore, the $\overline{\text{READY}}$ delay time for normal write contention is the same as for no access contention.
3. For access invlolving SCR write, $\overline{\text{READY}}$ delay time depends on the negative edge of $\overline{\text{RDS}}$ or $\overline{\text{WRS}}$ of the connected port. Please keep this in mind when designing a system. For details, see "S-DPRAM Bit width Expansion" and "Processing Mode".

$\overline{\text{READY}}$ Delay Time, Access Contention (FIFO Mode)

No.	Item	Contention	Preceding Access	Subsequent Access	Min	Max	Unit	Figure
R11	Read	Read-read	FIFO read Normal read FIFO read	FIFO read FIFO read Normal read	—	355	ns	Fig 49
R12		Write-read	FIFO write Normal write FIFO write	FIFO read FIFO read Normal read	—	355	ns	
R13		Write (SCR)-read	SCR write	FIFO read	—	250	ns	
R14	Write	Read-write	FIFO read Normal read FIFO read FIFO read	FIFO write FIFO write Normal write SCR write	—	370	ns	Fig 50
R15		Write-write	FIFO write Normal write FIFO write FIFO write	FIFO write FIFO write Normal write SCR write	—	370	ns	
R16		Write (SCR)-write	SCR write	FIFO write	—	250	ns	

$\overline{\text{READY}}$ Delay Time, Port Busy (FIFO Mode)

No.	Item	Contention	Preceding Access	Subsequent Access	Min	Max	Unit	Figure
R17	Read	Write-read	FIFO write Normal write FIFO write	FIFO read FIFO read Normal read	—	285	ns	Fig 53
R19	Write	Write-write	FIFO write Normal write FIFO write FIFO write	FIFO write FIFO write Normal write SCR write	—	285	ns	Fig 54

5

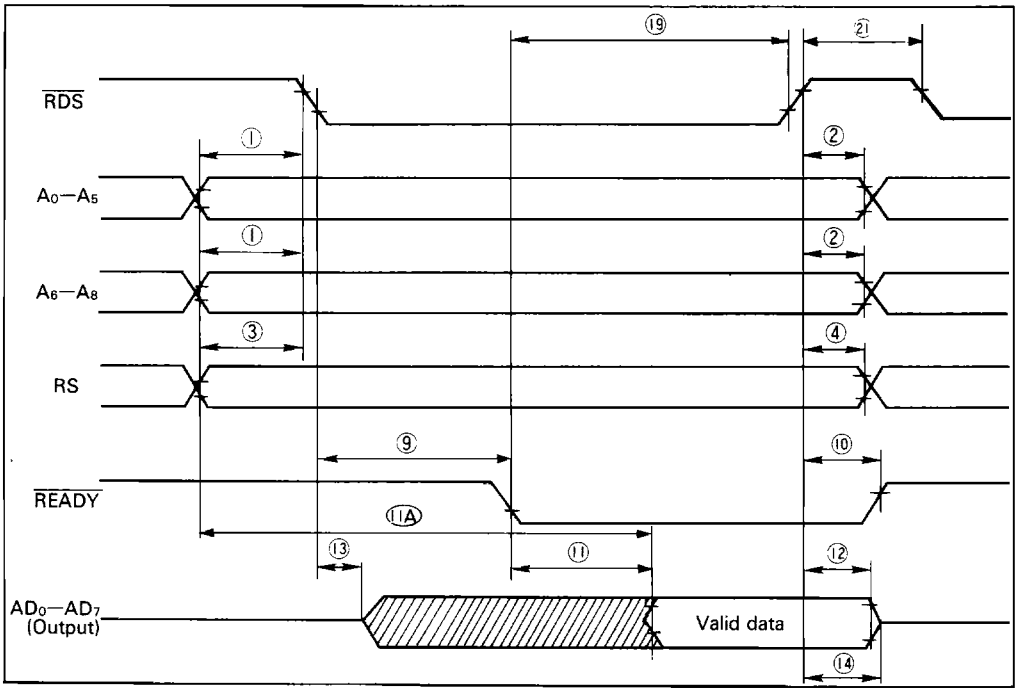


Figure 36. Non-Multiplexed Bus Read Timing

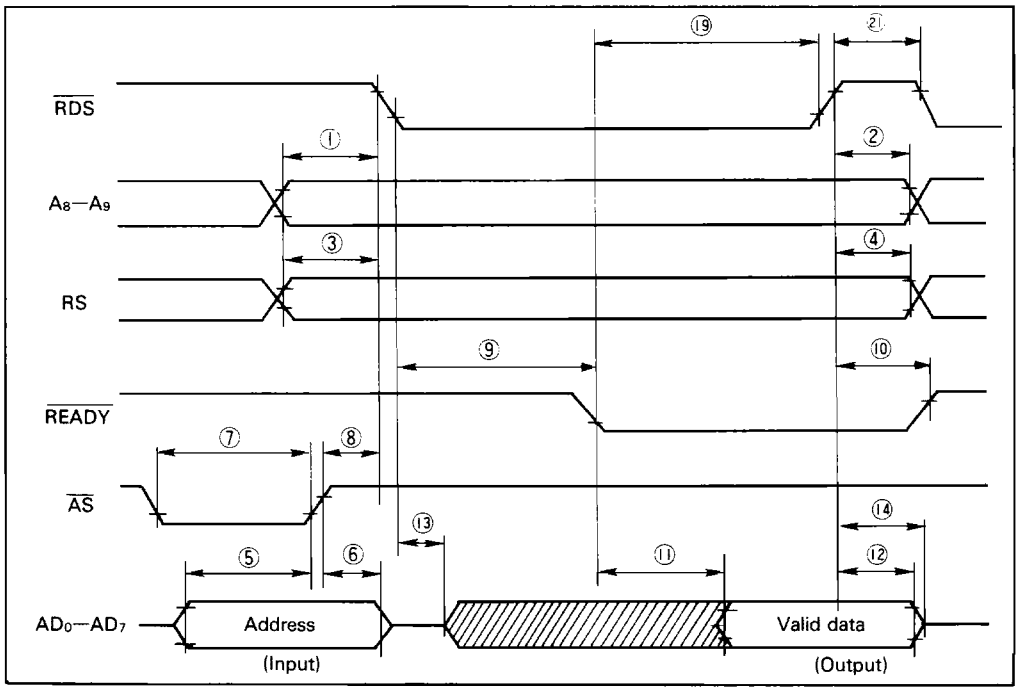


Figure 37. Multiplexed Bus Read Timing

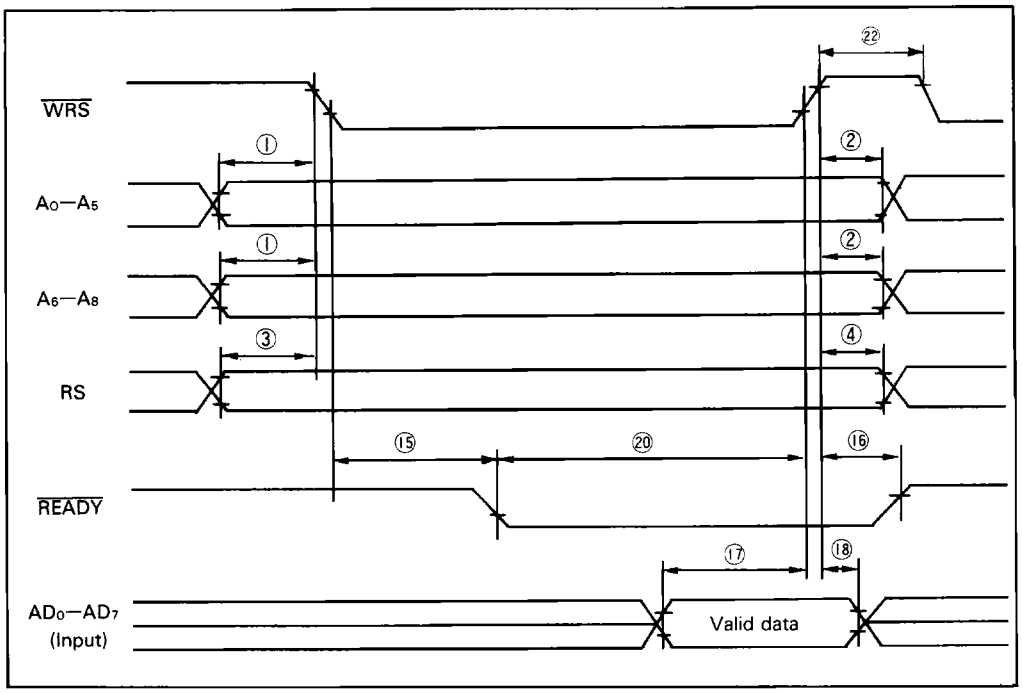


Figure 38. Non-Multiplexed Bus Write Timing

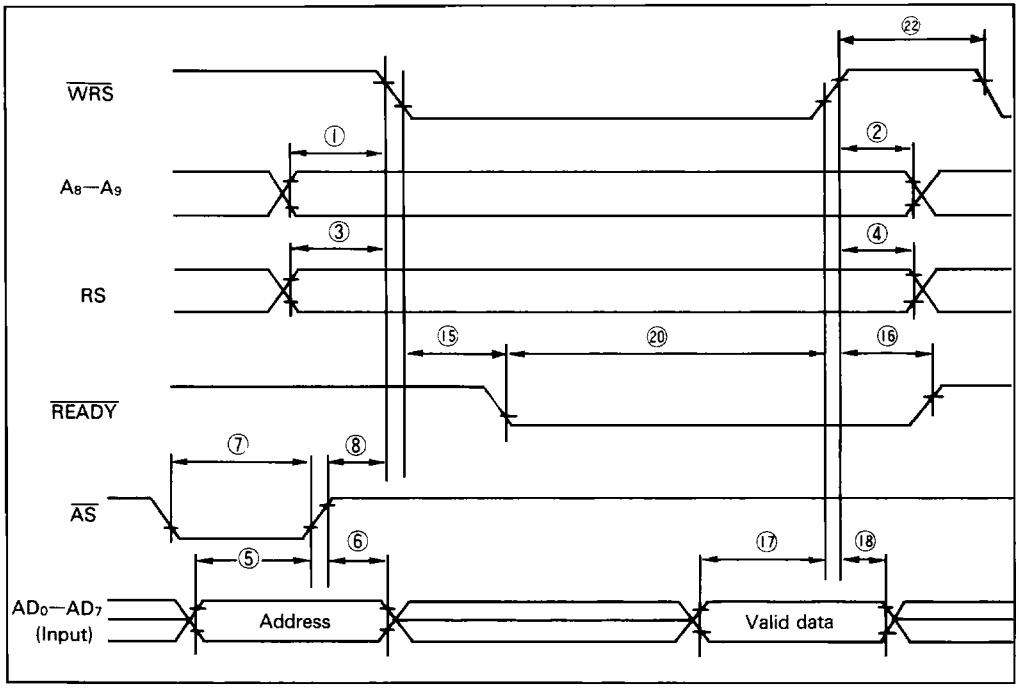


Figure 39. Multiplexed Bus Write Timing

5

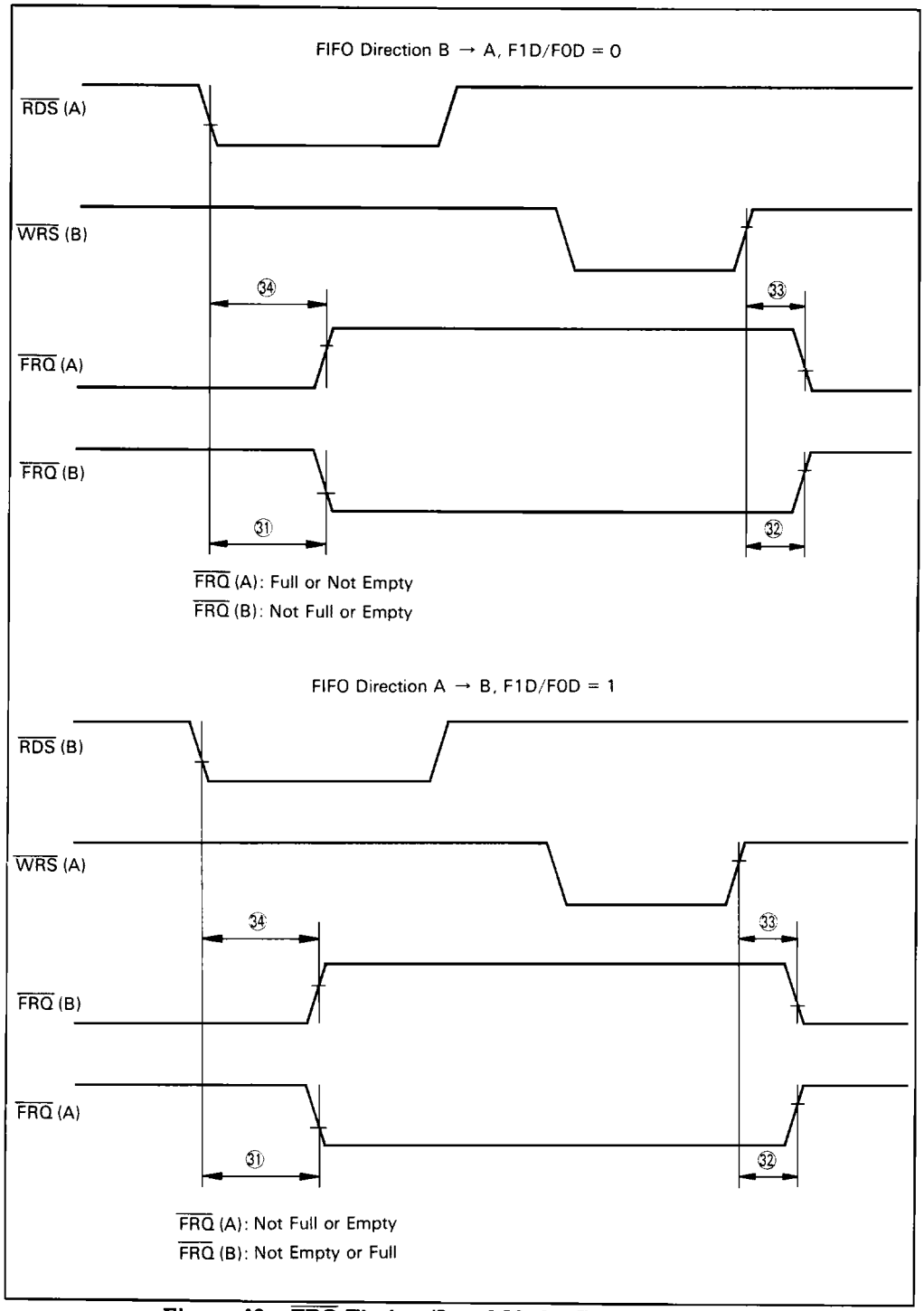


Figure 40. FRQ Timing (Level Mode, FR1M/FR0M = 0)

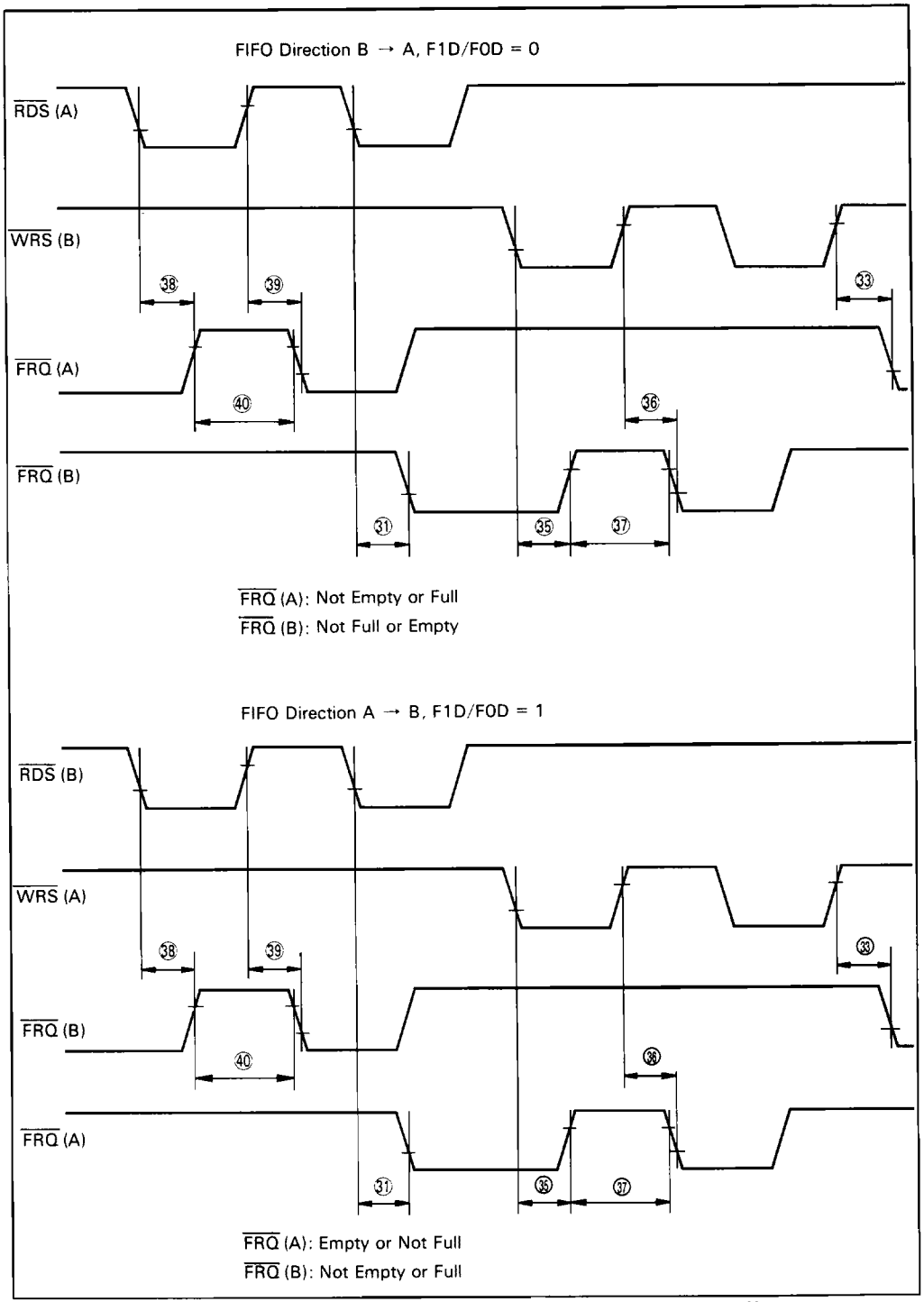


Figure 41. $\overline{FRQ}$ Timing (Pulse Mode, FR1M/FR0M = 1)

5

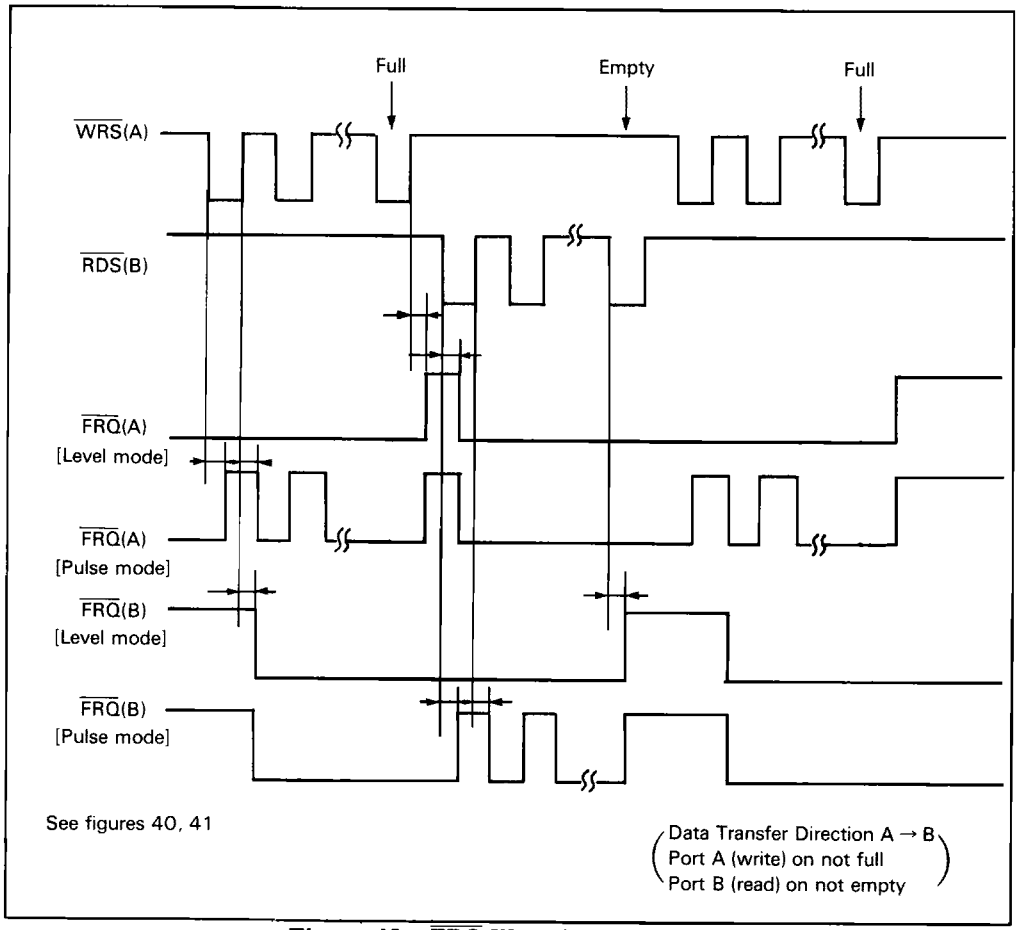


Figure 42. $\overline{\text{FRQ}}$ Waveform Example

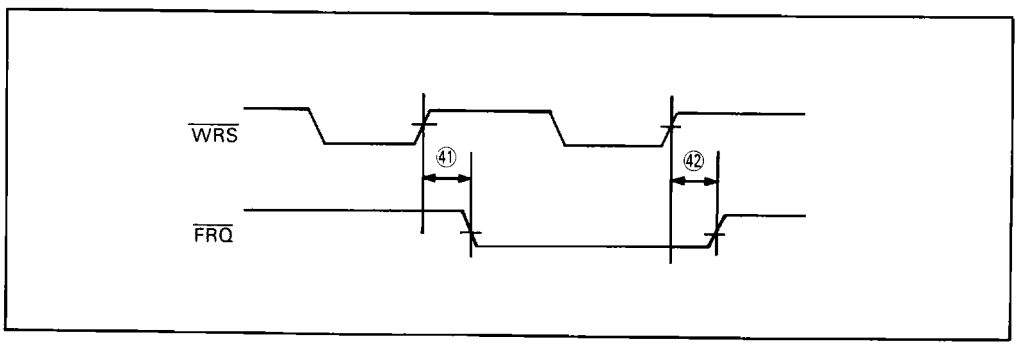


Figure 43. $\overline{\text{FRQ}}$ Timing (FRC Register Write)

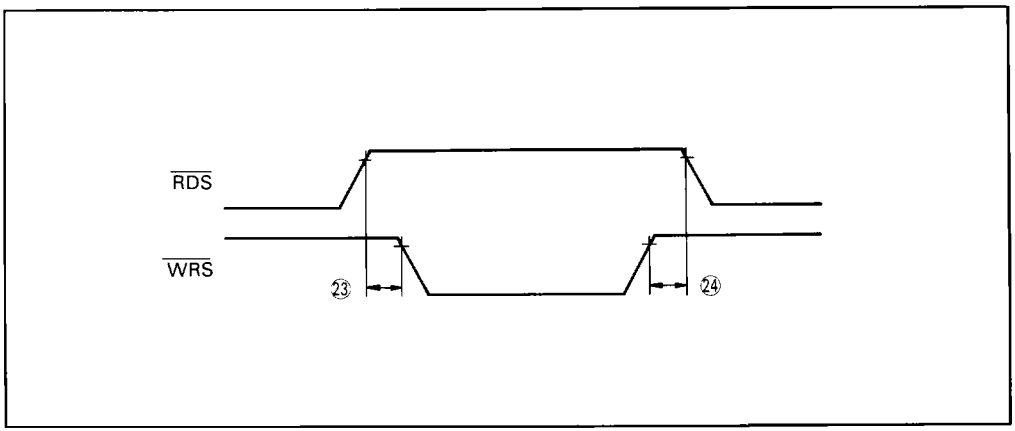


Figure 44. Recovery Time between $\overline{RDS}$ and $\overline{WRS}$ (Same Port)

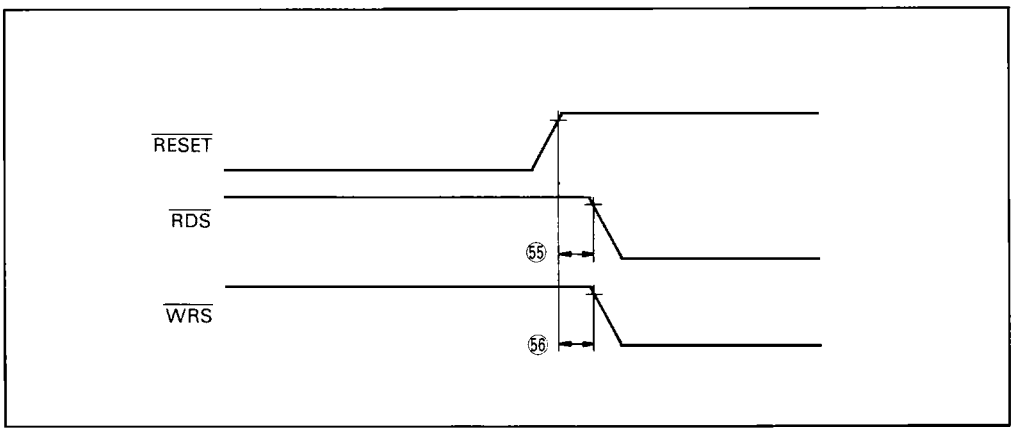


Figure 45. $\overline{RDS}$, $\overline{WRS}$ Wait Time from $\overline{RESET}$ Negate

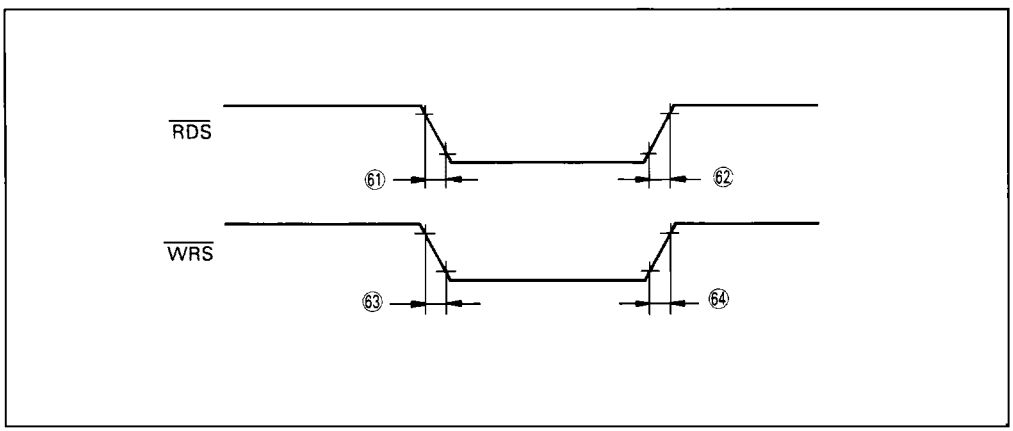
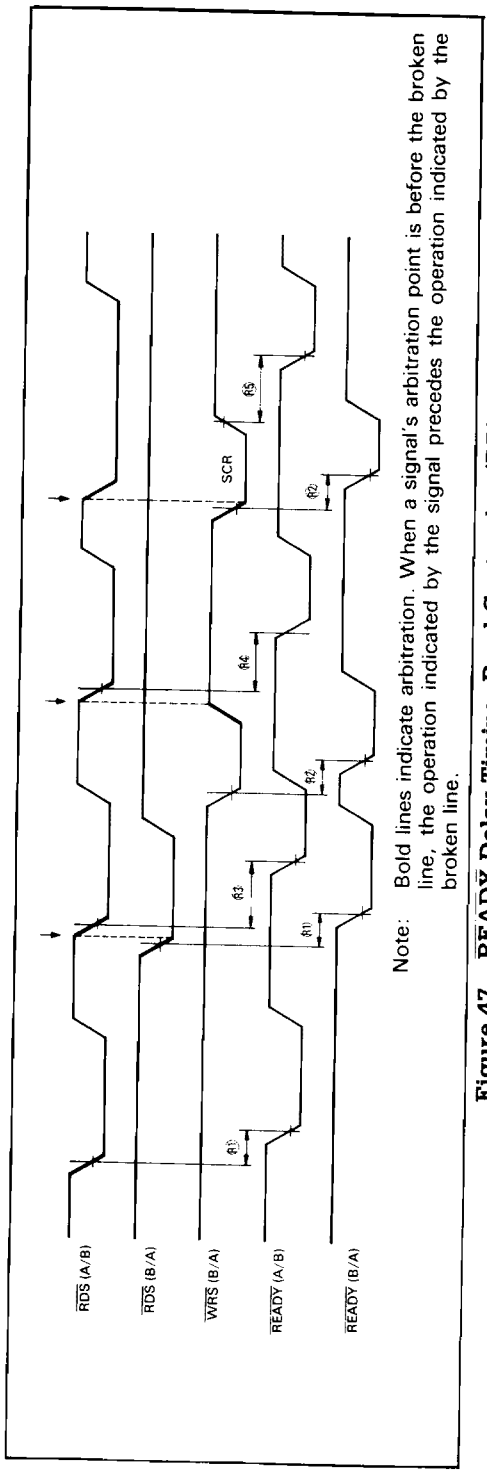


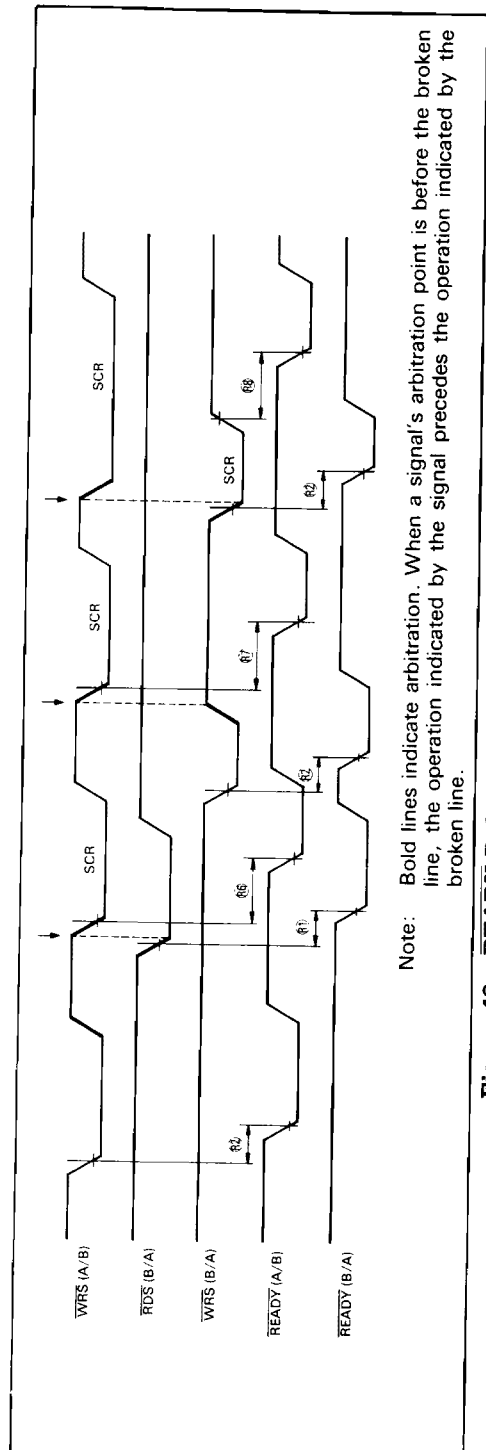
Figure 46. $\overline{RDS}$, $\overline{WRS}$ Timing

5



Note: Bold lines indicate arbitration. When a signal's arbitration point is before the broken line, the operation indicated by the signal precedes the operation indicated by the broken line.

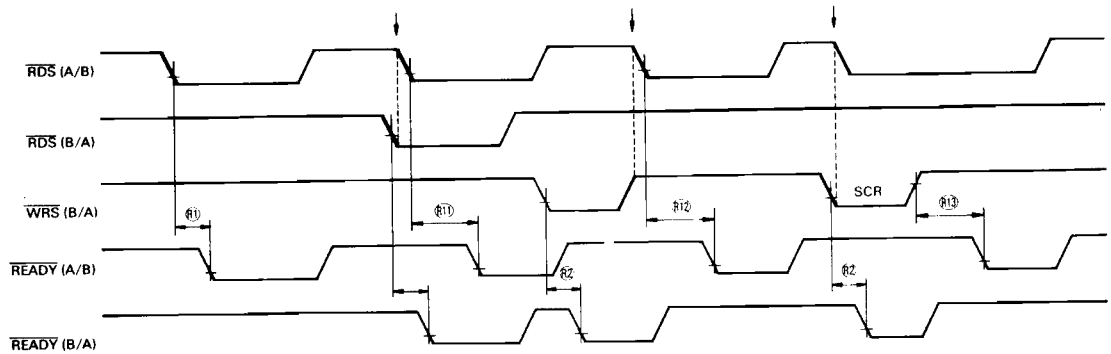
Figure 47. $\overline{\text{READY}}$ Delay Timing, Read Contention (DPRAM Mode)



Note: Bold lines indicate arbitration. When a signal's arbitration point is before the broken line, the operation indicated by the signal precedes the operation indicated by the broken line.

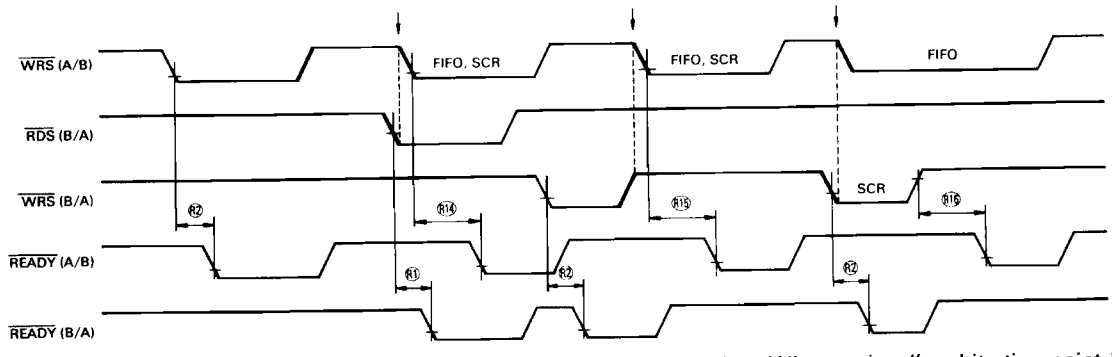
Figure 48. $\overline{\text{READY}}$ Delay Timing, Write Contention (DPRAM Mode)

HITACHI
Hitachi America, Ltd. • Hitachi Plaza • 2000 Sierra Point Pkwy. • Brisbane, CA 94005-1819 • (415) 589-8300



Note: Bold lines indicate arbitration. When a signal's arbitration point is before the broken line, the operation indicated by the signal precedes the operation indicated by the broken line.

Figure 49. $\overline{\text{READY}}$ Delay Timing, Read Contention (FIFO Mode)



Note: Bold lines indicate arbitration. When a signal's arbitration point is before the broken line, the operation indicated by the signal precedes the operation indicated by the broken line.

Figure 50. $\overline{\text{READY}}$ Delay Timing, Write Contention (FIFO Mode)

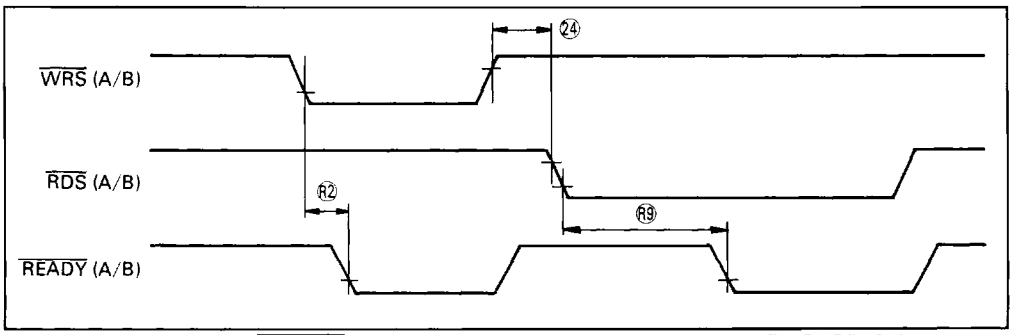


Figure 51. $\overline{\text{READY}}$ Delay Timing, Read Port Busy (DPRAM Mode)

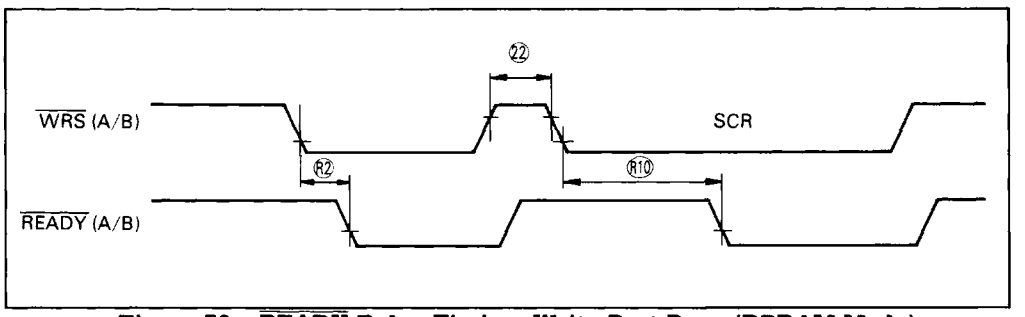


Figure 52. $\overline{\text{READY}}$ Delay Timing, Write Port Busy (DPRAM Mode)

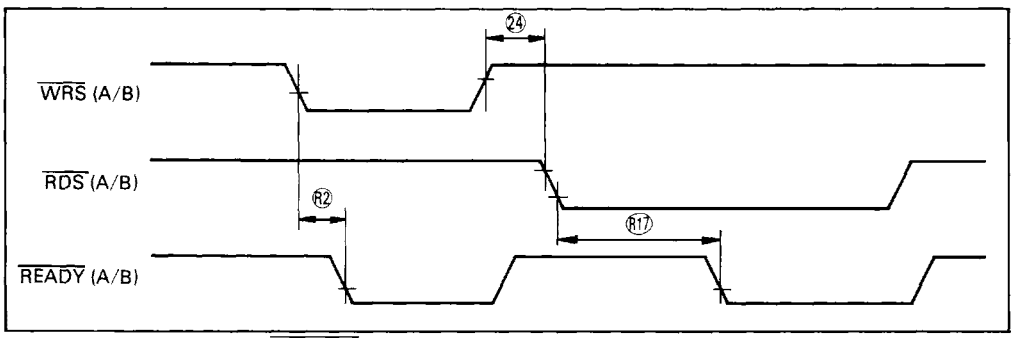


Figure 53. $\overline{\text{READY}}$ Delay Timing, Read Port Busy (FIFO Mode)

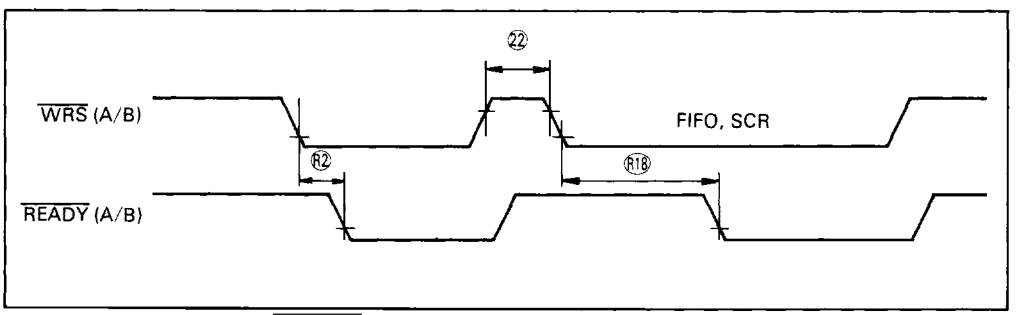


Figure 54. $\overline{\text{READY}}$ Delay Timing, Write Port Busy (FIFO Mode)

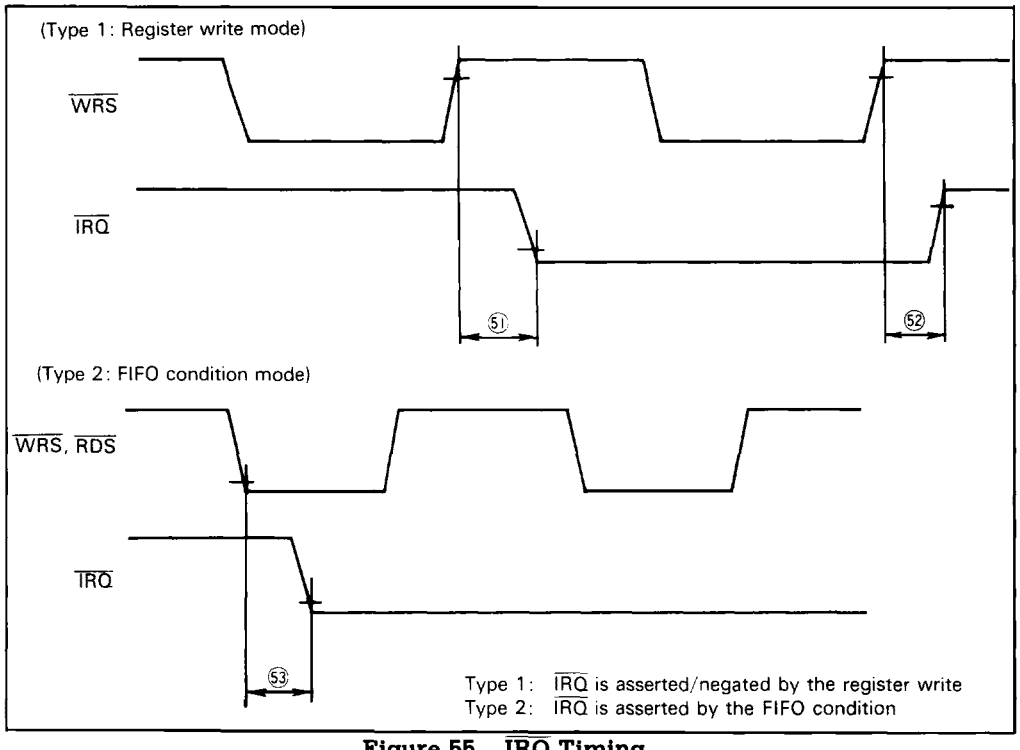


Figure 55. IRQ Timing

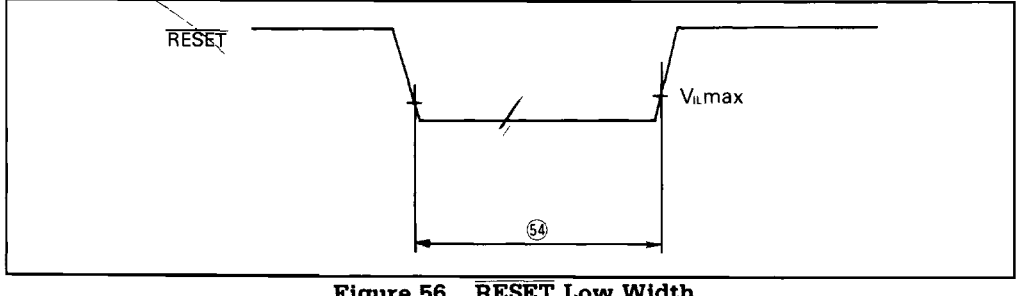


Figure 56. RESET Low Width

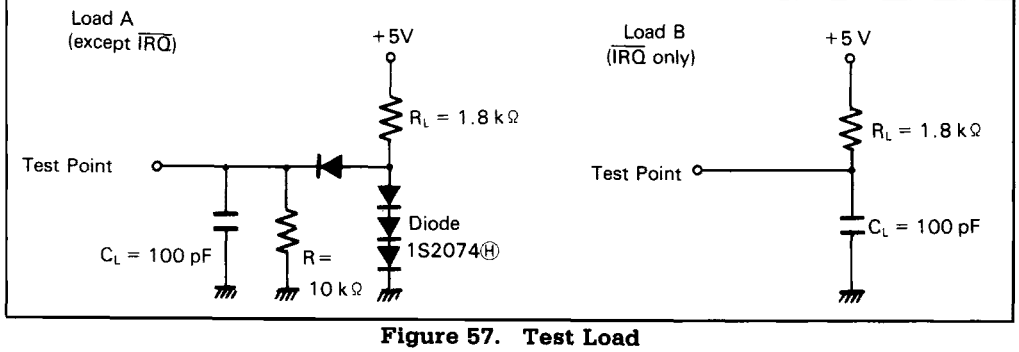


Figure 57. Test Load

Refer to application note (No. 68—3—18) for detail of this product.

5