
DMA Controller Architecture

Figures	v
Section 1. DMA Controller Architecture	1
DMA Controller Organization	2
System Interfaces	4
States	4
Idle State	5
Program State	5
Transfer State	6
DMA Controller Resets	7
Programming Considerations	8
Section 2. PIO Interface	9
PIO Interface Registers	10
System Interface Registers	11
Function Register	11
Byte Pointer	12
Operation Control Registers	13
Arbitration Level Registers	13
DMA Mask Register	14
DMA Status Register	15
Transfer Control Registers	16
DMA Mode Registers	16
I/O-Memory Address Registers	17
Memory Address Registers	18
Transfer Count Registers	18
Data Holding Register	18
Data Structures	19
Address	19
Transfer Count	19
Operations	20
All Operations	20
Memory-Verify Operation	21
Memory-to-I/O Transfer Operation	22
I/O-to-Memory Transfer Operation	23
Commands	24
Function Register Commands	25
Write Function	25
Immediate Commands	25
Set DMA Mask	25
Reset DMA Mask	26
Reset DMA Controller	26

Execute Commands	27
Read or Write I/O Address	27
Write Memory Address	28
Read Memory Address	28
Write Transfer Count	29
Read Transfer Count	29
Read DMA Status	30
Read or Write DMA Mode	30
Read or Write Arbitration Level	31
PIO Programming Model	32
Initialization	32
Termination	32
Section 3. SCB Interface	33
SCB Interface Registers	34
System Interface Registers	35
Function Register	35
Extended Function Register	36
Byte Pointer	36
SCB Control Registers	37
Control-Block Address Registers	37
Attention Register	38
Subsystem Control Register	38
Command Busy/Status Register	39
Indirect List Address Registers	40
Indirect List Count Registers	40
Operation Control Registers	41
Arbitration Level Registers	41
DMA Mask Register	42
DMA Status Register	43
Transfer Control Registers	44
DMA Mode Register	44
I/O-Memory Address Registers	46
Memory Address Registers	47
Transfer Count Registers	47
Data Holding Register	47
Data Structures	48
Transfer Structures	48
Address	48
Transfer Count	48
Control Structure	48
Command Word	49
Enable Word 1	50
Address 1	51
Address 2	51
Transfer Count	51

Indirect Lists	52
Operations	54
DMA Operations	54
Memory Verify Operation	54
Memory-to-I/O Transfer Operation	55
I/O-to-Memory Transfer Operation	56
SCB Fetch Operations	57
Control Block Fetch Operation	57
Indirect List Fetch Operation	58
Error Processing	59
Error Conditions	59
Error-Terminated DMA Operation	60
Error Recovery	60
Commands	61
Function Register Commands	62
Read or Write Function	62
Read or Write Extended Function	62
Immediate Commands	63
Set DMA Mask	63
Reset DMA Mask	63
Reset DMA Controller	64
Reset DMA Channel	64
Execute Commands	65
Read I/O-Memory Address	65
Read Memory Address	66
Read Transfer Count	67
Read DMA Status	67
Read DMA Mode	68
Read or Write Arbitration Level	68
Read DMA Busy	69
Restore FIFO Data	70
Read FIFO Byte Count	71
Read or Write Control-Block Address	72
Read Indirect List Address	73
Read Indirect List Count	74
Write Attention Code	75
Write Subsystem Control	75
Read Command Busy/Status	75
SCB Operation Programming Model	76
Initialization	76
Termination	76
Appendix A. Architecture Compliance	77
DMA Controller Levels	77
Type 1	77
Type 2	77

Appendix B. FIFO	79
FIFO Implementation Requirements	79
FIFO Buffer Implementation	79
FIFO Buffer Registers	80
FIFO Byte Count Register	80
FIFO Data Buffer	80
Extended Function Register	81
DMA Mode Registers	81
Transfer Count Register	82
State After Reset	82
FIFO Commands	82
Save FIFO Data	83
Restore FIFO Data	84
Read FIFO Byte Count	85
Read or Write FIFO Data	86
Reset DMA Controller	86
Reset DMA Channel	86
FIFO Operation Program Model	87
Reading from the FIFO buffer	87
Writing to the FIFO buffer	87
 Index	 89

Figures

1.	Structure of DMA Architecture	2
2.	Register Comparison – PIO to SCB Interfaces	3
3.	I/O Addresses Assignment	4
4.	I/O Address Assignment	4
5.	DMA Register Reset Initialization	7
6.	Default Arbitration-Level Values	8
7.	I/O Addresses and Registers – PIO Interface	10
8.	DMA Operation Selection	17
9.	PIO Commands	24
10.	I/O Addresses and Registers – SCB Interface	34
11.	DMA Operation Selection	45
12.	Control Block Structure	49
13.	Indirect List Structure	52
14.	SCB-Interface Commands	61
15.	Required Function Commands	79
16.	Registers Affected by Implementing FIFO Buffer	80
17.	Additional Extended Function Commands	81
18.	Residual Data Control Field	81
19.	FIFO Register Initialization	82
20.	Commands Affected by Implementing FIFO Buffer	82

Notes:

Section 1. DMA Controller Architecture

The direct-memory-access (DMA) controller is a system component that has the single task of managing the transfer of data between two addressable locations within the system address space. Using the DMA controller to manage data transfers relieves the system master of having to perform these transfers.

The DMA controller consist of eight channels. Each channel can be programmed to perform specific operations using specific parameters. Each channel can be set to respond to a specific arbitration level.

The DMA operations supported by each channel include data-transfer operations and the memory-verify operation. Data-transfer operations move data between an I/O device and memory. Memory-verify operation performs a memory read cycle but does not perform a write cycle or transfer any data.

The DMA controller architecture provides two interfaces to control DMA operations: programmed-I/O (PIO) and subsystem control block (SCB). The PIO interface supports a 24-bit memory address and an 8- or 16-bit data width through a set of registers and I/O ports. The SCB interface supports a 32-bit memory address and an 8-, 16-, or 32-bit data width through a control-block interface.

The DMA controller for all systems based on the Micro Channel^{*} architecture must support the PIO interface. In addition, the Type 2 DMA controller provides the SCB interface.

* Trademark of the IBM Corporation.

DMA Controller Organization

The architecture for the DMA controller consists of two interfaces; each interface provides its own method of performing DMA operations.

The PIO interface, which uses a set of I/O registers to control the DMA operations, is described starting on page 9.

The SCB interface, which uses control blocks to control DMA operations, is described starting on page 33.

The relationship of the functional interfaces that support these architectures is shown in the following figure.

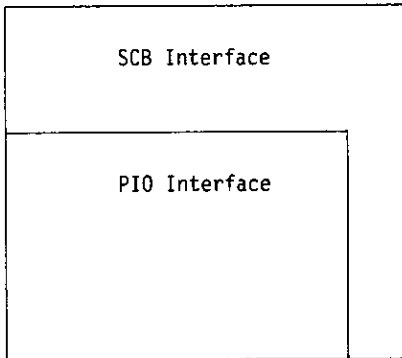


Figure 1. Structure of DMA Architecture

The following compares the organization of the two interfaces at the register level.

Register Name	Number of Registers	PIO		SCB	
		Size	R/W	Size	R/W
I/O-Memory Address	8	16	R/W	32	R
Memory Address	8	24	R/W	32	R
Transfer Count	8	16	R/W	32	R
DMA Mode	8	8	R/W	16	R
DMA Mask	8	8	W	8	R
Arbitration Level	8	8	R/W	8	R/W
DMA Status	1	16	R	16	R
Function	1	8	W	8	R/W
Extended Function	1	-	-	8	R/W
Control Block Address	8	-	-	32	R/W
Attention	1	-	-	8	W
Subsystem Control	1	-	-	8	R
Command Busy/Status	1	-	-	8	W
Indirect List Address	8	-	-	32	R
Indirect List Count	8	-	-	32	R
		R/W	Read and write	R	Read only
		-	Not available	W	Write only

Figure 2. Register Comparison—PIO to SCB Interfaces

System Interfaces

The I/O addresses assigned to the DMA controller are not accessible through the channel. These registers can be directly accessed only through I/O read and write operations from the system master.

The following are the I/O addresses assigned to the DMA controller.

I/O Address (hex)	Description
0000–000F	AT-compatible interface
0010–001F	PIO and SCB interfaces
0081–008F	AT-compatible interface
00C0–00DF	AT-compatible interface

Figure 3. I/O Addresses Assignment

The following are the I/O addresses used by the PIO and SCB interfaces.

I/O Address	Description	PIO	SCB
0010–0017	Reserved		
0018	Function register	WO	R/W
0019	Extended Function register	-	R/W
001A	Execute Function port	R/W	R/W
001B	Reserved		
001C	Attention register	-	WO
001D	Subsystem Control register	-	WO
001E	Reserved		
001F	Command Busy/Status register	-	RO

Figure 4. I/O Address Assignment

States

The DMA controller operates on the registers and data based on its current state. The DMA controller has three states:

- Idle state
- Program state
- Transfer state

The controller, acting on behalf of different DMA channels, cannot be in more than one state at a time. In each state, the DMA controller monitors various conditions to determine the next operation to perform and what state to enter next.

Idle State

The idle state is the default state for the DMA controller. The DMA controller enters this state after it is reset or after exiting from one of the other two states. While in the idle state, the DMA controller monitors the address bus and control signals to determine if an I/O cycle is occurring and monitors the arbitration bus and ARB/-GNT to determine if an arbitration cycle is occurring.

If an I/O cycle selects a DMA controller register, the DMA controller enters the program state and prepares to process the command.

During an arbitration cycle, the DMA controller monitors the arbitration bus. It enters the transfer state if the system channel is granted to an arbitration level that matches a value in an Arbitration Level register and the DMA channel is enabled.

When the DMA controller enters the idle state after a system reset, its registers are initialized to a default state. (For information on the default state for these register, see "DMA Controller Resets" on page 7.)

Program State

The DMA controller enters the program state whenever it detects an I/O cycle with an address within the address range assigned to the DMA controller.

The DMA controller is in the program state whenever commands are being issued that modify or return the contents of its registers. These commands are defined by the interfaces for PIO and SCB architectures.

The DMA controller exits from the program state after each I/O cycle is completed and returns to the appropriate state, either the idle state or the transfer state.

Transfer State

The DMA controller is in the transfer state whenever it is driving the control signals and the address bus. It drives these signals while performing a DMA transfer operation (data transfer or memory-verify) or an SCB fetch operation (a control block or an indirect list).

The DMA controller enters the transfer state from the idle state when all of the following conditions exist:

- An arbitrating device has been granted control of the system channel.
- The arbitration level matches the value in one of the Arbitration Level registers.
- The DMA channel with the matching arbitration level is enabled.

If more than one DMA channel meets these conditions, the lower-numbered DMA channel gets control and starts its DMA operation. After the lower-numbered channel has completed its operation, the next channel gets control and starts its operation. This continues until all operations for that arbitration level are completed.

Note: Although the DMA controller supports assigning the same level to more than one channel, IBM recommends that each DMA channel be assigned a unique arbitration level.

The DMA controller exits from the transfer state and returns to the idle state after an end-of-transfer is signaled or if a DMA error is detected. The end-of-transfer is signaled:

- After the DMA controller completes a single transfer for a nonbursting device
- After the DMA controller reaches the terminal count
- After a bursting device stops driving the 'burst' signal and the DMA controller has completed the current transfer

DMA Controller Resets

The DMA controller controller can perform up to four levels of reset. The local bus is held not-ready until the reset operation has been completed.

Hardware reset

is performed when the system is first powered-on. This level of reset initializes all the registers, functions, and state machines in the DMA controller to a known state.

Controller reset

is performed by issuing the DMA Controller Reset command. This reset is similar to the hardware reset except that it initializes only the DMA controller registers and functions. It leaves system functions, such as arbitration and refresh, intact.

Channel reset

is performed by issuing the DMA Channel Reset command. This command allows the programmer to reset a specific DMA channel without interfering with the other DMA channels. This reset affects only the registers, fields, and logic that pertain to the DMA channel being reset. For registers containing fields for more than one DMA channel, only the fields for the specified channel are reset. All DMA logic used during an active DMA transfer, including the state machine are also reset. The channel reset is only supported by the SCB interface.

Bus timeout reset

is performed when a DMA channel has gained control of the system channel and a bus timeout occurs. This reset is similar to a channel reset except that DMA Status register is not reset.

The following shows the DMA registers and their content after a DMA hardware reset (H), controller reset (S), channel reset (C), or bus timeout reset (T).

Register	Reset Type	State After Reset
Function		Not initialized
Extended Function		Not initialized
Byte Pointer	H,S,C	Points to low byte
DMA Status	H,S,C	All status is cleared
Arbitration Level	H	Each channel is set to its channel number (see Figure 6)
DMA Mask	H,S,C,T	Channels disabled (hex FF)
DMA Mode		Not initialized
I/O-Memory Address		Not initialized
Memory Address		Not initialized
Transfer Count		Not initialized
Command Busy/Status	H,S,C	No errors and Attention register not busy
Attention	H,S,C	No active requests
Control Block Address		Not initialized
Indirect List Address		Not initialized
Indirect List Count		Not initialized
FIFO Byte Count	H,S,C,T	FIFO empty (hex 00)

Figure 5. DMA Register Reset Initialization

The system initializes the Arbitration Level registers during a system or DMA controller reset to the following values.

DMA Channel	Arbitration Level
Channel 0	0
Channel 1	1
Channel 2	2
Channel 3	3
Channel 4	4
Channel 5	5
Channel 6	6
Channel 7	7

Figure 6. Default Arbitration-Level Values

Programming Considerations

For both the PIO and SCB interfaces, use the following sequences for proper DMA operation.

1. Disable interrupts when using any multiple-byte commands.
2. Initialize all DMA register values before enabling the I/O device.

For SCB operations, the control block must be set up in memory and the request enqueued before enabling the I/O device.

Section 2. PIO Interface

In the programmed-I/O (PIO) interface, the system master uses a set of programmable registers to:

- Specify the type of operation
- Establish the addresses and direction
- Specify the number of transfers for a given operation
- Select the channel to be used

Each channel uses its own 24-bit memory address register and 16-bit transfer counter to define each DMA operation. Data width for each transfer can be set to 8 or 16 bits. Therefore, each channel can be set up to perform 64K transfers (up to 128KB) within a single, contiguous block of memory addresses from 0 to 16MB (2^{24}), independent of the setting of the other DMA channels.

Each channel is preset to respond to an arbitration level equal to its channel number. In addition, the arbitration levels for DMA channels 0 and 4 can be modified to respond to other values.

PIO Interface Registers

The PIO interface registers control and monitor the state of the DMA controller and each DMA channel in the DMA controller. The registers can be divided into three functional categories: system interface, operational interface, and transfer control.

The following shows the registers and ports used, and the I/O addresses associated, with the PIO interface. The page number indicates where the register is described.

I/O Address	Register	R/W	Page Number	
System Interface Registers				
0018	Function	W	11	
001A	Execute Function port	R/W	na	
	Byte pointer	na	12	
Operation Control Registers				
	Arbitration Level	R/W	13	
	DMA Mask	W	14	
	DMA Status	R	15	
Transfer Control Registers				
	DMA Mode	R/W	16	
	I/O-Memory Address	R/W	17	
	Memory Address	R/W	18	
	Transfer Count	R/W	18	
	Data Holding	na	18	
	R/W	Read and write	R	Read only
	na	Not available	W	Write only

Figure 7. I/O Addresses and Registers – PIO Interface

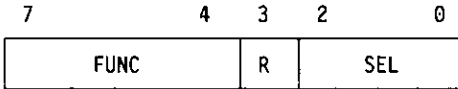
Note: Programs and hardware writing to reserved areas should ensure that the areas are set to 0 (unless otherwise indicated); programs and hardware reading these areas should treat reserved values as *don't care*.

System Interface Registers

These registers control the overall interface to the DMA controller and each of the DMA channels.

Function Register

The Function register is an 8-bit, write-only register that is used to select the DMA channel and specify the command to be performed. The Function register is loaded using the Write Function command (see page 25).



FUNC	Function
SEL	Select
R	Reserved

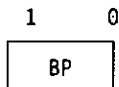
FUNC The Function field (bits 7–4) selects a command to be performed by the specified DMA channel. (For information on command selection, see “Commands” on page 24.)

SEL The Select field (bits 2–0) specifies the DMA channel that is to perform the PIO command.

Byte Pointer

The byte pointer is a 2-bit, internal register that is used to point to the individual bytes in multiple-byte registers. The byte pointer is increased by 1 for each read or write operation to the Execute Function port (hex 001A).

The contents of the byte pointer cannot be accessed. The byte pointer is cleared (set to point to the first byte of a register) automatically after a Write Function command (see page 25).



After the byte pointer is cleared, it points to bits 7 – 0 of a multiple-byte register. As each byte in the register is accessed, the byte pointer is increased to point to the next byte within the register. After the last byte (high byte) within the register is accessed, the byte pointer is undefined and must be cleared (set to point to the first byte).

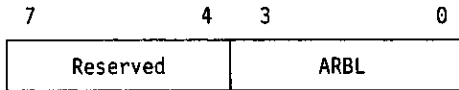
Operation Control Registers

These registers are used to control the operation of and maintain the status for each DMA channel.

Arbitration Level Registers

There are eight Arbitration Level registers, one for each DMA channel. Each register is an 8-bit register that contains the arbitration level assigned to the specific DMA channel. The registers for DMA channels 0 and 4 are read/write. The registers for the other DMA channels are not accessible.

The registers are accessed using the Read or Write Arbitration Level command (see page 31).



ARBL Arbitration Level

ARBL The Arbitration Level field (bits 3–0) specifies the arbitration level for the specific DMA channel.

DMA Mask Register

The DMA Mask register is an 8-bit, write-only register that contains one field for each DMA channel. The individual fields determine whether DMA operations are enabled or disabled for that channel. If DMA operations are enabled for a DMA channel, the DMA controller enters the transfer state when the system channel is granted to the arbitration level for that DMA channel.

The individual fields are set to 1 using the Set DMA Mask command, and set to 0 using the Reset DMA Mask commands (see page 25). All fields are set to 1 after performing a Reset DMA Controller command (see page 26).

7	6	5	4	3	2	1	0
M7	M6	M5	M4	M3	M2	M1	M0

M7-M0 Mask for Channels 7-0

M7 – M0 The Mask fields (bits 7 – 0) are used to enable and disable DMA operations on each of the DMA channels. When a Mask field is set to 1, the mask is set and the channel is disabled. When a Mask field is set to 0, the mask is reset and the channel is enabled. The Mask field is set to 1 by the DMA controller after the DMA channel has completed or terminated a DMA transfer operation.

DMA Status Register

The DMA Status register is a 16-bit, read-only register that contains the status for the eight DMA channels. As each byte is read, its status is cleared (the byte is set to 00). Performing a Reset DMA Controller command clears the status in both bytes.

The DMA Status register can be read using the Read DMA Status command (see page 30).

15	14	13	12	11	10	9	8
TC7	TC6	TC5	TC4	RQ7	RQ6	RQ5	RQ4
7	6	5	4	3	2	1	0
TC3	TC2	TC1	TC0	RQ3	RQ2	RQ1	RQ0

TC7-TC0 Terminal Count for Channels 7-0
RQ7-RQ0 DMA Request for Channels 7-0

- TC7-0** The Terminal Count fields (bits 15-12 and 7-4) indicate whether the corresponding DMA channel has completed a DMA operation (transfer count has passed 0) since the status was previously cleared. If the field is 1, the channel has completed a DMA operation; if the field is 0, the channel has not completed a DMA operation.
- RQ7-0** The DMA Request fields (bits 11-8 and 3-0) indicate whether the corresponding DMA channel has transferred data since the status was previously cleared. If the field is 1, the channel has been active; if the field is 0, the channel has not been active.

Transfer Control Registers

These registers determine the operation of the DMA channel and are used to control the source and destination addresses of each transfer and the number bytes transferred.

DMA Mode Registers

There are eight DMA Mode registers, one register for each DMA channel. Each register is an 8-bit, read/write register that controls the operation of that DMA channel.

The register is accessed using the Read or Write DMA Mode command (see page 30).

7	6	5	4	3	2	1	0
R	S16	R	DEC2	XFER	R	IOA	

S16	Size 16
DEC2	Decrement 2
XFER	Transfer Control
IOA	I/O Address
R	Reserved

- S16** The Size 16 field (bit 6) specifies the minimum data width used during the I/O cycle and determines whether the transfer count is the number of bytes or words to be transferred. When the field is set to 1, the transfer count represents the number of words to transfer. When the field is set to 0, the transfer count represents the number of bytes to transfer.

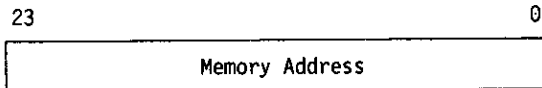
When a 16-bit transfer is performed to an I/O device, the I/O address is aligned on a word boundary (address bit 0 is 0), regardless of the address in the I/O-Memory Address register.

- DEC2** The Decrement 2 field (bit 4) specifies whether the DMA controller increases or decreases the address in the Memory Address register. When the field is set to 1, the Memory Address register is decreased by 1 for each byte transferred. When the field is set to 0, the Memory Address register is increased.

Memory Address Registers

There are eight Memory Address registers, one for each DMA channel. Each register is a 24-bit, read/write register that contains the memory address used during a DMA transfer. The memory address is increased or decreased as data is transferred.

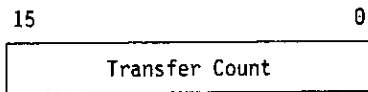
The register is read using the Read Memory Address command (see page 28) and is loaded using the Write Memory Address command (see page 28).



Transfer Count Registers

There are eight Transfer Count registers, one for each DMA channel. Each register is a 16-bit, read/write register that contains the number of transfer cycles to be performed. The register is decreased by the appropriate amount for each transfer cycle.

The Transfer Count register is read using the Read Transfer Count command (see page 29) and is loaded using the Write Transfer Count command (see page 29).



Data Holding Register

The DMA controller uses the Data Holding register to temporarily store the data being transferred. The minimum size of the Data Holding register is 2 bytes and the maximum is 255 bytes.

Data Structures

The DMA controller defines two transfer-control structures used by the PIO interface, which define the physical limits of the transfer.

Address

The memory addressing is limited to 24 bits. Memory addresses that are increased beyond hex 0FFFFFFF or decreased below 00 are undefined and cause compatibility problems in systems with 32-bit addressing.

When decreasing the memory address, the starting memory address must be aligned on a word boundary for DMA transfers using 16-bit data widths.

The I/O address can be aligned on a byte or word boundary. However, the DMA controller forces the I/O address to align on a word boundary when the width is specified as 16-bits.

Transfer Count

The transfer count is a 16-bit, zero-base value; a transfer count of hex 00 results in one transfer cycle.

Operations

The DMA operation to be performed is determined by the values loaded into the DMA Mode register, and determines how certain registers are used. The following describes how each of the registers is used for each type of operation.

All Operations

The following registers have the same usage for all DMA operations.

Arbitration Level Register: This register is used to associate an arbitration level to a DMA channel. If the DMA channel is enabled, the DMA controller starts a DMA operation when the corresponding arbitration level is granted control of the system channel. If more than one DMA channel has the same arbitration level, the operations are performed for all channels, starting with the lowest DMA channel.

If the winning arbitration level matches the level specified for a DMA channel that is disabled, the DMA controller remains in the idle state. (A bus timeout will occur if another master does not take control of the system channel.)

DMA Mask Register: The appropriate Mask field must be set to 0 (enabled) for the DMA transfer operation to begin. The DMA controller sets the field to 1 (disabled) after it completes the DMA operation.

DMA Status Register: The DMA Status register indicates whether or not a DMA channel was active and whether or not a DMA operation has been completed.

The DMA controller sets a DMA Request field to 1 every time the DMA channel starts an operation. It sets the Terminal Count field to 1 after the terminal count changes from 0000 to hex FFFF to indicate that the DMA channel has completed an operation.

Memory-Verify Operation

During the memory-verify operation, the DMA controller performs a memory-read cycle, but does not perform an I/O- or memory-write cycle. The operation begins when the arbitrating device is granted control of the system channel, if the DMA channel is enabled.

Note: All registers must be loaded with the appropriate values before starting the DMA operation.

DMA Mode Register: For memory-verify operations, the DMA Mode register is set to the following. The Size 16, Decrement 2, and I/O Address fields are set to the appropriate values, depending on the options desired.

7	6	5	4	3	2	1	0
0	S16	0	DEC2	0	0	0	IOA

I/O-Memory Address Register: This register is not used.

Memory Address Register: The Memory Address register points to the next memory location and is increased or decreased the appropriate amount, depending on the setting of the Decrement 2 field and the width of the transfer.

Transfer Count Register: The Transfer Count register contains the number of bytes or words that remain to be transferred (the Size 16 field determines whether the count value represents bytes or words). The count value is decreased the appropriate amount after the memory-read cycle. The DMA operation is completed (TC is driven active) when the transfer count goes from 0000 to hex FFFF.

Memory-to-I/O Transfer Operation

The memory-to-I/O transfer operation reads data from a location in memory and writes the data to an I/O device. The operation begins when the arbitrating device is granted control of the system channel, if the DMA channel is enabled.

Note: All registers must be loaded with the appropriate values before starting the DMA operation.

DMA Mode Register: For memory-to-I/O transfer operations, the DMA Mode register is set to the following. The Size 16, Decrement 2, and I/O Address fields are set to the appropriate values, depending on the options desired.

7	6	5	4	3	2	1	0
0	S16	0	DEC2	0	1	0	IOA

I/O-Memory Address Register: If the I/O Address field is set to 1, the DMA controller drives the address specified in this register onto the address bus during the I/O-write cycle. If the I/O Address field is set to 0, this register is not used, and the DMA controller drives address hex 0000 onto the address bus.

If the Size 16 field is set to 1, the DMA controller forces the I/O address to a word boundary (address bit 0 is driven to 0) and writes at least one word to the I/O address before performing another memory-read cycle or ending the transfer.

The I/O address remains unchanged during DMA operations.

Memory Address Register: The Memory Address register points to the next memory location and is increased or decreased the appropriate amount, depending on the setting of the Decrement 2 field and the width of the transfer.

Transfer Count Register: The Transfer Count register contains the number of bytes or words that remain to be transferred (the Size 16 field determines whether the count value represents bytes or words). The count value is decreased the appropriate amount after the I/O-write cycle. The DMA operation is completed (TC is driven active) when the transfer count goes from 0000 to hex FFFF.

I/O-to-Memory Transfer Operation

The I/O-to-memory transfer operation reads data from an I/O device and writes the data to a location in memory. The operation begins when the arbitrating device is granted control of the system channel, if the DMA channel is enabled.

Note: All registers must be loaded with the appropriate values before starting the DMA operation.

DMA Mode Register: For I/O-to-memory transfer operations, the DMA Mode register is set to the following. The Size 16, Decrement 2, and I/O Address fields are set to the appropriate values, depending on the options desired.

7	6	5	4	3	2	1	0
0	S16	0	DEC2	1	1	0	IOA

I/O-Memory Address Register: If the I/O Address field is set to 1, the DMA controller drives the address specified in this register onto the address bus during the I/O-read cycle. If the I/O Address field is set to 0, this register is not used, and the DMA controller drives address hex 0000 onto the address bus.

If the Size 16 field is set to 1, the DMA controller forces the I/O address to a word boundary (address bit 0 is driven to 0) and reads at least one word from the I/O address before performing the memory-write cycle.

The I/O address remains unchanged during DMA operations.

Memory Address Register: The Memory Address register points to the next memory location and is increased or decreased the appropriate amount, depending on the setting of the Decrement 2 field and the width of the transfer.

Transfer Count Register: The Transfer Count register contains the number of bytes or words that remain to be transferred (the Size 16 field determines whether the count value represents bytes or words). The count value is decreased the appropriate amount after the memory-write cycle. The DMA operation is completed (TC is driven active) when the transfer count goes from 0000 to hex FFFF.

Commands

These commands modify the state of the DMA controller, set parameters for other commands or DMA operations, or return information on the state of the DMA controller. There are two categories of commands through the Function register:

- Immediate commands
- Execute commands

All commands consist of an I/O write operation to the Function register (at address hex 0018). The immediate commands are those functions selected through the Function register that directly cause the DMA channel or DMA controller to perform some operation. The execute commands are those functions selected through the Function register that are performed when the Execute Function port (at hex 001A) is accessed.

The following shows the relationship between the functions selected and the command category.

Function (hex)	Command	Category
00	Read or write I/O address	Execute
01	Reserved	
02	Write memory address	Execute
03	Read memory address	Execute
04	Write transfer count	Execute
05	Read transfer count	Execute
06	Read DMA status	Execute
07	Read or write DMA mode	Execute
08	Read or write arbitration level	Execute
09	Set DMA mask	Immediate
0A	Reset DMA mask	Immediate
0B, 0C	Reserved	
0D	Reset DMA controller	Immediate
0E, 0F	Reserved	

Figure 9. PIO Commands

Function Register Commands

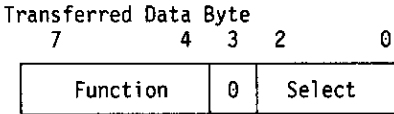
The PIO interface supports only one function-register command.

Write Function

This command loads the transferred data byte into the Function register. The Function field specifies the command to be performed and the Select field specifies the DMA channel.

This command sets the byte pointer to binary 00.

Write to Port 0018



Immediate Commands

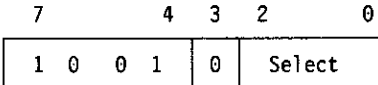
For the immediate commands, writing to the Function register causes the operation to be performed. The Select field specifies the DMA channel. These commands are performed immediately and do not require additional I/O accesses to the Execute Function register. The immediate commands are:

- Set DMA Mask
- Reset DMA Mask
- Reset DMA Controller

Set DMA Mask

This command sets the Mask field to 1 for the specified DMA channel. Setting the Mask field to 1 disables the DMA channel. The DMA channel is specified in the Select field.

Write to Port 0018



Reset DMA Mask

This command sets the Mask field to 0 for the specified DMA channel. Setting this field to 0 enables the DMA channel. The DMA channel is specified in the Select field.

Write to Port 0018

7	4	3	2	0
1	0	1	0	0
Select				

Reset DMA Controller

When it receives this command, the DMA controller:

- Terminates all transfer operations for all DMA channels
- Sets the byte pointer to binary 00
- Sets all mask fields to 1 to disable the DMA channels
- Sets all fields in the DMA Status register to 0

The Select field does not affect this command and should be set to 0.

Write to Port 0018

7	4	3	2	0
1	1	0	1	0
Select				

Execute Commands

These commands are issued in two parts. First, the Function register is loaded with the appropriate values. Then, the Execute Function port (hex 001A) is accessed to complete the command. The number of accesses is determined by the number of bytes in the register.

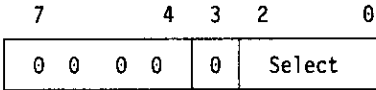
When multiple-byte registers are being accessed, the byte pointer determines which byte in the register is accessed. The Write Function command resets the byte pointer to point to byte 0 (bits 7–0).

Read or Write I/O Address

The read command returns the contents of the I/O-Memory Address register in byte increments. The write command loads the transferred data byte into the appropriate byte of the I/O-Memory Address register.

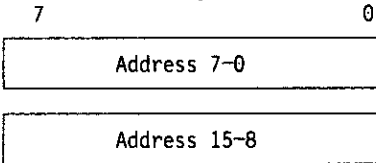
The DMA channel is specified in the Select field.

Write to Port 0018



Access Port 001A

Transferred Data Bytes

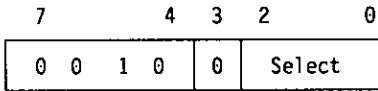


Write Memory Address

This command loads each transferred data byte into the appropriate byte of the Memory Address register.

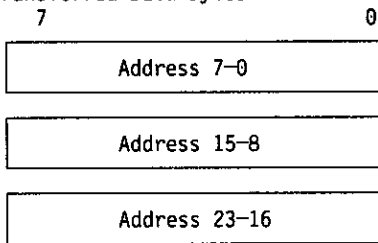
The DMA channel is specified in the Select field.

Write to Port 0018



Write to Port 001A

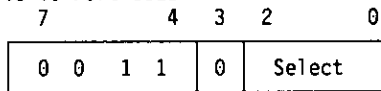
Transferred Data Bytes



Read Memory Address

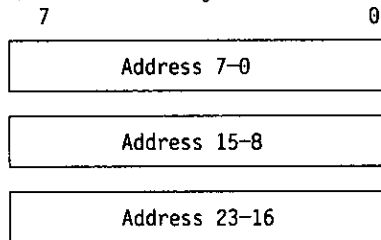
This command returns the content of the Memory Address register in byte increments. The DMA channel is specified in the Select field.

Write to Port 0018



Read Port 001A

Transferred Data Bytes

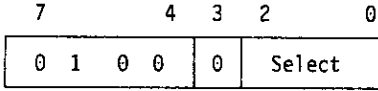


Write Transfer Count

This command loads each transferred data byte into the appropriate byte of the Transfer Count register.

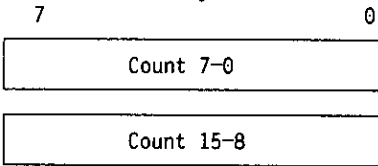
The DMA channel is specified in the Select field.

Write to Port 0018



Write to Port 001A

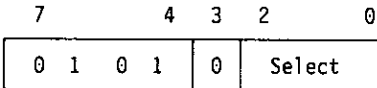
Transferred Data Bytes



Read Transfer Count

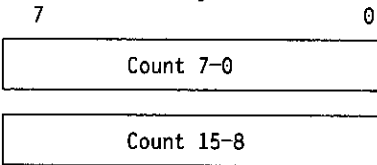
This command returns the content of the Transfer Count register in byte increments. The DMA channel is specified in the Select field.

Write to Port 0018



Read Port 001A

Transferred Data Bytes

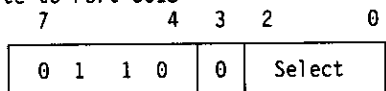


Read DMA Status

This command returns each byte of the DMA Status register and clears that byte of the DMA Status register (the byte is set to hex 00).

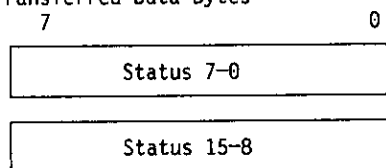
The Select field does not affect this command and should be set to 0.

Write to Port 0018



Read Port 001A

Transferred Data Bytes

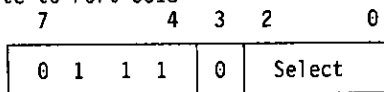


Read or Write DMA Mode

The read command returns the content of the DMA Mode register. The write command loads the transferred data byte into the DMA Mode register.

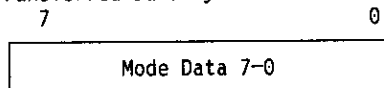
The DMA channel is specified in the Select field.

Write to Port 0018



Access Port 001A

Transferred Data Bytes

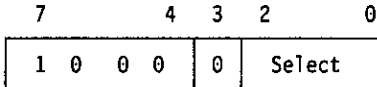


Read or Write Arbitration Level

The read command returns the contents of the Arbitration Level register. The write command loads the transferred data byte into the Arbitration Level register.

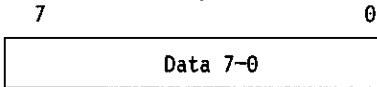
The DMA channel is specified in the Select field. Only DMA channels 0 and 4 support programmable arbitration levels; all other values of the Select field are reserved.

Write to Port 0018



Access Port 001A

Transferred Data Bytes



PIO Programming Model

Each of the eight DMA channels can be programmed independently and can operate concurrently. The DMA controller accepts commands only from the system master and accepts commands to a DMA channel regardless of whether the channel is transferring data or not.

Initialization

The following steps are performed to initialize a DMA channel:

1. Disable the DMA channel (set the Mask field to 1)
2. Load operational-control registers, if required
3. Load the transfer-control registers with the appropriate values
4. Enable the DMA channel (set the Mask field to 0)

The following is an example showing the programming of DMA channel 2 using the PIO interface commands. In this example, we want to transfer data from an I/O address (hex *mmnn*) to a memory location starting at an address (hex *xxyyzz*) and counting up a specific number of bytes (hex *aabb*).

Program Step	Address/Data
Set mask	0018h, 92h
Write I/O address (hex <i>mmnn</i>)	0018h, 02h 001Ah, <i>nnh</i> 001Ah, <i>mmh</i>
Write memory address (hex 00 <i>xxyyzz</i>)	0018h, 22h 001Ah, <i>zzh</i> 001Ah, <i>yyh</i> 001Ah, <i>xxh</i>
Write transfer count (hex 00 <i>aabb</i>)	0018h, 42h 001Ah, <i>bbh</i> 001Ah, <i>aa</i> h
Write DMA mode	0018h, 72h 001Ah, 0Dh
Reset mask	0018h, A2h

Termination

If a DMA transfer is terminated before the terminal count has been reached, the DMA channel must be disabled. Disabling the channel ensures that the DMA channel does not enter the transfer state accidentally when arbitration levels are being shared.

Section 3. SCB Interface

The SCB interface uses a form of the Locate mode described in the Subsystem Control Block (SCB)¹ architecture. The system master uses the I/O registers to control the interface, report status, and enqueue control blocks; it uses the control blocks to manage DMA operations.

The control block specifies the type of operation, the source and destination addresses of the data, and other parameters. The source or destination addresses can point to a data location or to an indirect list that points to several locations. The width of each transfer can be 8, 16, 24, or 32 bits and is determined dynamically by the signal protocols used on the system channel.

All channels support 32-bit memory addressing, 32-bit transfer counts, and data chaining. Therefore, each operation can transfer an unlimited amount of data to any location within the memory address space from 0 through 4GB (2^{32}).

Each DMA channel has a local arbiter that can be programmed independently to request the system channel and arbitrate for a device.

The SCB architecture is an extension to the registers and operations of the PIO architecture.

¹ callout".The SCB architecture is defined in the Personal Systems/2 Hardware Interface TechnicalReference—Architectures.

SCB Interface Registers

The SCB interface registers control and monitor the state of the DMA controller and each DMA channel in the DMA controller. The registers can be divided into four functional categories: system interface, SCB control, operational control, and transfer control.

The following shows the registers and ports used by, and the I/O addresses associated with, the SCB interface. The page number indicate where the register is described.

I/O Address	Register	R/W	Page Number
System Interface Registers			
0018	Function	R/W	35
0019	Extended Function	R/W	36
001A	Execute Function port	R/W	na
	Byte pointer	na	36
SCB Control Registers			
001C	Control Block Address	R/W	37
	Attention	W	38
001D	Subsystem Control	W	38
001F	Command Busy/Status	R	39
	Indirect List Address	R	40
	Indirect List Count	R	40
Operation Control Registers			
	Arbitration Level	R/W	41
	DMA Mask	R/W	42
	DMA Status	R	43
Transfer Control Registers			
	DMA Mode	R	44
	I/O-Memory Address	R	46
	Memory Address	R	47
	Transfer Count	R	47
	Data Holding	R	47

Figure 10. I/O Addresses and Registers – SCB Interface

Note: Programs and hardware writing to reserved areas should ensure that the areas are set to 0 (unless otherwise indicated); programs and hardware reading these areas should treat them as *don't care*.

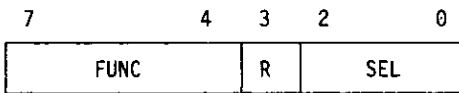
System Interface Registers

These registers control the overall interface to the DMA controller and each of the DMA channels. They are also used to report the status of each of the DMA operations.

Function Register

The Function register is an 8-bit, read/write register that is used to select the DMA channel and specify the command to be performed.

This register is accessed using the Read or Write Function command (see page 62).



FUNC	Function
SEL	Select
R	Reserved

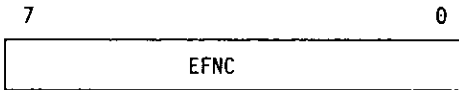
FUNC The Function field (bits 7 – 4) selects commands to be performed by the specified DMA channel. (For information on command selection, see “Commands” on page 61.)

SEL The Select field (bits 2 – 0) specifies the DMA channel that is to perform the command.

Extended Function Register

The Extended Function register is an 8-bit, read/write register that is used to specify additional commands.

This register is accessed using the Read or Write Extended Function commands (see page 62).



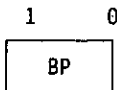
EFNC Extended Function

EFNC The Extended Function field (bits 7–0) defines additional commands to be performed. (For information on command selection, see “Commands” on page 61.)

Byte Pointer

The byte pointer is a 2-bit, internal register that is used to point to the individual bytes in multiple-byte registers. The byte pointer is increased by 1 for each read or write operation to the Execute Function port (hex 001A).

The contents of the byte pointer cannot be accessed. The byte pointer is cleared (set to point to the first byte of a register) automatically after the Write Function and Write Extended Function commands (see page 62).



After the byte pointer is cleared, it points to bits 7–0 of a multiple-byte register. As each byte in the register is accessed, the byte pointer is increased to point to the next byte within the register. After the last byte (high byte) within the register is accessed, the byte pointer is undefined and must be cleared (set to point to the first byte).

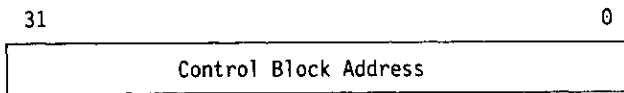
SCB Control Registers

These registers are used to enqueue and manage SCB-fetch operations (control block and indirect list).

Control-Block Address Registers

There are eight Control-Block Address registers, one for each DMA channel. Each register is a 32-bit, read/write register that contains the address of the control block being queued.

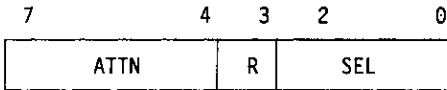
The Control-Block Address register is accessed using the Read or Write Control-Block Address command (see page 72).



Attention Register

The Attention register is an 8-bit, write-only register that is used to send an attention request to a DMA channel. The register is loaded using the Write Attention Code command (see page 75).

The Attention register is reset using the Reset DMA Controller and the Reset DMA Channel commands (see pages 64 and 64).



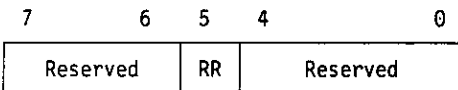
ATTN	Attention
SEL	Select
R	Reserved

ATTN The Attention field (bits 7 – 4) is used to initiate DMA operations using the SCB interface. The only request supported is the start control block request (hex 03), which causes the DMA controller to enqueue a request to start a control-block operation for the specified DMA channel.

SEL The Select field (bits 2 – 0) specifies the DMA channel that is to perform the request.

Subsystem Control Register

The Subsystem Control register is an 8-bit, write-only register that provides global control of the SCB interface. The register is loaded using the Write Subsystem Control command (see page 75).



RR	Reset Reject
----	--------------

RR The Reset Reject field (bit 5) restores the SCB interface to a state in which it can receive additional attention requests. When this field is set to a 1, the command-rejected condition is cleared, and the Busy and Reject fields are reset to 0. If the DMA controller is not in the command-rejected condition, setting this bit to 1 has no effect.

Command Busy/Status Register

The Command Busy/Status register is an 8-bit, read-only register that contains the status of the control block operation.

The contents of this register is returned using the Read Command Busy/Status command (see page 75). The fields are cleared using the Write Subsystem Control command (see page 75).

All fields in the register are cleared (set to 0) following the Reset DMA Controller or Reset DMA Channel commands and following a hardware reset.

7	5	4	3	1	0
Reserved		REJ	Reserved		BSY

REJ	Reject
BSY	Busy

REJ The Reject field (bit 4) indicates whether an error has been detected while processing a control block. When the field is 1, the DMA controller has detected an error while fetching or processing a control-block operation and the command is rejected. When the field is 0, the DMA controller did not detect an error.

An error condition also sets the Busy field to 1 (see "Error Processing" on page 59).

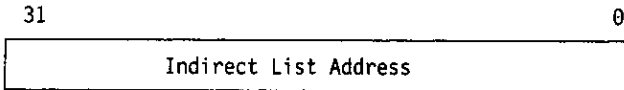
BSY The Busy field (bit 0) indicates whether the last write to the Attention register was processed. When the Attention register is loaded, the Busy field is set to 1. The DMA controller sets the field to 0 (not busy) after it has serviced the Attention register, making the register available to accept another request.

Programs must not write to the Attention register while either the Busy or the Reject fields are 1.

Indirect List Address Registers

There are eight Indirect List Address registers, one for each DMA channel. Each register is a 32-bit, read-only register that points to the next entry in the indirect list. The register is used only during operations involving data chaining.

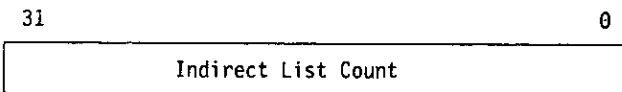
If data chaining is being used, the Indirect List Address register is loaded with the contents of the Address 2 field when the control block is fetched. The register can be read using the Read Indirect List Address command (see page 73).



Indirect List Count Registers

There are eight Indirect List Count registers, one for each DMA channel. Each Indirect List Count register is a 32-bit, read-only register that contains the number of bytes remaining in the indirect list. The register is used only during operations involving data chaining.

If data chaining is being used, the Indirect List Count register is loaded with the contents of the Transfer Count field while fetching the control block. The register can be read using the Read Indirect List Count command (see page 74).



Operation Control Registers

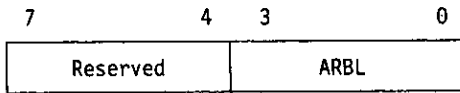
These registers are used to control the operation of and maintain the status for each DMA channel.

Arbitration Level Registers

There are eight Arbitration Level registers, one for each DMA channel. Each Arbitration Level register is an 8-bit, read/write register that contains the arbitration level assigned to the specific DMA channel.

The DMA controller compares the arbitration level used in winning control of the system channel to the value in each register. If the content of a register matches the winning arbitration level and DMA operations for that channel are enabled (not masked), the DMA controller starts the DMA operation.

The registers are accessed using the Read or Write Arbitration Level command (see page 68).



ARBL Arbitration Level

ARBL The Arbitration Level field (bits 3–0) specifies the arbitration level for this DMA channel. An arbitration level of hex F is reserved for the default master, and the DMA controller does not respond to this level.

DMA Mask Register

The DMA Mask register is an 8-bit, read/write register that contains one field for each DMA channel. The individual fields control whether DMA operations are enabled or disabled for that channel. If DMA operations are enabled for a DMA channel, the DMA controller enters the transfer state when it gains control of the system channel.

During normal operation, the DMA controller sets the Mask field to 0 when attention code 03 is written and sets the field to 1 when the DMA operation is completed.

The individual fields can also be set to 1 using the Set DMA Mask command, and set to 0 using the Reset DMA Mask command (see page 63). All mask fields are set to 1 after a DMA controller reset (see page 64).

The inverse of this register can be read using the Read DMA Busy command (see page 68).

7	6	5	4	3	2	1	0
M7	M6	M5	M4	M3	M2	M1	M0

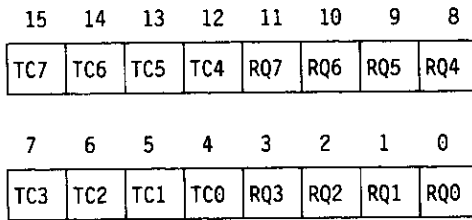
M7-M0 Mask for Channels 7-0

M7 – M0 The Mask fields (bits 7 – 0) are used to enable and disable DMA operations on each of the DMA channels. When a Mask field is set to 1, the mask is set and the channel is disabled. When a Mask field is set to 0, the mask is reset and the channel is enabled.

DMA Status Register

The DMA Status register is a 16-bit, read-only register that contains the status for the eight DMA channels. The DMA Status register can be read using the Read DMA Status command (see page 67).

As each byte is read, its status is cleared (the byte is set to 00). Performing a Reset DMA Controller command clears the status in both bytes.



TC7-TC0 Terminal Count for Channels 7-0
RQ7-RQ0 DMA Request for Channels 7-0

- TC7-0** The Terminal Count fields (bits 15-12 and 7-4) indicate whether the corresponding DMA channel has completed a DMA operation (transfer count has reached 0) since the status was previously cleared. If the field is 1, the channel has completed a DMA operation; if the field is 0, the channel has not completed a DMA operation.
- RQ7-0** The DMA Request fields (bits 11-8 and 3-0) indicate whether the corresponding DMA channel has transferred data since the status was previously cleared. If the field is 1, the channel has been active; if the field is 0, the channel has not been active.

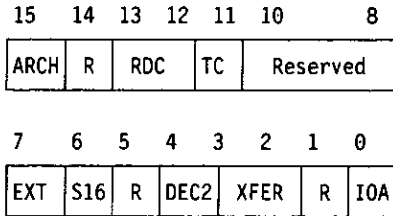
Transfer Control Registers

These registers determine the operation of the DMA channel and are used to control the source and destination addresses of each transfer and the number bytes transferred. During SCB-fetch operations, fields in the control block and indirect list are loaded into these registers. The registers are then used during the transfer.

DMA Mode Register

There are eight DMA Mode registers, one register for each DMA channel. Each register is a 16-bit, read-only register that controls the operation of that DMA channel. This register is loaded with values that are based on the Command word read when the control block is fetched.

The register is read using the Read DMA Mode command (see page 68).



ARCH	Architected	EXT	Extend Register
RDC	Residual Data Control	S16	Size 16
TC	Terminal Count	DEC2	Decrement 2
R	Reserved	XFER	Transfer Control
		IOA	I/O Address

ARCH The Architected field (bit 15) is loaded from the Architected field of the Command word and is used to indicate whether the control block follows the format defined by the SCB architecture. The DMA controller uses a unique format and does not use the architected format.

This field must be set to 0 to indicate that the format of the control block is specific to the DMA controller.

RDC The Residual Data Control field (bits 13 and 12) is used to control operation of a Data Holding register if it is implemented as a FIFO buffer. If the Data Holding register is not implemented as a FIFO, the DMA controller sets this

field to binary 00 (no residual data) during the SCB fetch operation.

This field is valid only when the Extend Register field is set to 1. (For information on FIFO implementations, see Appendix B, "FIFO" on page 79.)

TC The Terminal Count field (bit 11) controls whether the DMA controller drives the 'transfer count' signal (tc) active after it completes the transfer. This field is valid only when the Extend Register field is set to 1.

When the Terminal Count field is set to 1, the controller drives TC active; when the field is set to 0, the controller does not drive TC active

EXT The Extend Register field (bit 7) determines whether bits 15–8 in the DMA Mode register are used. When the field is set to 1, the fields in the high byte are used. When the field is set to 0, the fields in the high byte are ignored, and the DMA controller operates as if the high byte is set to hex 08 (terminal count enabled).

This field is set to 1 by the DMA controller during the SCB fetch operation.

S16 The Size 16 field (bit 6) to 0 by the DMA controller during the SCB fetch operation.

DEC2 The Decrement 2 field (bit 4) specifies whether the DMA controller increases or decreases the address in the Memory Address register. When the field is set to 1, the Memory Address register is decreased for each DMA transfer cycle. When the field is set to 0, the Memory Address register is increased.

XFER The Transfer Control field (bits 3 and 2) specifies the type of operation to be performed by the DMA channel.

Transfer Control	Operation Performed
00	Memory verify
01	Memory-to-I/O transfer (memory is the source)
10	Memory verify
11	I/O-to-memory transfer (memory is the destination)

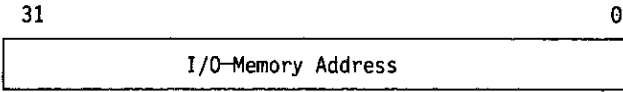
Figure 11. DMA Operation Selection

IOA The I/O Address field (bit 0) determines whether DMA controller uses the I/O address in the I/O-Memory Address register or I/O address hex 0000 during the DMA operation. When the field is 1, the DMA controller uses the address in the I/O-Memory Address register. When the field is 0, the DMA controller uses I/O address hex 0000 for the DMA transfer.

I/O-Memory Address Registers

There are eight I/O-Memory Address registers, one for each DMA channel. Each register is a 32-bit, read-only register that contains a 16-bit I/O address, which remains unchanged throughout the DMA operation. The I/O address is defined by bits 15 – 0; bits 31 – 16 are reserved and forced to hex 00.

The I/O-Memory Address register is loaded with the Address 1 field when the control block is fetched from memory (see page 57). The register is read using the Read I/O-Memory Address command (see page 65).

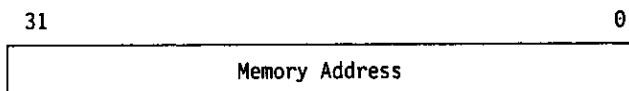


Memory Address Registers

There are eight Memory Address registers, one for each DMA channel. Each register is a 32-bit, read-only register that contains a memory address used during DMA transfers. The address is increased or decreased as data is transferred.

The Memory Address register is loaded with the content of the Address 2 field when fetching a control block or when fetching an entry from the indirect list.

The Memory Address register is read using the Read Memory Address command (see page 66).



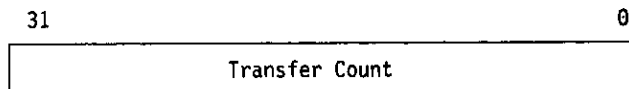
Transfer Count Registers

There are eight Transfer Count registers, one for each DMA channel. Each register is a 32-bit, read-only register that contains the number of bytes to be transferred.

The Transfer Count register is loaded with the Transfer Count field when fetching a control block or when fetching an entry from the indirect list.

The Transfer Count register is read using the Read Transfer Count command (see page 67).

Note: A transfer count of 0 is not supported and causes a control-block error.



Data Holding Register

The DMA controller uses the Data Holding register to temporarily store the data being transferred. The minimum size of the Data Holding register is 4 bytes and the maximum is 255 bytes.

Data Structures

The DMA controller defines two control structures used by the SCB interface. Transfer control structures define the physical limits of the transfer. SCB control structures specify the format of the command blocks and indirect lists.

Transfer Structures

The architecture defines two types of transfer-control structures: address and count.

Address

The address structures are either memory or I/O addresses.

The memory address is a 32-bit, byte-aligned address. The entire memory address space (4GB), which is represented by the 32-bit address, is accessible through the SCB interface. The memory address is undefined when increased beyond hex 0FFFFFFF or decreased below 0.

The I/O address is a 16-bit byte-aligned address.

Transfer Count

The transfer count is a 32-bit value that specifies the exact number of bytes to be transferred. Unlike the transfer count in the PIO interface (which is zero-based), the transfer count is one-based, and the transfer is completed when the count reaches 0 for the last block to be transferred.

Control Structure

The DMA controller implements a subset of the Locate mode from the Subsystem Control Block (SCB) architecture. The control block (the primary element) is a 16-byte data structure that is set up in memory to define the parameters for the DMA operation and to point to the data areas to use in the operation. The control block must be aligned on a doubleword boundary.

These control blocks specify the source and destination addresses by pointing to the location to be used (nonchained) or by pointing to an indirect list that points to the location to be used (data-chaining).

All information pertaining to the DMA operation is contained in the control block and related data structures. The operation is completed when all data for a specified block has been transferred and no blocks remain to be transferred.

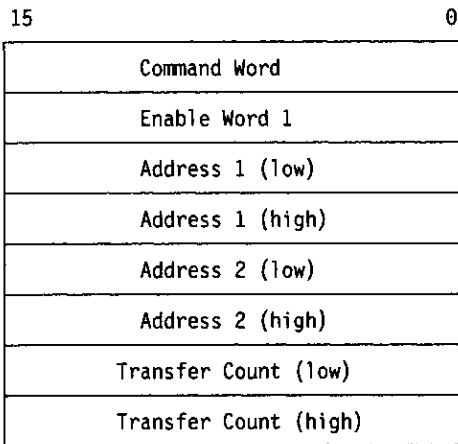
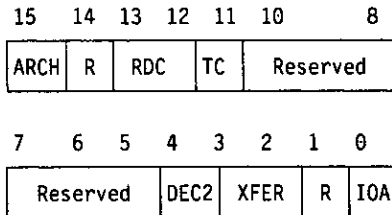


Figure 12. Control Block Structure

Command Word

The Command Word field is a 16-bit field that defines the parameters used in the DMA operation. Specific fields are loaded into the DMA Mode register during the fetch of the control block. (For definition of these fields, see "DMA Mode Register" on page 44.)



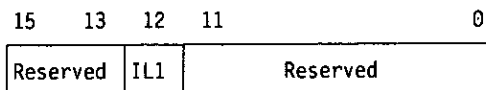
R Reserved

ARCH The Architected field (bit 15) is loaded into the Architected field of the DMA Mode register. This field must be set to 0 to indicate that the control block has a format that is unique to the DMA controller. If the field is set to 1, the DMA controller immediately returns an error indicating an unsupported control block.

- RDC** If the Data Holding register is implemented as a FIFO buffer, this field (bits 13 and 12) is loaded into the Residual Data Control field of the DMA Mode register. Otherwise, the field in the DMA Mode register is set to binary 00.
- TC** The Terminal Count field (bit 11) is loaded into the Terminal Count field of the DMA Mode register.
- DEC2** The Decrement 2 field (bit 4) is loaded into the Decrement 2 field of the DMA Mode register.
- XFER** The Transfer Control field (bits 3 and 2) is loaded into the Transfer Control field of the DMA Mode register.
- IOA** The I/O Address field (bit 0) is loaded into the I/O Address field of the DMA Mode register.

Enable Word 1

The Enable Word 1 is a 16-bit field that defines additional parameters to be used with this control block.



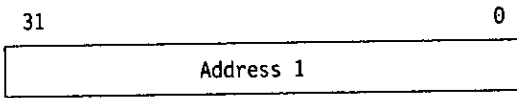
IL1 Indirect List 1

- IL1** The Indirect List 1 field (bit 12) specifies whether this control block uses data chaining or not. When the field is set to 1, data chaining is used, and the Address 2 field in the control block points to an indirect list, and the Transfer Count field specifies the number of bytes in the indirect list.

When the field is set to 0, the Address 2 field is a memory address to be used in the DMA operation and the Transfer Count field specifies the number of bytes to be transferred.

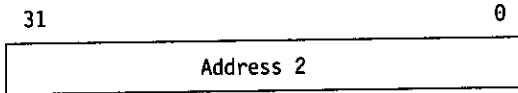
Address 1

The Address 1 field is a 32-bit field that is loaded into the I/O-Memory Address register during the fetch of the control block.



Address 2

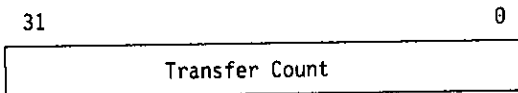
The Address 2 field is a 32-bit field that is loaded into the Memory Address register or the Indirect List Address register during the fetch of the control block. The register loaded is determined by the Indirect List 1 field in Enable Word 1.



Transfer Count

The Transfer Count field is a 32-bit field that is loaded into the Transfer Count register or Indirect List Count register (see pages 47 and 40). The register loaded is determined by the Indirect List 1 field in Enable Word 1.

Note: A transfer count of 0 is not supported and results in a control-block error condition.



Indirect Lists

The indirect list is a data structure set up in memory as a contiguous table that is aligned on a doubleword boundary. Each entry in the table consists of two doubleword fields (the minimum indirect list is one entry of eight bytes).

The address of the first entry in the indirect list and the number of bytes in the indirect list are specified in the Address 2 and Transfer Count fields of the control block.

Note: If the address of the indirect list is not aligned on a doubleword boundary or the count is not a multiple of 8 greater than 0, a control-block error occurs.

The format of the indirect list is shown in the following.

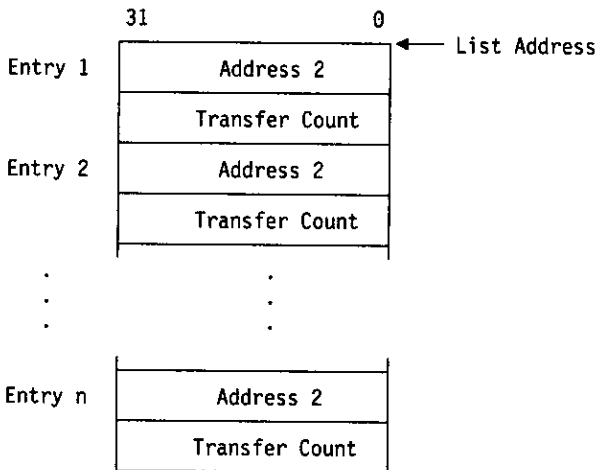


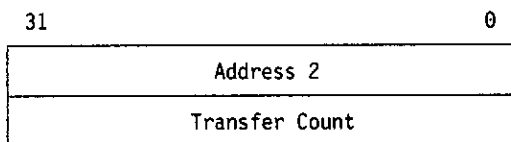
Figure 13. Indirect List Structure

Each entry consists of two doubleword fields.

The first field is the 32-bit Address 2 field that points to a contiguous block in memory that is used as the source or destination for the DMA transfer. This field is loaded into the Memory Address register during the fetch operation for this indirect-list entry.

The second field is the 32-bit Transfer Count field that specifies the number of bytes in the memory block. This field is loaded into the Transfer Count register during the fetch operation for this indirect-list entry.

Note: A count of 0 is not supported and results in a control-block error.



Operations

In the SCB interface, the DMA controller handles two types of operations, DMA operations and SCB-fetch operations.

DMA Operations

The DMA operation to be performed is determined by the values loaded into the DMA Mode register, and determines how certain registers are used. The following describes how each of the registers is used for each type of operation.

Arbitration Level Register: A virtual arbitration level capability is provided for each DMA channel.

DMA Mask Register: The Mask field is automatically set to 0 (DMA channel enabled) when a control block is enqueued with the Write Attention command. The Mask field is set to 1 by the DMA controller when the DMA operation is completed.

Transfer Count Register: When the control block is fetched, the DMA controller loads the Transfer Count register. This register keeps count of the number of bytes to be transferred for that part of the DMA operation.

For operations that do not involve an indirect list (nonchained operations), the register is loaded with the value in the Transfer Count field in the control block. The DMA operation is completed when the Transfer Count register reaches 0 and no bytes remain in the Data Holding register.

For operations that do involve an indirect list (data-chain operations), each fetch from the indirect list loads the Transfer Count register with the value in the Transfer Count field for that entry. The DMA operation is completed when the Indirect List Count and the Transfer Count registers reach 0 and no bytes remain in the Data Holding register.

Memory Verify Operation

The memory-verify operation reads data from a memory source and discards the data without writing the data. The operation begins when the arbitrating device is granted control of the system channel, if the DMA channel is enabled.

DMA Mode Register: The DMA controller loads the individual fields of the DMA Mode register with values based on the Command word that is read when the control block is fetched. The following is the contents of the DMA Mode register for a memory-verify operation.

15	14	13	12	11	10	9	8
0	0	0	0	TC	0	0	0

7	6	5	4	3	2	1	0
1	0	0	DEC2	0	0	0	0

I/O-Memory Address Register: This register is loaded with the contents of the Address 1 field when the control block is fetched. This register is not used for memory-verify operation.

Memory Address Register: For nonchained operations, this register is loaded with the value in the Address 2 field when the control block is fetched. For data-chained operations, each fetch from the indirect list loads this register with the value in the Address 2 field for that entry.

Memory-to-I/O Transfer Operation

The memory-to-I/O operation reads data from a location in memory and writes the data to an I/O or DMA device. The operation begins when the arbitrating device is granted control of the system channel, if the DMA channel is enabled.

DMA Mode Register: The DMA controller loads the individual fields of the DMA Mode register with values based on the Command word that is read when the control block is fetched. The following is the contents of the DMA Mode register for a memory-to-I/O transfer.

15	14	13	12	11	10	9	8
0	0	RDC	TC	0	0	0	0

7	6	5	4	3	2	1	0
1	0	0	DEC2	0	1	0	IOA

I/O-Memory Address Register: This register is loaded with the contents of the Address 1 field when the control block is fetched.

Memory Address Register: For nonchained operations, this register is loaded with the contents of the Address 2 field. For data-chained operations, each fetch from the indirect list loads this register with the contents of the Address 2 field for that entry.

I/O-to-Memory Transfer Operation

The I/O-to-memory operation reads data from an I/O or DMA device and writes the data to a location in memory. The operation begins when the arbitrating device is granted control of the system channel, if the DMA channel is enabled.

DMA Mode Register: The DMA controller loads the individual fields of the DMA Mode register with values based on the Command word that is read when the control block is fetched. The following is the contents of the DMA Mode register for a I/O-to-memory transfer.

15	14	13	12	11	10	9	8
0	0	RDC	TC	0	0	0	0
7	6	5	4	3	2	1	0
1	0	0	DEC2	1	1	0	IOA

I/O-Memory Address Register: This register is loaded with the contents of the Address 1 field when the control block is fetched.

Memory Address Register: For nonchained operations, this register is loaded with the value in the Address 2 field when the control block is fetched. For data-chained operations, each fetch from the indirect list loads this register with the value in the Address 2 field for that entry.

SCB Fetch Operations

The Locate mode in the SCB architecture defines a control structure that allows the system master to indirectly deliver DMA-transfer requests to the DMA controller using control blocks and indirect lists. The DMA controller contains an SCB sequencer, which fetches the control-block and indirect-list information from memory and stores the data in the appropriate DMA transfer registers. The following describes how the SCB sequencer uses the SCB control registers during the control-block fetch operation and subsequent indirect-list fetch operations.

Control Block Fetch Operation

A control-block fetch operation for an individual DMA channel is performed when the corresponding channel is granted control of the system channel. The control block is fetched at the beginning of the grant cycle before the DMA operation is started.

During a control-block fetch operation, the SCB sequencer reads the 16-byte control block from memory. The SCB sequencer interprets the various fields and uses the information to initialize the appropriate registers for the corresponding DMA channel.

- The Command word is used to initialize the DMA Mode register and to detect the type of control block.
- The Enable Word 1 field identifies whether or not the operation involves data chaining. If the data chaining is being used, the SCB sequencer enqueues a indirect-list-pending request for that DMA channel.
- The Address 1 field is used to initialize the I/O-Memory Address register.
- The Address 2 field is used to initialize the Memory Address register for nonchained operations and is used to initialize the Indirect List Address register for data-chained operations.
- The Transfer Count field is used to initialize the Transfer Count register for nonchained operations and is used to initialize the Indirect List Count register for data-chained operations.

Control Block Address Register: This register is loaded by the system master before the control block is enqueued (Write Attention command with request hex 03). After the control block is enqueued, this register must not be altered until the transfer has been completed (as indicated by the corresponding field returned by the Read DMA Busy command). Writing to this register while the DMA channel is busy can cause unpredictable results.

Attention Register: Because DMA operations and system master I/O operations to the DMA controller are mutually exclusive, the SCB sequencer interprets the attention request as it is written to the Attention register. The SCB sequencer sets the Busy field in the Command Busy/Status register to 1, services the Attention register, and resets the Busy field to 0 during a single I/O operation. Therefore, unless the Busy field was set to 1 because of an error, the Busy field is always 0 after the Write Attention command.

When a request code 03 is decoded, the SCB sequencer sets the DMA Mask field to 0 and internally enqueues a SCB-pending request for that DMA channel. The DMA controller allows all eight DMA channels to have SCB operations pending. The SCB-pending request is cleared when the control block is fetched for that DMA channel or after a Reset DMA Controller or Reset DMA Channel command (see pages 64 and 64).

Indirect List Fetch Operation

If data chaining is being used, an indirect-list fetch operation is performed immediately after the control-block fetch operation and before the DMA operation is started. During a single system grant cycle, the control block is fetched, the first entry in the indirect is fetched, and at least one DMA-transfer cycle occurs. If the indirect list contains multiple entries, additional indirect-list fetch operations are performed. The next entry is fetched immediately after the Transfer Count register goes to 0 and before the DMA controller releases the system channel.

Indirect List Address Register: During data-chained operations, the SCB sequencer uses this register to point to the memory location from which to fetch the next entry in the indirect list. After fetching the entry, the value in this register is increased by 8 to point to the next entry in the indirect list.

Indirect List Count Register: During data-chained operations, the SCB sequencer uses this register to count the number of bytes remaining in the indirect list. After fetching the entry, the value in this register is decreased by 8. When the value reaches 0, the SCB sequencer resets the indirect-list-pending request to indicate that no additional fetch operations are required. The Terminal Count field and the 'terminal count' signal are disabled until the indirect-list-pending request is reset (see "DMA Status Register" on page 43).

Error Processing

The DMA controller can detect certain errors that occur as a result of software or hardware errors. Although the DMA controller is a master, it normally relies on the DMA slave to compete for the system channel and to interrupt the system master. The DMA controller cannot directly notify the DMA slave that an error has been detected. Because the state of the DMA slave is undefined when the DMA operation is halted unexpectedly, the DMA controller relies on the central arbitration control point to force the DMA slave off the system channel and interrupt the system master.

Command Busy/Status Register: The Reject and Busy fields are both set to 1 when SCB-related error is detected.

Subsystem Control Register: This register is used to reset the Reject and Busy fields in the Command Busy/Status register.

Error Conditions

The DMA controller reports the following conditions as errors when detected.

- The control block address is not aligned on a doubleword boundary.
- The Architected field in the Command word is 1.
- The Transfer Count field in a control block or indirect list is 0.
- For data-chained operations, the Address 2 field (indirect-list address) is not aligned on a doubleword boundary, or the Transfer Count field (indirect-list count) is not a multiple of 8.

In addition, if the Extend Register field in the DMA Mode register is 0 during an SCB-related transfer operation, an error is reported. This field is set to 1 by the SCB sequencer and is checked at the beginning of each grant cycle. A value of 0 indicates that a Write DMA Mode command was issued before the current DMA operation has been completed (see “SCB Operation Programming Model” on page 76).

Error-Terminated DMA Operation

When an error condition is detected, the DMA controller uses the following procedures in terminating the DMA operation.

- The DMA controller immediately stops all activity for the current DMA channel and sets the corresponding field in the DMA Mask register to 1.
- The Reject and Busy fields in the Command Busy/Status register are set to 1.
- A bus-timeout request is sent to the central-arbitration control point (CACP).
- After the CACP generates a bus timeout, the DMA controller resets the DMA channel that was active at the time the error was detected (see “DMA Controller Resets” on page 7 for information on registers effected by the reset operation).

Error Recovery

The following procedure is used to recover from a error detected by the DMA controller.

1. Read the Arbitration register (hex 0090) to verify that a bus timeout has occurred and to identify the arbitration level. (Bit 5 is 1 if a timeout has occurred, and bits 3 – 0 are the arbitration level.)
2. Read the Command Busy/Status register to determine whether the error was detected by the DMA controller (the Reject field is 1).
3. If the DMA controller caused the error, reset the Command Busy/Status register by writing hex 20 to the Subsystem Control register, and take the appropriate actions to restore the state of the DMA slave involved.
4. Set bit 6 to 0 in the Arbitration register to reset the timeout condition.

5. If required, the DMA controller registers can be read to determine the cause of the error and the channel status at the time of the error.

Commands

These commands are used to modify the state of the DMA controller or to return information on the state of the DMA controller. There are two categories of commands through the Function and Extended Function registers:

- Immediate commands
- Execute commands

The function-register commands are I/O operations to the Function and Extended Function registers (at address hex 0018 and 0019). The immediate commands are those functions selected through the Function and Extended Function registers that directly cause the DMA channel or DMA controller to perform some action. Execute commands are those functions selected through the Function and Extended Function registers that are performed when the Execute Function port (at hex 001A) is accessed.

Extended Function	Function	Command	Type
	0	Read I/O-memory address	Execute
	1	Read or write control-block address	Execute
	3	Read memory address	Execute
	5	Read transfer count	Execute
	6	Read DMA status	Execute
	7	Read DMA mode	Execute
	8	Read or write arbitration level	Execute
	9	Set DMA mask	Immediate
	A	Reset DMA mask	Immediate
	D	Reset DMA controller	Immediate
00	F	Read DMA busy	Execute
02	F	Restore FIFO data	Execute
03	F	Read FIFO byte count	Execute
0B	F	Read indirect list address	Execute
0C	F	Read indirect list count	Execute
0D	F	Reset DMA channel	Immediate

Note: Any values not shown are reserved.

Figure 14. SCB-Interface Commands

Function Register Commands

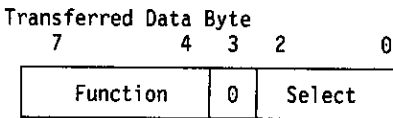
The Function and Extended Function registers are accessed through I/O addresses hex 0018 and 0019 in two single-byte operations.

Read or Write Function

The read command returns the contents of the Function register. The write command loads the transferred data byte into the register and clears the byte pointer.

The read command does not clear the byte pointer.

Access Port 0018

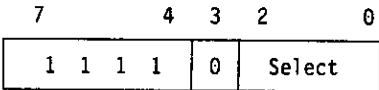


Read or Write Extended Function

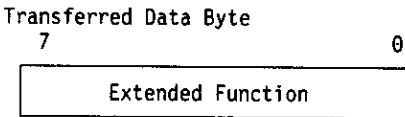
The read command returns the contents of the Extended Function register. The write command loads the transferred data byte into the Extended Function register.

The byte pointer is cleared.

Write to Port 0018



Access Port 0019



Immediate Commands

For these commands, writing to the Extended Function register causes the operation to be performed. The Select field specifies the DMA channel. These commands are performed immediately and do not require additional I/O accesses to the Execute Function port. The immediate commands are:

- Set DMA Mask
- Reset DMA Mask
- Reset DMA Controller
- Reset DMA Channel

Set DMA Mask

This command sets the Mask field to 1 for the specified DMA channel. Setting the mask field to 1 disables the DMA channel. The DMA channel is specified in the Select field.

The byte pointer is cleared.

Write to Port 0018

7	4	3	2	0
1	0	0	1	0
				Select

Note: This command is not required for DMA operations.

Reset DMA Mask

This command sets the Mask field to 0 for the specified DMA channel. Setting this field to 0 enables the DMA channel. The DMA channel is specified in the Select field.

The byte pointer is cleared.

Write to Port 0018

7	4	3	2	0
1	0	1	0	0
				Select

Note: This command is not required for DMA operations.

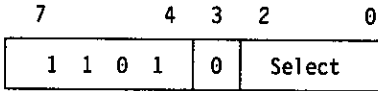
Reset DMA Controller

When it receives the command, the DMA controller:

- Terminates all transfer operations for all DMA channels
- Clears the byte pointer (set to binary 00)
- Sets the Attention register to hex 00
- Sets the Command Busy/Status register to hex 00
- Sets the DMA Status register to hex 00
- Sets the DMA Mask register to hex FF (all channels disabled)

The Select field does not affect this command and should be set to 0.

Write to Port 0018



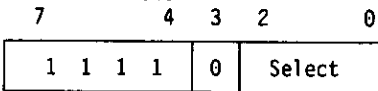
Reset DMA Channel

This command resets only those fields and operations associated with the specified DMA channel. A reset of the DMA channel:

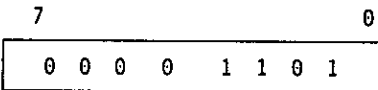
- Terminates all transfer operations for the specified channel
- Clears the byte pointer (set to binary 00)
- Sets the Mask field to 1 (disabled)
- Sets the DMA Request field and Terminal Count field to 0
- Sets the Attention register to hex 00
- Sets the Command Busy/Status register to hex 00

The DMA channel is specified in the Select field.

Write to Port 0018



Write to Port 0019



Execute Commands

These commands are issued in two parts. First, the Write Function or Write Extended Function command selects the function to perform and the specific register to access. Then, the Execute Function port (hex 001A) is accessed to complete the command. The number of accesses is determined by the number of bytes in the register.

When accessing multiple-byte registers, the byte pointer determines which byte in the register is accessed. The Write Function and Write Extended Function commands reset the byte pointer to point to byte 0 (bits 7–0).

Read I/O-Memory Address

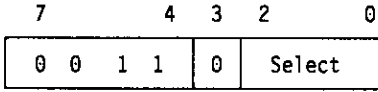
The read command returns the contents of the I/O-Memory Address register in byte increments. The DMA channel is specified in the Select field.

15	14	13	12	11	10	9	8
0	0	RDC	TC	0	0	0	
7	6	5	4	3	2	1	0
1	0	0	DEC2	1	1	0	IOA

Read Memory Address

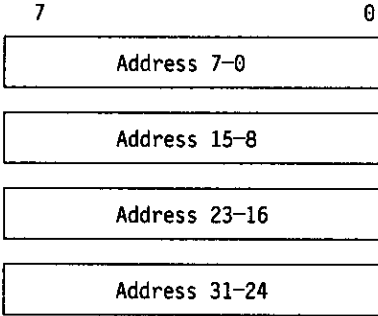
This command returns the contents of the Memory Address register in byte increments. The DMA channel is specified in the Select field.

Write to Port 0018



Read Port 001A

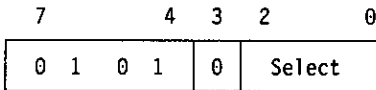
Transferred Data Bytes



Read Transfer Count

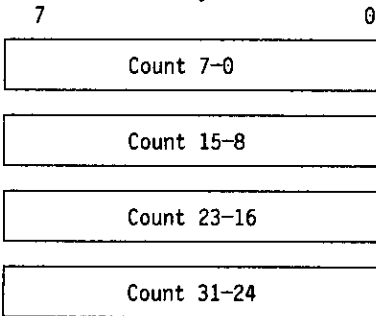
This command returns the contents of the Transfer Count register in byte increments. The DMA channel is specified in the Select field.

Write to Port 0018



Read Port 001A

Transferred Data Bytes

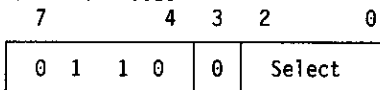


Read DMA Status

This command returns each byte of the DMA Status register and clears that byte of the DMA Status register (the byte is set to hex 00).

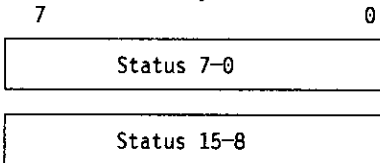
The Select field does not affect this command.

Write to Port 0018



Read Port 001A

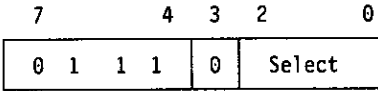
Transferred Data Bytes



Read DMA Mode

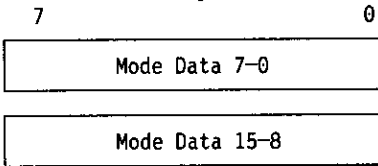
This command returns the contents of the DMA Mode register in byte increments. The DMA channel is specified in the Select field.

Write to Port 0018



Read Port 001A

Transferred Data Byte

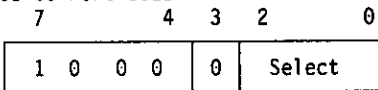


Read or Write Arbitration Level

The read command returns the contents of the Arbitration Level register. The write command loads the transferred data byte into the Arbitration Level register.

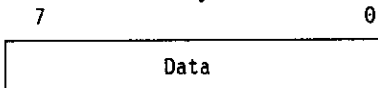
The DMA channel is specified in the Select field.

Write to Port 0018



Access Port 001A

Transferred Data Byte

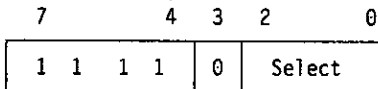


Read DMA Busy

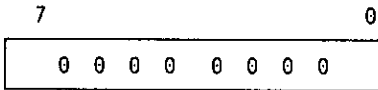
This command returns the inverse of the contents of the DMA Mask register. Each bit in the returned data byte indicates whether a DMA channel has an operation pending. If the bit is 0, the corresponding DMA channel does not have an operation pending (it is not busy); if the bit is 1, the channel has not completed the last operation (is busy).

The Select field has no effect on this command and should be set to 0.

Write to Port 0018

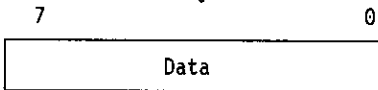


Write to Port 0019



Read Port 001A

Transferred Data Byte

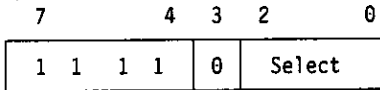


Restore FIFO Data

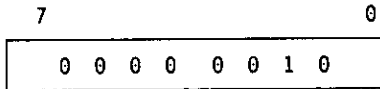
DMA controllers that do not implement a FIFO buffer must ensure that this command is treated as a no-op command. (For information on FIFO implementations, see Appendix B, "FIFO" on page 79.)

Note: This command must be performed before attempting to read the FIFO Byte Count register.

Write to Port 0018

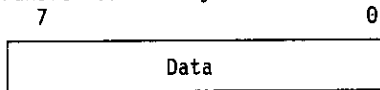


Write to Port 0019



Read Port 001A

Transferred Data Byte

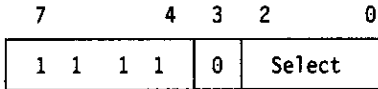


Read FIFO Byte Count

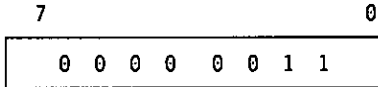
DMA controllers that do not implement a FIFO buffer must ensure that this command returns a hex 00. (For information on FIFO implementations, see Appendix B, "FIFO" on page 79.)

Note: The Restore FIFO Data command must be performed before attempting to read the FIFO byte count.

Write to Port 0018

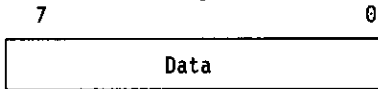


Write to Port 0019



Read Port 001A

Transferred Data Byte

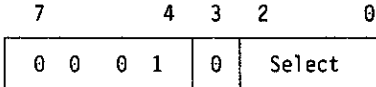


Read or Write Control-Block Address

The read command returns the contents of the Control-Block Address register in byte increments. The write command loads the transferred data bytes into the appropriate bytes of the Control-Block Address register. The DMA channel is specified in the Select field.

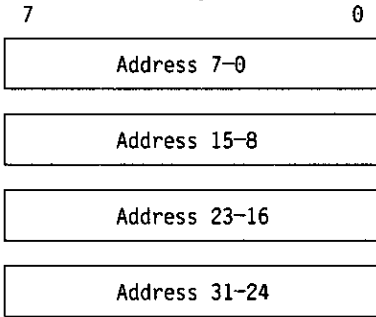
Note: When an indirect list is being used, the data returned by the read command is valid only if the corresponding DMA Request field is 0 (data has not been transferred). Otherwise, the address returned is always valid.

Write to Port 0018



Access Port 001A

Transferred Data Bytes

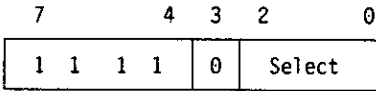


Read Indirect List Address

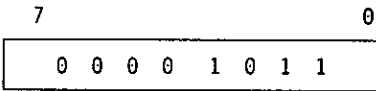
This command returns the contents of the Indirect List Address register in byte increments. The DMA channel is specified in the Select field.

Note: The data returned by this command is valid only if the corresponding DMA Request field is 1.

Write to Port 0018

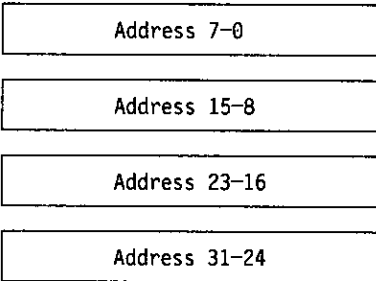


Write to Port 0019



Read Port 001A

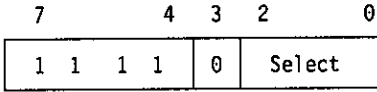
Transferred Data Bytes



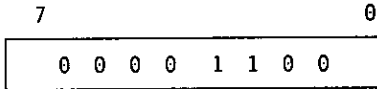
Read Indirect List Count

This command returns the contents of the Indirect List Count register in byte increments. The DMA channel is specified in the Select field.

Write to Port 0018

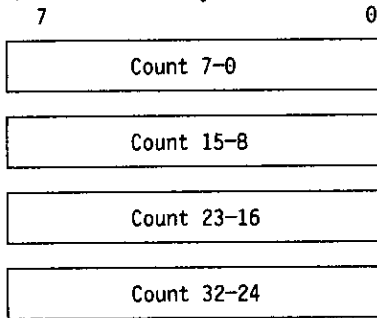


Write to Port 0019



Read Port 001A

Transferred Data Byte



Write Attention Code

This command loads the transferred data byte into the Attention register. The DMA channel is specified in the Select field.

Write to Port 001C

Transferred Data Byte

7 4 3 2 0

0	0	1	1	0	Select	
---	---	---	---	---	--------	--

Write Subsystem Control

This command clears the Busy and Reject fields in the Command Busy/Status register.

Write to Port 001D

Transferred Data Byte

7 6 5 4 0

0	0	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Read Command Busy/Status

This command returns the contents of the Command Busy/Status register.

Read Port 001F

Transferred Data Byte

7 0

Data							
------	--	--	--	--	--	--	--

SCB Operation Programming Model

Each of the eight DMA channels can be programmed independently and can operate concurrently. The DMA controller accepts commands only from the system master and accepts commands to a DMA channel regardless of whether the channel is transferring data or not.

Initialization

A form of the Locate mode is used to deliver DMA transfer requests to the DMA controller. Although the actual DMA transfer is defined using the control block, the system master uses commands to check the status, to load the operational control registers, and to deliver the SCB request. The following is the procedure for starting a DMA operation using the SCB interface.

1. Set up the control block and the indirect list in memory.
2. Issue the Read DMA Busy command to ensure that the DMA channel is not busy with a previous transfer. If it is busy, wait until the current transfer is completed or terminated.
3. Issue the Read Command Busy/Status command to ensure that the Attention register is not busy.
4. Load the Control-Block Address register with the address of the control block.
5. Load the Attention register with request code 3 and the DMA channel number.
6. Program the DMA slave to compete for the system channel.

When the arbitration level is granted control of the system channel, the control block is fetched, the DMA transfer registers are initialized, and the DMA operation is started.

Termination

The DMA controller does not require additional steps if the DMA channel operation is allowed to reach terminal count. However, if a DMA channel operation is terminated before terminal count is reached, a Reset DMA Channel command must be issued (see page 64). This command clears all remaining operations enqueued for the specified DMA channel. Other actions can cause subsequent operations using the same DMA channel to have unpredictable results.

Appendix A. Architecture Compliance

For all PS/2 systems that are based on the Micro Channel architecture, the DMA controller must provide support for the PIO interface (see Section 2, "PIO Interface" on page 9). For those systems with 32-bit Micro Channel connectors, the DMA controller can also provide support for the SCB interface (see Section 3, "SCB Interface" on page 33).

DMA Controller Levels

Type 1

The Type 1 DMA controllers provides the PIO interface and does not use a FIFO buffer.

Type 2

The Type 2 DMA controllers provides both the PIO and SCB interfaces and implements the Data Holding register as a FIFO buffer.

Notes:

Appendix B. FIFO

Basically, DMA controller operations involving a FIFO buffer are the same as those operations without a FIFO buffer. However, DMA applications that are concerned with residual data in the FIFO buffer must take the presence of the buffer into consideration when recovering from errors.

FIFO Implementation Requirements

A DMA controller that provides the SCB interface must implement the commands shown in Figure 15. These commands act on the Data Holding register and, when the Data Holding register is implemented as a FIFO buffer, these commands return values in support of the FIFO buffer. For descriptions of the commands, see "FIFO Commands" on page 82.

Extended Function	Function	Command	Command Class
02	F	Restore FIFO Data	Execute
03	F	Read FIFO Byte Count	Execute

Figure 15. Required Function Commands

FIFO Buffer Implementation

The DMA controller that provides the SCB interface can be implemented with a FIFO buffer. When it is implemented, the FIFO buffer replaces the Holding register; the FIFO buffer must be at least four bytes (the same minimum as the Data Holding register). The additional registers and commands are described in the following.

If residual data handling is supported, the DMA controller can implement the FIFO buffer in either of two ways:

- A single FIFO buffer and FIFO Byte Count register that is shared by all channels and eight backup buffers that are used to save and restore the content of the FIFO buffer for each of the DMA channels.
- Eight FIFO buffers and eight FIFO Byte Count registers, one set for each DMA channel.

FIFO Buffer Registers

The FIFO buffers are not required by the architecture. However, when implementing these buffers, existing registers need to be modified and new registers added. This section describes the changes required to implement the Data Holding register as a FIFO buffer.

The following are the registers that must be added or changed to support a FIFO buffer.

Register	R/W	Page Number
Additional Registers		
FIFO Byte Count	R	80
FIFO Data	R/W	80
Changed Registers		
Extended Function	R/W	36
DMA Mode	R	44
Transfer Count	R	47

Figure 16. Registers Affected by Implementing FIFO Buffer

FIFO Byte Count Register

The FIFO Byte Count register is an 8-bit, read-only register that contains the number of bytes of data in the FIFO buffer. This register can be read using the Read FIFO Byte Count command (see page 85).

FIFO Data Buffer

The FIFO data buffer is a read-write buffer that is used to temporarily store the data being transferred. For the SCB interface, the minimum size of the buffer (when implemented) is 4 bytes. The maximum size is 255 bytes.

The content of the FIFO buffer can be read with the Read FIFO Data command and can be written with the Write FIFO Data command.

Extended Function Register

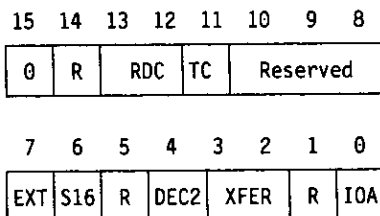
This register (defined on page 36) must provide the following additional extended functions in support of the FIFO buffer.

Extended Function (hex)	Command to Perform
02	Save or restore FIFO data
03	Read FIFO byte count
04	Read or write FIFO data buffer

Figure 17. Additional Extended Function Commands

DMA Mode Registers

This register (defined on page 44) defines the Residual Data Control field in support of the FIFO buffer. The field is loaded from the Residual Data Control field in the Command word during an SCB fetch operation.



RDC Residual Data Control
R Reserved

RDC The Residual Data Control field (bits 13 and 12) specifies whether the FIFO buffer will hold residual data when the DMA controller releases the channel before completing the DMA transfer operation.

RDC Field	Description
0 0	Residual data disabled The FIFO buffer contains data only while controlling the system channel. All data is transferred before the system channel is released.
0 1	Residual data disabled The FIFO buffer contains data only while controlling the system channel. All data is transferred before the system channel is released.
1 0	Residual data partially enabled For I/O-to-memory transfers, all data is transferred before the system channel is released. However, DMA transfers from memory allow data to be held in the FIFO after the controller has released the system channel.
1 1	Residual data fully enabled All operations allow data being transferred to be held in the FIFO buffer after the controller has released the system channel.

Figure 18. Residual Data Control Field

Transfer Count Register

When using a FIFO buffer, the Transfer Count register operates as before; however, the DMA transfer is not complete until all data is transferred from the FIFO buffer (the FIFO Byte Count register is 0).

State After Reset

After a reset operation, the state of the additional registers associated with the FIFO buffer is shown in the following table.

Register	Value	Description
FIFO Byte Count	00	FIFO empty
FIFO Data	-	FIFO empty
FIFO Backup	-	Not initialized

Figure 19. FIFO Register Initialization

FIFO Commands

This section describes the commands that must be added or changed to support implementing a FIFO buffer.

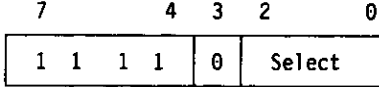
Restore FIFO Data

This command moves the content of the backup buffer into the FIFO buffer. The Select field specifies the DMA channel.

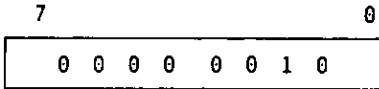
The transferred data byte is always 00.

Note: This command must be performed before attempting to read the FIFO byte count or data buffer.

Write to Port 0018

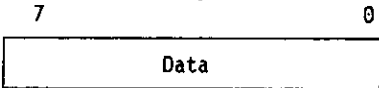


Write to Port 0019



Read Port 001A

Transferred Data Byte

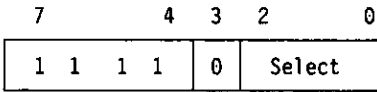


Read FIFO Byte Count

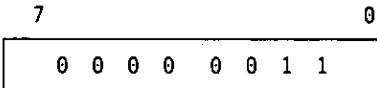
The read command returns the contents of the FIFO Byte Count register. The DMA channel is specified in the Select field.

Note: The restore command must be performed before attempting to read the FIFO byte count.

Write to Port 0018

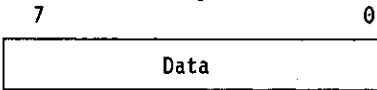


Write to Port 0019



Read Port 001A

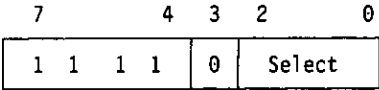
Transferred Data Byte



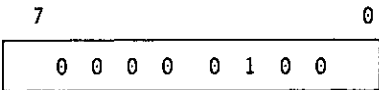
Read or Write FIFO Data

The read command returns a byte from the FIFO buffer and decreases the FIFO byte count by 1. The write command loads the transferred data byte into the FIFO buffer and increases the FIFO byte count by 1. The Select field specifies the DMA channel associated with the data being read or written.

Write to Port 0018

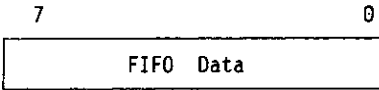
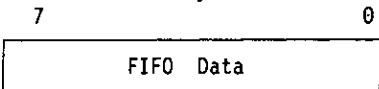


Write to Port 0019



Access Port 001A

Transferred Data Bytes



Reset DMA Controller

In addition to the operations defined in "Reset DMA Controller" on page 64, the DMA controller using a FIFO buffer must also set the FIFO Byte Count register to hex 00.

Reset DMA Channel

In addition to the operations defined in "Reset DMA Channel" on page 64, the DMA controller using a FIFO buffer must also set the FIFO Byte Count register to hex 00 (the Backup FIFO register, if implemented, is not affected).

FIFO Operation Program Model

The following shows a programming model for operations that work with the contents of the FIFO buffer. These operations are used when an application is recovering residual data that may be in the FIFO buffer.

Reading from the FIFO buffer

- Issue the Restore FIFO Buffer command
- Issue the Read FIFO Byte Count command
- If *byte count* is greater than 0
 - Then issue the Read DMA FIFO Data command for *byte count* iterations.
 - Otherwise, the FIFO buffer is empty or does not exist.

Note: This procedure does not require that the programmer know the actual size of the FIFO buffer or even if one is implemented.

Writing to the FIFO buffer

- Determine the state of the FIFO buffer
 - If state of the FIFO buffer is unknown, reset the DMA channel.
 - Otherwise, read the FIFO byte count to ensure that the FIFO buffer is empty or at a know state.
- Write data to the buffer up to the actual size of the FIFO buffer.
- Save the FIFO data, specifying the DMA channel.

Note: This procedure requires that the programmer know the size and residual support provided in the FIFO implementation.

Notes:

Index

A

address field, control block 51
arbitration level 13
arbitration level, read/write 31, 68
arbitration level, SCB 41
architected field 49
attention field 38
attention register 38, 58
attention, write 75

B

base commands 24
 categories 24
 execute 27
 read current transfer count 29
 read memory address 28
 read status 30
 read/write arb level 31
 read/write I/O 27
 read/write mode 30
 reset controller 26
 set/reset mask 25
 write base memory 28
 write base transfer count 29
 write function 25
base memory address, write 28
base transfer count, write 29
bursting device 6
bus, arbitration 5
busy field 39
busy status, read 75
busy, read 69
byte pointer 12
byte pointer, SCB 36

C

categories, command 24, 61
chaining 48
chaining data field 50

command word 49
command, control block word 49
commands, base 24
commands, SCB 61
 read busy status 75
 read DMA busy 69
 read FIFO count 71, 85
 read I/O 65
 read indirect list address 73
 read indirect list count 74
 read memory address 66
 read status 67
 read transfer count 67
 read-write FIFO 86
 read/write arb level 68
 read/write control-block
 address 72
 read/write function 62
 read/write mode 68
 restore FIFO 70
 save/restore FIFO 83
 set/reset mask 63
 write attention 75
 write control byte 75
commands, immediate 25, 63
control block address register 58
control blocks 48
control byte, write 75
control registers, SCB 37
control-block address register 37
control-block address,
 read/write 72
control-block error indicator 39
controller reset 64
controller states 4
current transfer count, read 29

D

data chaining 48, 50
data holding register 18, 47
data structures 48

decrement 2 field 45
decrement address field 16
DMA controller, reset command 26
DMA mask register 14, 42
DMA mask, set/reset 25, 63
DMA mode register 16, 44
DMA mode, read 68
DMA mode, read/write 30
DMA request field 15, 43
DMA status register 15, 43
DMA status, read 30, 67

E

enable word 1 50
end of transfer (EOT) 6
error field, control-block 39
execute commands, base 27
execute commands, SCB 65
Extend Register field 45
extended registers
 See registers, SCB

F

fields, extended
 address 1 and 2 51
 architected 49
 command word 49
 decrement 2 45
 DMA request 43
 enable word 1 50
 I/O address 46
 indirect list 1 50
 terminal count 43
 transfer control 45
 transfer count 51

fields, PIO

 decrement address 16
 DMA request 15
 function 11
 I/O address 17
 select 11
 size 16 16
 terminal count 15
 transfer control 17

FIFO count, read 71, 85
function command, read/write 62
function command, write 25
function field (PIO) 11
function register 11

G

granted control 5

H

holding register, data 18, 47

I

I/O address command, read 65
I/O address command,
 read/write 27
I/O address field 17, 46
I/O address register 17
I/O to memory transfers 23
idle state 5
immediate commands, base 25
immediate commands,
 extended 63
indirect list 1 field 50
indirect list address register 40, 59
indirect list address, read 73
indirect list count 40, 59
indirect list count, read 74
indirect lists 52

L

list address, read indirect 73
list count, read indirect 74
list, indirect 50

M

mask register 14, 42
mask, read 69
mask, set/reset 63
mask, set/reset DMA 25
memory address register 18
memory address, write 28

memory to I/O transfers 22, 55
memory verify operations 21, 54
mode register 16, 44

N

nonbursting device 6
nonchained data 48

O

operating states 4
operation control registers 13, 41
operations, PIO
 arbitration level 13
 data holding 18
 DMA mask 14
 DMA mode 16
 DMA status 15
 function 11
 I/O address 17
 I/O to memory 23
 memory address 18
 memory to I/O 22
 memory verify 21
 transfer count 18
operations, SCB
 I/O to memory 56
 memory to I/O 55
 memory verify 54

P

PIO commands 24
PIO fields
 See fields, PIO
PIO interface
PIO operations
 See operations, PIO
PIO registers
 See operations, PIO
pointer, byte 12
program state 5

R

read busy 69
read busy status 75
read DMA mode 68
read DMA status 30, 67
read FIFO count 71, 85
read I/O-memory address 65
read indirect list address 73
read indirect list count 74
read memory address 28, 66
read-write FIFO data 86
read/write arb level 31, 68
read/write control-block
 address 72
read/write DMA mode 30
read/write function command 62
read/write I/O address 27
registers, SCB
 arbitration level 41
 data holding 47
 DMA mask 42
 DMA mode 44
 DMA status 43
reject field 39
reject field reset, busy field
 reset 38
reset controller 26, 64
reset mask 26, 63
restore FIFO 70, 84

S

save or restore FIFO 83
SCB control registers 37
SCB control structures 48
SCB fields
 See fields, extended
SCB interface
SCB operations
 See operations, SCB
SCB registers
 See registers, SCB
select field 11
select field, SCB 38
set mask 25, 63
size 16 16

- states, operating 4
- status register, busy 39
- status, PIO 15
- status, read DMA 30, 67
- status, SCB 43
- structures, control block 48
- subsystem control register 38
- system interface registers 11, 35

T

- terminal count 15, 43
- transfer control field 17, 45
- transfer control registers 16, 44
- transfer count field 51
- transfer count registers 18
- transfer count, read 29, 67
- transfer count, write 29
- transfer size 16 16
- transfer state 6

V

- verify memory operations 21, 54

W

- word, command 49
- write arb level 31, 68
- write attention 75
- write base memory address 28
- write base transfer count 29
- write control byte 75
- write control-block address 72
- write DMA mode 30
- write function command 25
- write I/O address 27