Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	SH-2
Core Size	32-Bit Single-Core
Speed	125MHz
Connectivity	EBI/EMI, Ethernet, FIFO, SCI, SIO
Peripherals	DMA, POR, WDT
Number of I/O	78
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 1.89V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	176-UFBGA
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/ds76191w125bgv

Bit	Bit name	Default	Read/ Write	Description
0	T	Undefined	R/W	T Indicates true (1) or false (0) in the following instructions: MOVT, CMP/cond, TAS, TST, BT (BT/S), BF (BF/S), SETT, CLRT Indicates carry, borrow, overflow, or underflow in the following instructions: ADDV, ADDC, SUBV, SUBC, NEGC, DIV0U, DIV0S, DIV1, SHAR, SHAL, SHLR, SHLL, ROTR, ROTL, ROTCR, ROTCL

- Global-base register (GBR)

  This register indicates a base address in GBR indirect addressing mode. The GBR indirect addressing mode is used for data transfer of the on-chip peripheral module registers and logic operations.
- Vector-base register (VBR)

  This register indicates the base address of the exception handling vector table.

2.2.3 System Registers

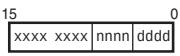
There are four 32-bit system registers, designated two multiply and accumulate registers (MACH and MACL), a procedure register (PR), and program counter (PC).

- Multiply and accumulate registers (MAC)

  This register stores the results of multiplication and multiply-and-accumulate operation.
- Procedure register (PR)

  This register stores the return-destination address from subroutine procedures.
- Program counter (PC)

  The PC indicates the point which is four bytes (two instructions) after the current execution instruction.

Instruction Format	Source Operand	Destination Operand	Sample Instruction
nm type 	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register direct	nnnn: register indirect	MOV.L Rm,@Rn
	mmmm: post-increment register indirect (multiply-and-accumulate operation) nnnn: * post-increment register indirect (multiply-and-accumulate operation)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm+,Rn
	mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@-Rn
	mmmm: register direct	nnnn: index register indirect	MOV.L Rm,@(R0,Rn)
md type 	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 type 	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd type 	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn

Power-On Reset by WDT: When TCNT of the WDT overflows while a setting is made so that a power-on reset can be generated in watchdog timer mode of the WDT, this LSI enters the power-on reset state.

If a reset caused by the signal input on the $\overline{\text{RES}}$ pin and a reset caused by a WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the WOVF bit in RSTCSR is cleared to 0. When the power-on reset exception handling caused by the WDT is started, the CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in the PC and SP, then the program starts.

5.2.3 H-UDI Reset

The H-UDI reset is generated by issuing the H-UDI reset assert command. The CPU operation is described below. For details, see section 21, User Debugging Interface (H-UDI).

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3 to I0) in the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in PC and SP, then the program starts.

6.2 Input/Output Pins

Table 6.1 shows the INTC pin configuration.

Table 6.1 Pin Configuration

Name	Abbr.	I/O	Function
Non-maskable interrupt input pin	NMI	Input	Input of non-maskable interrupt request signal
Interrupt request input pins	IRQ0 to IRQ7	Input	Input of maskable interrupt request signals

6.3 Register Descriptions

The interrupt controller has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Interrupt control register 0 (ICR0)
- IRQ control register (IRQCR)
- IRQ status register (IRQSR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)

Setting					Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]			BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	01 (12 bits)	01 (9 bits)			10 (16 bits)	01 (12 bits)	10 (10 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function	Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
A11	A20	L/H*[1]	A10/AP	Specifies address/ precharge	A11	A21	L/H*[1]	A10/AP	Specifies address/ precharge
A10	A19	A10	A9	Address	A10	A20	A10	A9	Address
A9	A18	A9	A8		A9	A19	A9	A8	
A8	A17	A8	A7		A8	A18	A8	A7	
A7	A16	A7	A6		A7	A17	A7	A6	
A6	A15	A6	A5		A6	A16	A6	A5	
A5	A14	A5	A4		A5	A15	A5	A4	
A4	A13	A4	A3		A4	A14	A4	A3	
A3	A12	A3	A2		A3	A13	A3	A2	
A2	A11	A2	A1		A2	A12	A2	A1	
A1	A10	A1	A0		A1	A11	A1	A0	
A0	A9	A0		Unused	A0	A10	A0		Unused
Example of memory connection					Example of memory connection				
One 128-Mbit product (2 Mwords x 16 bits x 4 banks, 9-bit column product)					One 256-Mbit product (4 Mwords x 16 bits x 4 banks, 10-bit column product)				

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification

Table 7.18 shows the relationship between the access size and the number of bursts.

Table 7.18 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bytes	4

Figures 7.14 and 7.15 show timing charts in burst read. In burst read, the ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is latched at the rising edge of the external clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READ command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, other banks can be accessed. The number of Tap cycles is specified by bits WTRP1 and WTRP0 in CS3WCR.

In this LSI, wait cycles can be inserted by specifying bits in CSnWCR to connect the SDRAM with variable frequencies. Figure 7.15 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READA command is output can be specified using bits WTRCD1 and WTRCD0 in CS3WCR. When bits WTRCD1 and WTRCD0 is set to one cycle or more, a Trw cycle where the NOP command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles from the Tc1 cycle where the READA command is output to the Td1 cycle where the read data is latched can be specified by bits A3CL1 and A3CL0 bits in CS3WCR in CS3WCR. This number of cycles corresponds to the synchronous DRAM CAS latency. The CAS latency for the synchronous DRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as one to four cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the synchronous DRAM.

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock (Pϕ). The overflow period that is shown inside the parenthesis in the table is the value when the peripheral clock (Pϕ) is 25 MHz.
0	CKS0	0	R/W	000: Pϕ (10 µs) 001: Pϕ /4 (41 µs) 010: Pϕ /16 (164 µs) 011: Pϕ /32 (328 µs) 100: Pϕ /64 (655 µs) 101: Pϕ /256 (2.62 ms) 110: Pϕ /1024 (10.49 ms) 111: Pϕ /4096 (41.94 ms) Note: If bits CKS2 to CKS0 are modified when the WDT is operating, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not operating.

9.2.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers is given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction. When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 9.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

8. If counter i reaches or exceeds n, the maximum specified time has elapsed and we can judge that the E-DMAC has stopped due to a transmit underflow. Initialize the EtherC and E-DMAC modules by setting the software-reset bit SWR in the E-DMAC mode register (EDMR). After re-making initial settings for the Ethernet module, initialize the transmit/receive descriptors and transmit/receive buffers.

Bit	Bit Name	Initial value	R/W	Description
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	Select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral operating clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS1 and CKS0. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

Note: * Only 0 can be written, to clear the flag.

14.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS1 and CKS0 in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock.

When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by a power-on reset and a transition to standby mode.

14.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset and is initialized to H'FFFF in standby mode.

14.5 Usage Notes

14.5.1 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 14.5 shows the timing to clear the CMCNT counter.

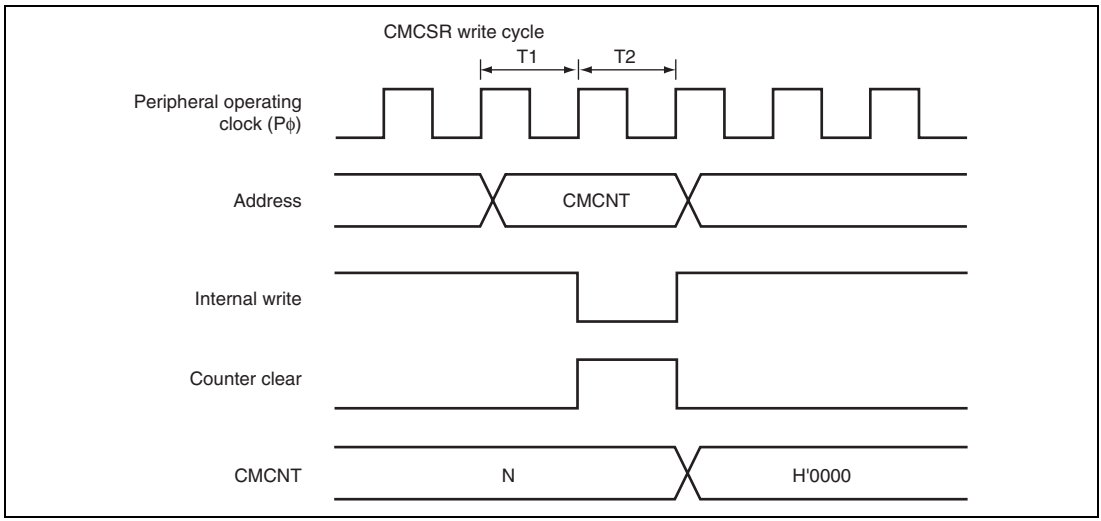


Figure 14.5 Conflict between Write and Compare-Match Processes of CMCNT

15.3.6 Serial Control Register (SCSCR)

SCSCR is a 16-bit register that operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI). Serial transmit data in the transmit FIFO data register (SCFTDR) is send to the transmit shift register (SCTSR). Then, the TDFE flag in the serial status register (SCFSR) is set to1 when the number of data in SCFTDR becomes less than the number of transmission triggers. At this time, a TXI is requested. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled* 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled Note: * The TXI interrupt request can be cleared by writing a greater number of transmit data than the specified transmission trigger number to SCFTDR and by clearing the TDFE bit to 0 after reading 1 from the TDFE bit, or can be cleared by clearing this bit to 0.

15.6 Serial Port Register (SCSPTR) and SCIF Pins

The relationship between SCSPTR and the SCIF pins is shown in figures 15.19 to 15.23.

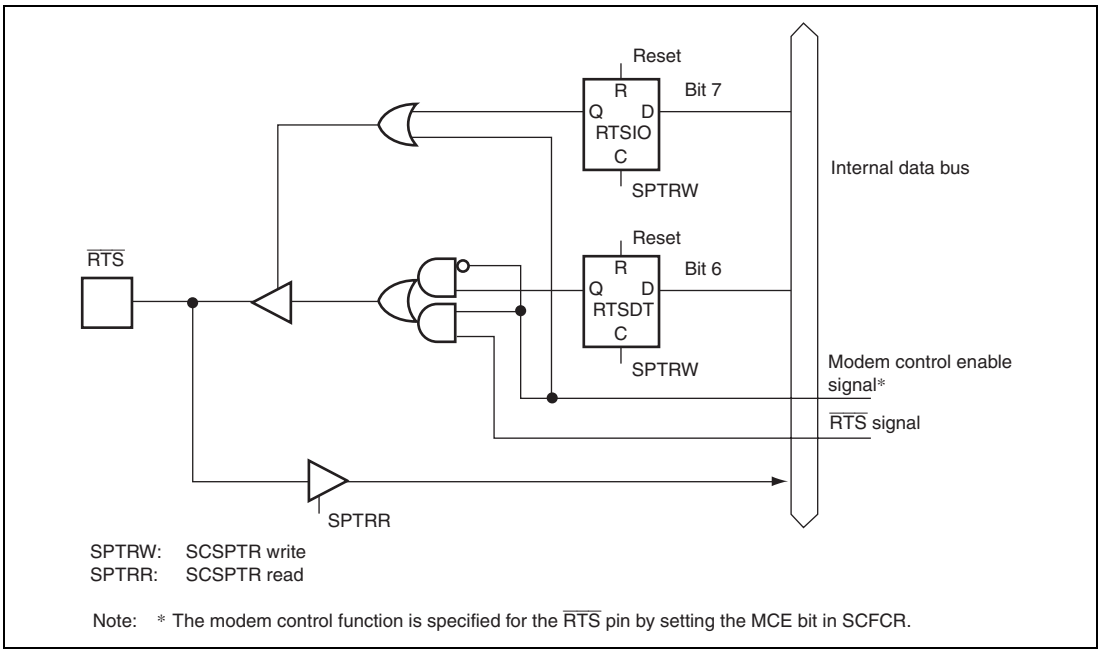


Figure 15.19 RTSIO Bit, RTSDT Bit, and $\overline{\text{RTS}}$ Pin

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit/receive data, slot position for control data, and FIFO request threshold value	
2	Set the SCKE bit to 1 in SICTR	Set operation start for baud rate generator	Note: Serial clock will not be output from the pin until communication is actually started.
3	Set the FSE bit in SICTR to 1	Set the start for frame synchronous signal output	
4	Set the TXE, and RXE bit in SICTR to 1	Enable transmission and reception	Note: Communication is actually started after SITDR has been written.
5	TDREQ = 1? No / Yes		
6	Set SITDR register	Set transmit data	
7	Transmit SITDR from SIOFTXD synchronously with SIOFSYNC		Transmission
8	Transfer complete? No / Yes	* Please check the TFEMP bit in SISTR (the transmit FIFO is empty?) and build a waiting loop to check the communication finished.	
9	Clear the TXE bit in SICTR to 0	Disable transmission	End of transmission
10	Clear the FSE bit in SICTR to 0	To be prepared for the transmission/ reception that is resumed later, set the FSE bit to '0' to synchronize the frame in this LSI.	
11	Set the MSSEL bit in SISCR to 1 Set the BPRS to 00000 and the BRDV to 111 in SISCR Apply a pulse to bits TxRST and RxRST in the SICTR (input 0→1→0) Set the SISCR register to set the baud rate and the master clock source again	To be prepared for the transmission/ reception that is resumed later, initialize inside the baud rate generator.	
12	Change communication mode ? No / Yes END	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
13	With FSE = 0, TXE = 0, and RXE = 0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

Note: * When interrupts due to transmit data underflow are enabled, after setting the no. 6 transmit data, the TXE bit should be set to 1.

Figure 16.9 (2) Transmission Operation in Master Mode (Example of Half-Duplex Transmission by the CPU with TDMAE=0)

16.4.8 Interrupts

The SIOF has one type of interrupt.

Interrupt Sources: Interrupts can be issued by several sources. Each source is shown as an SIOF status in SISTR. Table 16.12 lists the SIOF interrupt sources.

Table 16.12 SIOF Interrupt Sources

No.	Classification	Bit Name	Function Name	Description
1	Transmission	TDREQ	Transmit FIFO transfer request	The transmit FIFO stores data of specified size or more.
2		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
3	Reception	RDREQ	Receive FIFO transfer request	The receive FIFO stores data of specified size or more.
4		RFFUL	Receive FIFO full	The receive FIFO is full.
5	Control	TCRDY	Transmit control data ready	The transmit control register is ready to be written.
6		RCRDY	Receive control data ready	The receive control data register stores valid data.
7	Error	TFUDF	Transmit FIFO underflow	Serial data transmit timing has arrived while the transmit FIFO is empty.
8		TFOVF	Transmit FIFO overflow	Write to the transmit FIFO is performed while the transmit FIFO is full.
9		RFOVF	Receive FIFO overflow	Serial data is received while the receive FIFO is full.
10		RFUDF	Receive FIFO underflow	The receive FIFO is read while the receive FIFO is empty.
11		FSERR	FS error	A synchronous signal is input before the specified bit number has been passed (in slave mode).
12		SAERR	Assign error	The same slot is specified in both serial data and control data.

Whether an interrupt is issued or not as the result of an interrupt source is determined by the SIIER settings. If an interrupt source is set to 1 and the corresponding bit in SIIER is set to 1, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
7	EIC6	0	R/W	External Interrupt Source
6	EIC5	0	R/W	These bits specify the source for interrupts generated by the EIR bit. These bits can be written to from both an external device and the on-chip CPU. By using these bits, fast execution of interrupt exception handling is possible. These bits are completely under software control, and their values have no effect on the operation of this LSI.
5	EIC4	0	R/W	
4	EIC3	0	R/W	
3	EIC2	0	R/W	
2	EIC1	0	R/W	
1	EIC0	0	R/W	
0	EIR	0	R/W	External Interrupt Request While this bit is 1, the $\overline{\text{HIFINT}}$ pin is asserted to issue an interrupt request to an external device from this LSI.

17.4.7 HIF Address Register (HIFADR)

HIFADR is a 32-bit register which indicates the address in HIFRAM to be accessed by an external device. When using the LOCK bit setting in HIFMCR to specify consecutive access of HIFRAM, auto-increment (+4) or auto-decrement (-4) of the address, according to the AI/AD bit setting in HIFMCR, is performed automatically, and HIFADR is updated. HIFADR can be only read by the on-chip CPU. Access to HIFADR by an external device should be performed with HIFADR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	A9 to A2	All 0	R/W*	HIFRAM Address Specification These bits specify the address of HIFRAM to be accessed by an external device, with 32-bit boundary.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit can be only written to by an external device when the HIFRS pin is low. It cannot be written to by the on-chip CPU.

Bit	Bit Name	Initial Value	R/W	Description
13	SCMFDA	0	R/W	I Bus Cycle Condition Match Flag A When the I bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel A does not match 1: The I bus cycle condition for channel A matches
12	SCMFDB	0	R/W	I Bus Cycle Condition Match Flag B When the I bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel B does not match 1: The I bus cycle condition for channel B matches
11	PCTE	0	R/W	PC Trace Enable 0: Disables PC trace 1: Enables PC trace
10	PCBA	0	R/W	PC Break Select A Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution. 0: PC break of channel A is set before instruction execution 1: PC break of channel A is set after instruction execution
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is included in the break condition of channel B. 0: No data bus condition is included in the condition of channel B 1: The data bus condition is included in the condition of channel B

Table 21.3 External pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name	I/O
	from TDI		303	PE02/HIFDREQ/RXD_SIO0/-	IN
332	PD06/IRQ6/RxD2/DACK1	IN	302	PE01/HIFRDY/SIOMCLK0/-	IN
331	PD05/IRQ5/TxD2/DREQ1	IN	301	PE00/HIFEBL/SCK_SIO0/-	IN
330	PD04/IRQ4/SCK1/-	IN	300	PC17/MDC/-/-	IN
329	PD03/IRQ3/RxD1/DACK0	IN	299	PC16/MDIO/-/-	IN
328	PD02/IRQ2/TxD1/DREQ0	IN	298	PC15/CRS/-/-	IN
327	PD01/IRQ1/-/TEND1	IN	297	PC18/LNKSTA	IN
326	PD00/IRQ0/-/TEND0	IN	296	PD06/IRQ6/RxD2/DACK1	OUT
325	PE08/$\overline{\text{HIFCS}}$	IN	295	PD05/IRQ5/TxD2/DREQ1	OUT
324	PE24/HIFD15/CTS1/D31	IN	294	PD04/IRQ4/SCK1/-	OUT
323	PE23/HIFD14/RTS1/D30	IN	293	PD03/IRQ3/RxD1/DACK0	OUT
322	PE22/HIFD13/CTS0/D29	IN	292	PD02/IRQ2/TxD1/DREQ0	OUT
321	PE21/HIFD12/RTS0/D28	IN	291	PD01/IRQ1/-/TEND1	OUT
320	PE20/HIFD11/SCK1/D27	IN	290	PD00/IRQ0/-/TEND0	OUT
319	PE19/HIFD10/RxD1/D26	IN	289	PE08/$\overline{\text{HIFCS}}$	OUT
318	PE18/HIFD09/TxD1/D25	IN	288	PE24/HIFD15/CTS1/D31	OUT
317	PE17/HIFD08/SCK0/D24	IN	287	PE23/HIFD14/RTS1/D30	OUT
316	PE16/HIFD07/RxD0/D23	IN	286	PE22/HIFD13/CTS0/D29	OUT
315	PE15/HIFD06/TxD0/D22	IN	285	PE21/HIFD12/RTS0/D28	OUT
314	PE14/HIFD05/-/D21	IN	284	PE20/HIFD11/SCK1/D27	OUT
313	PE13/HIFD04/-/D20	IN	283	PE19/HIFD10/RxD1/D26	OUT
312	PE12/HIFD03/-/D19	IN	282	PE18/HIFD09/TxD1/D25	OUT
311	PE11/HIFD02/-/D18	IN	281	PE17/HIFD08/SCK0/D24	OUT
310	PE10/HIFD01/-/D17	IN	280	PE16/HIFD07/RxD0/D23	OUT
309	PE09/HIFD00/-/D16	IN	279	PE15/HIFD06/TxD0/D22	OUT
308	PE07/HIFRS	IN	278	PE14/HIFD05/-/D21	OUT
307	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0/-	IN	277	PE13/HIFD04/-/D20	OUT
306	PE05/$\overline{\text{HIFRD}}$	IN	276	PE12/HIFD03/-/D19	OUT
305	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0/-	IN	275	PE11/HIFD02/-/D18	OUT
304	PE03/HIFMD	IN	274	PE10/HIFD01/-/D17	OUT

Data loaded into the output pin boundary scan register in the Capture-DR state is not used for external circuit testing (it is replaced by a shift operation).

The upper four bits of the instruction code are 0000.

IDCODE: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the IDCODE mode stipulated by JTAG. When the H-UDI is initialized ($\overline{\text{TRST}}$ is asserted or TAP is in the Test-Logic-Reset state), the IDCODE mode is entered.

CLAMP, HIGHZ: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the CLAMP or HIGHZ mode stipulated by JTAG.

21.5.2 Points for Attention

- Boundary scan mode does not cover clock-related system signals (EXTAL, XTAL, CKIO, and CK_PHY), E10A-related signals ($\overline{\text{RES}}$ and $\overline{\text{ASEMD}}$), and H-UDI-related signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$).
- When the EXTEST, CLAMP, and HIGHZ commands are set, fix the $\overline{\text{RES}}$ pin low.
- When a boundary scan test for other than BYPASS and IDCODE is carried out, fix the $\overline{\text{ASEMD}}$ pin high.

21.6 Usage Notes

- An H-UDI command, once set, will not be modified as long as another command is not re-issued from the H-UDI. If the same command is given continuously, the command must be set after a command (BYPASS, etc.) that does not affect LSI operations is once set.
- Because LSI operations are suspended in standby mode, H-UDI commands are not accepted. To hold the state of the TAP before and after standby mode, the TCK signal must be high during standby mode transition.
- The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

(2) Auto-negotiation

The purpose of the Auto-negotiation function is to automatically configure the PHY to the optimum link parameters based on the capabilities of its link partner. Auto-negotiation is a mechanism for exchanging configuration information between two link-partners and automatically selecting the highest performance mode of operation supported by both sides. Auto-negotiation is fully defined in clause 28 of the IEEE 802.3 specification.

Once auto-negotiation has completed, information about the resolved link can be passed back to the controller (EtherC) via the Serial Management Interface (SMI). The results of the negotiation process are reflected in the Link Partner Ability Register (Register 5).

The auto-negotiation protocol is a purely physical layer activity and proceeds independently of the MAC controller (EtherC).

The advertised capabilities of the PHY are stored in register 4 of the SMI registers. The default advertised by the core PHY is determined by user-defined on-chip signal options. (i.e. the configuration of PHY-IF)

The following blocks are activated during an Auto-negotiation session:

- Auto-negotiation (digital)
- 100M ADC (analog)
- 100M PLL (analog)
- 100M equalizer/BLW/clock recovery (DSP)
- 10M SQUELCH (analog)
- 10M PLL (analog)
- 10M Transmitter (analog)

When enabled, auto-negotiation is started by the occurrence of one of the following events:

- Module reset (co_resetb of PHY-IF)
- PHY power on reset
- Software reset
- Power-down reset
- Link status down
- Setting register 0, bit 9 high (auto-negotiation restart)

Appendix

A. Port States in Each Pin State

Table A.1 Port States in Each Pin State

		Reset State		Power-Down Mode		
Classification	**Abbr.**	**Power-On (HIFMD = Low)**	**Power-On (HIFMD = High)**	**Software Standby**	**Sleep**	**H-UDI Module Standby**
Clock	EXTAL	I	I	I	I	I
	XTAL	O*[1]	O*[1]	O*[1]	O*[1]	O*[1]
	CKIO	O*[1]	O*[1]	ZO*[5]	O*[1]	O*[1]
	CK_PHY	I	I	I	I	I
System control	$\overline{\text{RES}}$	I	I	I	I	I
Operating mode control	MD5, MD3 to MD0	I	I	I	I	I
Interrupt	NMI	I	I	I	I	I
	IRQ7 to IRQ0	—	—	I	I	I
Address bus	A25 to A16	—	—	ZHL*[4]	O	O
	A15 to A0	O	O	ZHL*[4]	O	O
Data bus	D31 to D16	—	—	Z	IO	IO
	D15 to D0	Z	Z	Z	IO	IO
Bus control	$\overline{\text{WAIT}}$	—	—	Z	I	I
	$\overline{\text{IOIS16}}$	—	—	Z	I	I
	CKE	—	—	ZO*[2]	O	O
	$\overline{\text{CAS}}$, $\overline{\text{RAS}}$	—	—	ZO*[2]	O	O
	$\overline{\text{WE0}}$/DQMLL	H	H	ZH*[4]	O	O
	$\overline{\text{WE1}}$/DQMLU/ $\overline{\text{WE}}$	H	H	ZH*[4]	O	O
	$\overline{\text{WE2}}$/DQMUL/ $\overline{\text{ICIORD}}$	—	—	ZH*[4]	O	O