Section 6 Data Transfer Controller

6.1 Overview

The H8/532 chip includes a data transfer controller (DTC) that can be started by designated interrupts to transfer data from a source address to a destination address located in page 0. These addresses include in particular the registers of the on-chip supporting modules and I/O ports. Typical uses of the DTC are to change the setting of a control register of an on-chip supporting module in response to an interrupt from that module, or to transfer data from memory to an I/O port or the serial communication interface. Once set up, the transfer is interrupt-driven, so it proceeds independently of program execution, although program execution temporarily stops while each byte or word is being transferred.

6.1.1 Features

The main features of the DTC are listed below.

- The source address and destination address can be set anywhere in the 64k-byte address space of page 0.
- The DTC can be programmed to transfer one byte or one word of data per interrupt.
- The DTC can be programmed to increment the source address and/or destination address after each byte or word is transferred.
- After transferring a designated number of bytes or words, the DTC generates a CPU interrupt with the vector of the interrupt source that started the DTC.
- This designated data transfer count can be set from 1 to 65,536 bytes or words.

6.1.2 Block Diagram

Figure 6-1 shows a block diagram of the DTC.

The four DTC control registers (DTMR, DTSR, DTDR, and DTCR) are invisible to the CPU, but corresponding information is kept in a register information table in memory. A separate table is maintained for each DTC interrupt type. When an interrupt requests DTC service, the DTC loads its control registers from the table in memory, transfers the byte or word of data, and writes any altered register information back to memory.

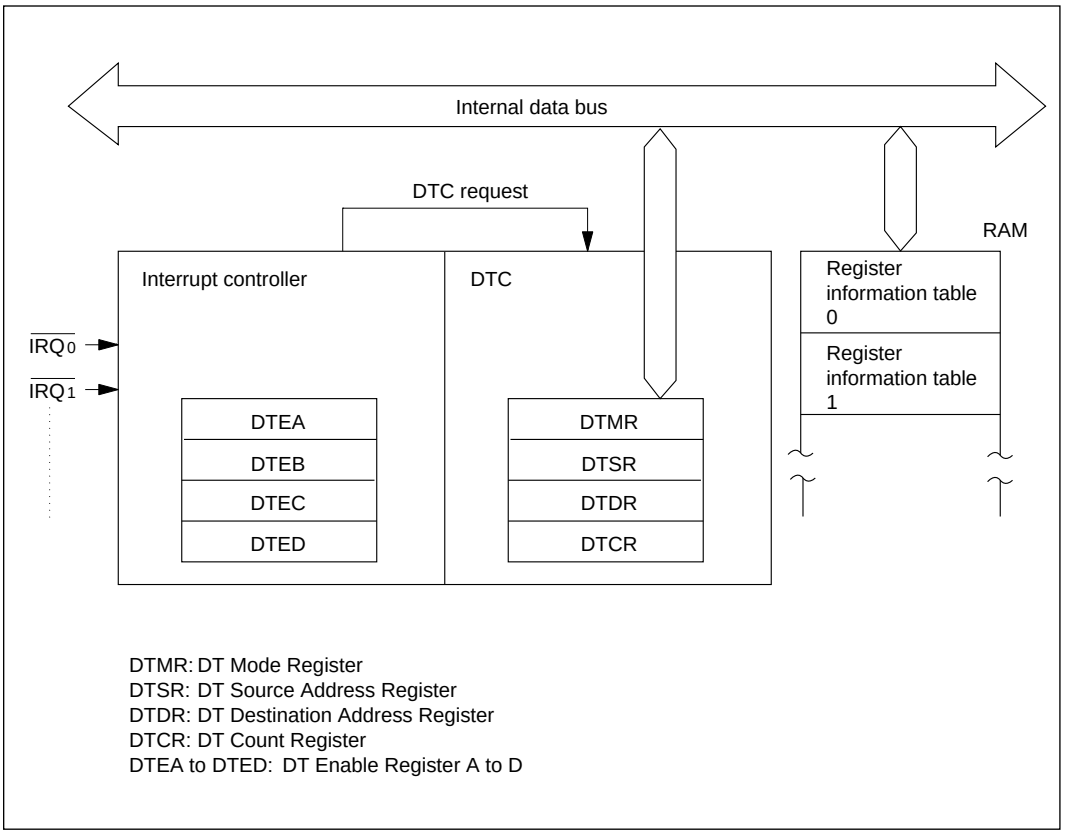


Figure 6-1 Block Diagram of Data Transfer Controller

6.1.3 Register Configuration

The four DTC control registers are listed in table 6-1. These registers are not located in the address space and cannot be written or read by the CPU. To set information in these registers, a program must write the information in a table in memory from which it will be loaded by the DTC.

Table 6-1 Internal Control Registers of the DTC

Name	Abbreviation	Read/Write
Data transfer mode register	DTMR	Disabled
Data transfer source address register	DTSR	Disabled
Data transfer destination address register	DTDR	Disabled
Data transfer count register	DTCR	Disabled

Starting of the DTC is controlled by the four data transfer enable registers, which are located in high addresses in page 0. Table 6-2 lists these registers.

Table 6-2 Data Transfer Enable Registers

Name		Abbreviation	Read/Write	Address	Initial Value
Data transfer	A	DTEA	R/W	H'FFF4	H'00
enable	B	DTEB	R/W	H'FFF5	H'00
register	C	DTEC	R/W	H'FFF6	H'00
	D	DTED	R/W	H'FFF7	H'00

6.2 Register Descriptions

6.2.1 Data Transfer Mode Register (DTMR)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Sz	SI	DI	—	—	—	—	—	—	—	—	—	—	—	—	—
Read/Write	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

The data transfer mode register is a 16-bit register, the first three bits of which designate the data size and specify whether to increment the source and destination addresses.

Bit 15—Sz (Size): This bit designates the size of the data transferred.

Bit 15

Sz	Description
0	Byte transfer
1	Word transfer* (two bytes at a time)

* For word transfer, the source and destination addresses must be even addresses.

Bit 14—SI (Source Increment): This bit specifies whether to increment to source address.

Bit 14

SI	Description
0	Source address is not incremented.
1	1) If Sz = 0: Source address is incremented by +1 after each data transfer. 2) If Sz = 1: Source address is incremented by +2 after each data transfer.

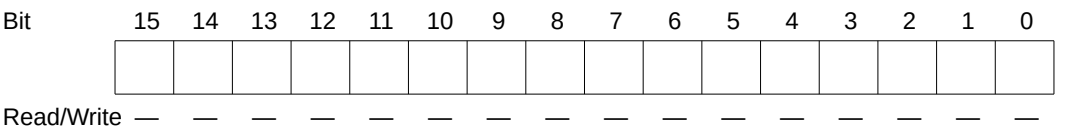
Bit 13—DI (Destination Increment): This bit specifies whether to increment to destination address.

Bit 13

DI	Description
0	Destination address is not incremented.
1	1) If Sz = 0: Destination address is incremented by +1 after each data transfer. 2) If Sz = 1: Destination address is incremented by +2 after each data transfer.

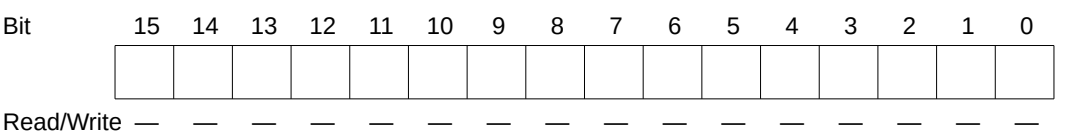
Bits 12 to 0—Reserved Bits: These bits are reserved.

6.2.2 Data Transfer Source Address Register (DTSR)



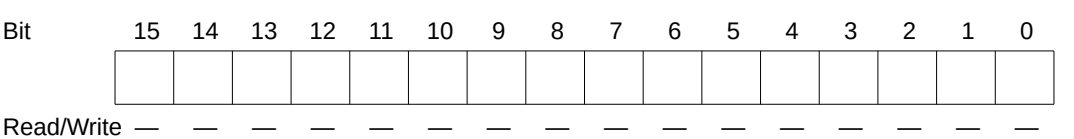
The data transfer source register is a 16-bit register that designates the data transfer source address. For word transfer this must be an even address. In the maximum mode, this address is implicitly located in page 0.

6.2.3 Data Transfer Destination Register (DTDR)



The data transfer destination register is a 16-bit register that designates the data transfer destination address. For word transfer this must be an even address. In the maximum mode, this address is implicitly located in page 0.

6.2.4 Data Transfer Count Register (DTCR)



The data transfer count register is a 16-bit register that counts the number of bytes or words of data remaining to be transferred. The initial count can be set from 1 to 65,536. A register value of 0 designates an initial count of 65,536.

The data transfer count register is decremented automatically after each byte or word is transferred. When its value reaches 0, indicating that the designated number of bytes or words have been transferred, a CPU interrupt is generated with the vector of the interrupt that requested the data transfer.

6.2.5 Data Transfer Enable Registers A to D (DTEA to DTED)

These four registers designate whether an interrupt starts the DTC. The bits in these registers are assigned to interrupts as indicated in table 6-3. No bits are assigned to the NMI, FOVI, OVI, and ERI interrupts, which cannot request data transfers.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6-3 Assignment of Data Transfer Enable Registers

	Interrupt Source					Interrupt Source				
Register	**Module**	**Bits 7 to 4**				**Module**	**Bits 3 to 0**			
		7	6	5	4		3	2	1	0
DTEA	$\overline{IRQ_0}$	—	—	—	IRQ_0	$\overline{IRQ_1}$	—	—	—	IRQ_1
DTEB	16-Bit FRT1	—	OCIB	OCIA	ICI	16-Bit FRT2	—	OCIB	OCIA	ICI
DTEC	16-Bit FRT3	—	OCIB	OCIA	ICI	8-Bit Timer	—	—	CMIB	CMIA
DTED	SCI	—	TXI	RXI	—	A/D converter	—	—	—	ADI

Note: Bits marked "—" should always be cleared to "0."

If the bit for a certain interrupt is set to "1," that interrupt is regarded as a request for DTC service. If the bit is cleared to "0," the interrupt is regarded as a CPU interrupt request.

Only the 16 interrupts indicated in table 6-3 can request DTC service. DTE bits not assigned to any interrupt (indicated by "—" in table 6-3) should be left cleared to "0."

• **Note on Timing of DTE Modifications:** The interrupt controller requires two system clock (ø) periods to determine the priority level of an interrupt. Accordingly, when an instruction modifies a data transfer enable register, the new setting does not take effect until the third state after that instruction has been executed.

6.3 Data Transfer Operation

6.3.1 Data Transfer Cycle

When started by an interrupt, the DTC executes the following data transfer cycle:

1. From the DTC vector table, the DTC reads the address at which the register information table for that interrupt is located in memory.
2. The DTC loads the data transfer mode register and source address register from this table and reads the data (one byte or word) from the source address.
3. If so specified in the mode register, the DTC increments the source address register and writes the new source address back to the table in memory.
4. The DTC loads the data transfer destination address register and writes the byte or word of data to the destination address.
5. If so specified in the mode register, the DTC increments the destination address register and writes the new destination address back to the table in memory.
6. The DTC loads the data transfer count register from the table in memory, decrements the data count, and writes the new count back to memory.
7. If the data transfer count is now 0, the DTC generates a CPU interrupt. The interrupt vector is the vector of the interrupt type that started the DTC.

At an appropriate point during this procedure the DTC also clears the interrupt request by clearing the corresponding flag bit in the status register of the on-chip supporting module to “0.” (For IRQ_0 or IRQ_1, the DTC clears an internal latch.)

But the DTC does not clear the data transfer enable bit in the data transfer enable register. This action, if necessary, must be taken by the user-coded interrupt-handling routine invoked at the end of the transfer.

The data transfer cycle is shown in a flowchart in figure 6-2.

For the steps from the occurrence of the interrupt up to the start of the data transfer cycle, see section 5.4.1, “Interrupt Handling Flow.”

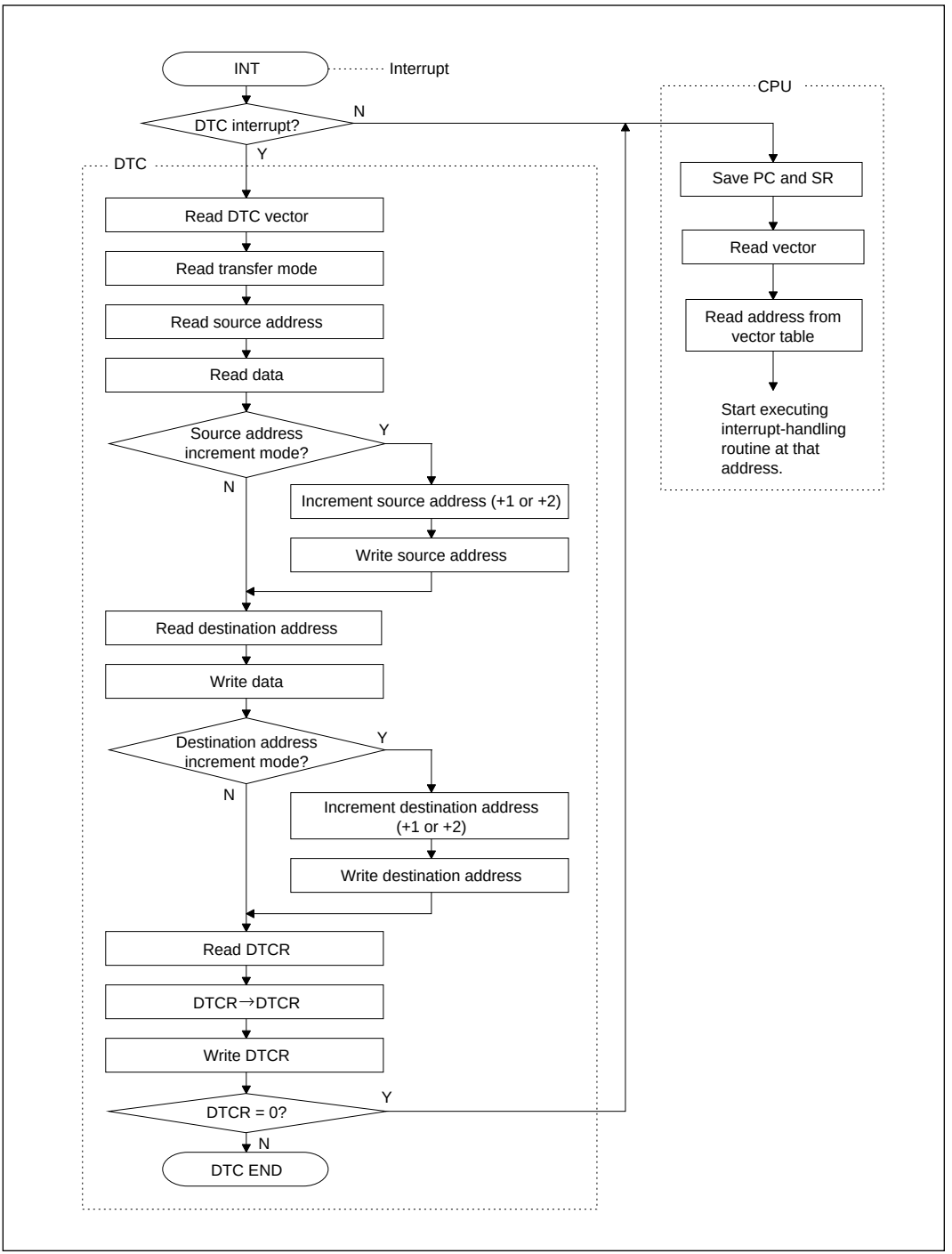
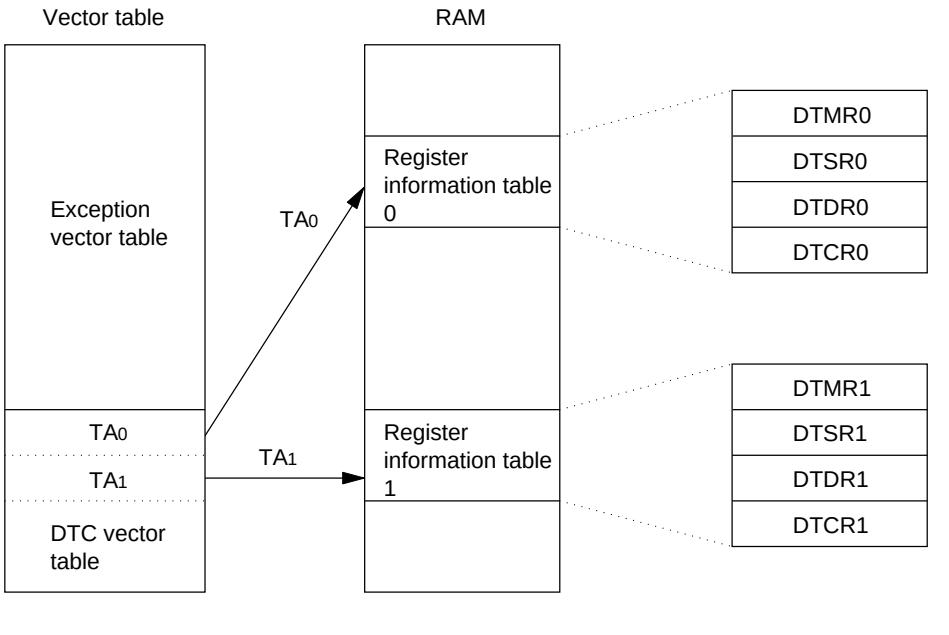


Figure 6-2 Flowchart of Data Transfer Cycle

6.3.2 DTC Vector Table

The DTC vector table is located immediately following the exception vector table at the beginning of page 0 in memory. For each interrupt that can request DTC service, the DTC vector table provides a pointer to an address in memory where the table of DTC control register information for that interrupt is stored. The register information tables can be placed in any available locations in page 0.



Note: TA_0, TA_1, ...: Addresses of DTC register information tables in memory.

In the normal case the register information tables are placed on a RAM. If the software does not need to modify the register information (addresses are fixed and transfer count is 1), it can be placed on ROM.

Figure 6-3 DTC Vector Table

In minimum mode, each entry in the DTC vector table consists of two bytes, pointing to an address in page 0. In maximum mode, for compatibility reasons, each DTC vector table entry consists of four bytes but the first two bytes are ignored; the last two bytes point to an address which is implicitly assumed to be in page 0, regardless of the current page specifications.

Figure 6-4 shows one DTC vector table entry in minimum and maximum mode.

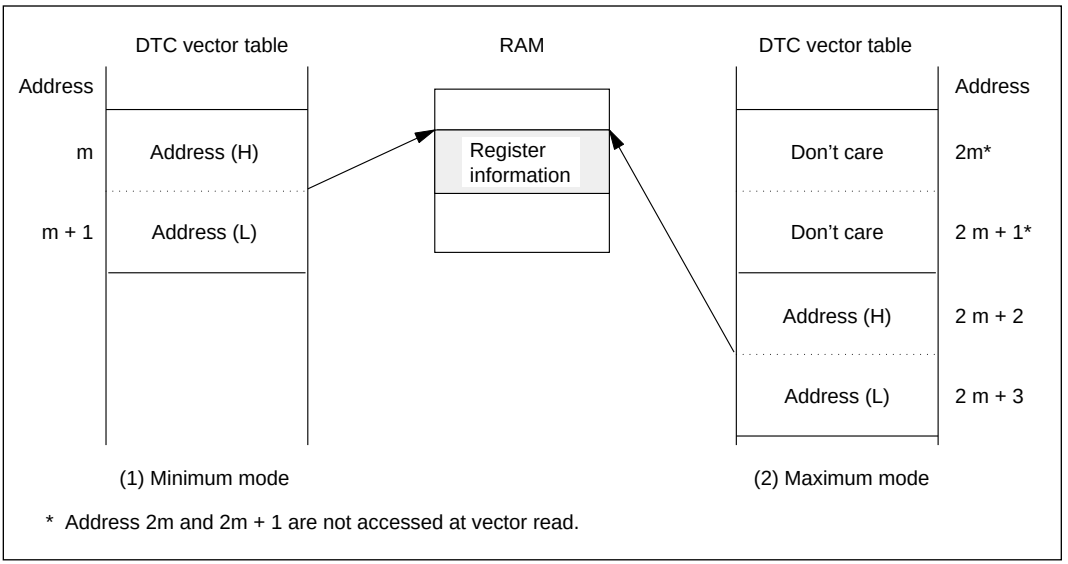


Figure 6-4 DTC Vector Table Entry

Table 6-4 lists the addresses of the entries in the DTC vector table for each interrupt.

Table 6-4 Addresses of DTC Vectors

		Address of DTC Vector	
Interrupt		**Minimum Mode**	**Maximum Mode**
IRQ_0		H'0080 - H'0081	H'0100 - H'0103
IRQ_1		H'0082 - H'0083	H'0104 - H'0107
16-Bit	ICI	H'0088 - H'0089	H'0110 - H'0113
free-running	OCIA	H'008A - H'008B	H'0114 - H'0117
timer 1	OCIB	H'008C - H'008D	H'0118 - H'011B
(FRT1)	FOVI	—	—
16-Bit	ICI	H'0090 - H'0091	H'0120 - H'0123
free-running	OCIA	H'0092 - H'0093	H'0124 - H'0127
timer 2	OCIB	H'0094 - H'0095	H'0128 - H'012B
(FRT2)	FOVI	—	—
16-Bit	ICI	H'0098 - H'0099	H'0130 - H'0133
free-running	OCIA	H'009A - H'009B	H'0134 - H'0137
timer 3	OCIB	H'009C - H'009D	H'0138 - H'013B
(FRT3)	FOVI	—	—

Table 6-4 Addresses of DTC Vectors (cont)

Interrupt		Address of DTC Vector	
		Minimum Mode	Maximum Mode
8-Bit	CMIA	H'00A0 - H'00A1	H'0140 - H'0143
timer	CMIB	H'00A2 - H'00A3	H'0144 - H'0147
	OVI	—	—
Serial	ERI	—	—
communication	RXI	H'00AA - H'00AB	H'0154 - H'0157
interface	TXI	H'00AC - H'00AD	H'0158 - H'015B
A/D converter	ADI	H'00B0 - H'00B1	H'0160 - H'0163

6.3.3 Location of Register Information in Memory

For each interrupt, the DTC control register information is stored in four consecutive words in memory in the order shown in figure 6-5.

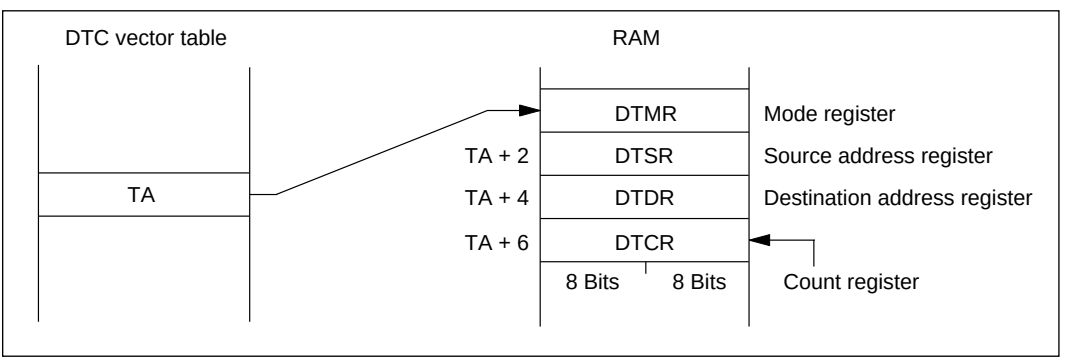


Figure 6-5 Order of Register Information

6.3.4 Length of Data Transfer Cycle

Table 6-5 lists the number of states required per data transfer, assuming that the DTC control register information is stored in on-chip RAM. This is the number of states required for loading and saving the DTC control registers and transferring one byte or word of data. Two cases are considered: a transfer between on-chip RAM and a register belonging to an I/O port or on-chip supporting module (i.e., a register in the register field from addresses H'FF80 to H'FFFF); and a transfer between such a register and external RAM.

Table 6-5 Number of States per Data Transfer

Increment Mode		On-Chip RAM ↔ Module or I/O Register		External RAM ↔ Module or I/O Register	
Source (SI)	Destination (DI)	Byte Transfer	Word Transfer	Byte Transfer	Word Transfer
0	0	31	34	32	38
0	1	33	36	34	40
1	0	33	36	34	40
1	1	35	38	36	42

Note: Numbers in the table are the number of states.

The values in table 6-5 are calculated from the formula:

$$N = 26 + 2 \times SI + 2 \times DI + M_S + M_D$$

Where M_S and M_D have the following meanings:

M_S: Number of states for reading source data

M_D: Number of states for writing destination data

The values of M_S and M_D depend on the data location as follows:

① Byte or word data in on-chip RAM: ⇨ 2 states

② Byte data in external RAM or register field: ⇨ 3 states

③ Word data in external RAM or register field: ⇨ 6 states

If the DTC control register information is stored in external RAM, $20 + 4 \times SI + 4 \times DI$ must be added to the values in table 6-5.

The values given above do not include the time between the occurrence of the interrupt request and the starting of the DTC. This time includes two states for the interrupt controller to check priority and a variable wait until the end of the current CPU instruction. At maximum, this time equals the sum of the values indicated for items No. 1 and 2 in table 6-6.

If the data transfer count is 0 at the end of a data transfer cycle, the number of states from the end of the data transfer cycle until the first instruction of the user-coded interrupt-handling routine is executed is the value given for item No. 3 in table 6-6.

Table 6-6 Number of States before Interrupt Service

No.	Reason for Wait		Number of States: Minimum Mode	Number of States: Maximum Mode
1	Interrupt priority decision and comparison with mask level in CPU status register		2 states	
2	Maximum number of states to completion of current instruction	Instruction is in on-chip memory	38 (LDM instruction specifying all registers)	
		Instruction is in external memory	74 + 16m (LDM instruction specifying all registers)	
3	Saving of PC and SR or PC, CP, and SR and instruction prefetch	Stack is in on-chip RAM	16	21
		Stack is in external memory	28 + 6m	41 + 10m

m: Number of wait states inserted in external memory access

6.4 Procedure for Using the DTC

A program that uses the DTC to transfer data must do the following:

1. Set the appropriate DTMR, DTSR, DTDR, and DTCR register information in the memory location indicated in the DTC vector table.
2. Set the data transfer enable bit of the pertinent interrupt to “1,” and set the priority of the interrupt source (in the interrupt priority register) and the interrupt mask level (in the CPU status register) so that the interrupt can be accepted.
3. Set the interrupt enable bit in the control register for the interrupt source. (For IRQ_0 and IRQ_1, the control register is the port 1 control register, P1CR.)

Following these preparations, the DTC will be started each time the interrupt occurs. When the number of bytes or words designated by the DTCR value have been transferred, after transferring the last byte or word, the DTC generates a CPU interrupt.

The user-coded interrupt-handling routine must take action to prepare for or disable further DTC data transfer: by readjusting the data transfer count, for example, or clearing the interrupt enable bit. If no action is taken, the next interrupt of the same type will start the DTC with an initial data transfer count of 65,536.

6.5 Example

Purpose: To receive 128 bytes of serial data via the serial communication interface.

Conditions:

- Operating mode: Minimum mode
- Received data are to be stored in consecutive addresses starting at H'FC00.
- DTC control register information for the RXI interrupt is stored at addresses H'FB80 to H'FB87.
- Accordingly, the DTC vector table contains H'FB at address H'00AA and H'80 at address H'00AB.
- The desired interrupt mask level in the CPU status register is 4, and the desired SCI interrupt priority level is 5.

Procedure

1. The user program sets DTC control register information in addresses H'FB80 to H'FB87 as shown in table 6-7.

Table 6-7 DTC Control Register Information Set in RAM

Address	Register	Description	Value Set
H'FB80	DTMR	Byte transfer Source address fixed Increment destination address	H'2000
H'FB82	DTSR	Address of SCI receive data register	H'FFDD
H'FB84	DTDR	Address H'FC00	H'FC00
H'FB86	DTCR	Number of bytes to be received: 128	H'0080

2. The program sets the RI (SCI Receive Interrupt) bit in the data transfer enable register (bit 5 of register DTED) to “1.”
3. The program sets the interrupt mask in the CPU status register to 4, and the SCI interrupt priority in bits 6 to 4 of interrupt priority register IPRD to 5.
4. The program sets the SCI to the appropriate receive mode, and sets the receive interrupt enable (RIE) bit in the serial control register (SCR) to “1” to enable receive interrupts.
5. Thereafter, each time the SCI receives one byte of data, it requests an RXI interrupt, which the interrupt controller directs toward the DTC. The DTC transfers the byte from the SCI’s receive data register (RDR) into RAM, and clears the interrupt request before ending.

6. When 128 bytes have been transferred (DTCR = 0), the DTC generates a CPU interrupt. The interrupt type is RXI.
7. The user-coded RXI interrupt-handling routine processes the received data and disables further data transfer (by clearing the RIE bit, for example).

Figure 6-6 shows the DTC vector table and data in RAM for this example.

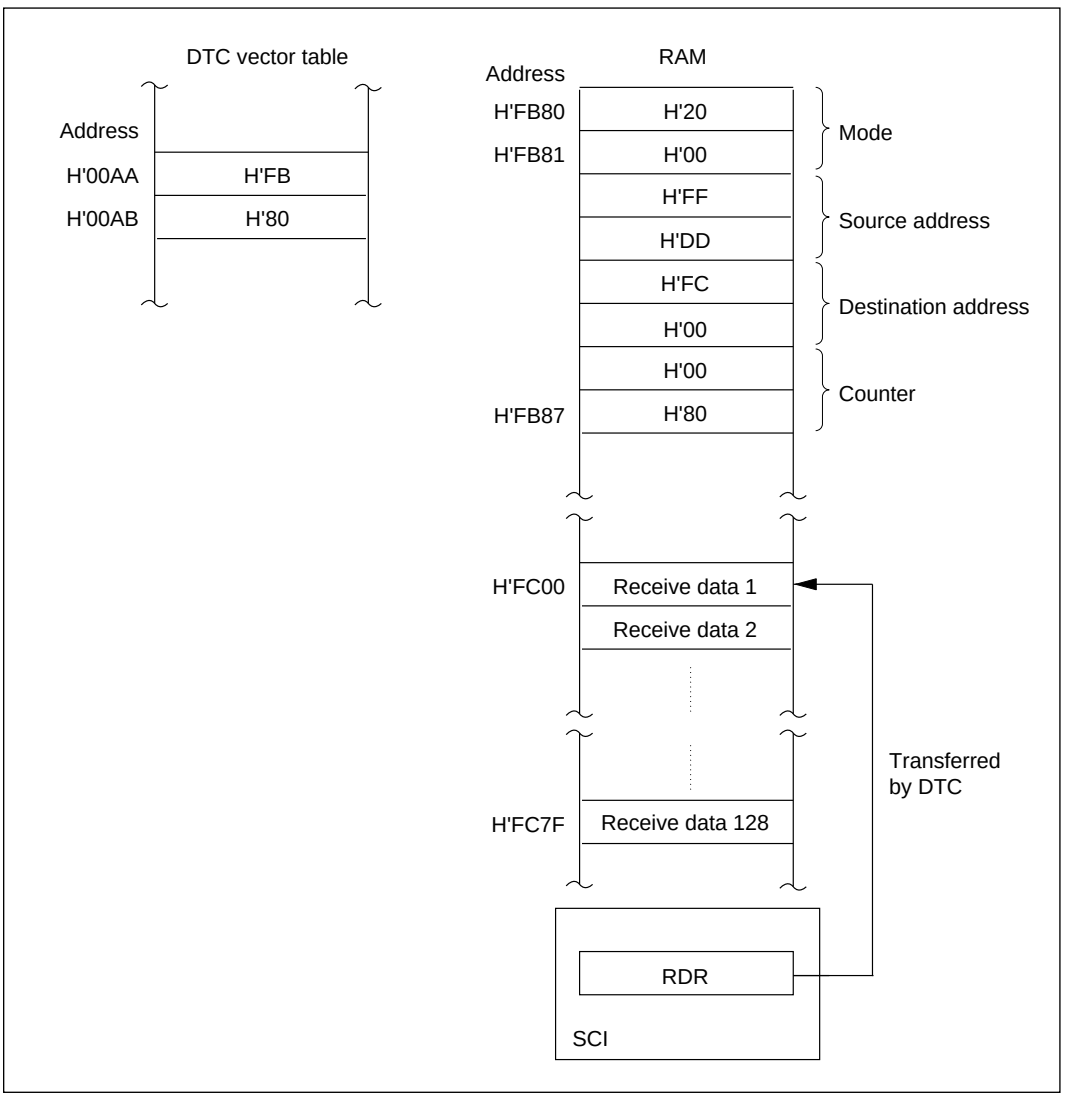


Figure 6-6 Use of DTC to Receive Data via Serial Communication Interface