Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	Z80180
Number of Cores/Bus Width	1 Core, 8-Bit
Speed	6MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8018006vsc00tr

$\overline{\text{RTS0}}$. *Request to Send 0 (Output, Active Low).* This output is a programmable modem control signal for ASCI channel 0.

RXA0, RXA1. *Receive Data 0 and 1 (Inputs, Active High).* These signals are the receive data to the ASCI channels.

RXS. *Clocked Serial Receive Data (Input, Active High).* This line is the receiver data for the CSIO channel. RXS is multiplexed with the $\overline{\text{CTS1}}$ signal for ASCI channel 1.

ST. *Status (Output, Active High).* This signal is used with the $\overline{\text{M1}}$ and $\overline{\text{HALT}}$ output to decode the status of the CPU machine cycle. Table 1 provides status summary.

Table 1. Status Summary

ST	$\overline{\text{HALT}}$	$\overline{\text{M1}}$	Operation
0	1	0	CPU operation (1st Op Code fetch)
1	1	0	CPU operation (2nd Op Code and 3rd Op Code fetch)
1	1	1	CPU operation (MC[2] except for Op Code fetch)
0	X[1]	1	DMA operation
0	0	0	HALT mode
1	0	1	SLEEP mode (including SYSTEM STOP mode)

1. X = Don't care
2. MC = Machine cycle

$\overline{\text{TEND0}}$, $\overline{\text{TEND1}}$. *Transfer End 0 and 1 (Outputs, Active Low).* This output is asserted active during the last write cycle of a DMA operation. It is used to indicate the end of the block transfer. $\overline{\text{TEND0}}$ in multiplexed with CKA1.

TEST. *Test (Output, not on DIP version).* This pin is for test and must be left open.

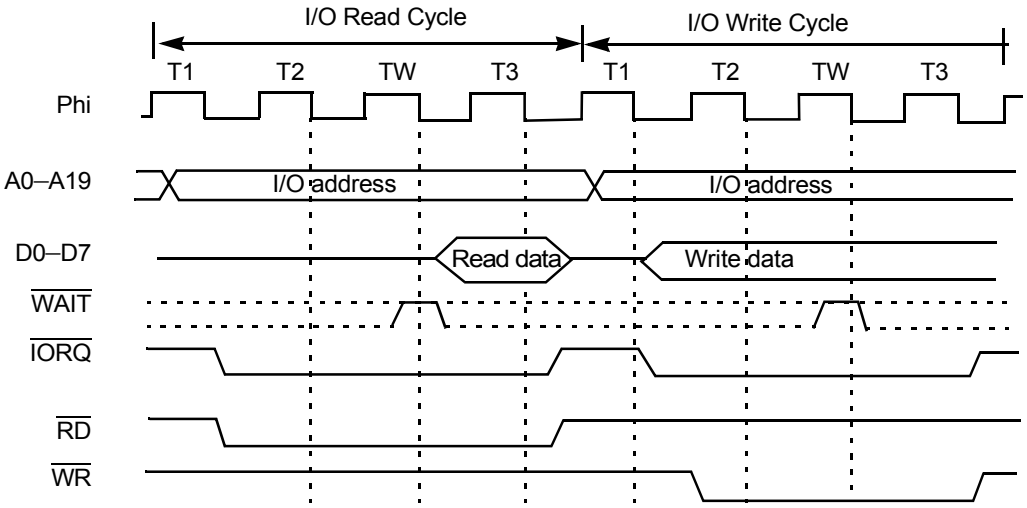

Figure 13. I/O Read/Write Timing Diagram

Basic Instruction Timing

An instruction may consist of a number of machine cycles including Op Code fetch, operand fetch, and data read/write cycles. An instruction may also include cycles for internal processes which make the bus IDLE. The example in Figure 14 illustrates the bus timing for the data transfer instruction LD (IX+d),g.

Table 6. I/O Address Map for Z80180-Class Processors Only (Continued)

	Register	Mnemonic	Address		
			Binary	Hex	Page
INT	IL Register (Interrupt Vector Low Register)	IL	XX110011	33H	67
	INT/TRAP Control Register	ITC	XX110100	34H	68
	Reserved		XX110101	35H	
Refresh	Refresh Control Register	RCR	XX110110	36H	88
	Reserved		XX110111	37H	
MMU	MMU Common Base Register	CBR	XX111000	38H	61
	MMU Bank Base Register	BBR	XX111001	39H	62
	MMU Common/Bank Area Register	CBAR	XX111010	3AH	60
I/O	Reserved		XX111011	3BH	
			↕	↕	
			XX111101	3DH	
	Operation Mode Control Register	OMCR	XX111110	3EH	15
	I/O Control Register	ICR	XX111111	3FH	42

vector table can be relocated on 32 byte boundaries. IL is initialized to 00H during RESET.

Interrupt Vector Low Register (IL: 33H)

Bit	7	6	5	4	3	2	1	0
Bit/Field	IL7	IL6	IL5	?				
R/W	R/W	R/W	R/W	?				
Reset	00H	00H	00H	?				

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

Bit Position	Bit/Field	R/W	Value	Description
7–5	IL7–5	R/W		The IL register is an internal I/O register which is programmed with the OUT0 instruction and can be read using the IN0 instruction.
4–0	?	N/A		Interrupt source dependent code

INT/TRAP Control Register (ITC)

ITC is used to handle TRAP interrupts and to enable or disable the external maskable interrupt inputs $\overline{INT0}$, $\overline{INT1}$ and $\overline{INT2}$.

Figure 35. $\overline{\text{NMI}}$ Timing

INT0 - Maskable Interrupt Level 0

The next highest priority external interrupt after $\overline{\text{NMI}}$ is $\overline{\text{INT0}}$. $\overline{\text{INT0}}$ is sampled at the falling edge of the clock state prior to T3 or T1 in the last machine cycle. If $\overline{\text{INT0}}$ is asserted LOW at the falling edge of the clock state prior to T3 or T1 in the last machine cycle, $\overline{\text{INT0}}$ is accepted. The interrupt is masked if either the IEF1 flag or the ITEO (Interrupt Enable 0) bit in ITC are reset to `0`. After $\overline{\text{RESET}}$ the state is as follows:

1. IEF1 is `0`, so $\overline{\text{INT0}}$ is masked
2. ITE0 is `1`, so $\overline{\text{INT0}}$ is enabled by execution of the El (Enable Interrupts) instruction

The $\overline{\text{INT0}}$ interrupt is unique in that 3 programmable interrupt response modes are available - Mode 0, Mode 1 and Mode 2. The specific mode is selected with the IM 0, IM 1 and IM 2 (Set Interrupt Mode) instructions. During $\overline{\text{RESET}}$, the Z8X180 is initialized to use Mode 0 for $\overline{\text{INT0}}$. The 3 interrupt response modes for $\overline{\text{INT0}}$ are:

- Mode 0–Instruction fetch from data bus
- Mode 1–Restart at logical address `0038H`
- Mode 2–Low-byte vector table address fetch from data bus

INT0 Mode 0

During the interrupt acknowledge cycle, an instruction is fetched from the data bus (DO–D7) at the rising edge of T3. Often, this instruction is one of the eight single byte RST (RESTART) instructions which stack the PC and restart execution at a fixed logical address. However, multibyte instructions can be processed if the interrupt acknowledging device can provide a multibyte response. Unlike all other interrupts, the PC is not automatically stacked:

memory mapped I/O. transfers, the CKA0/$\overline{\text{DREQ0}}$ pin automatically functions as input pin or output pin even if it has been programmed as output pin for CKA0. And the CKA1/$\overline{\text{TEND0}}$ pin functions as an input or an output pin for $\overline{\text{TEND0}}$ by setting CKA1D to 1 in CNTLA1.

To initiate memory to/from I/O (and memory to/from memory mapped I/O) DMA transfer for channel 0, perform the following operations:

1. Load the memory and I/O or memory mapped I/O source and destination addresses into SAR0 and DAR0.

   I/O addresses (not memory mapped I/O are limited to 16 bits (A0–A15). Make sure that bits A16, A17 and A19 are 0 (A18 is a don't care) to correctly enable the external $\overline{\text{DREQ0}}$ input.

2. Specify memory to/from I/O or memory to/from memory mapped I/O mode and address increment/decrement in the SM0, SM1, DM0 and DM1 bits of DMODE.

3. Load the number of bytes to transfer in BCR0.

4. Specify whether $\overline{\text{DREQ0}}$ is edge- or level-sense by programming the DMS0 bit of DCNTL.

5. Enable or disable DMA termination interrupt with the DIE0 bit in DSTAT.

6. Program DE0: = 1 (with $\overline{\text{DWEO}}$ = 0 in the same access) in DSTAT and the DMA operation begins under the control of the $\overline{\text{DREQ0}}$ input.

Memory to ASCI - Channel 0

Channel 0 has extra capability to support DMA transfer to/from the on-chip two channel ASCI. In this case, the external $\overline{\text{DREQ0}}$ input is not used for DMA timing. Rather, the ASCI status bits are used to generate an internal $\overline{\text{DREQ0}}$ The TDRE (Transmit Data Register Empty) bit and the RDRF (Receive Data Register Full) bit are used to generate an internal

ASCI Receive Shift Register 0,1(RSR0, 1)

This register receives data shifted in on the RXA pin. When full, data is automatically transferred to the ASCI Receive Data Register (RDR) if it is empty. If RSR is not empty when the next incoming data byte is shifted in, an overrun error occurs. The RSR is not program-accessible.

ASCI Receive Data Register 0,1 (RDR0, 1: I/O Address = 08H, 09H)

When a complete incoming data byte is assembled in RSR, it is automatically transferred to the RDR if RDR is empty. The next incoming data byte can be shifted into RSR while RDR contains the previous received data byte. Thus, the ASCI receiver on Z80180 is double-buffered.

ASCI Receive Data Register Ch. 0 (RDR0: 08H)

Bit	7	6	5	4	3	2	1	0
Bit/Field	ASCI Receive Channel 0							
R/W	R/W							
Reset	0							
Note: R = Read W = Write X = Indeterminate ? = Not Applicable								

ASCI Receive Data Register Ch. 1 (RDR1: 09H)

Bit	7	6	5	4	3	2	1	0
Bit/Field	ASCI Receive Channel 1							
R/W	R/W							
Reset	0							
Note: R = Read W = Write X = Indeterminate ? = Not Applicable								

\

On the Z8S180 and Z8L180-class processors are quadruple buffered. The ASCI Receive Data Register is a read-only register. However, if RDRF =

The error flags (PE, FE, and OVRN bits) are also held at 0. Even after the $\overline{DCD0}$ input goes Low, these bits do not resume normal operation until the status register (STAT0, is read. This first read of (STAT0, while enabling normal operation, still indicates the $\overline{DCD0}$ input is High ($\overline{DCD0}$ bit = 1) even though it has gone Low. Thus, the STAT0 register must be read twice to ensure the $\overline{DCD0}$ bit is reset to 0:

$\overline{RTS0}$: Request to Send 0 (Output)

$\overline{RTS0}$ allows the ASCI to control (start/stop) another communication devices transmission (for example, by connection to that device's $\overline{CTS}$ input). $\overline{RTS0}$ is essentially a 1-bit output port, having no side effects on other ASCI registers or flags.

$\overline{CTS1}$: Clear to Send 1 (Input)

Channel 1 $\overline{CTS1}$ input is multiplexed with Clocked Serial Receive Data (RXS). The $\overline{CTS1}$ function is selected when the $\overline{CTS1E}$ bit in STAT1 is set to 1. When enabled, the $\overline{CTS1}$ operation is equivalent to CTS0,

Modem control signal timing is depicted in Figure 53 and Figure 54.

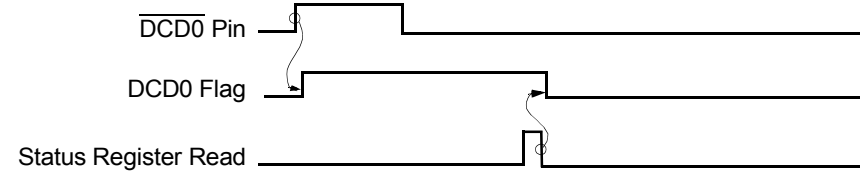

Figure 53. $\overline{DCD0}$ Timing Diagram

2^ss depends on the three least significant bits of the CNTLB register, as described in Table 21.

Table 21. 2^ss Values

ss2	ss1	ss0	2^ss
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	External Clock from CKA0

The ASCIs require a 50% duty cycle when CKA is used as an input. Minimum High and Low times on CKA0 are typical of most CMOS devices.

RDRF is set, and if enabled, an Rx Interrupt or DMA REquest is generated when the receiver transfers a character from the Rx Shift Register to the RX FIFO. The FIFO provides a margin against overruns. When the is more than one character in the FIFO, and software or a DMA channel reads a character, RDRF either remains set or is cleared and then immediately set again. For example, if a receive interrupt service routine does not real all the characters in the RxFIFO, RDRF and the interrupt request remain asserted.

The Rx DMA request is disabled when any of the error flags PE or FE or OVRN are set, so that software can identify with which character the problem is associated.

If Bit 7, RDRF Interrupt Inhibit, is set to 1, the ASCI does not request a Receive interrupt when its RDRF flag is 1. Set this bit when programming a DMA channel to handle the receive data from an ASCI. The other

Timer Data Register 0L (TMDR0L: 0CH)

Bit	7	6	5	4	3	2	1	0
Bit/Field	Timer Data							
R/W	R/W							
Reset	0							
Note: R = Read W = Write X = Indeterminate ? = Not Applicable								

Timer Data Register 0H (TMDR0H: 0DH)

Bit	7	6	5	4	3	2	1	0
Bit/Field	Timer Data							
R/W	R/W							
Reset	0							
Note: R = Read W = Write X = Indeterminate ? = Not Applicable								

Timer Reload Register (RLDR: I/O Address = CH0: 0EH, 0FH, CHI, 16H, 17H)

PRT0 and PRT1 each contain 16-bit Timer Reload Registers (RLDR). RLDR0 and RLDR1 are each accessed as low and high byte registers (RLDR0H, RLDR0L and RLDR1H, RLDR1L). During RESET, RLDR0 and RLDR1 are set to FFFFH

When the TMDR counts down to 0, it is automatically reloaded with the contents of RLDR.

Flag Registers (F, F')

The flag registers store status bits (described in the next section) resulting from executed instructions.

General Purpose Registers (BC, BC', DE, DE', HL, HL')

The General Purpose Registers are used for both address and data operation. Depending on the instruction, each half (8 bits) of these registers (B, C, D, E, H, and I) may also be used.

Interrupt Vector Register (I)

For interrupts that require a vector table address to be calculated ($\overline{\text{INT0}}$ Mode 2, $\overline{\text{INT1}}$, $\overline{\text{INT2}}$, and internal interrupts), the Interrupt Vector Register (I) provides the most significant byte of the vector table address. I is cleared to 00H during reset.

R Counter (R)

The least significant seven bits of the R counter (R) count the number of instructions executed by the Z80180. R increments for each CPU Op Code fetch cycle (each $\overline{\text{M1}}$ cycle). R is cleared to 00H during reset.

Index Registers (IX, and IY)

The Index Registers are used for both address and data operations. For addressing, the contents of a displacement specified in the instruction are added to or subtracted from the Index Register to determine an effective operand address.

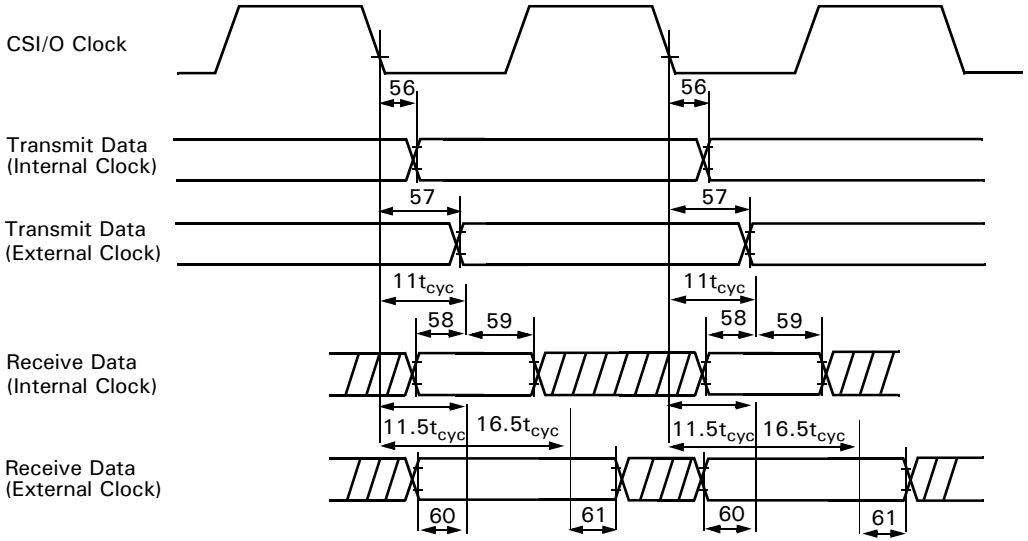

Figure 90. CSI/O Receive/Transmit Timing Diagram

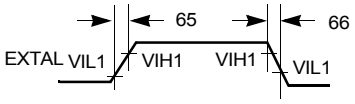

Figure 91. External Clock Rise Time and Fall Time

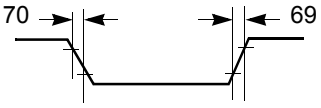

Figure 92. Input Rise Time and Fall Time (Except EXTAL, $\overline{\text{RESET}}$)

Table 38. Arithmetic and Logical Instructions (8-bit) (Continued)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flags					
			Immed	Ext	Ind	Reg	Regl	Imp	Rel				7	6	4	2	1	0
													S	Z	H	P/V	N	C
OR	OR g	10 110 g				S		D		1	4	Ar + gr→Ar	↑	↑	R	P	R	R
	OR (HL)	10 110 110					S	D		1	6	Ar + $(HL)_M$→Ar	↑	↑	R	P	R	R
	OR m	11 110 110	S					D		2	6	Ar + m→Ar	↑	↑	R	P	R	R
		<m>																
	OR (IX + d)	11 011 101			S			D		3	14	Ar + $(IX + d)_M$→Ar	↑	↑	R	P	R	R
		10 110 110																
		<d>																
	OR (IY + d)	11 111 101			S			D		3	14	Ar + $(IY + d)_M$→Ar	↑	↑	R	P	R	R
		10 110 110																
		<d>																
SUB	SUB g	10 010 g				S		D		1	4	Ar-gr→Ar	↑	↑	↑	V	S	↑
	SUB (HL)	10 010 110					S	D		1	6	Ar-$(HL)_M$→Ar	↑	↑	↑	V	S	↑
	SUB m	11 010 110	S					D		2	6	Ar-m→Ar	↑	↑	↑	V	S	↑
		<m>																
	SUB (IX + d)	11 011 101			S			D		3	14	Ar-$(IX + d)_M$-c→Ar	↑	↑	↑	V	S	↑
		10 011 110																
		<d>																
	SUB (IY + d)	11 111 101			S			D		3	14	Ar-$(IY + d)_M$-c→Ar	↑	↑	↑	V	S	↑
		10 010 110																
		<d>																

Table 40. Arithmetic Instructions (16-bit)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flags					
			Immed	Ext	Ind	Reg	Regl	Imp	Rel				7	6	4	2	1	0
													S	Z	H	P/V	N	C
ADD	ADD HL,ww	00 ww1 001				S		D		1	7	$HL_R + ww_R \rightarrow HL_R$	•	•	X	•	R	↑
	ADD IX,xx	11 011 101 00 xx1 001				S		D		2	10	$IX_R + xx_R \rightarrow {}^*IX_R$	•	•	X	•	R	↑
	ADD IY,yy	11 111 101 00 yy1 001				S		D		2	10	$IY_R + yy_R \rightarrow IY_R$	•	•	X	•	R	↑
ADC	ADC HL,ww	11 101 101 01 ww1 010				S		D		2	10	$HL_R + ww_R + c \rightarrow HL_R$	↑	↑	X	V	R	↑
DEC	DEC ww	00 ww1 011				S/D				1	4	$ww_R - 1 \rightarrow \bullet ww_R$	•	•	•	•	•	•
	DEC IX	11 011 101 00 101 011						S/D		2	7	$1X_R - 1 \rightarrow IX_R$	•	•	•	•	•	•
	DEC IY	11 111 101 00 101 011						S/D		2	7	$1Y_R - 1 \rightarrow IY_R$	•	•	•	•	•	•
INC	INC ww	00 ww 0011				S/D				1	4	$ww_R + 1 \rightarrow ww_R$	•	•	•	•	•	•
	INC IX	11 011 101 00 100 011						S/D		2	7	$1X_R + 1 \rightarrow IX_R$	•	•	•	•	•	•
	INC IY	11 111 101 00 100 011						S/D		2	7	$1Y_R + 1 \rightarrow IY_R$	•	•	•	•	•	•
SBC	SBC HL ww	11 101 101 01 ww0 010				S		D		2	10	$HL_R - ww_R - c \rightarrow HL_R$	↑	↑	X	V	S	↑

MNEMONICS	Bytes	Machine Cycles	States
LD g,g'	1	2	4
LD SP,HL	1	2	4
LD SP,IX	2	3	7
LD SP,IY	2	3	7
MLT ww"	2	13	17
NEG	2	2	6
NOP	1	1	3
OR (HL)	1	2	6
OR (IX+d)	3	6	14
OR (IY+d)	3	6	14
OR m	2	2	6
OR g	1	2	4
OTDM**	2	6	14
OTDMR**	2	8	16 (If Br ≠ 0)
	2	6	14 (If Br = 0)
OTDR	2	6	14 (If Br ≠ 0)
	2	4	12 (If Br = 0
OTIM**	2	6	14
OTIMR**	2	8	16 (If Br ≠ 0)
	2	6	14 (If Br = 0)
OTIR	2	6	14 (If Br ≠ 0)
	2	4	12 (If Br = 0)
OUTD	2	4	12
OUTI	2	4	12
OUT (m),A	2	4	10
OUT (C),g	2	4	10
OUT0 (m),g **	3	5	13
POP IX	2	4	12
POP IY	2	4	12
POP zz	1	3	9

Table 56. Pin Status During RESET and LOW POWER OPERATION Modes

Symbol	Pin Function	Pin Status in Each Operation Mode			
		RESET	SLEEP	IOSTOP	SYSTEM STOP
$\overline{WAIT}$	—	IN (N)	IN (N)	IN (A)	IN (N)
$\overline{BUSACK}$	—	1	OUT	OUT	OUT
$\overline{BUSREQ}$	—	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{RESET}$	—	0	IN (A)	IN (A)	IN (A)
$\overline{NMI}$	—	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{INT}_0$	—	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{INT}_1$	—	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{INT}_2$	—	IN (N)	IN (A)	IN (A)	IN (A)
ST	—	1	1	OUT	1
A0–A17, A19	—	Z	1	A	1
A18/TOUT	A18	Z	1	A	1
	TOUT	Z	OUT	H	H
D0–D7	—	Z	Z	A	Z
$\overline{RTS}$0	—	1	H	OUT	H
$\overline{CTS}$0	—	IN (N)	IN (A)	IN (N)	N (N)
$\overline{DCD}$0	—	IN (N)	IN (A)	IN (N)	IN (N)
TXA0	—	1	OUT	H	H
RXA0	—	IN (N)	IN (A)	IN (N)	IN (N)
CKA0/$\overline{DREQ}$0	CKA0 (Internal Clock Mode)	Z	OUT	Z	Z

Table 57. Internal I/O Registers (Continued)

Register	Mnemonics	Address		Remarks
ASCI Status Channel 0:	STAT0	0	4	

bit	RDRF	OVRN	PE	FE	RIE	$\overline{DCD0}$	TDRE	TIE
during RESET	0	0	0	0	invalid	*	**	0
R/W	R	R	R	R	R/W	R	R	R/W

Transmit Interrupt Enable
Transmit Data Register Empty
Data Carrier Detect
Receive Interrupt Enable
Framing Error
Parity Error
Overrun Error
Receive Data Register Full

** $\overline{CTS_0}$ Pin	TDRE
L	1
H	0

* $\overline{DCD_0}$: Depending on the condition of $\overline{DCD_0}$ Pin.

ASCI Status Channel 1:	STAT1	0	5

bit	RDRF	OVRN	PE	FE	RIE	CTS1E	TDRE	TIE
during RESET	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R/W	R	R	R/W

Transmit Interrupt Enable
Transmit Data Register Empty
$\overline{CTS1}$ Enable
Receive Interrupt Enable
Framing Error
Parity Error
Overrun Error
Receive Data Register Full