12.0 Intel® 6 Series Express Chipset MAC Programming Interface

12.0.1 Registers Byte Ordering

This section defines the structure of registers that contain fields carried over the network. Some examples are L2, L3, L4 fields and MACsec fields.

The following example is used to describe byte ordering over the wire (hex notation):

Last First

...,06, 05, 04, 03, 02, 01, 00

where each byte is sent with the LSbit first. That is, the bit order over the wire for this example is

Last First

..., 0000 0011, 0000 0010, 0000 0001, 0000 0000

The general rule for register ordering is to use Host Ordering (also called little endian). Using the above example, a 6-byte fields (e.g. MAC address) is stored in a CSR in the following manner:

	Byte 3	Byte 2	Byte 1	Byte0
DW address (N)	0x03	0x02	0x01	0x00
DW address (N+4)	...	...	0x05	0x04

The exceptions listed below use network ordering (also called big endian). Using the above example, a 16-bit field (e.g. EtherType) is stored in a CSR in the following manner:

	Byte 3	Byte 2	Byte 1	Byte0
(DW aligned)	...	...	0x00	0x01
or				
(Word aligned)	0x00	0x01	...	...

The following exceptions use network ordering:

- All ETherType fields

Note: The "normal" notation as it appears in text books, etc. is to use network ordering. Example: Suppose a MAC address of 00-A0-C9-00-00-00. The order on the network is 00, then A0, then C9, etc. However, the host ordering presentation would be

Byte 3	Byte 2	Byte 1	Byte0

DW address (N)	00	C9	A0	00
DW address (N+4)	...	...	00	00

12.0.2 Register Conventions

All registers in the LAN Controller are defined to be 32 bits, so write cycles should be accessed as 32 bit double-words, There are some exceptions to this rule:

- Register pairs where two 32 bit registers make up a larger logical size

Reserved bit positions: Some registers contain certain bits that are marked as "reserved". These bits should never be set to a value of "one" by software. Reads from registers containing reserved bits may return indeterminate values in the reserved bit-positions unless read values are explicitly stated. When read, these reserved bits should be ignored by software.

Reserved and/or undefined addresses: any register address not explicitly declared in this specification should be considered to be reserved, and should not be written to. Writing to reserved or undefined register addresses may cause indeterminate behavior. Reads from reserved or undefined configuration register addresses may return indeterminate values unless read values are explicitly stated for specific addresses. Reserved fields within defined registers are defined as Read-Only (RO). When writing to these registers the RO fields should be set to their init value. Reading from reserved fields may return indeterminate values.

Initial values: most registers define the initial hardware values prior to being programmed. In some cases, hardware initial values are undefined and will be listed as such via the text "undefined", "unknown", or "X". Some such configuration values may need to be set via NVM configuration or via software in order for proper operation to occur; this need is dependent on the function of the bit. Other registers may cite a hardware default which is overridden by a higher-precedence operation. Operations which may supersede hardware defaults may include a valid NVM load, completion of a hardware operation (such as hardware auto-negotiation), or writing of a different register whose value is then reflected in another bit.

For registers that should be accessed as 32 bit double words, partial writes (less than a 32 bit double word) will not take effect (i.e. the write is ignored). Partial reads will return all 32 bits of data regardless of the byte enables.

Note: Partial reads to read-on-clear registers (e.g. ICR) can have unexpected results since all 32 bits are actually read regardless of the byte enables. Partial reads should not be done.

Note: All statistics registers are implemented as 32 bit registers. Though some logical statistics registers represent counters in excess of 32-bits in width, registers must be accessed using 32-bit operations (e.g. independent access to each 32-bit field).

Note: The LAN Controller supports a single memory access at a time.

See special notes for Multicast Table Arrays in their specific register definitions.

12.0.3 PCI Configuration and Status Registers - CSR Space

12.0.3.1 PCI Register Map

All configuration registers are listed in the table below. These registers are ordered by grouping and are not necessarily listed in order that they appear in the address space.

Register Based Legend:

RW – Read Write register.

RO – Read Only Register.

RO/CR – Read Only Register, Clear on Read.

RO/V – Read Only Register, Read status is not constant.

RW/RO – Read write by FW, Read only by SW.

R/WC – Read Write Clear registers. Writing '0's has no affect. Writing '1's clears the appropriate fields (see detailed description of the specific registers).

RW/V – Read Write register This bit self-clears immediately.

RW/SN – Read Write register initial value loaded from NVM.

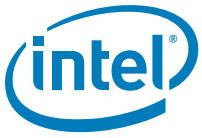
RC/WC – Read Clear/ Write Clear registers. Writing '0's has no affect. Writing '1's clears the appropriate fields. Read may also clear the register depending on enablement (see specific registers).

RWC/CR/V – Read Write register clear on read, clear on write.

WO – Write only registers. Reading from these registers does not reflect any meaningful data. Mostly it would be all zero's (see detailed description of the specific registers).

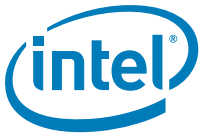
Table 12-7. Register Summary

Offset	Abbreviation	Name	RW
General Register Descriptions			
0x00000	CTRL	Device Control Register	RW
0x00008	STATUS	Device Status Register	RO
0x0000C	STRAP	Strapping Option Register	RO
0x00018	CTRL_EXT	Extended Device Control Register	RW
0x00020	MDIC	MDI Control Register	RW
0x00024	FEXTNVM4	Future Extended4 NVM Register	RW
0x00028	FEXTNVM	Future Extended NVM Register	RW
0x0002C	FEXT	Future Extended Register	RW
0x00030	FEXTNVM2	Future Extended2 NVM Register	RW
0x00034	KUMCTRLSTA	Kumeran control and status registers	RW
0x00038	BUSNUM	Device and Bus Number	RO
0x000F8	LTRV	Latency Tolerance Reporting Value	RW
0x000FC	LPIC	Low Power Idle Control	RW
0x00170	FCTTV	Flow Control Transmit Timer Value	RW
0x05F40	FCRTV	Flow Control Refresh Threshold Value	RW
0x00F00	EXTCNF_CTRL	Extended Configuration Control	RW



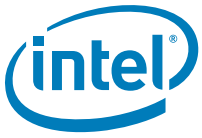
Offset	Abbreviation	Name	RW
0x00F08	EXTCNF_SIZE	Extended Configuration Size	RW
0x00F10	PHY_CTRL	PHY Control Register	RW
0x00F18	PCIEANACFG	PCIE Analog Configuration	RW
0x01000	PBA	Packet Buffer Allocation	RW
0x01008	PBS	Packet Buffer Size	RW
0x05B00	DCR	DMA Control Register	RW
Interrupt Register Descriptions			
0x000C0	ICR	Interrupt Cause Read Register	RC/ WC
0x000C4	ITR	Interrupt Throttling Register	RW
0x000C8	ICS	Interrupt Cause Set Register	WO
0x000D0	IMS	Interrupt Mask Set/Read Register	RW
0x000D8	IMC	Interrupt Mask Clear Register	WO
0x000E0	Mask - IAM	Interrupt Acknowledge Auto	RW
Receive Register Descriptions			
0x00100	RCTL	Receive Control Register	RW
0x00104	RCTL1	Receive Control Register 1	RW
0x02008	ERT	Early Receive Threshold	RW
0x02170 + n*0x4 [n=0..1]	PSRCTL	Packet Split Receive Control Register	RW
0x02160	FCRTL	Flow Control Receive Threshold Low	RW
0x02168	FCRTH	Flow Control Receive Threshold High	RW
0x02800 + n*0x100[n=0..1]	RDBAL	Receive Descriptor Base Address Low queue	RW
0x02804 + n*0x100[n=0..1]	RDBAH	Receive Descriptor Base Address High queue	RW
0x02808 + n*0x100[n=0..1]	RDLEN	Receive Descriptor Length queue	RW
0x02810 + n*0x100[n=0..1]	RDH	Receive Descriptor Head queue	RW
0x02818 + n*0x100[n=0..1]	RDT	Receive Descriptor Tail queue	RW
0x02820 + n*0x100[n=0..1]	RDTR	Interrupt Delay Timer (Packet Timer)	RW
0x02828 + n*0x100[n=0..1]	RXDCTL	Receive Descriptor Control	RW
0x0282C	RADV	Receive Interrupt Absolute Delay Timer	RW
0x02C00	RSRPD	Receive Small Packet Detect Interrupt	RW
0x02C08	RAID	Receive ACK Interrupt Delay Register	RW
0x02C10	CPUVEC	CPU Vector Register	RW
0x05000	RXCSUM	Receive Checksum Control	RW
0x05008	RFCTL	Receive Filter Control Register	RW

Offset	Abbreviation	Name	RW
0x05200-0x0527C	MTA[31:0]	Multicast Table Array	RW
0x05400 + 8*n (n=0…6)	RAL	Receive Address Low	RW
0x05404 + 8*n (n=0…6)	RAH	Receive Address High	RW
0x05438 + 8*n (n=0…3)	SRAL	Shared Receive Address Low	RW
0x0543C + 8*n (n=0…2)	SRAH	Shared Receive Address High 0…2	RW
0x05454	SHRAH[3]	Shared Receive Address High 3	RW
0x05818	MRQC	Multiple Receive Queues Command Register	RW
0x05864	RSSIM	RSS Interrupt Mask Register	RW
0x05868	RSSIR	RSS Interrupt Request Register	RW
0x05C00 + 4*n (n=0…31)	RETA	Redirection Table	RW
0x05C80 + 4*n (n=0…9)	RSSRK	Random Key Register	RW
Transmit Register Descriptions			
0x00400	TCTL	Transmit Control Register	RW
0x00410	TIPG	Transmit IPG Register	RW
0x00458	AIT	Adaptive IFS Throttle	RW
0x03800 + n*0x100[n=0..1]	TDBAL	Transmit Descriptor Base Address Low	RW
0x03804 + n*0x100[n=0..1]	TDBAH	Transmit Descriptor Base Address High	RW
0x03808 + n*0x100[n=0..1]	TDLEN	Transmit Descriptor Length	RW
0x03810 + n*0x100[n=0..1]	TDH	Transmit Descriptor Head	RW
0x03818 + n*0x100[n=0..1]	TDT	Transmit Descriptor Tail	RW
0x03840 + n*0x100[n=0..1]	TARC	Transmit Arbitration Count	RW
0x03820	TIDV	Transmit Interrupt Delay Value	RW
0x03828 + n*0x100[n=0..1]	TXDCTL	Transmit Descriptor Control	RW
0x0382C	TADV	Transmit Absolute Interrupt Delay Value	RW
Statistic Register Descriptions			
0x04000	CRCERRS	CRC Error Count	RO
0x04004	ALGNERRC	Alignment Error Count	RO
0x0400C	RXERRC	RX Error Count	RO
0x04010	MPC	Missed Packets Count	RO



Offset	Abbreviation	Name	RW
0x0403C	CEXTERR	Carrier Extension Error Count	RO
0x04040	RLEC	Receive Length Error Count	RO
0x04048	XONRXC	XON Received Count	RO
0x0404C	XONTXC	XON Transmitted Count	RO
0x04050	XOFFRXC	XOFF Received Count	RO
0x04054	XOFFTXC	XOFF Transmitted Count	RO
0x04058	FCRUC	FC Received Unsupported Count	RO
0x04074	GPRC	Good Packets Received Count	RO
0x04078	BPRC	Broadcast Packets Received Count	RO
0x0407C	MPRC	Multicast Packets Received Count	RO
0x04080	GPTC	Good Packets Transmitted Count	RO
0x04088	GORCL	Good Octets Received Count Low	RO
0x0408C	GORCH	Good Octets Received Count High	RO
0x04090	GOTCL	Good Octets Transmitted Count Low	RO
0x04094	GOTCH	Good Octets Transmitted Count High	RO
0x040A0	RNBC	Receive No Buffers Count	RO
0x040A4	RUC	Receive Undersize Count	RO
0x040A8	RFC	Receive Fragment Count	RO
0x040AC	ROC	Receive Oversize Count	RO
0x040B0	RJC	Receive Jabber Count	RO
0x040B4	MNGPRC	Management Packets Received Count	RO
0x040B8	MNGPDC	Management Packets Dropped Count	RO
0x040BC	MNGPTC	Management Packets Transmitted Count	RO
0x040D8	TCBPD	Tx Circuit Breaker Packets Dropped	RO
0x040C0	TORL	Total Octets Received Low	RO
0x040C4	TORH	Total Octets Received High	RO
0x040C8	TOTL	Total Octets Transmitted	RO
0x040CC	TOTH	Total Octets Transmitted	RO
0x040D0	TPR	Total Packets Received	RO
0x040D4	TPT	Total Packets Transmitted	RO
0x040F0	MPTC	Multicast Packets Transmitted Count	RO
0x040F4	BPTC	Broadcast Packets Transmitted Count	RO
0x040F8	TSCTC	TCP Segmentation Context Transmitted Count	RO
0x04100	IAC	Interrupt Assertion Count	RO
Management Register Descriptions			
0x05800	WUC	Wake Up Control Register	RW
0x05808	WUFC	Wake Up Filter Control Register	RW
0x05810	WUS	Wake Up Status Register	RW

Offset	Abbreviation	Name	RW
0x5838	IPAV	IP Address Valid	RW
0x05840 + 8*n (n=1…3)	IP4AT	IPv4 Address Table	RW
0x05880 + 4*n (n=0…3)	IP6AT	IPv6 Address Table	RW
0x05F00 + 8*n (n=0…7)	FFLT	Flexible Filter Length Table	RW
0x09000 + 8*n (n=0…127)	FFMT	Flexible Filter Mask Table	RW
0x09800 + 8*n (n=0…127)	FFVT	Flexible Filter Value Table	RW
0x09804 + 8*n (n=0…127)	FFVT2	Flexible Filter Value Table	RW
0x0B620	TSYNCRXCTL	RX Time Sync Control register	RW
0x0B624	RXSTMPL	RX timestamp Low	RO
0x0B628	RXSTMPH	RX timestamp High	RO
0x0B62C	RXSATRL	RX timestamp attributes low	RO
0x0B630	RXSATRH	RX timestamp attributes high	RO
0x0B634	RXMTRL	RX message type register low	RW
0x0B638	RXUDP	RX UDP port	RW
0x0B614	TSYNCTXCTL	TX Time Sync Control register	RW
0x0B618	TXSTMPL	TX timestamp value Low	RO
0x0B61C	TXSTMPH	TX timestamp value High	RO
0x0B600	SYSTIML	System time register Low	RO
0x0B604	SYSTIMH	System time register High	RO
0x0B608	TIMINCA	Increment attributes register	RW
0x0B60C	TIMADJL	Time adjustment offset register low	RW
0x0B610	TIMADJH	Time adjustment offset register high	RW
Diagnostic Register Descriptions			
0x02410	RDFH	Receive Data FIFO Head Register	RW
0x02418	RDFT	Receive Data FIFO Tail Register	RW
0x02420	RDFHS	Receive Data FIFO Head Saved Register	RW
0x02428	RDFTS	Receive Data FIFO Tail Saved Register	RW
0x02430	RDFPC	Receive Data FIFO Packet Count	RW
0x03410	TDFH	Transmit Data FIFO Head Register	RW
0x03418	TDFT	Transmit Data FIFO Tail Register	RW
0x03420	TDFHS	Transmit Data FIFO Head Saved Register	RW
0x03428	TDFTS	Transmit Data FIFO Tail Saved Register	RW
0x03430	TDFPC	Transmit Data FIFO Packet Count	RW
0x10000 - 0x15FFC	PBM	Packet Buffer Memory	RW



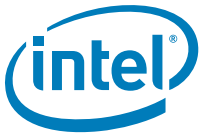
Offset	Abbreviation	Name	RW
0x0C000 - 0x0C3FC	GMD	Ghost Memory Data	RW
0x0C400 - 0x0C5FC	DMD	Descriptor Memory Data	RW
Hidden Testability Register Descriptions			
0x05B60	MNGCCR	MNG CSR Control register	RW
0x05B64	MNGCAR	MNG CSR Address Register	RW
0x05B68	MNGCDO	MNG CSR Data Out Register	RO
0x05B6C	MNGCDI	MNG CSR Data In Register	RW
0x05F04	CHIKN	Debug Chicken Register	RW
MACsec Register Descriptions			
0x0B000	LSECTXCAP	MACsec TX Capabilities register	RW
0x0B300	LSECRXCAP	MACsec RX Capabilities register	RW
0x0B004	LSECTXCTRL	MACsec TX Control register	RW
0x0B304	LSECRXCTRL	MACsec RX Control register	RW
0x0B008	LSECTXSCL	MACsec TX SCI Low	RW
0x0B00C	LSECTXSCH	MACsec TX SCI High	RW
0x0B010	LSECTXSA	MACsec TX SA	RW
0x0B018	LSECTXPN0	MACsec TX SA PN 0	RW
0x0B01C	LSECTXPN1	MACsec TX SA PN 1	RW
0x0B01C	LSECTXKEY0 [n]	MACsec TX Key 0 0 + 4*n (n=0…3) 0x0B02	WO
0x0B01C	LSECTXKEY1 [n]	MACsec TX Key 1 0 + 4*n (n=0…3) 0x0B03	WO
0x0B3D0 + 4*n (n=0…3)	LSECRXSCL[n]	MACsec RX SCI Low	RW
0x0B3E0 + 4*n (n=0…3)	LSECRXSCH[n]	MACsec RX SCI High	RW
0x0B310 + 4*n (n=0…7)	LSECRXSA[n]	MACsec RX SA	RW
0x0B330 + 4*n (n=0…7)	LSECRXSAPN	MACsec RX SA PN	RW
0x0B350 + 0x10*n (n=0…7) + 4*m (m=0..3)	LSECRXKEY[n,m]	MACsec RX Key	WO
0x04300	LSECTXUT	Tx Untagged Packet Counter	RC
0x04304	LSECTXPKTE	Encrypted Tx Packets	RC
0x04308	LSECTXPKTP	Protected Tx Packets	RC
0x0430C	LSECTXOCTE	Encrypted Tx Octets	RC
0x04310	LSECTXOCTP	Protected Tx Octets	RC
0x04314	LSECRXUT	MACsec Untagged RX Packet	RC
0x0431C	LSECRXOCTE	MACsec RX Octets Decrypted	RC

Offset	Abbreviation	Name	RW
0x04320	LSECRXOCTP	MACsec RX Octets Validated	RC
0x04324	LSECRXBAD	MACsec RX Packet with Bad Tag	RC
0x04328	LSECRXNOSCI	MACsec RX Packet No SCI	RC
0x0432C	LSECRXUNSCI	MACsec RX Packet Unknown SCI count	RC
0x04330	LSECRXUNCH	MACsec RX Unchecked Packets	RC
0x04340 + 4*n (n=0…3)	LSECRXDELAY[n]	MACsec RX Delayed Packets	RC
0x04350 + 4*n (n=0…3)	LSECRXLATE[n]	MACsec RX Late Packets	RC
0x04360 + 4*n (n=0…7)	LSECRXOK[n]	MACsec RX Packet OK	RC
0x043A0 + 4*n (n=0…7)	LSECRXINV[n]	MACsec Check RX Invalid	RC
0x04380 + 4*n [n=0...7]	LSECRXNV[n]	MACsec RX Not valid count	RC
0x043C0 + 4*n (n=0…3)	LSECRXNUSA[n]	MACsec RX Not Using SA	RC
0x043D0 + 4*n (n=0…3)	LSECRXUNSA[n]	MACsec RX Unused SA	RC

12.0.3.2 General Register Descriptions

12.0.3.2.1 Device Control Register - CTRL (0x00000; RW)

Bit	Type	Default	Description
0	RW/SN	1	**Full Duplex (FD)**. 0 – half duplex 1 – full duplex. Controls the MAC duplex setting when explicitly setting by software. Loaded from the NVM word 13h.
1	RO	0	**Reserved**. Write as 0 for future compatibility
2	RW	0	**Master Disable**. When set, the LAN Controller blocks new master requests on the PCI device. Once no master requests are pending by this function, the *Master Enable Status* bit is cleared.
5:3	RO	0	**Reserved.** Write as 0 for future compatibility.
6	RO	1	**Reserved.**
7	RO	0	**Reserved.** Must be set to ‘0’.
9:8	RW	10b	**Speed selection (SPEED)**. These bits may determine the speed configuration and are written by software after reading the PHY configuration through the MDIO interface. These signals are ignored when Auto-Speed Detection is enabled. (00)b – 10Mb/s (01)b – 100Mb/s (10)b – 1000Mb/s (11)b – not used



10	RO	0	**Reserved**. Write as 0 for future compatibility.
11	RW/SN	0	**Force Speed (FRCSPD)**. This bit is set when software wants to manually configure the MAC speed settings according to the SPEED bits above. When using a PHY device, note that the PHY device must resolve to the same speed configuration or software must manually set it to the same speed as the MAC. The value is loaded from word 13h in the NVM. Note that this bit is superseded by the CTRL_EXT.SPD_BYPS bit which has a similar function.
12	RW	0	**Force Duplex (FRCDPLX)**. When set to 1, software may override the duplex indication from the PHY that is indicated in the FDX to the MAC. Otherwise, the duplex setting is sampled from the PHY FDX indication into the MAC on the asserting edge of the PHY LINK signal. When asserted, the CTRL.FD bit sets duplex.
13	RO	0	Reserved.
15:14	RO	0	Reserved. Reads as 0.
16	RW	0	**LANPHYPC override** When set to 1 this bit provides the SW driver the ability to control the LANPHYPC pin value.
17	RW	0	**LANPHYPC Value** When LANPHYPC override is set to 1 this bit will define the value of the LANPHYPC pin.
23:18	RO	0x0	Reserved.
24	RW	0	**LCD Power Down (LCDPD)**. When the bit is cleared to '0', the LCD power down setting is controlled by the internal logic of the LAN controller. When set to '1' and the CTRL_EXT.PHYPDEN is set as well, the LAN controller sets the external LCD to power down mode using the LANPHYPC.
26	RW/V	0	**Host Software Reset (SWRST)**. This bit performs a reset to the PCI data path and the relevant shared logic. Writing 1 initiates the reset. This bit is self-clearing.
27	RW	0	**Receive Flow Control Enable (RFCE)**. Indicates the device will respond to the reception of flow control packets. If Auto-Negotiation is enabled, this bit is set to the negotiated duplex value.
28	RW	0	**Transmit Flow Control Enable (TFCE)**. Indicates the device will transmit flow control packets (XON & XOFF frames) based on receiver fullness. If Auto-Negotiation is enabled, this bit is set to the negotiated duplex value.
29	RO	0	**Reserved.**
30	RW	0	**VLAN Mode Enable (VME)**. When set to 1, all packets transmitted from LAN Controller that have VLE set will be sent with an 802.1Q header added to the packet. The contents of the header come from the transmit descriptor and from the VLAN type register. On receive, VLAN information is stripped from 802.1Q packets.

31	RW/V	0	**LAN Connected Device Reset (LCD_RST).** Controls a 0 – normal (operational) 1 – reset to PHY is asserted. The LCD_RST functionality is gated by the FWSM.RSPCIPHY bit. If the FWSM.RSPCIPHY bit is not set to '1', then setting the LCD_RST has no impact. For proper operation Software or Firmware must also set the SWRST bit in the register at the same time. This bit is self-clearing.

Note: Fields loaded from the NVM are set by the NVM only if the signature bits of the NVM's Initialization Control Word match (01)b.

This register, as well as the Extended Device Control register (CTRL_EXT), controls the major operational modes for the device. While software writes to this register to control device settings, several bits (such as FD and SPEED) may be overridden depending on other bit settings and the resultant link configuration is determined by the PHY's Auto-Negotiation resolution.

The FD (duplex) and SPEED configurations of the device are normally determined from the link configuration process. Software may specifically override/set these MAC settings via certain bits in a forced-link scenario; if so, the values used to configure the MAC must be consistent with the PHY settings.

Manual link configuration is controlled through the PHY's MII management interface.

Host Software Reset (bit 26), may be used to globally reset the entire host data path . This register is provided primarily as a last-ditch software mechanism to recover from an indeterminate or suspected hung hardware state. Most registers (receive, transmit, interrupt, statistics, etc.), and state machines will be set to their power-on reset values, approximating the state following a power-on or PCI reset. One internal configuration register, the Packet Buffer Allocation (PBA) register, retains its value through a Software reset.

Note: To ensure that global device reset has fully completed and that the controller will respond to subsequent accesses, one must wait approximately 1 microsecond after setting before attempting to check to see if the bit has cleared or to access (read or write) any other device register.

Note: This register's address is reflected also at address 0x00004 for legacy reasons. Neither the SW driver nor the FW should use it since it may be unsupported in next generations.

12.0.3.2.2 Device Status Register - STATUS (0x00008; RO)

Bits	Attribute	Default	Description
0	RO/V	X	**Full Duplex (FD)**. 0 – half duplex 1 – full duplex Reflects duplex setting of the MAC and/or link.
1	RO/V	X	**Link up (LU)**. 0 – no link established 1 – link established For this to be valid, the *Set Link Up* bit of the *Device Control Register* (CTRL.SU) must be set.

3,2	RO/V	00b	**PHY Type Indication (PHYTYPE).** Indicates the LAN Connected Device attached to LAN controller and the resulting mode of operation of the MAC/LAN Connected Device Link buses. PHYTYPE PHY-Device 00 82579 01 Reserved 10 Reserved. 11 Reserved This field is loaded from the Shared Init control word in the NVM.
4	RO/V	X	**Transmission Paused (TXOFF).** Indication of pause state of the transmit function when symmetrical flow control is enabled.
5	RO/V	1	**PHY Power Up not (PHYPWR).** RO bit that indicates the power state of the PHY. 0 – The PHY is powered on in the active state. 1 – The PHY is in the power down state The PHYPWR bit is valid only after PHY reset is asserted. Note: The PHY Power Up indication reflects the status of the LANPHYPC signaling to the LCD.
7:6	RO/V	X	**Link speed setting (SPEED).** This bit reflects the speed setting of the MAC and/or link. (00)b – 10Mb/s (01)b – 100Mb/s (10)b – 1000Mb/s (11)b – 1000Mb/s
8	RO/V	X	**Master Read Completions Blocked**. This bit is set when the device receives a completion with an error (EP = 1 or status unsuccessful); It is cleared on PCI reset.
9	RW/V/C	0	**LAN Init Done.** This bit is Asserted following completion of the LAN initialization from the FLASH. See "LAN Init Done Event" section for a complete description. Software is expected to clear this field to make it usable for the next Init done event.
10	RW/V/C	1	**PHY Reset Asserted (PHYRA).** This bit is R/W. The HW sets this bit following the assertion of LCD reset (either HW or in-band). The bit is cleared on writing '0' to it.
18:11	RO	0	**Reserved**.
19	RO/V	1	**Master Enable Status**. Cleared by LAN Controller when the *Master Disable* bit is set and no master requests are pending by this function, otherwise this bit is set. This bit indicates that no master requests will be issued by this function as long as the *Master Disable* bit is set.
29:20	RO	0	**Reserved**. Reads as 0.
30	RO	0	**Reserved.**
31	RO/SN	1	**Clock Control ¼ (CLK_CNT_1_4)**. This bit is loaded from the NVM word 13h and indicates the device supports lowering its DMA clock to ¼ of its value.

FD reflects the actual MAC duplex configuration. This normally reflects the duplex setting for the entire link, as it normally reflects the duplex configuration negotiated between the PHY and link partner (copper link) or MAC and link partner (fiber link).

Link up provides a useful indication of whether something is attached to the port. Successful negotiation of features/link parameters results in link activity. The link startup process (and consequently the duration for this activity after reset) may be several 100's of ms. It reflects whether the PHY's LINK indication is present. Refer to section for more details.

TXOFF indicates the state of the transmit function when symmetrical flow control has been enabled and negotiated with the link partner. This bit is set to 1 when transmission is paused due to the reception of an XOFF frame. It is cleared upon expiration of the pause timer or the receipt of an XON frame.

SPEED indicates the actual MAC speed configuration. These bits normally reflect the speed of the actual link, negotiated by the PHY and link partner, and reflected internally from the PHY to the MAC. These bits may represent the speed configuration of the MAC only, if the MAC speed setting has been forced via software (CTRL.SPEED). Speed indications are mapped as shown below:

(00)b – 10Mb/s

(01)b – 100Mb/s

(10)b – 1000Mb/s

(11)b – 1000Mb/s

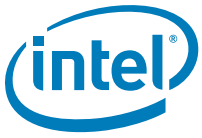
12.0.3.2.3 Strapping Option Register - STRAP (0x0000C; RO)

This register reflects the values of the soft strapping options fetched from the NVM descriptor in the CH space.

Attribute	Bit(s)	Initial Value	Description
RO	0	1	**Reserved.**
RO	5:1	0	**LAN NVM Size (NVMS)**. LAN NVM Space size is indicated in multiples of 4KB. LAN NVM size may very from 4KB to 128KB while 0 value means 4KB.
RO	16	0	**LC SMBus address enable (LCSMBADDEN)**
RO	23:17	0	**LC SMBus address (LCSMBADD)**
RO	24	0	**LCD SMBus address enable (LCDSMBADDEN)**
RO	31:25	0	**LCD SMBus address (LCDSMBADD)**

12.0.3.2.4 Extended Device Control Register - CTRL_EXT (0x00018; RW)

Bits	Type	Default	Description
11:0	RO	0	**Reserved**.
12	RW/V	1	**MACsec Clock Gate (LSecCK).** When cleared, the MACsec logic gets its clocks. When the LSecCK is set the MACsec logic (including all its CSR registers) do not get any clocks. This bit is loaded from NVM word 13h.
14:13	RO	0	**Reserved.**



15	RW	0	**Speed Select Bypass (SPD_BYPS).** When set to 1, all speed detection mechanisms are bypassed, and the device is immediately set to the speed indicated by CTRL.SPEED. This provides a method for software to have full control of the speed settings of the device when the change takes place by overriding the hardware clock switching circuitry.
18:16	RO	0	**Reserved.**
19	RW/SN	0	**Dynamic Clock Gating (DynCK)**. When set, this bit enables dynamic clock gating of the DMA and MAC units. Also see the description of the DynWakeCK in this register. The bit is loaded from NVM word 13h.
20	RW/SN	1	**PHY Power Down Enable (PHYPDEN)**. When set, this bit enables the LCD to enter a low-power state when the LAN controller is at the DMoff / D3 or Dr with no WoL. This bit is loaded from word 13h in the NVM.
21	RO	0	**Reserved**.
22	RO	1	**Reserved**.
23	RO	0	**Reserved.**
24	RO	1	**Reserved.**
25	RW	0	**DMA Clock Control (DMACKCTL)**. Controls the DMA clock source in none GbE mode (10/100 and no Link). In GbE mode the DMA clock source is always Kumeran PLL divided by 2. In nominal operation this bit should be in the default state in which the DMA clock source in none GbE is mosc_clk. In test mode the DMACKCTL and PLLGateDis should be set to 1 and CLK_CNT_1_4 in the NVM should not be set. In this mode the DMA clock source is Kumeran PLL divided by 2.
26	RW	0	**Disable Static Kumeran PLL Gating (PLLGateDis**). By default the PLL is functional only when Kumeran link is required, and inactive when it is not required (at non GbE mode if Jordan is available).When set to 1 the Kumeran PLL is always active.
27	RW	0	**IAME**. When the IAM (interrupt acknowledge auto-mask enable) bit is set, a read or write to the ICR register will have the side effect of writing the value in the IAM register to the IMC register. When this bit is 0, the feature is disabled.
28	RW	0	**Driver loaded (DRV_LOAD).** This bit should be set by the driver after it was loaded, Cleared when the driver unloads or soft reset. The MNG controller loads this bit to indicate to the manageability controller that the driver has loaded.
29	RW	0	**INT_TIMERS_CLEAR_ENA**. When set this bit enables the clear of the interrupt timers following an IMS clear. In this state, successive interrupts will occur only after the timers will expire again. When clear, successive interrupts following IMS clear may happen immediately.
30	RO	0	**Reserved**.
31	RO	0	**Reserved**. Reads as 0.

This register provides extended control of device functionality beyond that provided by the Device Control register (CTRL).

Note: If software uses the EE_RST function and desires to retain current configuration information, the contents of the control registers should be read and stored by

software. Control register values are changed by a read of the NVM which occurs upon assertion of the EE_RST bit.

Note: The EEPROM reset function may read configuration information out of the NVM which affects the configuration of PCI configuration space BAR settings. The changes to the BAR's are not visible unless the system is rebooted and the BIOS is allowed to re-map them.

Note: The SPD_BYPS bit performs a similar function as the CTRL.FRCSPD bit in that the device's speed settings are determined by the value software writes to the CRTL.SPEED bits. However, with the SPD_BYPS bit asserted, the settings in CTRL.SPEED take effect immediately rather than waiting until after the device's clock switching circuitry performs the change.

12.0.3.2.5 MDI Control Register - MDIC (0x00020; RW)

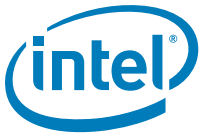
Bits	Type	Default	Description
15:0	RW/V	X	**Data (DATA).** In a Write command, software places the data bits and the MAC shifts them out to the LAN Connected Device. In a Read command, the MAC reads these bits serially from the LAN Connected Device and software can read them from this location.
20:16	RW/V	0	**LAN Connected Device Register address (REGADD).** i.e., Reg 0, 1, 2, … 31.
25:21	RW/V	0	**LAN Connected Device Address (PHYADD).**
27:26	RW/V	0	**Op-code (OP)**. 01 for MDI Write 10 for MDI Read. Other values are reserved.
28	RW/V	1	**Ready bit (R)**. Set to 1 by LAN Controller at the end of the MDI transaction (i.e., indicates a Read or Write has been completed). It should be reset to 0 by software at the same time the command is written.
29	RW/V	0	**Interrupt Enable (I)**. When set to 1 by software, it will cause an Interrupt to be asserted to indicate the end of an MDI cycle.
30	RW/V	0	**Error (E)**. This bit set is to 1 by the HW when it fails to complete an MDI read. Software should make sure this bit is clear (0) before making a MDI read or write command.
31	RO	0	**Reserved**. Write as 0 for future compatibility.

This register is used by software to read or write MDI (Management Data Interface) registers in a GMII/MII LAN Connected Device.

Note: Internal logic uses the MDIC to communicate w the LCD. All fields in these registers are indicated as "/V" since the internal logic may use them to access the LCD. Since the HW uses this register, all HW, SW and FW must use semaphore logic (the Ownership flags) before accessing the MDIC.

For an MDI Read cycle the sequence of events is as follows:

1. the CPU performs a write cycle to the MII register with:
   — Ready = 0
   — Interrupt Enable bit set to 1 or 0.



- Op-Code = 10b (read)
- PHYADD = the LAN Connected Device address from the MDI register
- REGADD = the register address of the specific register to be accessed (0 through 31)

2. the MAC applies the following sequence on the MDIO signal to the LAN Connected Device:
   - <PREAMBLE><01><10><PHYADD><REGADD><Z>

where the Z stands for the MAC tri-stating the MDIO signal.

3. the LAN Connected Device returns the following sequence on the MDIO signal:
   - <0><DATA><IDLE>
4. the MAC discards the leading bit and places the following 16 data bits in the MII register.
5. LAN Controller asserts an Interrupt indicating MDI "Done", if the Interrupt Enable bit was set.
6. LAN Controller sets the Ready bit in the MII register indicating the Read is complete.
7. the CPU may read the data from the MII register and issue a new MDI command.

For an MDI Write cycle the sequence of events is as follows:

1. the CPU performs a write cycle to the MII register with:
   - Ready = 0
   - Interrupt Enable bit set to 1 or 0.
   - Op-Code = 01b (write)
   - PHYADD = the LAN Connected Device address from the MDI register
   - REGADD = the register address of the specific register to be accessed (0 through 31)
   - Data = specific Data for Desired Control of LAN Connected Device
2. the MAC applies the following sequence on the MDIO signal to the LAN Connected Device:
   - <PREAMBLE><01><01><PHYADD><REGADD><10><DATA><IDLE>
3. the LAN Controller asserts an Interrupt indicating MDI “Done” if the Interrupt Enable bit was set.
4. the LAN Controller sets the Ready bit in the MII register to indicate Step 2. has been completed.
5. the CPU may issue a new MDI command.

Note: An MDI Read or Write may take as long as 64 microseconds from the CPU Write to the Ready bit assertion.

If an invalid opcode is written by software, the MAC will not execute any accesses to the LAN Connected Device registers.

If the LAN Connected Device does not generate a zero as the second bit of the turnaround cycle for reads, the MAC will abort the access, set the E (error) bit, write 0xFFFF to the data field to indicate an error condition, and set the ready bit.

Accessing LCD Wakeup register using MDIC

A new page is defined in the LAN Connected Device to hold the wakeup register space.

When SW wants to configure the wakeup state (either read or write to these registers) the MDIO page should be set to 800 (for host accesses). While the page remains the same Wakeup register access is enabled.

While the page is set to the Wakeup page the address field is no longer translated as reg_addr (register address) but as an instruction. If the given address is in [0..15] range meaning PHY registers, the functionality remains unchanged.

There are 2 valid instructions:

Address Set – 0x11 – Wakeup space address is set for either reading or writing.

Data cycle – 0x12 – Wakeup space accesses read or write cycle.

For the LAN Connected Device, in the wake area Read cycle, the sequence of events is as follows:

Setting page 0x0800

1. the Driver performs a write cycle to the MDI register with:
   - Ready = 0
   - Op-Code = 01b (write)
   - PHYADD = the LAN Connected Device address from the MDI register
   - REGADD = page setting
   - DATA = 0x0800 (Wakeup page)

Address setting

2. the Driver performs a write cycle to the MDI register with:
   - Ready = 0
   - Op-Code = 01b (write)
   - PHYADD = the LAN Connected Device address from the MDI register
   - REGADD = 0x11 (Address set)
   - DATA = XXXX (Address of the register to be read)

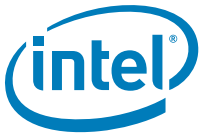
Reading a register

1. the Driver performs a write cycle to the MDI register with:
   - Ready = 0
   - Op-Code = 10b (read)
   - PHYADD = the LAN Connected Device address from the MDI register
   - REGADD = 0x12 (data cycle for read)
   - DATA = YYYY (Data will be valid when the ready bit is set)

For the LAN Connected Device, in the wake area Write cycle, the sequence of events is as follows:

Setting page 0x0800

1. the Driver performs a write cycle to the MDI register with:



- Ready = 0
- Op-Code = 01b (write)
- PHYADD = the LAN Connected Device address from the MDI register
- REGADD = page setting
- DATA = 0x0800 (Wakeup page)

Address setting

2. the Driver performs a write cycle to the MDI register with:
   - Ready = 0
   - Op-Code = 01b (write)
   - PHYADD = the LAN Connected Device address from the MDI register
   - REGADD = 0x11 (Address set)
   - DATA = XXXX (Address of the register to be read)

Writing a register

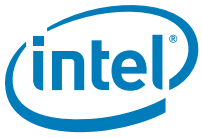
3. the Driver performs a write cycle to the MDI register with:
   - Ready = 0
   - Op-Code = 01b (write)
   - PHYADD = the LAN Connected Device address from the MDI register
   - REGADD = 0x12 (data cycle for write)
   - DATA = YYYY (Data to be written to the register)

12.0.3.2.6 Future Extended NVM Register - FEXTNVM (0x00028; RW)

This register is initialized to the HW default only at LAN PWR Good reset. Software should not modify these fields to values other than their recommended values. Bits 15:0 of this register are loaded from the NVM word 19h and bits 31:16 are loaded from the NVM word 1Ah.

Bits	Type	Default	Description
0	RW/SN	0	Reserved
1	RW/SN	0	**dma_clk_enable_d.** Enable dynamic clock stop. When this bit is set to 1, clk is always ticking, default value is '0' (HW and NVM).
2	RW/SN	0	**wake_dma_clk_enable_d.** Enable dynamic clock stop. When this bit is set to 1, clk is always ticking, default value is '0' (HW and NVM).
3	RW/SN	0	**gpt_clk_enable_d.** Enable dynamic clock stop. When this bit is set to 1, clk is always ticking, default value is '0' (HW and NVM).
4	RW/SN	0	**mac_clk_enable_d.** Enable dynamic clock stop. When this bit is set to 1, clk is always ticking, default value is '0' (HW and NVM).
5	RW/SN	0	**m2k_clk_enable_d.** Enable dynamic clock stop. When this bit is set to 1, clk is always ticking, default value is '0' (HW and NVM).

Bits	Type	Default	Description
6	RW/SN	0	**Invalid image CSUM.** When cleared this bit indicates to the LAD NVM programming tools (eeupdate..) that the Image CSUM needs to be corrected. When set the CSUM is assumed to be correct.
7	RW/SN	0	**Reserved.**
8	RW/SN	0	**Reserved.**
9	RW/SN	0	**Reserved.**
10	RW/SN	0	**Enable MDIO Watchdog Timer (MDIOWatchEna)**. When set to '0', the 100msec MDIO watchdog timer is enabled.
11	RW/SN	0	**Reserved.**
12	RW/SN	0	**Reserved.**
13	RW/SN	0	**Reserved.**
14	RW/SN	0	**Reserved.**
15	RW/SN	0	**Reserved.**
16	RW/SN	0	**Reserved**.
17	RW/SN	0	**Reserved**.
18	RW/SN	0	**Reserved**.
19	RW/SN	0	**Reserved**.
20	RW/SN	0	**Disable CLK gate Enable Due to D3hot**. When set it disables assertion of bb_clkgaten due to D3hot. Default NVM setting is '0'.
21	RW/SN	0	**LAN Disable Mode.** When set to '1', legacy flow managed by BIOS routine should be performed to disable the Lan. Otherwise, the whole flow will be managed by hardware when Lan-Disable RTC well bit is set to '1'in ICH9. Default NVM setting is '0'.
22	RW/SN	0	**Reserved**.
23	RW/SN	0	**Reserved**.
24	RW/SN	0	**Reserved**.
25	RW/SN	0	**Reserved**.
26	RW/SN	0	**Reserved**.
27	RW/SN	0	**SW LCD Config Enable**. This bit has no impact on the hardware but rather influences the software flow. The software should initialize the LCD using the "Extended Configuration" image in the NVM only when both the "SW LCD Config Enable" bit is set and the "LCD Write Enable" bit in the EXTCNF_CTRL register is cleared.
28	RW/SN	0	Reserved.
29	RW/SN	0	Reserved.
30	RW/SN	0	Enables assertion of "gbe_pmcmsus_powerdown_rdy_mosc" in WoL enabled configuration.
31	RW/SN	0	Enables assertion of "gbe_pmcmsus_powerdown_rdy_mosc" in D3/DMoff/no_wake configuration.



12.0.3.2.7 Future Extended Register - FEXT (0x0002C; RW)

This register is initialized to the HW default only at LAN PWR Good reset. Software should not modify these fields to values other than their recommended values.

Bits	Type	Default	Description
0	RO	0	**Reserved.**
1	RO	0	**Reserved.**
2	RO/V	0	**Reserved.**
3	RO/V	0	**LAN PHY Power Control (PHYPC)**. Set to 1 indicates external power to theLAN Controller is On. 0 – external power is off.
4	RW	0	**Reserved.**
5	RW	0	**Reserved.**
6	RW	0	**Reserved.**
7	RW	0	**Reserved.**
8	RW	0	**HW SW CRC mismatch trigger** – when set to 1 the LC generates a trigger signal whenever there is a mismatch between the software calculated CRC and the hardware calculated CRC. This feature is ignored when CRC calculation is off-loaded to HW.
9	RW	0	Write disable Ghost and DMA RAMs on CRC mismatch – when set to 1: disable any writes to the following rams in the event of CRC mismatch until reset: Ghost read pci descriptor Ghost read pci data the four rams in the descriptor engine the packet buffer
10	RW	0	When set to 1: Enables the data visibility of the Ghost read PCI descriptor and PCI data RAMs to the NOA.
11	RW	0	Visibility in/out read data select (1 = in) FEXT.10 must be set to 1.
12	RW	0	Visibility data/desc read ram select (1 = data) FEXT.10 must be set to 1
13	RW	0	When set to 1: The Ghost read RAMs are readable by the slave bus.
14	RW	0	**Reserved.**
15	RW	0	**Reserved.**
16	RW	0	**Reserved.**
17	RW	0	**Reserved.**
31:18	RW	0x00	**Future Extended.** Reserved for future setting.

12.0.3.2.8 Future Extended NVM 2 - FEXTNVM2 (0x00030; RW)

This register is initialized to the HW default only at LAN PWR Good reset. Software should not modify these fields to values other than their recommended values. Bits 15:0 of this register are loaded from the NVM word 24h and bits 31:16 are loaded from the NVM word 25h.

Bits	Type	Default	Description
31:0	RW/SN	0	**Reserved.**

12.0.3.2.9 Future Extended NVM 3 - FEXTNVM3 (0x0003C; RW)

This register is initialized to the HW default only at LAN PWR Good reset. Software should not modify these fields to values other than their recommended values. Bits 15:0 of this register are loaded from the NVM word 26h and bits 31:16 are loaded from the NVM word 27h.

Bits	Type	Default	Description
31:0	RW/SN	0	**Reserved.**

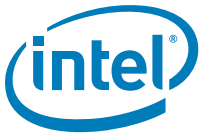
12.0.3.2.10 Future Extended NVM 4- FEXTNVM4 (0x00024; RW)

This register is initialized to the HW default only at LAN PWR Good reset. Software should not modify these fields to values other than their recommended values. Bits 15:0 of this register are loaded from the NVM word 2Ah and bits 31:16 are loaded from the NVM word 2Bh.

Bits	Type	Default	Description
2:0	RW/SN	000	**Reserved.**
3	RW/SN	0	**Reserved.**
5:4	RW/SN	00	**Reserved.**
6	RW/SN	0	**Reserved.**
7	RW/SN	0	**Reserved.**
14:8	RW/SN	0x0	**Reserved.**
15	RW/SN	0	**Reserved.**
23:16	RW/SN	0x0	**Reserved.**
25:24	RW/SN	00	**Reserved.**
26	RW/SN	0	**Reserved.**
27	RW/SN	0	**Reserved.**
28	RW/SN	0	**Reserved.**
29	RW/SN	0	Enable PLL shut on 1000Mbps link up. When set to 0 and 1000Mbps link is up the LAN Controller will not approve PLL shut in K1, when set to 1 PLL shut on K1 will not be gated in 1000Mbps speed. This bit is loaded from NVM word 2Bh bit 13
31:30	RW/SN	000	**Reserved.**

12.0.3.2.11 Device and Bus Number - BUSNUM (0x00038; RO)

Bit	Type	Default	Description
28:0	RO	0x00	**Reserved**.



10:8	RO	000b	**Function Number.** The LAN controller is a single PCI function being function 0.
15:11	RO	0x19	**Device Number.** During nominal operation the LAN controller has a predefined Device number equal to 25 (0x19).
23:16	RO	0x00	**Bus Number.** The LAN controller captures its bus number during host configuration write cycles type 0 aimed at the device. This field is initialized by LAN power good reset, PCI reset and D3 to D0 transition.
31:24	RO	0x00	**Reserved**.

12.0.3.2.12 Flow Control Transmit Timer Value - FCTTV (0x00170; RW)

Bit	Type	Default	Description
15:0	RW	X	**Transmit Timer Value (TTV)**. to be included in XOFF frame.
31:16	RO	0	**Reserved**. Read as 0. Should be written to 0 for future compatibility.

The 16-bit value in the TTV field is inserted into a transmitted frame (either XOFF frames or any PAUSE frame value in any software transmitted packets). It counts in units of slot time. If software wishes to send an XON frame, it must set TTV to 0 prior to initiating the PAUSE frame.

Note: The LAN Controller uses a fixed slot time value of 64 byte times.

12.0.3.2.13 Flow Control Refresh Threshold Value - FCRTV (0x05F40; RW)

Bit	Type	Default	Description
15:0	RW	X	**Flow Control Refresh Threshold (FCRT)**. This value indicates the threshold value of the flow control shadow counter. When the counter reaches this value, and the conditions for a pause state are still valid (buffer fullness above low threshold value), a pause (XOFF) frame is sent to the link partner. The FCRTV timer count interval is the same as other flow control timers and counts at slot times of 64 byte times. If this field contains a zero value, the Flow Control Refresh is disabled.
31:16	RO	0	**Reserved**.

12.0.3.2.14 Extended Configuration Control - EXTCNF_CTRL (0x00F00; RW)

Bit	Type	Default	Description
0	RW/SN	0	**LCD Write Enable.** When set, enables the Extended LAN Connected Device Configuration area in the LAN Controller. When disabled, the Extended LAN Connected Device Configuration area is ignored. Loaded from NVM word 14h.
1	RW/SN	0	**Reserved.**
2	RW/SN	0	**Reserved.**

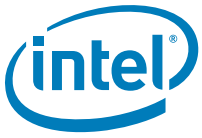
3	RW/SN	1	**OEM Write Enable.** When set, enables auto load of the OEM bits from the PHY_CTRL register to the PHY. Loaded from NVM word 14h.
4	RO	0	**Reserved**.
5	RW/V	0	**SW Semaphore FLAG (SWFLAG).** This bit is set by the device driver to gain access permission to shared CSR registers with the firmware and hardware The bit is initialized on power-up PCI reset and software reset.
6	RO/V	0	**MDIO HW Ownership.** HW requests access to MDIO. Part of the arbitration scheme for MDIO access. This is a RO bit.
7	RW/V	0	Gate Phy Configuration (PPW, SKU read, OEM configuration).
15:8	RO	0x0	**Reserved**.
27:16	RW/SN	0x001	**Extended Configuration Pointer**. Defines the base address (in DW) of the Extended Configuration area in the NVM.
31:28	RW	0	**Reserved**.

12.0.3.2.15 Extended Configuration Size - EXTCNF_SIZE (0x00F08; RW)

Bit	Type	Default	Description
31:24	RO	0x0	**Reserved**.
23:16	RW/SN	0x0	**Extended LCD Length**. Size (in DW) of the Extended LAN Connected Device Configuration area loaded from Extended Configuration word 2 in the NVM. If an extended configuration area is disabled by "LCD Write Enable" field in word 14h in the NVM, this length must be set to zero.
15:0	RW/SN	0x0	**Reserved**.

12.0.3.2.16 +-PHY Control Register - PHY_CTRL (0x00F10; RW) This register is initialized to the HW default at LAN PWR Good reset.

Bit	Type	Default	Description
31:29	RO	0x0	**Reserved**
28:25	RO	0x0	**SKU read data**. These four bits contain the SKU value read from the LANConnected Device SKU register. Using these bits, the SKU mechanism determines the Device ID.
24	RO	0x0	**Reserved**.
23	RO	0x0	**SKU done**. This bit indicates the termination of SKU read.
22	RW	0x0	**Reserved**.
21	RW	0	**Reserved**
20	RW	0	**Reserved.**
19:17	RW	0x2	**Reserved.**
16	RW	0	**Reserved.**
15:8	RO	0x0	**Reserved**
7	RW/SN	0	**B2B Ena**. Enables SPD in Back To Back link setup. Bit is initialized by word 17h bit 15 in the NVM.



Bit	Type	Default	Description
6	RW/SN	0	**Global GbE Disable**. Prevents PHY auto negotiating 1000 Mb/s link in all power states (including D0a). Bit is initialized by word 17h bit 14 in the NVM.
5:4	RO	00b	**Reserved**.
3	RW/SN	1	**GbE Disable at non D0a**. Prevents PHY from auto negotiating 1000Mb/s link in all power states except D0a (DR, D0u and D3). Bit is initialized by word 17h bit 11 in the NVM. This bit must be set since GbE is not supported in Sx by the platform.
2	RW/SN	1	**LPLU in non D0a**. Enables PHY to negotiate for slowest possible link (Reverse AN) in all power states except D0a (DR, D0u and D3). Bit is initialized by word 17h bit 10 in the NVM.
1	RW/SN	0	**LPLU in D0a**. Enables PHY to negotiate for slowest possible link (Reverse AN) in all power states (including D0a). This bit overrides the LPLU in non D0a bit. Bit is initialized by word 17h bit 9 in the NVM.
0	RW/SN	0	**SPD Ena**. Enables PHY Smart Power Down mode. Bit is initialized by word 17h bit 8 in the NVM.

12.0.3.2.17 PCIE Analog Configuration - PCIEANACFG (0x00F18; RW)

Bit	Type	Default	Description
31:6	RO	0	**Reserved**.
5:0	RW/SN	0x0	**Reserved**.

12.0.3.2.18 Packet Buffer Allocation - PBA (0x01000; RW)

Bit	Type	Reset	Description
4:0	RW	0x12	**Receive packet buffer allocation (RXA)**. Defines the size of the Rx buffer in K byte units. Default is 18Kbytes.
15:5	RO	X	**Reserved**.
20:16	RO	0x	**Transmit packet buffer allocation (TXA)**. Defines the size of the Tx buffer in K byte units. This field is read only and equals to the Packet Buffer Size (PBS) minus RXA (the default value of the PBS is KB).
31:21	RO	X	**Reserved**.

This register sets the on-chip receive and transmit storage allocation ratio.

Programming this register does not automatically re-load or initialize internal packet-buffer RAM pointers. The software must reset both transmit and receive operation (using the global device reset CTRL.SWRST bit) after changing this register in order for it to take effect. The PBA register itself will not be reset by assertion of the Software reset, but will only be reset upon initial hardware power-on.

If Early Receive functionality is not enabled (indicate field/register), the Receive packet buffer should be larger than the max expected received packet + 32B.

For best performance the transmit buffer allocation should be set to accept two full sized packets.

Transmit Packet Buffer size should be configured to be more than 4 KB.

12.0.3.2.19 Packet Buffer Size - PBS (0x01008; RW)

Bit	Type	Reset	Description
4:0	RW	0x12	**Receive packet buffer allocation (RXA)**. Defines the size of the Rx buffer in K byte units. Default is 18Kbytes.
15:5	RO	X	**Reserved**.
20:16	RO	0x	**Transmit packet buffer allocation (TXA)**. Defines the size of the Tx buffer in K byte units. This field is read only and equals to the Packet Buffer Size (PBS) minus RXA (the default value of the PBS is KB).
31:21	RO	X	**Reserved**.

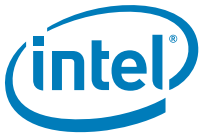
This register sets the on-chip receive and transmit storage allocation size, The allocation value is read/write for the lower 6 bits. The division between transmit and receive is done according to the PBA register.

Note: Programming this register does not automatically re-load or initialize internal packet-buffer RAM pointers. The software must reset both transmit and receive operation (using the global device reset CTRL.SWRST bit) after changing this register in order for it to take effect. The PBS register itself will not be reset by assertion of the Software reset, but will only be reset upon initial hardware power-on.

Note: Programming this register should be aligned with programming the PBA register HW operation, if PBA and PBS are not coordinated is not determined.

12.0.3.2.20 Packet Buffer ECC Status - PBECCSTS (0x0100C; RW)

Bit	Type	Reset	Description
7:0	RC	0x0	Corr_err_cnt - Correctable Error Count: This counter is increment every time a correctable error is detected, the counter stops counting when reaching 0xff. Cleared by read.
15:8	RC	0x0	uncorr_err_cnt - Uncorrectable Error Count: This counter is increment every time an uncorrectable error is detected, the counter stops counting when reaching 0xff. Cleared by read.
16	RW	0	ECC enable
17	RW	0	Stop on first Error (SOFE). When set the ECC test will capture the failing address into LFA
19:18	RO	0x0	**Reserved.** read as zero.
31:20	RO	0x0	Last failure address (LFA). When Stop on first Error (SOFE) bit is set to 1, when there is ECC failure, the LFA register will capture the failing address of the failure



12.0.3.2.21 Packet Buffer ECC Error Inject - PBEEI (0x01004; RW)

Bit	Type	Reset	Description
0	RW	0	Inject an error on TX Buffer on header line When this bit is set an error will be injected in the next write cycle to a header line of the TX buffer. Auto cleared by HW when error is injected if PBECCINJ.ENECCADD is clear (0).
1	RW	0	Inject an error on TX Buffer on data line When this bit is set an error will be injected in the next write cycle to a data line of the TX buffer. Auto cleared by HW when error is injected if PBECCINJ.ENECCADD is clear (0).
2	RW	0	Inject an error on RX Buffer on header line When this bit is set an error will be injected in the next write cycle to a header line of the RX buffer. Auto cleared by HW when error is injected if PBECCINJ.ENECCADD is clear (0).
3	RW	0	Inject an error on RX Buffer on data line When this bit is set an error will be injected in the next write cycle to a data line of the RX buffer. Auto cleared by HW when error is injected if PBECCINJ.ENECCADD is clear (0).
15:4	RO	0x0	Reserved
23:16	RW	0	Error 1 bit location (value of 0xFF - No Error injection on this bit)
31:24	RW	0x0	Error 2 bit location (value of 0xFF - No Error injection on this bit)

12.0.3.2.22 Packet Buffer ECC Injection - PBECCINJ (0x01010; RW)

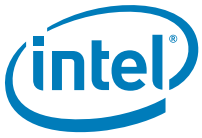
Bit	Type	Reset	Description
11:0	RW	0	Address 0 Injection - Error injection first address in packet buffer.
23:12	RW	0	Address 1 Injection - Error injection secondaddress in packet buffer.
24	RW	0	Enable ECC injection to address (ENACCADD). When set to 0 the addresses for ECC injection from this register will be ignored
31:25	RO	0x0	Reserved

12.0.3.3 Interrupt Register Descriptions

12.0.3.3.1 Interrupt Cause Read Register - ICR (0x000C0; RC/WC)

This register is Read-Clear or Write-Clear (see details after the tables). If enabled, read access also clears the ICR content after it is posted to the SW. Otherwise a Write cycle is required to clear the relevant bit fields. Write a '1' clears the written bit while writing '0' has no affect (with the exception of the INT_ASSERTED bit as detailed below).

Bit	Type	Default	Description
0	RWC/CR/V	0	**Transmit Descriptor Written Back (TXDW)**. Set when hardware processes a descriptor with either RS set. If using delayed interrupts (IDE set), the interrupt is delayed until after one of the delayed-timers (TIDV or TADV) expires.
1	RWC/CR/V	0	**Transmit Queue Empty (TXQE)**. Set when, the last descriptor block for a transmit queue has been used. When configured to use more than one transmit queue this interrupt indication will be issued if one of the queues is empty and will not be cleared until all the queues have valid descriptors.
2	RWC/CR/V	0	**Link Status Change (LSC)**. This bit is set whenever the link status changes (either from up to down, or from down to up). This bit is affected by the LINK indication from the PHY.
3	RO	0	**Reserved**.
4	RWC/CR/V	0	**Receive Descriptor Minimum Threshold hit (RXDMT0)**. Indicates that the minimum number of receive descriptors RCTL.RDMTS are available and software should load more receive descriptors.
5	RWC/CR/V	0	**Disable SW Write Access (DSW).** The DSW bit indicates that the FW changed the status of the DISSW or the DISSWLNK bits in the FWSM register.
6	RWC/CR/V	0	**Receiver Overrun (RXO).** Set on receive data FIFO overrun. Could be caused either because there are no available buffers or because receive bandwidth is inadequate.
7	RWC/CR/V	0	**Receiver Timer Interrupt (RXT0).** Set when the timer expires.
8	RWC/CR/V	0	**Reserved.**
9	RWC/CR/V	0	**MDIO Access Complete (MDAC)**. Set when the MDIO access is completed.
11:10	RO	0	**Reserved**.
12	RWC/CR/V	0	**PHY Interrupt (PHYINT).** Set when the LAN Connected Device generates an interrupt.
13	RO	0	**Reserved**.
14	RWC/CR/V	0	**MACsec Packet Number (LSECPN).** The Tx Packet Number hit the "PN exhaustion threshold" as defined in the LSECTXCTRL register and the host is the KaY.
15	RWC/CR/V	0	**Transmit Descriptor Low Threshold hit (TXD_LOW).** Indicates that the descriptor ring has reached the threshold specified in the Transmit Descriptor Control register.
16	RWC/CR/V	0	**Small Receive Packet Detected (SRPD).** Indicates that a packet of size < RSRPD.SIZE register has been detected and transferred to host memory. The interrupt is only asserted if RSRPD.SIZE register has a non-zero value.



Bit	Type	Default	Description
17	RWC/CR/V	0	**Receive ACK Frame Detected (ACK).** Indicates that an ACK frame has been received and the timer in RAID.ACK_DELAY has expired.
18	RWC/CR/V	0	**Reserved.**
19	RWC/CR/V	0	**Reserved.**
20	RWC/CR/V	0	**Reserved.**
21	RWC/CR/V	0	**Reserved**.
22	RWC/CR/V	0	**ECC Error (ECCER).** Indicates an uncorrectable EEC error had occured
30:23	RO	0	**Reserved**. Reads as 0.
31	RWC/CR/V	0	**Interrupt Asserted (INT_ASSERTED).** This bit is set when the LAN port has a pending interrupt. If the Interrupt is enabled in the PCI configuration space an Interrupt is asserted.

This register contains all interrupt conditions for LAN Controller. Whenever an interrupt causing event occurs, the corresponding interrupt bit is set in this register. An interrupt is generated whenever one of the bits in this register is set, and the corresponding interrupt is enabled via the Interrupt Mask Set/Read Register (see Section 12.0.3.4.5).

Whenever an interrupt causing event occurs, all timers of delayed interrupts are cleared and their cause event is set in the ICR.

- Read ICR register is affected differently in the following cases:
  - Case 1 - Interrupt Mask Register equals 0x0000 (mask all) - ICR content will be cleared.
  - Case 2 - Interrupt was asserted (ICR.INT_ASSERTED=1) - ICR content will be cleared and Auto Mask is active, meaning, the IAM register is written to the IMC register.
  - Case 3 - Interrupt was not asserted (ICR.INT_ASSERTED=0) - Read has no side affect.

Writing a 1 to any bit in the register will also clear that bit. Writing a 0 to any bit will have no effect on that bit. The INT_ASSERTED bit is a special case. Writing a 1 or 0 to this bit has no affect. It is cleared only when all interrupt sources are cleared.

12.0.3.3.2 Interrupt Throttling Register - ITR (0x000C4; RW)

Bit	Type	Default	Description
15:0	RW	0	**INTERVAL**. Minimum inter-interrupt interval. The interval is specified in 256ns units. Zero disables interrupt throttling logic.
31:16	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.

Software can use this register to pace (or even out) the delivery of interrupts to the host CPU. This register provides a guaranteed inter-interrupt delay between interrupts asserted by the network controller, regardless of network traffic conditions. To independently validate configuration settings, software can use the following algorithm to convert the inter-interrupt interval value to the common 'interrupts/sec' performance metric:

Interrupts/sec = $(256 \times 10^{-9}\text{sec} \times \text{interval})^{-1}$

For example, if the interval is programmed to 500d, the network controller guarantees the CPU will not be interrupted by the network controller for 128 microseconds from the last interrupt.

Inversely, inter-interrupt interval value can be calculated as:

inter-interrupt interval = $(256 \times 10^{-9}\text{sec} \times \text{interrupts/sec})^{-1}$

The optimal performance setting for this register is very system and configuration specific. An initial suggested range for the interval value is 65--5580 (28B - 15CC).

Note: When working at 10/100Mbps and running at ¼ clock the interval time is multiplied by four.

12.0.3.3.3 Interrupt Cause Set Register - ICS (0x000C8; WO)

Bit	Type	Default	Description
0	WO	X	**TXDW**. Sets Transmit Descriptor Written Back.
1	WO	X	**TXQE**. Sets Transmit Queue Empty.
2	WO	X	**LSC**. Sets Link Status Change.
3	RO	X	**Reserved**.
4	WO	X	**RXDMT**. Sets Receive Descriptor Minimum Threshold hit.
5	WO	X	**DSW**. Sets Block SW Write accesses.
6	WO	X	**RXO**. Sets Receiver Overrun. Set on receive data FIFO overrun.
7	WO	X	**RXT**. Sets Receiver Timer Interrupt.
8	WO	X	**Reserved.**
9	WO	X	**MDAC**. Sets MDIO Access Complete Interrupt.
11:10	RO	X	**Reserved**.
12	WO	X	**PHYINT**. Sets PHY Interrupt.
13	RO	X	**Reserved**.
14	WO	X	**LSECPN.** Sets MACsec Packet Number Interrupt.
15	WO	X	**TXD_LOW**. Transmit Descriptor Low Threshold Hit.
16	WO	X	**SRPD**. Small Receive Packet Detected and Transferred.
17	WO	X	**ACK**. Set Receive ACK frame detected.
18	WO	X	**MNG.** Set the Manageability Event Interrupt.
19	WO	X	**Reserved.**
20	WO	X	**Reserved.**
21	RO	X	**Reserved**.
22	WO	X	**ECCER** Set uncorrectable EEC error.
31:23	RO	X	**Reserved**. Should be written with 0 to ensure future compatibility.

Software uses this register to set an interrupt condition. Any bit written with a "1" sets the corresponding interrupt. This results in the corresponding bit being set in the Interrupt Cause Read Register (see Section 12.0.3.4), and an interrupt is generated if one of the bits in this register is set, and the corresponding interrupt is enabled via the Interrupt Mask Set/Read Register (see Section 12.0.3.4.5).

Bits written with "0" are unchanged.

12.0.3.3.4 Interrupt Mask Set/Read Register - IMS (0x000D0; RW)

Bit	Type	Default	Description
0	RWS	0	**TXDW.** Sets mask for Transmit Descriptor Written Back.
1	RWS	0	**TXQE.** Sets mask for Transmit Queue Empty.
2	RWS	0	**LSC.** Sets mask for Link Status Change.
3	RO	0	**Reserved.**
4	RWS	0	**RXDMT0.** Sets mask for Receive Descriptor Minimum Threshold hit.
5	RWS	0	**DSW.** Sets mask for Block SW Write accesses.
6	RWS	0	**RXO.** Sets mask for Receiver Overrun. Set on receive data FIFO overrun.
7	RWS	0	**RXT0.** Sets mask for Receiver Timer Interrupt.
8	RWS	0	**Reserved.**
9	RWS	0	**MDAC.** Sets mask for MDIO Access Complete Interrupt.
11:10	RO	0	**Reserved**.
12	RWS	0	**PHYINT.** Sets mask for PHY Interrupt.
13	RO	0	**Reserved.**
14	RWS	0	**LSECPN.** Sets the mask for MACsec Packet Number Int.
15	RWS	0	**TXD_LOW.** Sets the mask for Transmit Descriptor Low Threshold hit.
16	RWS	0	**SRPD.** Sets mask for Small Receive Packet Detection.
17	RWS	0	**ACK.** Sets the mask for Receive ACK frame detection.
18	RWS	0	**MNG.** Sets mask for Manageability Event Interrupt.
19	RWS	0	**Reserved.**
20	RWS	0	**Reserved.**
21	RO	0	**Reserved**.
22	RWS	0	**ECCER** Sets mask for uncorrectable EEC error
31:23	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.

Reading this register returns which bits have an interrupt mask set. An interrupt is enabled if its corresponding mask bit is set to 1, and disabled if its corresponding mask bit is set to 0. An interrupt is generated whenever one of the bits in this register is set, and the corresponding interrupt condition occurs. The occurrence of an interrupt condition is reflected by having a bit set in the Interrupt Cause Read Register (see Section 12.0.3.4).

A particular interrupt may be enabled by writing a 1 to the corresponding mask bit in this register. Any bits written with a 0 are unchanged.

Note: If software desires to disable a particular interrupt condition that had been previously enabled, it must write to the Interrupt Mask Clear Register (see Section 12.0.3.4.6), rather than writing a 0 to a bit in this register.

When the CTRL_EXT.INT_TIMERS_CLEAR_ENA bit is set, then following writing all 1's to the IMS register (enable all Interrupts) all interrupt timers are cleared to their initial value. This auto clear provides the required latency before the next INT event.

12.0.3.3.5 Interrupt Mask Clear Register - IMC (0x000D8; WO)

Bit	Type	Default	Description
0	WO	0	**TXDW.** Clears mask for Transmit Descriptor Written Back.
1	WO	0	**TXQE.** Clears mask for Transmit Queue Empty.
2	WO	0	**LSC.** Clears mask for Link Status Change.
3	RO	0	**Reserved.**
4	WO	0	**RXDMT0.** Clears mask for Receive Descriptor Minimum Threshold hit.
5	WO	0	**DSW.** Clears mask for Block SW Write accesses.
6	WO	0	**RXO.** Clears mask for Receiver Overrun.
7	WO	0	**RXT0.** Clears mask for Receiver Timer Interrupt.
8	WO	0	**Reserved.**
9	WO	0	**MDAC.** Clears mask for MDIO Access Complete Interrupt.
11:10	RO	0	**Reserved**. Reads as 0.
12	WO	0	**PHYINT.** Clears PHY Interrupt.
13	RO	0	**Reserved.**
14	WO	0	**LSECPN.** Clears the MACsec Packet Number Interrupt.
15	WO	0	**TXD_LOW.** Clears the mask for Transmit Descriptor Low Threshold hit.
16	WO	0	**SRPD.** Clears mask for Small Receive Packet Detect Interrupt.
17	WO	0	**ACK.** Clears the mask for Receive ACK frame detect Interrupt.
18	WO	0	**MNG.** Clears mask for the Manageability Event Interrupt.
19	WO	0	**Reserved.**
20	WO	0	**Reserved.**
21	RO	0	**Reserved**.
22	WO	0	**ECCER** Clears the mask for uncorrectable EEC error
31:23	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.

Software uses this register to disable an interrupt. Interrupts are presented to the bus interface only when the mask bit is a one and the cause bit is a one. The status of the mask bit is reflected in the Interrupt Mask Set/Read Register, and the status of the cause bit is reflected in the Interrupt Cause Read register (see Section 12.0.3.4).

Software blocks interrupts by clearing the corresponding mask bit. This is accomplished by writing a 1 to the corresponding bit in this register. Bits written with 0 are unchanged (i.e. their mask status does not change).

In summary, the sole purpose of this register is to allow software a way to disable certain, or all, interrupts. Software disables a given interrupt by writing a 1 to the corresponding bit in this register.

12.0.3.3.6 Interrupt Acknowledge Auto-Mask - IAM (0x000E0; RW)

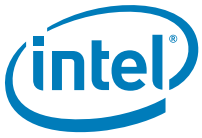
Bit	Type	Default	Description
0-31	RW	0	**IAM_VALUE**. When the CTRL_EXT.IAME bit is set and the ICR.INT_ASSERTED=1, an ICR read or write will have the side effect of writing the contents of this register to the IMC register.

12.0.3.4 Receive Register Descriptions

12.0.3.4.1 Receive Control Register - RCTL (0x00100; RW)

Bit	Type	Default	Description
0	RO	0	**Reserved**. This bit represented a hardware reset of the receive-related portion of the device in previous controllers, but is no longer applicable. Only a full device reset CTRL.SWRST is supported. Write as 0 for future compatibility.
1	RW	0	**Enable (EN)**. The receiver is enabled when this bit is 1. Writing this bit to 0, stops reception after receipt of any in progress packets. All subsequent packets are then immediately dropped until this bit is set to 1. Note that this bit controls only DMA functionality to the host. Packets are counted by the statistics even when this bit is cleared.
2	RW	0	**Store bad packets (SBP)**. 0 – do not store bad packets 1 – store bad packets Note that CRC errors before the SFD are ignored. Any packet must have a valid SFD in order to be recognized by the device (even bad packets). Note: Packet errors will not be routed to the MNG even if this bit is set.
3	RW	0	**Unicast promiscuous enable (UPE)**. 0 – disabled 1 – enabled
4	RW	0	**Multicast promiscuous enable (MPE)**. 0 – disabled 1 – enabled
5	RW	0	**Long packet enable (LPE)**. 0 – disabled 1 – enabled
7:6	RW	00b	**Reserved**.
9:8	RW	0	**Receive Descriptor Minimum Threshold Size (RDMTS)**. The corresponding interrupt is set whenever the fractional number of free descriptors becomes equal to RDMTS. Table 12-8 below lists which fractional values correspond to RDMTS values. See Section 12.0.3.5.8 for details regarding RDLEN.
11:10	RW	0	**Descriptor Type (DTYP).** 00 – Legacy or Extended descriptor type 01 – Packet Split descriptor type 10 and 11 – Reserved

Bit	Type	Default	Description
13:12	RW	0	**Multicast Offset (MO)**. This determines which bits of the incoming multicast address are used in looking up the bit vector. 00 – [47:38] 01 – [46:37] 10 – [45:36] 11 – [43:34]
14	RW	0	**Reserved.**
15	RW	0	**Broadcast Accept Mode (BAM)**. 0 – ignore broadcast (unless it matches through exact or imperfect filters) 1 – accept broadcast packets.
17:16	RW	0	**Receive Buffer Size (BSIZE)**. RCTL.BSEX – 0 00 – 2048 Bytes 01 – 1024 Bytes 10 – 512 Bytes 11 – 256 Bytes RCTL.BSEX – 1 00 – reserved 01 – 16384 Bytes 10 – 8192 Bytes 11 – 4096 Bytes BSIZE is only used when DTYP – 00. When DTYP – 01, the buffer sizes for the descriptor are controlled by fields in the PSRCTL register. BSIZE is not relevant when the FLXBUF is other than 0, in that case, FLXBUF determines the buffer size.
21:18	RO	0	**Reserved.** Should be written with 0.
22	RW	0	**Reserved.**
23	RW	0	**Pass MAC Control Frames (PMCF)**. 0 – do not (specially) pass MAC control frames. 1 – pass any MAC control frame (type field value of 0x8808) that does not contain the pause opcode of 0x0001.
24	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.
25	RW	0	**Buffer Size Extension (BSEX)**. Modifies buffer size indication (BSIZE above). 0 – Buffer size is as defined in BSIZE 1 – Original BSIZE values are multiplied by 16.
26	RW	0	**Strip Ethernet CRC from incoming packet (SECRC)**. 0 – does not strip CRC 1 – Strips CRC The stripped CRC is not DMA'd to host memory and is not included in the length reported in the descriptor
30:27	RW	0	**FLXBUF.** Determines a flexible buffer size. When this field is "0000", the buffer size is determined by BSIZE. If this field is different from "0000", the receive buffer size is the number represented in Kbytes: i.e. "0001" = 1KB (1024 Bytes).



Bit	Type	Default	Description
31	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.

LPE controls whether long packet reception is permitted. Hardware discards long packets if LPE is 0. A long packet is one longer than 1522 bytes. If LPE is 1, the maximum packet size that the device can receive is 9018 bytes.

RDMTS{1,0} determines the threshold value for free receive descriptors according to the following table:

Table 12-8. RDMTS Values

RDMTS	Free Buffer Threshold
00	½
01	¼
10	1/8
11	Reserved

BSIZE controls the size of the receive buffers and permits software to trade-off descriptor performance versus required storage space. Buffers that are 2048 bytes require only one descriptor per receive packet maximizing descriptor efficiency. Buffers that are 256 bytes maximize memory efficiency at a cost of multiple descriptors for packets longer than 256 bytes.

PMCF controls the DMA function of MAC control frames (other than flow control). A MAC control frame in this context must be addressed to either the MAC control frame multicast address or the station address, match the type field and NOT match the PAUSE opcode of 0x0001. If PMCF = 1 then frames meeting this criteria will be DMA'd to host memory.

The SECRC bit controls whether the hardware strips the Ethernet CRC from the received packet. This stripping occurs prior to any checksum calculations. The stripped CRC is not DMA'd to host memory and is not included in the length reported in the descriptor.

12.0.3.4.2 Receive Control Register 1 - RCTL1 (0x00104; RW)

Bit	Type	Default	Description
7:0	RO	0	**Reserved**. This bit represented a hardware reset of the receive-related portion of the device in previous controllers, but is no longer applicable. Only a full device reset CTRL.SWRST is supported. Write as 0 for future compatibility.
9:8	RW	0	**Receive Descriptor Minimum Threshold Size (RDMTS)**. The corresponding interrupt is set whenever the fractional number of free descriptors becomes equal to RDMTS. Table 12-8 lists which fractional values correspond to RDMTS values. See Section 12.0.3.5.8 for details regarding RDLEN.
11:10	RW	0	**Descriptor Type (DTYP).** 00 – Legacy or Extended descriptor type 01 – Packet Split descriptor type 10 and 11 – Reserved The value of RCTL1.DTYP should be the same as RCTL.DTYP (same descriptor types used in both descriptor queues).

15:12	RO	0	**Reserved**.
17:16	RW	0	**Receive Buffer Size (BSIZE)**. RCTL.BSEX – 0 00 – 2048 Bytes 01 – 1024 Bytes 10 – 512 Bytes 11 – 256 Bytes RCTL.BSEX – 1 00 – reserved 01 – 16384 Bytes 10 – 8192 Bytes 11 – 4096 Bytes BSIZE is only used when DTYP – 00. When DTYP – 01, the buffer sizes for the descriptor are controlled by fields in the PSRCTL register. BSIZE is not relevant when the FLXBUF is other than 0, in that case, FLXBUF determines the buffer size.
24:18	RO	0	**Reserved.** Should be written with 0.
25	RW	0	**Buffer Size Extension (BSEX)**. Modifies buffer size indication (BSIZE above). 0 – Buffer size is as defined in BSIZE 1 – Original BSIZE values are multiplied by 16.
26	RW	0	**Reserved.** Should be written with 0.
30:27	RW	0	**FLXBUF.** Determine a flexible buffer size. When this field is "0000", the buffer size is determined by BSIZE. If this field is different from "0000", the receive buffer size is the number represented in Kbytes: i.e. "0001" – 1KB (1024 Bytes).
31	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.

This register is used to configure queue1 registers when working in VMDq mode.

12.0.3.4.3 Early Receive Threshold - ERT (0x02008; RW)

Bit	Type	Default	Description
12:0	RW	0x0000	**Receive Threshold Value (RxThreshold)**. This threshold is in units of 8 bytes.
13	RW	0	**Reserved.** When SW activates the "Early Receive" mechanism (by setting the RxThreshold field to a non-zero value) it must set this bit as well.
14	RO	0	**Reserved.**
21:15	RW	0	**Reserved.**
31:22	RO	0	**Reserved.** Reads as 0. Should be written to 0 for future compatibility.

This register contains the RxThreshold value. This threshold determines how many bytes of a given packet should be in the LAN Controller's on-chip receive packet buffer before it attempts to begin transmission of the frame on the host bus. This register allows software to configure the "early receive" mode.

This field has a granularity of 8 bytes. So, if this field is written to 0x20, which corresponds to a threshold of 256 (decimal) bytes. If the size of a given packet is smaller than the threshold value, or if this register is set to 0, then LAN Controller will start the PCI transfer only after the entire packet is contained in LAN Controller's receive packet buffer. LAN Controller examines this register on a cycle by cycle basis to determine if there is enough data to start a transfer for the given frame over the PCI bus.

Once the device acquires the bus, it will attempt to DMA all of the data collected in the internal receive packet buffer so far.

The only negative affect of setting this value too low is that it will cause additional PCI bursts for the packet. In other words, this register allows software to trade-off latency versus bus utilization. Too high a value will effectively eliminate the early receive benefits (at least for short packets) and too low a value will deteriorate PCI bus performance due to a large number of small bursts for each packet. The RUTEC statistic counts certain cases where the ERT has been set too low, and thus provides software a feedback mechanism to better tune the value of the ERT.

It should also be noted that this register will have an effect only when the receive packet buffer is nearly empty (the only data in the packet buffer is from the packet that is currently on the wire).

Note: When Early receive is used in parallel to the Packet split feature the minimum value of the ERT register should be bigger than the header size to enable the actual packet split.

Note: Early Receive should be enabled only when working in a Jumbo frames enabled environment and the ERT.RxThreshold should be set to 0xFA so that only packets bigger than 2K bytes would trigger the early receive mechanism.

12.0.3.4.4 Packet Split Receive Control Register - PSRCTL (0x02170 + n*0x4 [n=0..1]; RW)

Bit	Type	Default	Description
6:0	RW	2	**Receive Buffer Size for Buffer 0 (BSIZE0).** The value is in 128 byte resolution. Value can be from 128 bytes to 16256 bytes (15.875 Kbytes). Default buffer size is 256B. SW should not program this field to a zero value.
7	RO	0	**Reserved.** Should be written with 0 to ensure future compatibility.
13:8	RW	4	**Receive Buffer Size for Buffer 1 (BSIZE1).** The value is in 1K resolution. Value can be from 1K byte to 63K bytes. Default buffer size is 4KB. SW should not program this field to a zero value.
15:14	RO	0	**Reserved.** Should be written with 0 to ensure future compatibility.
21:16	RW	4	**Receive Buffer Size for Buffer 2 (BSIZE2).** The value is in 1K resolution. Value can be from 1K byte to 63K bytes. Default buffer size is 4KB. SW may program this field to any value.
23:22	RO	0	**Reserved.** Should be written with 0 to ensure future compatibility.

Bit	Type	Default	Description
29:24	RW	0	**Receive Buffer Size for Buffer 3 (BSIZE3)**. The value is in 1K resolution. Value can be from 1K byte to 63K bytes. Default buffer size is 0KB. SW may program this field to any value.
31:30	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.

Note: If SW sets a buffer size to zero, all buffers following that one must be set to zero as well. Pointers in the Receive descriptors to buffers with a zero size should be set to anything but NULL pointers.

12.0.3.4.5 Flow Control Receive Threshold Low - FCRTL (0x02160; RW)

Bit	Type	Default	Description
2:0	RO	0	**Reserved**. The underlying bits might not be implemented in all versions of the chip. Must be written with 0.
15:3	RW	0	**Receive Threshold Low (RTL)**. FIFO low water mark for flow control transmission.
30:16	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.
31	RW	0	**XON Enable (XONE)**. 0 – disabled 1 – enabled.

This register contains the receive threshold used to determine when to send an XON packet. It counts in units of bytes. The lower 3 bits must be programmed to 0 (8 byte granularity). Software must set XONE to enable the transmission of XON frames. Whenever hardware crosses the receive high threshold (becoming more full), and then crosses the receive low threshold and XONE is enabled (= 1), hardware transmits an XON frame.

Note that flow control reception/transmission are negotiated capabilities by the Auto-Negotiation process. When the device is manually configured, flow control operation is determined by the RFCE and TFCE bits of the device control register.

This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x00168.

12.0.3.4.6 Flow Control Receive Threshold High - FCRTH (0x02168; RW)

Bit	Type	Default	Description
2:0	RO	0	**Reserved**. The underlying bits might not be implemented in all versions of the chip. Must be written with 0.
15:3	RW	0	**Receive Threshold High (RTH)**. FIFO high water mark for flow control transmission.
31:16	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register contains the receive threshold used to determine when to send an XOFF packet. It counts in units of bytes. This value must be at least 8 bytes less than the maximum number of bytes allocated to the Receive Packet Buffer (PBA, RXA), and the

lower 3 bits must be programmed to 0 (8 byte granularity). Whenever the receive FIFO reaches the fullness indicated by RTH, hardware transmits a PAUSE frame if the transmission of flow control frames is enabled.

Note that flow control reception/transmission are negotiated capabilities by the Auto-Negotiation process. When the device is manually configured, flow control operation is determined by the RFCE and TFCE bits of the device control register.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x00160.

12.0.3.4.7 Receive Descriptor Base Address Low queue - RDBAL (0x02800 + n*0x100[n=0..1]; RW)

Bit	Type	Default	Description
3:0	RO	0	**Reserved**. Ignored on writes. Returns 0 on reads.
31:4	RW	X	**Receive Descriptor Base Address Low** (RDBAL).

This register contains the lower bits of the 64 bit descriptor base address. The lower 4 bits are always ignored. The Receive Descriptor Base Address must point to a 16B aligned block of data.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, RDBAL[0] may also be accessed at its alias offset of 0x00110.

12.0.3.4.8 Receive Descriptor Base Address High queue - RDBAH (0x02804 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
31:0	RW	X	**Receive Descriptor Base Address** [63:32] (RDBAH).

This register contains the upper 32 bits of the 64 bit Descriptor base address.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, RDBAH[0] may also be accessed at its alias offset of 0x00114.

12.0.3.4.9 Receive Descriptor Length queue- RDLEN (0x02808 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
6:0	RO	0	**Reserved.** Ignore on write. Reads back as 0.
19:7	RW	0	**Descriptor Length (LEN)**
31:20	RO	0	**Reserved.** Reads as 0. Should be written to 0 for future compatibility.

This register sets the number of bytes allocated for descriptors in the circular descriptor buffer. It must be 128B aligned.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, RDLEN[0] may also be accessed at its alias offset of 0x00118.

Note: The descriptor ring must be equal to or larger then 8 descriptors.

12.0.3.4.10 Receive Descriptor Head queue - RDH (0x02810 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
15:0	RW/V	0	**Receive Descriptor Head** (RDH).
31:16	RO	0	**Reserved**. Should be written with 0

This register contains the head pointer for the receive descriptor buffer. The register points to a 16B datum. Hardware controls the pointer. The only time that software should write to this register is after a reset (hardware reset or CTRL.SWRST) and before enabling the receive function (RCTL.EN). If software were to write to this register while the receive function was enabled, the on-chip descriptor buffers may be invalidated and the hardware could be become confused.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, RDH[0] may also be accessed at its alias offset of 0x00120.

12.0.3.4.11 Receive Descriptor Tail queue - RDT (0x02818 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
15:0	RW	0	**Receive Descriptor Tail** (RDT).
31:16	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register contains the tail pointer for the receive descriptor buffer. The register points to a 16B datum. Software writes the tail register to add receive descriptors for the hardware to process.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, RDT[0] may also be accessed at its alias offset of 0x00128.

12.0.3.4.12 Interrupt Delay Timer (Packet Timer) - RDTR (0x02820 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
15:0	RW	0	**Receive Delay Timer.** Receive Packet delay timer measured in increments of 1.024 microseconds.
30:16	RO	0	**Reserved**. Reads as 0.
31	WO	0	**Flush Partial Descriptor Block (FPD).** when set to 1; ignored otherwise. Reads 0.

This register is used to delay interrupt notification for the receive descriptor ring by coalescing interrupts for multiple received packets. Delaying interrupt notification helps maximize the number of receive packets serviced by a single interrupt.

This feature operates by initiating a countdown timer upon successfully receiving each packet to system memory. If a subsequent packet is received *before* the timer expires, the timer is re-initialized to the programmed value and re-starts its countdown. If the timer expires due to *not* having received a subsequent packet within the programmed interval, pending receive descriptor write backs are flushed and a receive timer interrupt is generated.

Setting the value to 0 represents no delay from a receive packet to the interrupt notification, and results in immediate interrupt notification for each received packet.

Writing this register with FPD set initiates an immediate expiration of the timer, causing a write back of any consumed receive descriptors pending write back, and results in a receive timer interrupt in the ICR.

Receive interrupts due to a Receive Absolute Timer (RADV) expiration will cancel a pending RDTR interrupt. The RDTR countdown timer is reloaded but halted, so as to avoid generation of a spurious second interrupt after the RADV has been noted, but may be restarted by a subsequent received packet.

Note: FPD is self-clearing.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x00108.

12.0.3.4.13 Receive Descriptor Control - RXDCTL (0x02828 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
5:0	RW	0x00	**Prefetch Threshold** (PTHRESH).
7:6	RO	0x00	**Reserved**.
13:8	RW	0x00	**Host Threshold** (HTHRESH).
14	RW	0	**Reserved**.
15	RW	0	**Reserved**.
21:16	RW	0x01	**Write-back Threshold** (WTHRESH).
23:22	RO	0x00	**Reserved**.
24	RW	0x0	**Granularity (GRAN).** Units for the thresholds in this register. 0 – cache lines 1 – descriptors
31:25	RO	0x00	**Reserved**.

Note: This register was not fully validated. SW should set it to 0x0000 during nominal operation.

This register controls the fetching and write-back of receive descriptors. The three threshold values are used to determine when descriptors will be read from and written to host memory. The values may be in units of cache lines or descriptors (each descriptor is 16 bytes) based on the GRAN flag. If GRAN=0 (specifications are in cache-line granularity), the thresholds specified (based on the cache line size specified in the PCI configuration space CLS field) must not represent greater than 31 descriptors.

Note: When (WTHRESH = 0) or (WTHRESH = 1 and GRAN = 1) only descriptors with the 'RS' bit set will be written back.

PTHRESH is used to control when a prefetch of descriptors will be considered. This threshold refers to the number of valid, unprocessed receive descriptors the chip has in its on-chip buffer. If this number drops below PTHRESH, the algorithm will consider pre-fetching descriptors from host memory. This fetch will not happen however unless there are at least HTHRESH valid descriptors in host memory to fetch.

Note: HTHRESH should be given a non zero value when ever PTHRESH is used.

WTHRESH controls the write-back of processed receive descriptors. This threshold refers to the number of receive descriptors in the on-chip buffer which are ready to be written back to host memory. In the absence of external events (explicit flushes), the write-back will occur only after at least WTHRESH descriptors are available for write-back.

Note: Possible values:

GRAN = 1 (descriptor granularity):

PTHRESH = 0...31

WTHRESH = 0...31

HTHRESH = 0...31

GRAN = 0 (cache line granularity):

PTHRESH = 0...3 (for 16 descriptors cache line - 256 bytes)

WTHRESH = 0...3

HTHRESH = 0...4

Note: For any WTHRESH value other than 0 - The packet and absolute timers must get a non zero value for WTHRESH feature to take affect.

Note: Since the default value for write-back threshold is 1, the descriptors are normally written back as soon as one cache line is available. WTHRESH must contain a non-zero value to take advantage of the write-back bursting capabilities of LAN Controller.

Note: RXDCTL1 is only accessible when VMDq is enabled (MRQC.MRxQueue = 10).

12.0.3.4.14 Receive Interrupt Absolute Delay Timer- RADV (0x0282C; RW)

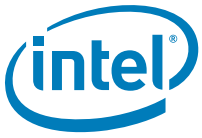
Bits	Type	Default	Description
15:0	RW	0	**Receive Absolute Delay Timer.** Receive Absolute delay timer measured in increments of 1.024 microseconds (0 = disabled)
31:16	RO	0	**Reserved**. Reads as 0

If the packet delay timer is used to coalesce receive interrupts, it will ensure that when receive traffic abates, an interrupt will be generated within a specified interval of no receives. During times when receive traffic is continuous, it may be necessary to ensure that no receive remains unnoticed for too long an interval. This register may be used to *ensure* that a receive interrupt occurs at some predefined interval after the first packet is received.

When this timer is enabled, a separate absolute countdown timer is initiated upon successfully receiving each packet to system memory. When this absolute timer expires, pending receive descriptor write backs are flushed and a receive timer interrupt is generated.

Setting this register to 0 disables the absolute timer mechanism (the RDTR register should be used with a value of 0 to cause immediate interrupts for all receive packets).

Receive interrupts due to a Receive Packet Timer (RDTR) expiration will cancel a pending RADV interrupt. If enabled, the RADV countdown timer is reloaded but halted, so as to avoid generation of a spurious second interrupt after the RDTR has been noted.



12.0.3.4.15 Receive Small Packet Detect Interrupt- RSRPD (0x02C00; RW)

Bits	Type	Default	Description
11:0	RW	0	**SIZE**. If the interrupt is enabled any receive packet of size <= SIZE will assert an Interrupt. SIZE is specified in bytes and includes the headers and the CRC. It does not include the VLAN header in size calculation if it is stripped.
31:12	RO	X	**Reserved**.

12.0.3.4.16 Receive ACK Interrupt Delay Register - RAID (0x02C08; RW)

Bits	Type	Default	Description
15:0	RW	0	**ACK_DELAY**. ACK delay timer measured in increments of 1.024 microseconds. When the Receive ACK frame detect Interrupt is enabled in the IMS register, ACK packets being received will use a unique delay timer to generate an interrupt. When an ACK is received, an absolute timer will load to the value of ACK_DELAY. The interrupt signal will be set only when the timer expires. If another ACK packet is received while the timer is counting down, the timer will not be reloaded to ACK_DELAY.
31:16	RO		**Reserved**.

If an immediate (non-scheduled) interrupt is desired for any received Ack frame, the ACK_DELAY should be set to 0.

12.0.3.4.17 CPU Vector Register - CPUVEC (0x02C10; RW)

Bits	Type	Default	Description
31:0	Res.	0x00	**Reserved**.

12.0.3.4.18 Receive Checksum Control - RXCSUM (0x05000; RW)

Bits	Type	Default	Description
7:0	RW	0x00	**Packet Checksum Start (PCSS)**.
8	RW	1	IP Checksum Offload Enable (IPOFL).
9	RW	1	TCP/UDP Checksum Offload Enable (TUOFL).
11:10	RO	0	Reserved.
12	RW	0	IP Payload Checksum Enable (IPPCSE).
13	RW	0	Packet Checksum Disable (PCSD).
14	RW	0	**Reserved**.
15	RW	0	**Reserved**.
31:16	RO	0	**Reserved.**

The Receive Checksum Control register controls the receive checksum off loading features of LAN Controller. LAN Controller supports the off loading of three receive checksum calculations: the Packet Checksum, the IP Header Checksum, and the TCP/UDP Checksum.

PCSD: The Packet Checksum and IP Identification fields are mutually exclusive with the RSS hash. Only one of the two options is reported in the Rx descriptor. The RXCSUM.PCSD affect is shown in the table below:

RXCSUM.PCSD	0 (Checksum Enable)	1 (Checksum Disable)
Legacy Rx Descriptor (RCTL.DTYP = 00b)	Packet Checksum is reported in the Rx Descriptor	Forbidden Configuration
Extended or Header Split Rx Descriptor (RCTL.DTYP = 01b)	Packet Checksum and IP Identification are reported in the Rx Descriptor	RSS Hash value is reported in the Rx Descriptor

PCSS IPPCSE: The PCSS and the IPPCSE control the Packet Checksum calculation. As noted above, the packet checksum shares the same location as the RSS field. The Packet checksum is reported in the Receive descriptor when the RXCSUM.PCSD bit is cleared.

If RXCSUM.IPPCSE cleared (the default value), the checksum calculation that is reported in the Rx Packet checksum field is the unadjusted “16 bit ones complement” of the packet. The Packet Checksum starts from the byte indicated by RXCSUM.PCSS (0 corresponds to the first byte of the packet), after VLAN stripping if enabled (by CTRL.VME). For example, for an Ethernet II frame encapsulated as an 802.3ac VLAN packet and with RXCSUM.PCSS set to 14, the packet checksum would include the entire encapsulated frame, excluding the 14-byte Ethernet header (DA, SA, Type/Length) and the 4-byte VLAN tag. The Packet Checksum will not include the Ethernet CRC if the RCTL.SECRC bit is set. Software must make the required offsetting computation (to back out the bytes that should not have been included and to include the pseudo-header) prior to comparing the Packet Checksum against the TCP checksum stored in the packet.

If the RXCSUM.IPPCSE is set, the Packet checksum is aimed to accelerate checksum calculation of fragmented UDP packets.

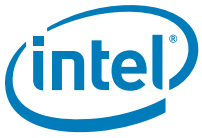
Note: The PCSS value should not exceed a pointer to IP header start or else it will erroneously calculate IP header checksum or TCP/UDP checksum.

RXCSUM.IPOFLD is used to enable the IP Checksum off-loading feature. If RXCSUM.IPOFLD is set to one, LAN Controller will calculate the IP checksum and indicate a pass/fail indication to software via the IP Checksum Error bit (IPE) in the ERROR field of the receive descriptor. Similarly, if RXCSUM.TUOFLD is set to one, LAN Controller will calculate the TCP or UDP checksum and indicate a pass/fail indication to software via the TCP/UDP Checksum Error bit (TCPE(. Similarly, if RFCTL.IPv6_DIS and RFCTL.IP6Xsum_DIS are cleared to zero and RXCSUM.TUOFLD is set to one, LAN Controller will calculate the TCP or UDP checksum for IPv6 packets. It then indicates a pass/fail condition in the TCP/UDP Checksum Error bit (RDESC.TCPE).

This applies to checksum off loading only. Supported Frame Types:

- Ethernet II
- Ethernet SNAP

This register should only be initialized (written) when the receiver is not enabled (e.g. only write this register when RCTL.EN = 0).



12.0.3.4.19 Receive Filter Control Register - RFCTL (0x05008; RW)

Bits	Type	Default	Description
0	RW	0	**iSCSI Disable (ISCSI_DIS).** Disable the ISCSI filtering for header split functionality.
1:5	RW	0	**iSCSI DWord count (ISCSI_DWC).** This field indicated the Dword count of the iSCSI header, which is used for packet split mechanism.
6	RW	0	**NFS Write disable (NFSW_DIS).** Disable filtering of NFS write request headers for header split functionality.
7	RW	0	**NFS Read disable (NFSR_DIS)**. Disable filtering of NFS read reply headers for header split functionality.
9:8	RW	00	**NFS Version** (NFS_VER). 00 – NFS version 2 01 – NFS version 3 10 – NFS version 4 11 – Reserved for future use
10	RW	0	**Reserved**.
11	RW	0	**Reserved**.
12	RW	0	**ACK accelerate disable (ACKDIS).** When this bit is set LAN Controller will not accelerate interrupt on TCP ACK packets.
13	RW	0	**ACK data Disable** (ACKD_DIS). 1 – LAN Controller will recognize ACK packets according to the ACK bit in the TCP header + No –CP data 0 – LAN Controller will recognize ACK packets according to the ACK bit only. This bit is relevant only if the ACKDIS bit is not set.
14	RW	0	**IP Fragment Split Disable (IPFRSP_DIS).** When this bit is set the header of IP fragmented packets will not be set.
15	RW	0	**Extended status Enable (EXSTEN).** When the EXSTEN bit is set or when the Packet Split receive descriptor is used, LAN Controller writes the extended status to the Rx descriptor.
16	RO	0	**Reserved**.
17	RO	0	**Reserved**.
31:18	RO	0	**Reserved**. Should be written with 0 to ensure future compatibility.

12.0.3.4.20 Multicast Table Array - MTA[31:0] (0x05200-0x0527C; RW)

Bits	Type	Default	Description
31:0	RW	X	**Bit Vector**. Word wide bit vector specifying 32 bits in the multicast address filter table.

There is one register per 32 bits of the Multicast Address Table for a total of 32 registers (thus the MTA[31:0] designation). The size of the word array depends on the number of bits implemented in the multicast address table. Software must mask to the desired bit on reads and supply a 32-bit word on writes.

Note: All accesses to this table must be 32-bit.

Note: These registers' addresses have been moved from where they were located in the 82542. However, for 82542 compatibility, these registers may also be accessed at their alias offsets of 0x00200-0x0027C.

The figure below diagrams the multicast lookup algorithm. The destination address shown represents the internally stored ordering of the received DA. Note that Byte 1 bit 0 indicated in this diagram is the first on the wire. The bits that are directed to the multicast table array in this diagram match a Multicast offset in the CTRL equals 00b. The complete multicast offset options are:

Multicast Offset	Bits directed to the Multicast Table Array
00b	DA[47:38] = Byte 6 bits 7:0, Byte 5 bits 7:6
01b	DA[46:37] = Byte 6 bits 6:0, Byte 5 bits 7:5
10b	DA[45:36] = Byte 6 bits 5:0, Byte 5 bits 7:4
11b	DA[43:34] = Byte 6 bits 3:0, Byte 5 bits 7:2

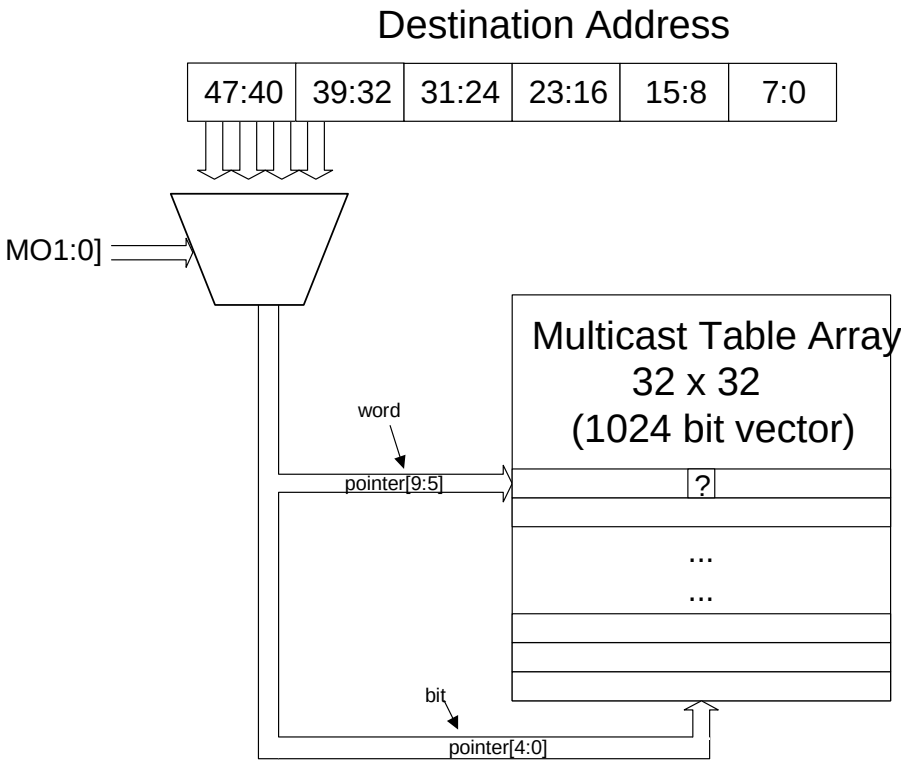
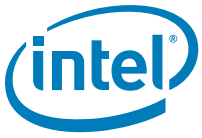


Figure 12-1. Multicast Table Array Algorithm

12.0.3.4.21 Receive Address Low - RAL (0x05400 + 8*n (n=0…6); RW)

While “n” is the exact unicast/Multicast address entry and it is equals to 0,1,…6



Bits	Type	Default	Description
31:0	RW	X	**Receive Address Low (RAL)**. The lower 32 bits of the 48 bit Ethernet address n (n=0, 1…6). RAL 0 is loaded from words 0 and 1 in the NVM.

Note: These registers' addresses have been moved from where they were located in the 82542. However, for 82542 compatibility, these registers may also be accessed at their alias offsets of 0x0040-0x000BC.

12.0.3.4.22 Receive Address High - RAH (0x05404 + 8*n (n=0…6); RW)

While "n" is the exact unicast/Multicast address entry and it is equals to 0,1,…6

Bits	Type	Default	Description
15:0	RW	X	**Receive Address High (RAH)**. The upper 16 bits of the 48 bit Ethernet address n (n=0, 1…6). RAH 0 is loaded from word 2 in the NVM.
17:16	RW	X	**Address Select (ASEL)**. Selects how the address is to be used. Decoded as follows: 00 – Destination address (must be set to this in normal mode) 01 – Source address 10 – Reserved 11 – Reserved
18	RW	0	**VMDq output index (VIND)** – defines the VMDq output index associated with a receive packet that matches this MAC address (RAH and RAL).
30:19	RO	0	**Reserved**. Reads as 0. Ignored on write.
31	RW	see description	**Address valid (AV)**. Cleared after master reset. If the NVM is present, the *Address Valid* field of *Receive Address Register 0* will be set to 1 after a software or PCI reset or NVM read. This bit is cleared by master (software) reset.

AV determines whether this address is compared against the incoming packet. AV is cleared by a master (software) reset.

ASEL enables the device to perform special filtering on receive packets.

Note: The first receive address register (RAR0) is also used for exact match pause frame checking (DA matches the first register). Therefore RAR0 should always be used to store the individual Ethernet MAC address of the adapter.

Note: These registers' addresses have been moved from where they were located in the 82542. However, for 82542 compatibility, these registers may also be accessed at their alias offsets of 0x0040-0x000BC.

After reset, if the NVM is present, the first register (Receive Address Register 0) will be loaded from the IA field in the NVM, its *Address Select* field will be 00, and its *Address Valid* field will be 1. If no NVM is present the *Address Valid* field will be 0. The *Address Valid* field for all of the other registers will be 0.

12.0.3.4.23 Shared Receive Address Low - SHRAL[n] (0x05438 + 8*n (n=0…3); RW)

Bits	Type	Default	Description
31:0	RW	X	**Receive Address Low (RAL)**. The lower 32 bits of the 48 bit Ethernet address n (n=0…3).

These registers may be WR locked by the LockMAC field in the FWSM register.

12.0.3.4.24 Shared Receive Address High 0…2 - SHRAH[n] (0x0543C + 8*n (n=0…2); RW)

Bits	Type	Default	Description
15:0	RW	X	**Receive Address High (RAH)**. The upper 16 bits of the 48 bit Ethernet address n (n=0…3).
17:16	RO	00	**Address Select (ASEL)**. Selects how the address is to be used. 00b means that it is used to decode the Destination MAC address.
18	RW	0	**VMDq output index (VIND)** – defines the VMDq output index associated with a receive packet that matches this MAC address (RAH and RAL).
30:19	RO	0	**Reserved**. Reads as 0. Ignored on write.
31	RW	0	**Address valid (AV)**. When this bit is set the relevant RAL,RAH are valid (compared against the incoming packet). Init trigger of this field depends on the state of the LockMAC state (see below).

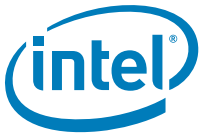
These registers may be WR locked by the LockMAC field in the FWSM register.

The description of SHRAH[9] is different and is described in Section 12.0.3.4.25.

12.0.3.4.25 Shared Receive Address High 3 - SHRAH[3] (0x05454; RW)

Bits	Type	Default	Description
15:0	RW	X	**Receive Address High (RAH)**. The upper 16 bits of the 48 bit Ethernet address n (n=0…3).
17:16	RO	00	**Address Select (ASEL)**. Selects how the address is to be used. 00b means that it is used to decode the Destination MAC address.
18	RW	0	**VMDq output index (VIND)** – defines the VMDq output index associated with a receive packet that matches this MAC address (RAH and RAL).
29:19	RO	0	**Reserved**. Reads as 0. Ignored on write.
30	RW	0	**All Nodes Multicast Address valid (MAV)**. The All Nodes Multicast address (33:33:00:00:00:01) is valid when this bit is set. Init trigger of this bit depends on the state of the LockMAC state (see below). Note that 0x33 is the first byte on the wire.
31	RW	0	**Address valid (AV)**. When this bit is set the relevant Address 3 is valid (compared against the incoming packet). Init trigger of this field depends on the state of the LockMAC state (see below).

These registers may be WR locked by the LockMAC field in the FWSM register.



12.0.3.4.26 Receive Address Initial CRC Calculation - RAICC[n] (0x05F50 + 4*n (n=0…10); RW)

Bits	Type	Reset	Description
31:0	RW	0x0	CRC32 calculation of the relevant configured RA address

12.0.3.4.27 Multiple Receive Queues Command register - MRQC (0x05818; RW)

Bits	Type	Default	Description
1:0	RW	0x00	**Multiple Receive Queues Enable** (**MRxQueue**). Enables support for Multiple Receive Queues and defines the mechanism that controls queue allocation. This field can be modified only when receive to host is not enabled (RCTL.EN = 0). 00b – Multiple Receive Queues are disabled 01b – Multiple Receive Queues as defined by MSFT RSS. The RSS Field Enable bits define the header fields used by the hash function. 10b – VMDq enable, enables VMDq operation as defined in section Receive Queuing for Virtual Machine Devices 11b – Reserved
2	Reserved.	0x0	**Reserved**.
15:3	RO	0x0	**Reserved**.
21:16	RW	0x0	**Reserved.**
31:22	RO	0x0	**Reserved**.

12.0.3.4.28 RSS Interrupt Mask Register - RSSIM (0x05864; RW)

Bits	Type	Default	Description
31:0	Reserved.	0x00	**Reserved**.

12.0.3.4.29 RSS Interrupt Request Register - RSSIR (0x5868; RW)

Bits	Type	Default	Description
31:0	Reserved.	0x00	Reserved.

12.0.3.4.30 Redirection Table - RETA (0x05C00 + 4*n (n=0…31); RW)

The redirection table is a 32 entry table. Each entry is composed of 4 Tags each 8-bits wide. Only the first or last 6 bits of each Tag are used (5 bits for the CPU index and 1 bit for Queue index).

Offset	31:24	23:16	15:8	7:0
0x05C00 + n*4	Tag 4*n+3	Tag 4*n+2	Tag 4*n+1	Tag 4*n

Bits	Type	Default	Description
4:0	RW	X	**CPU INDX 0**. CPU index for Tag 4*n (n=0,1,…31)
6:5	RO	X	**Reserved**.
7	RW	X	**QUE INDX 0.** Queue Index for Tag 4*n (n=0,1,…31)

Bits	Type	Default	Description
12:8	RW	X	**CPU INDX 1**. CPU index for Tag 4*n+1 (n=0,1,...31)
14:13	RO	X	**Reserved**.
15	RW	X	**QUE INDX 1.** Queue Index for Tag 4*n+1 (n=0,1,...31)
20:16	RW	X	**CPU INDX 2**. CPU index for Tag 4*n+2 (n=0,1,...31)
22:21	RO	X	**Reserved**.
23	RW	X	**QUE INDX 2.** Queue Index for Tag 4*n+2 (n=0,1,...31)
28:24	RW	X	**CPU INDX 3.** CPU index for Tag 4*n+3 (n=0,1,...31)
30:29	RO	X	**Reserved**.
31	RW	X	**QUE INDX 3**. Queue Index for Tag 4*n+3 (n=0,1,...31)

Note: RETA cannot be read when RSS is enabled.

12.0.3.4.31 Random Key Register - RSSRK (0x05C80 + 4*n (n=0...9); RW)

The RSS Random Key Register stores a 40 byte key (10 Dword entry table) used by the RSS hash function.

Bits	Type	Default	Description
7:0	RW	0x00	**K0.** Byte n*4 of the RSS random key (n=0,1,...9)
15:8	RW	0x00	**K1.** Byte n*4+1 of the RSS random key (n=0,1,...9)
23:16	RW	0x00	**K2.** Byte n*4+2 of the RSS random key (n=0,1,...9)
31:24	RW	0x00	**K3.** Byte n*4+3 of the RSS random key (n=0,1,...9)

12.0.3.5 Transmit Register Descriptions

12.0.3.5.1 Transmit Control Register - TCTL (0x00400; RW)

Bits	Type	Default	Description
0	RW	0	**IP Identification 15 bit** (IPID15). When '1', the IP Identification field will be incremented and wrapped around on 15-bit base. For example, if IP ID is equal to 0x7FFF then the next value will be 0x0000; if IP ID is equal to 0xFFFF then the next value will be 0x8000. When '0', the IP Identification field will be incremented and wrapped around on 16-bit base. In this case, the value following 0x7FFF is 0x8000, and the value following 0xFFFF is 0x0000. The purpose of this feature is to enable the software to manage 2 subgroups of connections.
1	RW	0	**Enable (EN)**. The transmitter is enabled when this bit is 1. Writing this bit to 0 will stop transmission after any in progress packets are sent. Data remains in the transmit FIFO until the device is re-enabled. Software should combine this with reset if the packets in the FIFO should be flushed.
2	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

Bits	Type	Default	Description
3	RW	1	**Pad short packets (PSP).** (with valid data, NOT padding symbols). 0 – do not pad; 1 – pad. Padding makes the packet 64B. This is not the same as the minimum collision distance. If Padding of short packets is allowed, the value in TX descriptor length field should be not less than 17 bytes.
11:4	RW	0x0F	**Collision Threshold (CT)**. This determines the number of attempts at retransmission prior to giving up on the packet (not including the first transmission attempt). While this can be varied, it should be set to a value of 15 in order to comply with the IEEE specification requiring a total of 16 attempts. The Ethernet back-off algorithm is implemented and clamps to the maximum number of slot-times after 10 retries. This field only has meaning when in half-duplex operation.
21:12	RW	0x3F	**Collision Distance (COLD)**. Specifies the minimum number of byte times which must elapse for proper CSMA/CD operation. Packets are padded with special symbols, not valid data bytes. Hardware checks and pads to this value plus one byte even in full-duplex operation. Default value is 64B – 512B times.
22	RW/V	0	**Software XOFF Transmission (SWXOFF)**. When set to a 1 the device will schedule the transmission of an XOFF (PAUSE) frame using the current value of the PAUSE timer. This bit self clears upon transmission of the XOFF frame.
23	RW	0	**Reserved**.
24	RW	0	**Re-transmit on Late Collision (RTLC)**. Enables the device to retransmit on a late collision event.
27:25	RW	0x0	**Reserved.** Used to be UNORTX and TXDSCMT in predecessors.
28	RO	1	**Reserved.**
30:29	RW	01	**Read Request Threshold (RRTHRESH).** These bits will define the threshold size for the intermediate buffer to determine when to send the read command to the Packet buffer. Threshold is defined as follow: RRTHRESH – 00b Threshold – 2 lines of 16 bytes RRTHRESH – 01b Threshold – 4 lines of 16 bytes RRTHRESH – 10b Threshold – 8 lines of 16 bytes RRTHRESH – 11b Threshold – No threshold (transfer data after all of the request is in the RFIFO)
31	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

Two fields deserve special mention: CT and COLD. Software may choose to abort packet transmission in less than the Ethernet mandated 16 collisions. For this reason, hardware provides CT.

Wire speeds of 1000 Mbps result in a very short collision radius with traditional minimum packet sizes. COLD specifies the minimum number of bytes in the packet to satisfy the desired collision distance. It is important to note that the resulting packet has special characters appended to the end. These are NOT regular data characters. Hardware strips special characters for packets that go from 1000 Mbps environments to 100 Mbps environments. Note that the hardware evaluates this field against the packet size in Full Duplex as well.

Note: While 802.3x flow control is only defined during full duplex operation, the sending of PAUSE frames via the SWXOFF bit is not gated by the duplex settings within the device.

Software should not write a 1 to this bit while the device is configured for half duplex operation.

RTLC configures the LAN Controller to perform retransmission of packets when a late collision is detected. Note that the collision window is speed dependent: 64 bytes for 10/100 Mb/s and 512 bytes for 1000Mb/s operation. If a late collision is detected when this bit is disabled, the transmit function assumes the packet is successfully transmitted. This bit is ignored in full-duplex mode.

12.0.3.5.2 Transmit IPG Register - TIPG (0x00410; RW)

Bits	Type	Default	Description
9:0	RW	0x8	**IPG Transmit Time (IPGT)**. Specifies the IPG length for back-to-back transmissions equal to [(IPGT+4) x 8] bit time.
19:10	RW	0x8	**IPG Receive Time 1 (IPGR1)**. Specifies the defer IPG part 1 (during which carrier sense is monitored). Equal to (IPGR1 x 8) when DJHDX=0 and equals to (IPGR1+2) x 8 when DJHDX=1.
29:20	RW	0x9	**IPG Receive Time 2 (IPGR2)**. Specifies the defer IPG. Equal to (IPGR2+3) x 8 when DJHDX=0 and equal to (IPGR2+5) x 8 when DJHDX=1.
31:30	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register controls the IPG (Inter Packet Gap) timer. IPGT specifies the IPG length for back-to-back transmissions in both full and half duplex. Note that an offset of 4 byte times is added to the programmed value to determine the total IPG. Therefore, a value of 8 is recommended to achieve a 12 byte time IPG.

IPGR1 specifies the portion of the IPG in which the transmitter will defer to receive events. This should be set to 2/3 of the total effective IPG, or 8.

IPGR specifies the total IPG time for non back-to-back transmissions (transmission following deferral) in half duplex.

An offset of 5 byte times is added to the programmed value to determine the total IPG after a defer event. Therefore, a value of 7 is recommended to achieve a 12 byte time effective IPG for this case. Note the IPGR should never be set to a value greater than IPGT. If IPGR is set to a value equal to or larger than IPGT, it will override the IPGT IPG setting in half duplex, resulting in inter packet gaps that are larger than intended by IPGT in that case. Full Duplex will be unaffected by this, and will always rely on IPGT only.

In summary, the recommended TIPG value to achieve 802.3 compliant minimum transmit IPG values in full and half duplex is 0x00702008.

12.0.3.5.3 Adaptive IFS Throttle - AIT (0x00458; RW)

Bits	Type	Default	Description
15:0	RW	0x0000	**Adaptive IFS value (AIFS)**. This value is in units of 8 nanoseconds.
31:16	RO	0x0000	**Reserved**. This field should be written with 0.

Adaptive IFS throttles back-to-back transmissions in the transmit packet buffer and delays their transfer to the CSMA/CD transmit function, and thus can be used to delay the transmission of back-to-back packets on the wire. Normally, this register should be set to 0. However, if additional delay is desired between back-to-back transmits, then this register may be set with a value greater than zero.

The Adaptive IFS field provides a similar function to the IPGT field in the TIPG register (see Section 12.0.3.6.2). However, it only affects the initial transmission timing, not re-transmission timing.

Note: If the value of the AdaptiveIFS field is less than the IPGTransmitTime field in the Transmit IPG registers then it will have no effect, as the chip will select the maximum of the two values.

12.0.3.5.4 Transmit Descriptor Base Address Low - TDBAL (0x03800 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
3:0	RO	0	**Reserved**. Ignored on writes. Returns 0 on reads
31:4	RW	X	**Transmit Descriptor Base Address Low** (TDBAL)

This register contains the lower bits of the 64 bit descriptor base address. The lower 4 bits are ignored. The Transmit Descriptor Base Address must point to a 16B aligned block of data.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, TDBAL[0] may also be accessed at its alias offset of 0x00420.

12.0.3.5.5 Transmit Descriptor Base Address High - TDBAH (0x03804 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
31:0	RW	X	**Transmit Descriptor Base Address** [63:32] (TDBAH).

This register contains the upper 32 bits of the 64 bit Descriptor base address.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, TDBAH[0] may also be accessed at its alias offset of 0x00424.

12.0.3.5.6 Transmit Descriptor Length - TDLEN (0x03808 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
6:0	RO	0	**Reserved**. Ignore on write. Reads back as 0.
19:7	RW	0	**Descriptor Length (LEN)**.
31:20	RO	0	**Reserved**. Reads as 0. Should be written to 0.

This register contains the descriptor length and must be 128B aligned.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, TDLEN[0] may also be accessed at its alias offset of 0x00428.

Note: The descriptor ring must be equal to or larger than 8 descriptors.

12.0.3.5.7 Transmit Descriptor Head - TDH (0x03810 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
15:0	RW/V	0	**Transmit Descriptor Head (TDH)**.
31:16	RO	0	**Reserved**. Should be written with 0.

This register contains the head pointer for the transmit descriptor ring. It points to a 16B datum. Hardware controls this pointer. The only time that software should write to this register is after a reset (hardware reset or CTRL.SWRST) and before enabling the transmit function (TCTL.EN). If software were to write to this register while the transmit function was enabled, the on-chip descriptor buffers may be invalidated and the hardware could be become confused.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, TDH[0] may also be accessed at its alias offset of 0x00430.

12.0.3.5.8 Transmit Descriptor Tail - TDT (0x03818 + n*0x100[n=0..1]; RW)

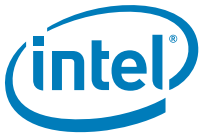
Bits	Type	Default	Description
15:0	RW	0	**Transmit Descriptor Tail (TDT)**.
31:16	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

Note: This register contains the tail pointer for the transmit descriptor ring. It points to a 16B datum. Software writes the tail pointer to add more descriptors to the transmit ready queue. Hardware attempts to transmit all packets referenced by descriptors between head and tail.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, TDT[0] may also be accessed at its alias offset of 0x00438.

12.0.3.5.9 Transmit Arbitration Count - TARC (0x03840 + n*0x100[n=0..1]; RW)

Bits	Type	Default	Description
6:0	RW	3	**Transmit Arbitration Count (COUNT).** number of packets that can be sent from queue 0 to make the N over M arbitration between the queues. Writing 0 to this register is forbidden.
7	RW	0	**Compensation mode (COMP).** when set to 1 the HW will compensate this queue according to the compensation ratio, if the number of packets in a TCP segmentation in queue 1 caused the counter in queue 1 to go below zero
9:8	RW	00	**Compensation Ratio (RATIO).** this value will determine the ratio between the number of packets transmitted on queue1 in a TCP segmentation offload to the number of compensated packets transmitted from queue 0 00 – 1/1 compensation ratio 01 – 1/2 compensation ratio 10 – 1/4 compensation ratio 11 – 1/8 compensation ratio



Bits	Type	Default	Description
10	RW	1	**Descriptor enable (ENABLE).** The ENABLE bit of the Transmit queue 0 should always be set.
26:11	RO	0	**Reserved.** Reads as 0. Should be written to 0 for future compatibility.
27	RW	0	**Reserved.** Reserved for Multiple Tx request disable. This bit should not be modified by SW.
31:28	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

The default hardware value for TARC0.COUNT is **3** (this value is also reflected after reset).

The counter is subtracted as a part of the transmit arbitration.

It is reloaded to its high (last written) value when it decreases below zero.

- Upon read, the hardware returns the current counter value.
- Upon write, the counter will update the high value in the **next** counter-reload.
- The counter may be decreased in chunks (when transmitting TCP segmentation packets). It should never roll because of that.
  The size of chunks is determined according to the TCP segmentation (number of packets sent).

When the counter reaches zero, other TX queues should be selected for transmission as soon as possible (usually after current transmission).

COMP is the enable bit to compensate between the two queues, when enabled (set to 1) the HW will compensate between the two queues if one of the queues is transmitting TCP segmentation packets and its counter went below zero, the HW will compensate the other queue according to the ratio in the TARC1.RATIO.

For example if the TARC0.COUNT reached (-5) after sending TCP segmentation packets and both TARC0.COMP and TARC1.COMP are enabled (set to 1) and TARC1.RATIO is 01 (1/2 compensation) TARC1.COUNT will be adjusted by adding 5/2=2 to the current count.

RATIO is the multiplier to compensate between the two queues. The compensation method is described in the explanation above.

For DHG 802.3p using qWAVE API the following configuration will be used:

TARC0: COUNT = 1, COMP = 0, RATIO = 00.

TARC1: COUNT = 4, COMP = 1, RATIO = 00.

12.0.3.5.10 Transmit Interrupt Delay Value - TIDV (0x03820; RW)

Bits	Type	Default	Description
15:0	RW	0	**Interrupt Delay Value (IDV)**. Counts in units of 1.024 microseconds. A value of 0 is not allowed.
30:16	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.
31	WO	0	**Flush Partial Descriptor Block (FPD).** when set to 1; ignored otherwise. Reads 0.

This register is used to delay interrupt notification for transmit operations by coalescing interrupts for multiple transmitted buffers. Delaying interrupt notification helps maximize the amount of transmit buffers reclaimed by a single interrupt. This feature *only* applies to transmit descriptor operations where (a) interrupt-based reporting is requested (RS set) and (b) the use of the timer function is requested (IDE is set).

This feature operates by initiating a countdown timer upon successfully transmitting the buffer. If a subsequent transmit delayed-interrupt is scheduled *before* the timer expires, the timer is re-initialized to the programmed value and re-starts its countdown. When the timer expires, a transmit-complete interrupt (ICR.TXDW) is generated.

Setting the value to 0 is not allowed. If an immediate (non-scheduled) interrupt is desired for any transmit descriptor, the descriptor IDE should be set to 0.

The occurrence of either an immediate (non-scheduled) or absolute transmit timer interrupt will halt the TIDV timer and eliminate any spurious second interrupts.

Transmit interrupts due to a Transmit Absolute Timer (TADV) expiration or an immediate interrupt (RS/RSP=1, IDE=0) will cancel a pending TIDV interrupt. The TIDV countdown timer is reloaded but halted, though it may be restarted by a processing a subsequent transmit descriptor.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x00440.

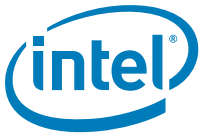
Writing this register with FPD set initiates an immediate expiration of the timer, causing a write back of any consumed transmit descriptors pending write back, and results in a transmit timer interrupt in the ICR.

Note: FPD is self-clearing.

12.0.3.5.11 Transmit Descriptor Control - TXDCTL (0x03828 + n*0x100[n=0..1]; RW)

Note: This register was not fully validated. SW should set it to 0x0000 during nominal operation.

Bits	Type	Default	Description
5:0	RW	0x00	**Prefetch Threshold** (PTHRESH).
7:6	RO	0x00	**Reserved**.
13:8	RW	0x00	**Host Threshold** (HTHRESH).
15:14	RO	0x00	**Reserved**.
21:16	RW	0x00	**Write-back Threshold** (WTHRESH).
23:22	RO	0x00	**Reserved**.
24	RW	0x0	**Granularity (GRAN)**. Units for the thresholds in this register. 0 – cache lines 1 – descriptors
31:25	RW	0x0	**Transmit descriptor Low Threshold** (LWTHRESH). Interrupt asserted when the number of descriptors pending service in the transmit descriptor queue (processing distance from the TDT) drops below this threshold.



This register controls the fetching and write-back of transmit descriptors. The three threshold values are used to determine when descriptors will be read from and written to host memory. The values may be in units of cache lines or descriptors (each descriptor is 16 bytes) based on the GRAN flag.

Note: When GRAN = 1 all descriptors will be written back (even if not requested).

PTHRESH is used to control when a prefetch of descriptors will be considered. This threshold refers to the number of valid, unprocessed transmit descriptors the chip has in its on-chip buffer. If this number drops below PTHRESH, the algorithm will consider pre-fetching descriptors from host memory. This fetch will not happen however, unless there are at least HTHRESH valid descriptors in host memory to fetch.

Note: HTHRESH should be given a non zero value when ever PTHRESH is used.

WTHRESH controls the write-back of processed transmit descriptors. This threshold refers to the number of transmit descriptors in the on-chip buffer which are ready to be written back to host memory. In the absence of external events (explicit flushes), the write-back will occur only after at least WTHRESH descriptors are available for write-back.

Possible values:

GRAN = 1 (descriptor granularity):

PTHRESH = 0..31

WTHRESH = 0..31

HTHRESH = 0..31

GRAN = 0 (cacheline granularity):

PTHRESH = 0..3 (for 16 descriptors cacheline - 256 bytes)

WTHRESH = 0..3

HTHRESH = 0..4

Note: For any WTHRESH value other than 0 - The packet and absolute timers must get a non zero value for the WTHRESH feature to take affect.

Note: Since the default value for write-back threshold is 0, descriptors are normally written back as soon as they are processed. WTHRESH must be written to a non-zero value to take advantage of the write-back bursting capabilities of the LAN Controller. If the WTHRESH is written to a non-zero value then all of the descriptors are written back consecutively no matter the setting of the RS bit.

Since write-back of transmit descriptors is optional (under the control of RS bit in the descriptor), not all processed descriptors are counted with respect to WTHRESH. Descriptors start accumulating after a descriptor with RS is set. Furthermore, with transmit descriptor bursting enabled, all of the descriptors are written back consecutively no matter the setting of the RS bit.

Note: Leaving this value at its default will cause descriptor processing to be similar to the 82542.

LWTHRESH controls the number of pre-fetched transmit descriptors at which a transmit descriptor-low interrupt (ICR.TXD_LOW) is reported. This may allow software to operate more efficiently by maintaining a continuous addition of transmit work, interrupting only when the hardware nears completion of all submitted work.

LWTHRESH specifies a multiple of 8 descriptors. An interrupt is asserted when the number of descriptors available transitions from (threshold level=8*LWTHRESH)+1 ◊ (threshold level=8*LWTHRESH). Setting this value to 0 will disable this feature.

12.0.3.5.12 Transmit Absolute Interrupt Delay Value-TADV (0x0382C; RW)

Bits	Type	Default	Description
15:0	RW	0	**Interrupt Delay Value (IDV)**. Counts in units of 1.024 microseconds. (0 – disabled)
31:16	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

The transmit interrupt delay timer (TIDV) may be used to coalesce transmit interrupts. However, it may be necessary to ensure that no completed transmit remains unnoticed for too long an interval in order ensure timely release of transmit buffers. This register may be used to *ensure* that a transmit interrupt occurs at some predefined interval after a transmit is completed. Like the delayed-transmit timer, the absolute transmit timer *only* applies to transmit descriptor operations where (a) interrupt-based reporting is requested (RS set) and (b) the use of the timer function is requested (IDE is set).

This feature operates by initiating a countdown timer upon successfully transmitting the buffer. When the timer expires, a transmit-complete interrupt (ICR.TXDW) is generated. The occurrence of either an immediate (non-scheduled) or delayed transmit timer (TIDV) expiration interrupt will halt the TADV timer and eliminate any spurious second interrupts.

Setting the value to 0 disables the transmit absolute delay function. If an immediate (non-scheduled) interrupt is desired for any transmit descriptor, the descriptor IDE should be set to 0.

12.0.3.6 Statistic Register Descriptions

Note: All Statistics registers reset when read. In addition, they stick at 0xFFFF_FFFF when the maximum value is reached.

Note: For the receive statistics it should be noted that a packet is indicated as "received" if it passes the device's filters and is placed into the packet buffer memory. A packet does not have to be DMA'd to host memory in order to be counted as "received".

Note: Due to divergent paths between interrupt-generation and logging of relevant statistics counts, it may be possible to generate an interrupt to the system for a noteworthy event prior to the associated statistics count actually being incremented. This is extremely unlikely due to expected delays associated with the system interrupt-collection and ISR delay, but might be observed as an interrupt for which statistics values do not quite make sense. Hardware guarantees that any event noteworthy of inclusion in a statistics count will be reflected in the appropriate count within 1 microsecond; a small time-delay prior to read of statistics may be necessary to avoid the potential for receiving an interrupt and observing an inconsistent statistics count as part of the ISR.

12.0.3.6.1 CRC Error Count - CRCERRS (0x04000; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**CEC.** CRC error count.

Counts the number of receive packets with CRC errors. In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusively) in length. If receives are not enabled, then this register will not increment.

12.0.3.6.2 RX Error Count - RXERRC (0x0400C; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**RXEC**. RX error count.

Counts the number of packets received in which RX_ER was asserted by the PHY. In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusively) in length. If receives are not enabled, then this register will not increment.

12.0.3.6.3 Missed Packets Count - MPC (0x04010; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**MPC**. Missed Packets Count.

Counts the number of missed packets. Packets are missed when the receive FIFO has insufficient space to store the incoming packet. This could be caused by too few buffers allocated, or because there is insufficient bandwidth on the IO bus. Events setting this counter cause RXO, the receiver overrun interrupt, to be set. This register does not increment if receives are not enabled.

Note: Note that these packets will also be counted in the Total Packets Received register as well as in Total Octets Received.

12.0.3.6.4 Carrier Extension Error Count - CEXTERR (0x0403C; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**CEXTERR**. Number of packets received with a carrier extension error.

This register counts the number of packets received in which the carrier extension error was signaled across the GMII interface. The PHY propagates carrier extension errors to the MAC when an error is detected during the carrier extended time of a packet reception. An extension error is signaled by the PHY by the encoding of 0x1F on the receive data inputs while RX_ER is asserted to the MAC. This register will only increment if receives are enabled and the device is operating at 1000Mb/s.

12.0.3.6.5 XON Received Count - XONRXC (0x04048; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**XONRXC**. Number of XON packets received.

This register counts the number of XON packets received. XON packets can use the global address, or the station address. This register will only increment if receives are enabled.

12.0.3.6.6 XON Transmitted Count - XONTXC (0x0404C; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**XONTXC**. Number of XON packets transmitted.

This register counts the number of XON packets transmitted. These can be either due to queue fullness, or due to software initiated action (using SWXOFF). This register will only increment if transmits are enabled.

12.0.3.6.7 XOFF Received Count - XOFFRXC (0x04050; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**XOFFRXC**. Number of XOFF packets received.

This register counts the number of XOFF packets received. XOFF packets can use the global address, or the station address. This register will only increment if receives are enabled.

12.0.3.6.8 XOFF Transmitted Count - XOFFTXC (0x04054; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**XOFFTXC**. Number of XOFF packets transmitted.

This register counts the number of XOFF packets transmitted. These can be either due to queue fullness, or due to software initiated action (using SWXOFF). This register will only increment if transmits are enabled.

12.0.3.6.9 FC Received Unsupported Count - FCRUC (0x04058; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**FCRUC**. Number of unsupported flow control frames received.

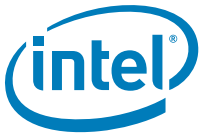
This register counts the number of unsupported flow control frames that are received.

The FCRUC counter is incremented when a flow control packet is received which matches either the reserved flow control multicast address (in FCAH/L) or the MAC station address, and has a matching flow control type field match (to the value in FCT), but has an incorrect opcode field. This register will only increment if receives are enabled.

12.0.3.6.10 Good Packets Received Count - GPRC (0x04074; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**GPRC**. Number of good packets received (of any legal length).

This register counts the number of good (non-erred) packets received of any legal length. The legal length for the received packet is defined by the value of LongPacketEnable (see 10.2.7.13 Receive Length Error Count). This register does not include received flow control packets and only counts packets that pass filtering. This register will only increment if receives are enabled. This register does not count packets counted by the *Missed Packet Count (MPC)* register.



12.0.3.6.11 Broadcast Packets Received Count - BPRC (0x04078; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**BPRC**. Number of broadcast packets received.

This register counts the number of good (non-erred) broadcast packets received. This register counts every broadcast packets received that passed filtering. This register does not count packets counted by the *Missed Packet Count (MPC)* register.

12.0.3.6.12 Multicast Packets Received Count - MPRC (0x0407C; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**MPRC**. Number of multicast packets received.

This register counts the number of good (non-erred) multicast packets received. This register does not count multicast packets received that fail to pass address filtering nor does it count received flow control packets. This register does not count packets counted by the *Missed Packet Count (MPC)* register.

12.0.3.6.13 Good Packets Transmitted Count - GPTC (0x04080; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**GPTC**. Number of good packets transmitted.

This register counts the number of good (non-erred) packets transmitted. A good transmit packet is considered one that is 64 or more bytes in length (from <Destination Address> through <CRC>, inclusively) in length. This does not include transmitted flow control packets. This register will only increment if transmits are enabled. This register does not count packets counted by the *Missed Packet Count (MPC)* register. The register counts clear as well as secure packets.

12.0.3.6.14 Good Octets Received Count - GORCL (0x04088; RO)

12.0.3.6.15 Good Octets Received Count - GORCH (0x0408C; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**GORCL**. Number of good octets received – lower 4 bytes.
31:0	RO/V	0	**GORCH**. Number of good octets received – upper 4 bytes.

These registers make up a logical 64-bit register which counts the number of good (non-erred) octets received. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusively. This register must be accessed using two independent 32-bit accesses. SW should first read the low register (GORCL) at this point the HW will snapshot the upper 32 bits to be read (GORCH).

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. Only packets that pass address filtering are counted in this register. This register will only increment if receives are enabled.

These octets do not include octets in received flow control packets.

12.0.3.6.16 Good Octets Transmitted Count - GOTCL (0x04090; RO);

12.0.3.6.17 Good Octets Transmitted Count - GOTCH (0x04094; RO);

Bits	Type	Default	Description
31:0	RO/V	0	**GOTCL**. Number of good octets transmitted – lower 4 bytes.
31:0	RO/V	0	**GOTCH**. Number of good octets transmitted – upper 4 bytes.

These registers make up a logical 64-bit register which counts the number of good (non-erred) packets transmitted. This register must be accessed using two independent 32-bit accesses. SW should first read the low register (GOTCL) at this point the HW will snapshot the upper 32 bits to be read (GOTCH).

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusively. This register counts octets in successfully transmitted packets which are 64 or more bytes in length. This register will only increment if transmits are enabled. The register counts clear as well as secure octets.

These octets do not include octets in transmitted flow control packets.

12.0.3.6.18 Receive No Buffers Count - RNBC (0x040A0; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**DefaultRNBC**. Number of receive no buffer conditions.

This register counts the number of times that frames were received when there were no available buffers in host memory to store those frames (receive descriptor head and tail pointers were equal). The packet will still be received if there is space in the FIFO. This register will only increment if receives are enabled.

This register does not increment when flow control packets are received.

12.0.3.6.19 Receive Undersize Count - RUC (0x040A4; RO)

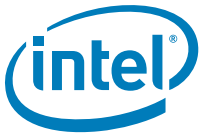
Bits	Type	Default	Description
31:0	RO/V	0	**RUC**. Number of receive undersize errors.

This register counts the number of received frames that passed address filtering, and were less than the minimum size (64 bytes from <Destination Address> through <CRC>, inclusively), and had a valid CRC. This register will only increment if receives are enabled.

12.0.3.6.20 Receive Fragment Count - RFC (0x040A8; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**RFC**. Number of receive fragment errors.

This register counts the number of received frames that passed address filtering, and were less than the minimum size (64 bytes from <Destination Address> through <CRC>, inclusively), but had a bad CRC (this is slightly different from the Receive Undersize Count register). This register will only increment if receives are enabled.



12.0.3.6.21 Receive Oversize Count - ROC (0x040AC; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**ROC**. Number of receive oversize errors.

This register counts the number of received frames that passed address filtering, and were greater than maximum size. Packets over 1522 bytes are oversized if LongPacketEnable is 0. If LongPacketEnable (LPE) is 1, then an incoming, packet is considered oversized if it exceeds 16384 bytes.

If receives are not enabled, this register will not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusively.

12.0.3.6.22 Receive Jabber Count - RJC (0x040B0; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**RJC**. Number of receive jabber errors.

This register counts the number of received frames that passed address filtering, and were greater than maximum size and had a bad CRC (this is slightly different from the Receive Oversize Count register).

Packets over 1522 bytes are oversized if LongPacketEnable is 0. If LongPacketEnable (LPE) is 1, then an incoming packet is considered oversized if it exceeds 16384 bytes.

If receives are not enabled, this register will not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusively.

12.0.3.6.23 Management Packets Received Count - MNGPRC (0x040B4; RO)

Bits	Type	Default	Description
15:0	RO/V	0	**Reserved**.
31:16	RO	0	**Reserved**.

This register counts the total number of packets received that pass the management filters or receive circuit breaker redirection. Any packets with errors are not counted, except for packets dropped because the management receive FIFO is full.

12.0.3.6.24 Management Packets Dropped Count - MNGPDC (0x040B8; RO)

Bits	Type	Default	Description
15:0	RO/V	0	**MPDC**. Number of management packets dropped.
31:16	RO	0	**Reserved**.

This register counts the total number of packets received that pass the management filters and then are dropped because the management receive FIFO is full.

12.0.3.6.25 Management Packets Transmitted Count - MNGPTC (0x040BC; RO)

Bits	Type	Default	Description
15:0	RO/V	0	**MPTC**. Number of management packets transmitted.
31:16	RO	0	**Reserved**.

This register counts the total number of packets that are transmitted that are received over the LAN Controller PCI-M interface.

12.0.3.6.26 Tx Circuit Breaker Packets Dropped -TCBPD (0x040D8; RO)

Bits	Type	Default	Description
15:0	RO/V	0	**TCBD**. Dropped redirected transmit CB packets. Packets are dropped due to lack of room in the redirection MNG FIFO. It could be either due to M-Link BW or no memory resources.
31:16	RO	0	**Reserved**.

12.0.3.6.27 Total Octets Received - TORL (0x040C0; RO);

12.0.3.6.28 Total Octets Received - TORH (0x040C4; RO);

Bits	Type	Default	Description
31:0	RO/V	0	**TORL**. Number of total octets received – lower 4 bytes.
31:0	RO/V	0	**TORH**. Number of total octets received – upper 4 bytes.

These registers make up a logical 64-bit register which count the total number of octets received. This register must be accessed using two independent 32-bit accesses. SW should first read the low register (TORL) at this point the HW will snapshot the upper 32 bits to be read (TORH). In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached.

All packets received that pass address filtering will have their octets summed into this register, regardless of their length, whether they are erred, or whether they are flow control packets. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusively. This register will only increment if receives are enabled.

Note: Broadcast rejected packets will be counted in this counter (in contradiction to all other rejected packets that are not counted).

12.0.3.6.29 Total Octets Transmitted - TOTL (0x040C8; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**TOTL.** Number of total octets transmitted – lower 4 bytes.

The TOTL and TOTH registers make up a logical 64-bit register which count the total number of octets transmitted. This register must be accessed using two independent 32-bit accesses. SW should first read the low register (TORL) at this point the HW will snapshot the upper 32 bits to be read (TOTH). In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached.

All transmitted packets will have their octets summed into this register, regardless of their length or whether they are flow control packets. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusively.

Octets transmitted as part of partial packet transmissions (e.g. collisions in half-duplex mode) are not included in this register. This register will only increment if transmits are enabled.

12.0.3.6.30 Total Octets Transmitted - TOTH (0x040CC; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**TOTH**. Number of total octets transmitted – upper 4 bytes

See explanation of the TOTL above.

12.0.3.6.31 Total Packets Received - TPR (0x040D0; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**TPR**. Number of all packets received.

This register counts the total number of all packets received. All packets received will be counted in this register, regardless of their length, whether they are erred, or whether they are flow control packets. This register will only increment if receives are enabled.

Note: Broadcast rejected packets will be counted in this counter (in contradiction to all other rejected packets that are not counted).

12.0.3.6.32 Total Packets Transmitted - TPT (0x040D4; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**TPT**. Number of all packets transmitted.

This register counts the total number of all packets transmitted. All packets transmitted will be counted in this register, regardless of their length, or whether they are flow control packets.

Partial packet transmissions (e.g. collisions in half-duplex mode) are not included in this register. This register will only increment if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMB, and packets generated by the ASF function.

12.0.3.6.33 Multicast Packets Transmitted Count - MPTC (0x040F0; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**MPTC**. Number of multicast packets transmitted.

This register counts the number of multicast packets transmitted. This register does not include flow control packets and increments only if transmits are enabled. Counts clear as well as secure traffic.

12.0.3.6.34 Broadcast Packets Transmitted Count - BPTC (0x040F4; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**BPTC**. Number of broadcast packets transmitted count.

This register counts the number of broadcast packets transmitted. This register will only increment if transmits are enabled. This register counts all packets, including standard and secure packets. (Management packets will never be more than 200 bytes).

12.0.3.6.35 TCP Segmentation Context Transmitted Count - TSCTC (0x040F8; RO)

Bits	Type	Default	Description
31:0	RO/V	0	**TSCTC**. Number of TCP Segmentation contexts transmitted count.

This register counts the number of TCP segmentation offload transmissions and increments once the last portion of the TCP segmentation context payload is segmented and loaded as a packet into the on-chip transmit buffer. Note that it is not a measurement of the number of packets sent out (covered by other registers). This register will only increment if transmits and TCP Segmentation offload are enabled.

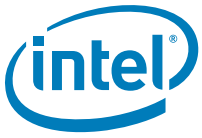
12.0.3.6.36 Interrupt Assertion Count - IAC (0x04100; RO)

Bits	Type	Default	Description
0-31	RO/V	0	**IAC**. This is a count of the interrupt assertions that have occurred. It counts the total number of interrupts generated in the system.

12.0.3.7 Management Register Descriptions

12.0.3.7.1 Wake Up Control Register - WUC (0x05800; RW)

Bits	Type	Default	Description
0	RW/ SN	0	**Advance Power Management Enable (APME).** 1 – APM Wakeup is enabled 0 – APM Wakeup is disabled Loaded from the NVM word 1Ah.
1	RW/V	0	**PME_En.** This read/write bit is used by the driver to access the PME_En bit of the *Power Management Control / Status Register* (PMCSR) without writing to PCI configuration space.
2	RWC	0	**PME_Status.** This bit is set when LAN Controller receives a wakeup event. It is the same as the PME_Status bit in the *Power Management Control* / Status Register (PMCSR). Writing a "1" to this bit will clear it, and also clear the PME_Status bit in the PMCSR.
3	RW	1	**Assert PME On APM Wakeup (APMPME).** If it is 1, LAN Controller will set the *PME_Status* bit in the *Power Management Control / Status Register* (PMCSR) and assert Host_Wake when APM Wakeup is enabled and LAN Controller receives a matching magic packet.
4	RW/ SN	0	**Link Status Change Wake Enable (LSCWE).** Enables wake on link status change as part of APM wake capabilities.



Bits	Type	Default	Description
5	RW/ SN	0	**Link Status Change Wake Override (LSCWO).** If "1", wake on Link Status Change does not depend on the LNKC bit in the Wake Up Filter Control Register (WUFC). Instead, it is determined by the APM settings in the WUC register.
6	RO	0	**Reserved.** Was "APM Flexible Filter Allocation (APMFFA)".
7	RO	0	**Reserved.** Was "Flexible APM filter Enable (FLEX_APM_FILTER_EN)".
8	RW/ SN	0	**Phy_Wake.** This bit indicates if the Phy connected to the LAN controller supports wakeup. This bit is loaded from NVM word 13h bit 8.
29:9	RO	0	**Reserved**. Reads as 0.
31:30	RO	0	**Reserved**.

The PME_Status bits are cleared in the following conditions:

- If there is VAUX then the PME Status bits should be cleared by:
  - — PWR Good
  - — Explicit Software Clear
- If there is NO VAUX then the PME Status bits should be cleared by:
  - — PWR Good
  - — PCI Reset de-assertion
  - — Explicit Software Clear

12.0.3.7.2 Wake Up Filter Control Register - WUFC (0x05808; RW)

Bits	Type	Default	Description
0	RW	0	**LNKC.** Link Status Change Wake Up Enable
1	RW	0	**MAG.** Magic Packet Wake Up Enable
2	RW	0	**EX.** Directed Exact Wake Up Enable
3	RW	0	**MC.** Directed Multicast Wake Up Enable
4	RW	0	**BC.** Broadcast Wake Up Enable
5	RW	0	**ARP.** ARP/IPv4 Request Packet Wake Up Enable
6	RW	0	**IPV4.** Directed IPv4 Packet Wake Up Enable
7	RW	0	**IPV6.** Directed IPv6 Packet Wake Up Enable
8	RO	0	**Reserved**.
9:14	RO	0	**Reserved**.
15	RW	0	**NoTCO.** Ignore TCO Packets for TCO. If the NoTCO bit is set, then any packet that passes the manageability packet filtering will not cause a Wake Up event even if it passes one of the Wake Up Filters.
16	RW	0	**FLX0.** Flexible Filter 0 Enable
17	RW	0	**FLX1.** Flexible Filter 1 Enable
18	RW	0	**FLX2.** Flexible Filter 2 Enable
19	RW	0	**FLX3.** Flexible Filter 3 Enable
22	RW	0	**FLX6.** Flexible Filter 6 Enable

Bits	Type	Default	Description
23	RW	0	**FLX7.** Flexible Filter 7 Enable
31:24	RO	0	**Reserved.**

This register is used to enable each of the pre-defined and flexible filters for wake up support. A value of 1 means the filter is turned on, and a value of 0 means the filter is turned off.

12.0.3.7.3 Wake Up Status Register - WUS (0x05810; RW)

Bits	Type	Default	Description
0	RW	0	**LNKC.** Link Status Changed
1	RW	0	**MAG.** Magic Packet Received
2	RW	0	**EX.** Directed Exact Packet Received. The packet's address matched one of the 7 pre-programmed exact values in the *Receive Address* registers.
3	RW	0	**MC.** Directed Multicast Packet Received. The packet was a multicast packet that hashed to a value corresponding to a 1 bit in the *Multicast Table Array*.
4	RW	0	**BC.** Broadcast Packet Received
5	RW	0	**ARP.** ARP/IPv4 Request Packet Received
6	RW	0	**IPV4.** Directed IPv4 Packet Received
7	RW	0	**IPV6.** Directed IPv6 Packet Received
15:8	RO	0	**Reserved**. Read as 0
16	RW	0	**FLX0.** Flexible Filter 0 Match
17	RW	0	**FLX1.** Flexible Filter 1 Match
18	RW	0	**FLX2.** Flexible Filter 2 Match
19	RW	0	**FLX3.** Flexible Filter 3 Match
20	RW	0	FLX4. Flexible Filter 4 Match
21	RW	0	FLX5. Flexible Filter 5 Match
22	RW	0	**FLX6.** Flexible Filter 6 Match
23	RW	0	**FLX7.** Flexible Filter 7 Match
31:24	RO	0	**Reserved.**

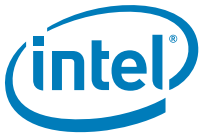
This register is used to record statistics about all Wake Up packets received. A packet that matches multiple criteria may set multiple bits. Writing a 1 to any bit will clear that bit.

This register will not be cleared when PCI_RST_N is asserted. It will only be cleared when LAN PWR Good is de-asserted or when cleared by the driver.

12.0.3.7.4 IP Address Valid - IPAV (0x5838; RW)

The IP Address Valid indicates whether the IP addresses in the IP Address Table are valid:

Bits	Type	Default	Description
0	RO	0	**Reserved**.



Bits	Type	Default	Description
1	RW	0	**V41**. IPv4 Address 1 Valid
2	RW	0	**V42**. IPv4 Address 2 Valid
3	RW	0	**V43**. IPv4 Address 3 Valid
4:15	RO	0x00	**Reserved**.
16	RW	0	**V60**. IPv6 Address Valid
31:17	RO	0x00	**Reserved**.

12.0.3.7.5 IPv4 Address Table - IP4AT (0x05840 + 8*n (n=1…3); RW)

The IPv4 Address Table is used to store the three IPv4 addresses for ARP/IPv4 Request packet and Directed IPv4 packet wake up. It is a 4 entry table with the following format:

Bits	Type	Default	Description
31:0	RW	X	**IPADD**. IP Address n (n=1, 2, 3)

The register at address 0x5840 (n=0) was used in predecessors and reserved in the LAN Controller.

12.0.3.7.6 IPv6 Address Table - IP6AT (0x05880 + 4*n (n=0…3); RW)

The IPv6 Address Table is used to store the IPv6 address for Directed IPv6 packet wake up and Manageability traffic filtering. The IP6AT has the following format:

Bits	Type	Default	Description
31:0	RW	X	**IPV6 Address**. IPv6 Address bytes n*4…n*4+3 (n=0, 1, 2, 3) while byte 0 is first on the wire and byte 15 is last.

The IP6AT may be used by both host and manageability engine. An interrupt mechanism is added to inform the manageability engine on any change of these registers by the host software.

12.0.3.7.7 Flexible Filter Length Table - FFLT (0x05F00 + 8*n (n=0…7); RW)

There are 8 flexible filters Lengths. The Flexible Filter Length Table stores the minimum packet lengths required to pass each of the Flexible Filters. Any packets that are shorter than the programmed length will not pass that filter. Each Flexible Filter will consider a packet that does not have any mismatches up to that point to have passed the Flexible Filter when it reaches the required length. It will not check any bytes past that point.

Bits	Type	Default	Description
10:0	RW	X	**LEN**. Minimum Length for Flexible Filter n.
31:11	RO	X	**Reserved**.

All reserved fields read as 0’s and ignore writes.

Note: Before writing to the Flexible Filter Length Table the driver must first disable the flexible filters by writing 0’s to the Flexible Filter Enable bits of the Wake Up Filter Control Register (WUFC.FLXn).

12.0.3.7.8 Flexible Filter Mask Table - FFMT (0x09000 + 8*n (n=0…127); RW)

There are 128 mask entries. The Flexible Filter Mask and Table is used to store the four 1-bit masks for each of the first 128 data bytes in a packet, one for each Flexible Filter. If the mask bit is 1, the corresponding Flexible Filter will compare the incoming data byte at the index of the mask bit to the data byte stored in the Flexible Filter Value Table.

Bits	Type	Default	Description
0	RW	X	**Mask 0**. Mask for filter 0 byte n (n=0, 1… 127)
1	RW	X	**Mask 1**. Mask for filter 1 byte n (n=0, 1… 127)
2	RW	X	**Mask 2**. Mask for filter 2 byte n (n=0, 1… 127)
3	RW	X	**Mask 3**. Mask for filter 3 byte n (n=0, 1… 127)
6	RW	X	**Mask 6**. Mask for filter 6 byte n (n=0, 1… 127)
7	RW	X	**Mask 7**. Mask for filter 7 byte n (n=0, 1… 127)
31:8	RO	X	**Reserved.**

Note: The table is organized to permit expansion to 8 (or more) filters and 256 bytes in a future product without changing the address map.

Note: Before writing to the Flexible Filter Mask Table the driver must first disable the flexible filters by writing 0's to the Flexible Filter Enable bits of the Wake Up Filter Control Register (WUFC.FLXn).

12.0.3.7.9 Flexible Filter Value Table - FFVT (0x09800 + 8*n (n=0…127); RW)

There are 128 filter values. The Flexible Filter Value is used to store the one value for each byte location in a packet for each flexible filter. If the corresponding mask bit is 1, the Flexible Filter will compare the incoming data byte to the values stored in this table.

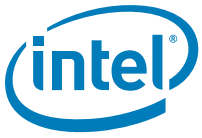
Bits	Type	Default	Description
7:0	RW	X	**Value 0**. Value of filter 0 byte n (n=0, 1… 127)
15:8	RW	X	**Value 1**. Value of filter 1 byte n (n=0, 1… 127)
23:16	RW	X	**Value 2**. Value of filter 2 byte n (n=0, 1… 127)
31:24	RW	X	**Value 3**. Value of filter 3 byte n (n=0, 1… 127)

Before writing to the Flexible Filter Value Table the driver must first disable the flexible filters by writing 0's to the Flexible Filter Enable bits of the Wake Up Filter Control Register (WUFC.FLXn).

12.0.3.7.10 Flexible Filter Value Table - FFVT (0x09800 + 8*n (n=0…127); RW)

There are 128 filter values. The Flexible Filter Value is used to store the one value for each byte location in a packet for each flexible filter. If the corresponding mask bit is 1, the Flexible Filter will compare the incoming data byte to the values stored in this table.

Bits	Type	Reset	Description
7:0	RW	X	**Value 0**. Value of filter 0 byte n (n=0, 1… 127)
15:8	RW	X	**Value 1**. Value of filter 1 byte n (n=0, 1… 127)
23:16	RW	X	**Value 2**. Value of filter 2 byte n (n=0, 1… 127)



31:24	RW	X	**Value 3**. Value of filter 3 byte n (n=0, 1… 127)

Before writing to the Flexible Filter Value Table the driver must first disable the flexible filters by writing 0's to the Flexible Filter Enable bits of the Wake Up Filter Control Register (WUFC.FLXn).

12.0.3.8 Time Sync Register Description

Note: All the registers bellow are for logical needs only, upon implementation their offset or structure can be changed according to the project specific needs.

12.0.3.8.1 RX Time Sync Control register - TSYNCRXCTL (0xB620; RW)

Bits	Type	Default	Description
0	(RO/ V)	0	**RXTT**. Rx timestamp valid equals '1' when a valid value for Rx timestamp is captured in the Rx timestamp register, clear by read of Rx timestamp register RXSTMPH.
3:1	RW	0x0	**Type**. Type of packets to timestamp - 000b – Time stamp L2 (V2) packets only (Sync or Delay_req depends on message type in Section 12.0.3.8.6 and packets with message ID 2 and 3) 001b – Time stamp L4 (V1) packets only (Sync or Delay_req depends on message type in Section 12.0.3.8.6) 010b – Time stamp V2 (L2 and L4) packets (Sync or Delay_req depends on message type in Section 12.0.3.8.6 and packets with message ID 2 and 3) 100b – Time stamp all packets (in this mode no locking is done to the value in the timestamp registers and no indications in receive descriptors will be transferred 101b - Time stamp all packets which message id bit 3 is zero, which means timestamp all event packets. This is applicable for V2 packets only. 011b, 110b and 111b – reserved
4	RW	0x0	**En**. Enable RX timestamp 0x0 – time stamping disabled. 0x1 – time stamping enabled.
31:5	RO	0x0	**Reserved.**

12.0.3.8.2 RX timestamp Low - RXSTMPL (0x0B624; RO)

Bits	Type	Default	Description
31:0	RO	0x0	**RXSTMPL**. Rx timestamp LSB value

12.0.3.8.3 RX timestamp High - RXSTMPH (0x0B628; RO)

Bits	Type	Default	Description
31:0	RO	0x0	**RXSTMPH**. Rx timestamp MSB value

12.0.3.8.4 RX timestamp attributes low - RXSATRL (0x0B62C; RO)

Bits	Type	Default	Description
31:0	RO	0x0	**SourceIDL**. Sourceuuid low The value of this register is in host order.

12.0.3.8.5 RX timestamp attributes high- RXSATRH (0x0B630; RO)

Bits	Type	Default	Description
15:0	RO	0x0	**SourceIDH**. Sourceuuid high The value of this register is in host order.
31:16	RO	0x0	**SequenceID**. SequenceI The value of this register is in host order.

12.0.3.8.6 RX message type register low - RXMTRL (0x0B634; RW)

Bits	Type	Default	Description
15:0	RW	0x88F7	PTP L2 EtherType to timestamp The value of this register is programmed/read in network order.
23:16	RW	0x0	V1 control to timestamp
31:24	RW	0x0	V2 messageId to timestamp

12.0.3.8.7 RX UDP port - RXUDP (0x0B638; RW)

Bits	Type	Default	Description
15:0	RW	0x0319	**UPORT**. UDP port number to time stamp The value of this register is programmed/read in network order.
31:16	RO	0x0	**Reserved.**

12.0.3.8.8 TX Time Sync Control register - TSYNCTXCTL (0x0B614; RW)

Bits	Type	Default	Description
0	RO/V	0	**TXTT**. Tx timestamp valid equals '1' when a valid value for Rx timestamp is captured in the Rx timestamp register, clear by read of Tx timestamp register TXSTMPH.
3:1	RO	0	**Reserved.**
4	RW	0	**EN**. Enable TX timestamp 0x0 – time stamping disabled. 0x1 – time stamping enabled.
5	RW	0	**Packet mode**. In this mode (deterministic SYSTIM) the SYSTIM register is incremented by TIMINCA.IV (increment value) every time a packet is time stamped. There is no significance whatsoever to the TIMINCA.IP (increment period) value.· TIMADJ (time adjustment) registers work as they do in normal mode: every time they're set to a certain value - that value is immediately (next clock – not packet) added (/subtracted) to the SYSTIM value. 0x0 - Packet mode disabled 0x1 - Packet mode enabled

Bits	Type	Default	Description
31:6	RO	0	**Reserved.**

12.0.3.8.9 TX timestamp value Low - TXSTMPL (0x0B618; RO)

Bits	Type	Default	Description
31:0	RO	0x0	**TXSTMPL**. Tx timestamp LSB value

12.0.3.8.10 TX timestamp value High - TXSTMPH (0x0B61C; RO)

Bits	Type	Default	Description
31:0	RO	0x0	**TXSTMPH**. Tx timestamp MSB value

12.0.3.8.11 System time register Low - SYSTIML (0x0B600; RO)

Bits	Type	Default	Description
31:0	RW	0x0	**STL**. System time LSB register

12.0.3.8.12 System time register High - SYSTIMH (0x0B604; RO)

Bits	Type	Default	Description
31:0	RW	0x0	**STH**. System time MSB register

12.0.3.8.13 Increment attributes register - TIMINCA (0x0B608; RW)

Bits	Type	Default	Description
23:0	RW	0x0	**IV**. Increment value – *incvalue*
31:24	RW	0x0	**IP**. Increment period – *incperiod*

12.0.3.8.14 Time adjustment offset register low - TIMADJL (0x0B60C; RW)

Bits	Type	Default	Description
31:0	RW	0x00	**TADJL**. Time adjustment value – Low

12.0.3.8.15 Time adjustment offset register high - TIMADJH (0x0B610;RW)

Bits	Type	Default	Description
30:0	RW	0x00	**TADJH**. Time adjustment value - High
31	RW	0x0	**Sign**. Sign (“0”=”+”, “1”=”-“)

12.0.3.9 Diagnostic Register Descriptions

LAN Controller contains several diagnostic registers. These registers allow software to directly access the contents of the LAN Controller’s internal Packet Buffer Memory (PBM), also referred to as FIFO space. These registers also give software visibility into what locations in the PBM that the HW currently considers to be the “head” and “tail” for both transmit and receive operations.

12.0.3.9.1 Receive Data FIFO Head Register - RDFH (0x02410; RW)

Bits	Type	Default	Description
12:0	RW	0	**Receive FIFO Head pointer (FIFO Head)**.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register stores the head pointer of the on–chip receive data FIFO. Since the internal FIFO is organized in units of 64 bit words, this field contains the 64 bit offset of the current Receive FIFO Head. So a value of "0x8" in this register corresponds to an offset of 8 QWORDS or 64 bytes into the Receive FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x08000. In addition, with the LAN Controller, the value in this register contains the offset of the Receive FIFO head, relative to the beginning of the entire PBM space. Alternatively, with the 82542, the value in this register contains the relative offset from the beginning of the Receive FIFO space (within the PBM space).

12.0.3.9.2 Receive Data FIFO Tail Register - RDFT (0x02418; RW)

Bits	Type	Default	Description
12:0	RW	0	**Receive FIFO Tail pointer (FIFO Tail)**.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

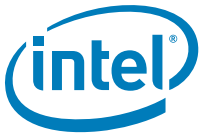
This register stores the tail pointer of the on–chip receive data FIFO. Since the internal FIFO is organized in units of 64 bit words, this field contains the 64 bit offset of the current Receive FIFO Tail. So a value of "0x8" in this register corresponds to an offset of 8 QWORDS or 64 bytes into the Receive FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x08008. In addition, with the LAN Controller, the value in this register contains the offset of the Receive FIFO tail, relative to the beginning of the entire PBM space. Alternatively, with the 82542, the value in this register contains the relative offset from the beginning of the Receive FIFO space (within the PBM space).

12.0.3.9.3 Receive Data FIFO Head Saved Register - RDFHS (0x02420; RW)

Bits	Type	Default	Description
12:0	RW	0	**FIFO Head**. A "saved" value of the Receive FIFO Head pointer.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register stores a copy of the Receive Data FIFO Head register in the case that the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.



12.0.3.9.4 Receive Data FIFO Tail Saved Register - RDFTS (0x02428; RW)

Bits	Type	Default	Description
12:0	RW	0	**FIFO Tail**. A "saved" value of the Receive FIFO Tail pointer.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register stores a copy of the Receive Data FIFO Tail register in the case that the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

12.0.3.9.5 Receive Data FIFO Packet Count - RDFPC (0x02430; RW)

Bits	Type	Default	Description
12:0	RW	0	**RX FIFO Packet Count**. The number of **received packets currently in the RX FIFO**.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register reflects the number of receive packets that are currently in the Receive FIFO. This register is available for diagnostic purposes only, and should not be written during normal operation.

12.0.3.9.6 Transmit Data FIFO Head Register - TDFH (0x03410; RW)

Bits	Type	Default	Description
12:0	RW/V	0x0900[1]	**FIFO Head.** Transmit FIFO Head pointer.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

1. The initial value equals PBA. RXA times 128.

This register stores the head pointer of the on–chip transmit data FIFO. Since the internal FIFO is organized in units of 64 bit words, this field contains the 64 bit offset of the current Transmit FIFO Head. So a value of "0x8" in this register corresponds to an offset of 8 QWORDS or 64 bytes into the Transmit FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x08010. In addition, with the LAN Controller, the value in this register contains the offset of the Transmit FIFO head relative to the beginning of the entire PBM space. Alternatively, with the 82542, the value in this register contains the relative offset from the beginning of the Transmit FIFO space (within the PBM space).

12.0.3.9.7 Transmit Data FIFO Tail Register - TDFT (0x03418; RW)

Bits	Type	Default	Description
12:0	RW/V	0x000[1]	**FIFO Tail.** Transmit FIFO Tail pointer.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

1. The initial value equals PBA. RXA times 128.

This register stores the head pointer of the on–chip transmit data FIFO. Since the internal FIFO is organized in units of 64 bit words, this field contains the 64 bit offset of the current Transmit FIFO Tail. So a value of "0x8" in this register corresponds to an offset of 8 QWORDS or 64 bytes into the Transmit FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Note: This register's address has been moved from where it was located in the 82542. However, for 82542 compatibility, this register may also be accessed at its alias offset of 0x08018. In addition, with the LAN Controller, the value in this register contains the offset of the Transmit FIFO tail relative to the beginning of the entire PBM space. Alternatively, with the 82542, the value in this register contains the relative offset from the beginning of the Transmit FIFO space (within the PBM space).

12.0.3.9.8 Transmit Data FIFO Head Saved Register - TDFHS (0x03420; RW)

Bits	Type	Default	Description
12:0	RW/V	0x000[1]	**FIFO Head**. A "saved" value of the Transmit FIFO Head pointer.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

1. The initial value equals PBA. RXA times 128.

This register stores a copy of the Transmit Data FIFO Head register in the case that the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

12.0.3.9.9 Transmit Data FIFO Tail Saved Register - TDFTS (0x03428; RW)

Bits	Type	Default	Description
12:0	RW/V	0x000[1]	**FIFO Tail**. A "saved" value of the Transmit FIFO Tail pointer.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

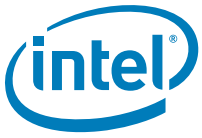
1. The initial value equals PBA. RXA times 128.

This register stores a copy of the Transmit Data FIFO Tail register in the case that the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

12.0.3.9.10 Transmit Data FIFO Packet Count - TDFPC (0x03430; RW)

Bits	Type	Default	Description
12:0	RW	0	**TX FIFO Packet Count**. The number of packets to be transmitted that are currently in the TX FIFO.
31:13	RO	0	**Reserved**. Reads as 0. Should be written to 0 for future compatibility.

This register reflects the number of packets to be transmitted that are currently in the Transmit FIFO. This register is available for diagnostic purposes only, and should not be written during normal operation.



12.0.3.9.11 Ghost Memory Data - GMD (0x0C000 - 0x0C3FC; RW)

Bits	Type	Default	Description
31:0	RW	X	**Data.** Ghost Memory Data.

Ghost memory data is available to diagnostics. Locations can be accessed as 32 bit words.

12.0.3.9.12 Descriptor Memory Data - DMD (0x0C400 - 0x0C5FC; RW)

Bits	Type	Default	Description
31:0	RW	X	**Data.** Descriptor Memory Data.

Descriptor memory data is available to diagnostics. Locations can be accessed as 32 bit words.

12.0.3.10 MACsec Register Descriptions

12.0.3.10.1 MACsec TX Capabilities register - LSECTXCAP (0x0B000; RW)

Bits	Type	Default	Description
2:0	RO	1b	**TX CA-supported** Number of CA's supported by the device.
6:3	RO	1b	**TX SC Capable.** Number of SC's supported by the device on the transmit data path. The LAN controller supports twice the number of SA's as the TX SC for seamless re-keying, i.e., 2 SA's.
15:7	RO	0x0	**Reserved.**
23:16	RO/V	0x0	**Tx LSEC Key SUM (LSECTXSUM).** A bit wise XOR of the LSECTXKEY 0 bytes and LSECTXKEY 1 bytes. This register may be used by KaY (the programming entity) to validate key programming.
31:24	RO/V	0x0	**Reserved.**

12.0.3.10.2 MACsec RX Capabilities register - LSECRXCAP (0x0B300; RW)

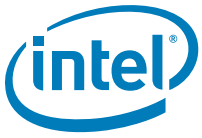
Bits	Type	Default	Description
2:0	RO	1b	**RX CA-supported** Number of CA's supported by the device.
6:3	RO	4b1b	**RX SC Capable.** Number of SC's supported by the device on the receive data path. The LAN controller supports twice SA's as the RX SC for seamless re-keying, i.e., 8 2 SA's.
15:7	RO	0x0	**Reserved.**
23:16	RO/V	0x0	**Rx LSEC Key SUM (LSECRXSUM).** A bit wise XOR of the Rx MACsec keys 0…7 as defined in registers LSECRXKEY [n, m]. Each byte is XORed with the respective byte of the other keys. This register may be used by KaY (the programming entity) to validate key programming.
31:24	RO	0x0	**Reserved.**

12.0.3.10.3 MACsec TX Control register - LSECTXCTRL (0x0B004; RW)

Bits	Type	Default	Description
1:0	RW	00b	**Enable Tx MACsec.** Enable Tx MACsec off loading. 00b – Disable Tx MACsec (Tx all packets w/o MACsec offload). 01b – Add integrity signature. 10b – Encrypt and add integrity signature. 11b – Reserved. When this field equals 00b (MACsec offload is disabled). The "Tx Untagged Packet" register is not incremented for transmitted packets when the "Enable Tx MACsec" equals 00b.
2	RW	0	PNID PN Increase Disable 0 - Normal operation 1 - PN is not incremented, used for testability mode only.
3	RW	0	**(Testability feature)** when set to 1 the HW ignores the ILSEC bit in the TX descriptor and transmits the packet as if the ILSEC bit was set.
4	RW	0b	**Reserved.**
5	RW	1b	**Always Include SCI.** This field controls whether SCI is explicitly included in the transmitted SecTag. 0b – False 1b – True, SCI is explicitly included
6	RW	0b	**Reserved.**
7	RW	1	**Reserved.**
31:8	RW	11..1b	**PN exhaustion threshold.** MSB of the threshold over which HW needs to interrupt the KaY to warn of TX SA PN exhaustion and will trigger a new SA renegotiation. Bits 7:0 of the threshold are all 1's.

12.0.3.10.4 MACsec RX Control register - LSECRXCTRL (0x0B304; RW

Bits	Type	Default	Description
1:0	RW	00b	**Reserved.**
3:2	RW	00b	**Reserved.**
4	RO	1b	**Reserved.**
5	RO	1b	**Reserved.**
6	RW	0b	**Reserved.**
7	RW	1	**Replay Protect.** Enable replay protection.
28:8	RO	0x0	**Reserved**
31:29	RW	0	**Reserved**.



12.0.3.10.5 MACsec TX SCI Low - LSECTXSCL (0x0B008; RW)

Bits	Type	Default	Description
31:0	RW	0b	**MAC Address SecY Low.** The 4 LS bytes of the MAC address copied to the SCI field in the MACsec header. The value of this register is programmed/read in host order.

12.0.3.10.6 MACsec TX SCI High - LSECTXSCH (0x0B00C; RW)

Bits	Type	Default	Description
15:0	RW	0b	**MAC Address SecY High.** The 2 MS bytes of the MAC address copied to the SCI field in the MACsec header. The value of this register is programmed/read in host order.
31:16	RW	0b	**Port Identifier.** Always zero for transmitted packets.

12.0.3.10.7 MACsec TX SA - LSECTXSA (0x0B010; RW)

Bits	Type	Default	Description
1:0	RW	0b	**AN0 – Association Number 0.** This 2 bit field is posted to the AN field in the transmitted MACsec header when SA 0 is active.
3:2	RW	0b	**AN1 – Association Number 1.** This 2 bit field is posted to the AN field in the transmitted MACsec header when SA 1 is active.
4	RW	0b	**SA Select (SelSA).** This bit selects between SA 0 or SA 1 smoothly, i.e., on a packet boundary. A value of '0' selects SA 0 and a value of '1' selects SA 1.
5	RO/V	0b	**Active SA (ActSA).** This bit indicates the active SA. The ActSA follows the value of the SelSA on a packet boundary. The KaY (the programming entity) may use this indication to retire the old SA.
6	RW	0	**Reserved**.
7	RW	0	**Reserved**.
31:8	RW	0x0	**Reserved**.

12.0.3.10.8 MACsec TX SA PN 0 - LSECTXPN0 (0x0B018; RW)

Bits	Type	Default	Description
31:0	RW	0b	**PN – Packet number.** This field is posted to the PN field in the transmitted MACsec header when SA 0 is active. It is initialized by the KaY at SA creation and then increments by 1 for each transmitted packet using this SA. Packets should never be transmitted if the PN repeats itself. In order to protect against such an event, the HW generates an LSECPN interrupt to KaY when the PN reaches the exhaustion threshold as defined in the LSECTXCTRL register. There is additional level of defense against repeating the PN. The HW will never transmit packets after the PN reaches a value of 0xFF...FF. In order to guarantee this, the HW clears the "Enable Tx MACsec" field in the LSECTXCTRL register to 00b once a packet is transmitted with a PN that equals to 0xFF...F0. The value of this register is programmed/read in host order.

12.0.3.10.9 MACsec TX SA PN 1 - LSECTXPN1 (0x0B01C; RW)

Bits	Type	Default	Description
31:0	RW	0b	**PN – Packet number.** This field is posted to the PN field in the transmitted MACsec header when SA 1 is active. It is initialized by the KaY at SA creation and then increments by 1 for each transmitted packet using this SA. Packets should never be transmitted if the PN repeats itself. In order to protect against such an event the HW generates an LSECPN interrupt to KaY when the PN reaches the exhaustion threshold as defined in the LSECTXCTRL register. There is additional level of defense against repeating the PN. The HW will never transmit packets after the PN reaches a value of 0xFF...FF. In order to guarantee this, the HW clears the "Enable Tx MACsec" field in the LSECTXCTRL register to 00b once a packet is transmitted with a PN that equals to 0xFF...F0. The value of this register is programmed/read in host order.

12.0.3.10.10 MACsec TX Key 0 - LSECTXKEY0 [n] (0x0B020 + 4*n (n=0...3); WO)

Bits	Type	Default	Description
31:0	WO	0x0	**LSEC Key 0.** Transmit MACsec key of SA 0. n – 0 LSEC Key defines bits 31:0 of the Tx MACsec Key n – 1 LSEC Key defines bits 63:32 of the Tx MACsec Key n – 2 LSEC Key defines bits 95:64 of the Tx MACsec Key n – 3 LSEC Key defines bits 127:96 of the Tx MACsec Key This field is WO for confidentiality protection. For data integrity check, the hash value may read the LSECTXSUM field in the LSECCAP register. If for some reason a read request is aimed to this register a value of all zeros will be returned. The value of this register is programmed/read in host order.

12.0.3.10.11 MACsec TX Key 1 - LSECTXKEY1 [n] (0x0B030 + 4*n (n=0…3); WO)

Bits	Type	Default	Description
31:0	WO	0x0	**LSEC Key 1.** Transmit MACsec key of SA 1. n – 0 LSEC Key defines bits 31:0 of the Tx MACsec Key n – 1 LSEC Key defines bits 63:32 of the Tx MACsec Key n – 2 LSEC Key defines bits 95:64 of the Tx MACsec Key n – 3 LSEC Key defines bits 127:96 of the Tx MACsec Key This field is WO for confidentiality protection. For data integrity check, the hash value may read the LSECTXSUM field in the LSECCAP register. If for some reason a read request is aimed to this register a value of all zeros will be returned. The value of this register is programmed/read in host order.

12.0.3.11 MACsec Rx SC Registers

The registers in this section relate to MACsec Receive SC context. There are 4 SC(s) in the receive data path defined as SC0, SC1, SC2 and SC3. The registers below with index n relates to the SC index, while n=0, 1, 2, 3.

12.0.3.11.1 MACsec RX SCI Low - LSECRXSCL (0x0B3D0 + 4*n (n=0…3); RW)

Bits	Type	Default	Description
31:0	RW	0b	**MAC Address SecY low.** The 4 LS bytes of the MAC address in the SCI field in the incoming packet that are compared with this field for SCI matching. Comparison result is meaningful only if the SC bit in the TCI header is set. Index n=0, 1, 2, 3 for SCI0, SCI1, SCI2 and SCI3 respectively. The value of this register is programmed/read in host order.

12.0.3.11.2 MACsec RX SCI High - LSECRXSCH (0x0B3E0 + 4*n (n=0…3); RW)

Bits	Type	Default	Description
15:0	RW	0b	**MAC Address SecY High.** The 2 MS bytes of the MAC address in the SCI field in the incoming packet that are compared with this field for SCI matching. Comparison result is meaningful only if the SC bit in the TCI header is set. Index n=0, 1, 2, 3 for SCI0, SCI1, SCI2 and SCI3 respectively. The value of this register is programmed/read in host order.
31:16	RW	0b	**Port Identifier.** The Port Number in the SCI field in the incoming packet that is compared with this field for SCI matching. Comparison result is meaningful only if the SC bit in the TCI header is set. Index n=0, 1, 2, 3 for SCI0, SCI1, SCI2 and SCI3 respectively. The value of this register is programmed/read in host order.

12.0.3.12 MACsec Rx SA Registers

The registers in this section relates to MACsec Receive SA context. There are 8 SA(s) in the receive data path defined as SA0 and SA1… SA7. The registers below with index n relates to the SA index, while n=0, 1… 7. SA0 and SA1 are assigned to SCI0, SA2 and SA3 are assigned to SCI1, SA4 and SA5 are assigned to SCI2 and SA6 and SA7 are assigned to SCI3.

12.0.3.12.1 MACsec RX SA - LSECRXSA (0x0B310 + 4*n (n=0…7); RW)

Bits	Type	Default	Description
1:0	RW	0b	**AN – Association Number.** This field is compared with the AN field in the TCI field in the incoming packet for match.
2	RW	0b	**SA Valid.** This bit is set or cleared by the KaY to validate or invalidate the SA.
3	RO/V	0b	**Frame received.** This bit is cleared when the SA Valid (bit 2) transitions from 0->1, and is set when a frame is received with this SA. When the Frame received bit is set the Retired bit of the other SA of the same SC is set. Note that a single frame reception with the new SA is sufficient to retire the old SA since we assume the Replay Window is 0.
4	RO/V	0b	**Retired.** When this bit is set the SA is invalid (retired). This bit is cleared when a new SA is configured by the KaY (SA Valid transition to 1). It is set to '1' when a packet is received with the other SA of the same SC. Note that a single frame reception with the new SA is sufficient to retire the old SA since we assume the Replay Window is 0.
31:5	RO	0	**Reserved.**

12.0.3.12.2 MACsec RX SA PN - LSECRXSAPN (0x0B330 + 4*n (n=0…7); RW)

Bits	Type	Default	Description
31:0	RW	0b	**PN – Packet number.** This register holds the PN field of the next incoming packet that uses this SA. The PN field in the incoming packet must be greater or equal to the PN register. The PN register is set by KaY at SA creation. It is updated by the hardware for each received packet using this SA to be Received PN + 1. The value of this register is programmed/read in host order.

12.0.3.12.3 MACsec RX Key - LSECRXKEY[n,m] (0x0B350 + 0x10*n (n=0…7) + 4*m (m=0…3); WO)

Bits	Type	Default	Description
31:0	WO	0x0	**LSEC Key.** Receive MACsec key of SA n, while n=0,1,2…7. m – 0 LSEC Key defines bits 31:0 of the Rx MACsec Key m – 1 LSEC Key defines bits 63:32 of the Rx MACsec Key m – 2 LSEC Key defines bits 95:64 of the Rx MACsec Key m – 3 LSEC Key defines bits 127:96 of the Rx MACsec Key This field is WO for confidentiality protection. For data integrity check, the KaY hash value may read the LSECRXSUM field in the LSECCAP registers. If for some reason a read request is aimed to this register a value of all zeros will be returned. The value of this register is programmed/read in host order.

12.0.3.13 MACsec Tx Port Statistics

These counters are defined by spec as 64bit while implementing only 32 bit in the hardware. The KaY must implement the 64 bit counter in SW by regularly polling the hardware statistic counters. The HW section of the statistics counter is cleared upon read action.

12.0.3.13.1 Tx Untagged Packet Counter - LSECTXUT (0x04300; RC)

This statistic implements the SecyStatsRxUntaggedPkts statistic of the 802.1ae MIB while in non strict mode. In strict mode, this implements the secyStatsRxNoTagPkts statistic of the 802.1ae MIB.

Bits	Type	Default	Description
31:0	RC	0x0	**Untagged Packet CNT.** Increments for each transmitted packet that is transmitted with the ILSec bit cleared in the packet descriptor while "Enable Tx MACsec" field in the LSECTXCTRL register is either 01b or 10b. The KaY must implement a 64 bit counter. It can do that by reading the LSECTXUT register regularly.

12.0.3.13.2 Encrypted Tx Packets - LSECTXPKTE (0x04304; RC)

Bits	Type	Default	Description
31:0	RC	0x0	**Encrypted Packet CNT.** Increments for each transmitted packet through the controlled port with E bit set (i.e. confidentiality was prescribed for this packet by SW/FW).

12.0.3.13.3 Protected Tx Packets - LSECTXPKTP (0x04308; RC)

Bits	Type	Default	Description
31:0	RC	0x0	**Protected Packet CNT.** Increments for each transmitted packet through the controlled port with E bit cleared (i.e. integrity only was prescribed for this packet by SW/FW).

12.0.3.13.4 Encrypted Tx Octets - LSECTXOCTE (0x0430C; RC)

Bits	Type	Default	Description
31:0	RC	0x0	**Encrypted Octet CNT.** Increments for each byte of user data through the controlled port with E bit set (i.e. confidentiality was prescribed for this packet by SW/FW).

12.0.3.13.5 Protected Tx Octets - LSECTXOCTP (0x04310; RC)

Bits	Type	Default	Description
31:0	RC	0x0	**Protected Octet CNT.** Increments for each byte of user data through the controlled port with E bit (i.e. integrity only was prescribed for this packet by SW/FW).

12.0.3.14 MACsec Rx Port Statistic Counters

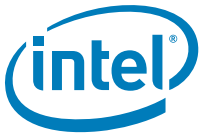
These counters are defined by spec as 64bit while implementing only 32 bit in the hardware. The KaY must implement the 64 bit counter in SW by regularly polling the hardware statistic counters.

12.0.3.14.1 MACsec Untagged RX Packet - LSECRXUT (0x04314; RC)

Bits	Type	Default	Description
31:0	RC	0b	**Untagged Packet CNT.** Increments for each packet received having no tag. Increments only when “Enable Rx MACsec” field in the LSECRXCTRL register is either 01b or 10b.

12.0.3.14.2 MACsec RX Octets Decrypted - LSECRXOCTD (0x0431C; RC)

Bits	Type	Default	Description
31:0	RC	0b	**Decrypted Rx Octet CNT.** The number of octets of User Data recovered from received frames that were both integrity protected and encrypted. This includes the octets from SecTag to ICV not inclusive. These counts are incremented even if the User Data recovered failed the integrity check or could not be recovered.



12.0.3.14.3 MACsec RX Octets Validated - LSECRXOCTV (0x04320; RC)

Bits	Type	Default	Description
31:0	RC	0b	**Validated Rx Octet CNT.** The number of octets of User Data recovered from received frames that were integrity protected but not encrypted.This includes the octets from SecTag to ICV not inclusive. These counts are incremented even if the User Data recovered failed the integrity check or could not be recovered.

12.0.3.14.4 MACsec RX Packet with Bad Tag - LSECRXBAD (0x04324; RC)

Bits	Type	Default	Description
31:0	RC	0b	**Bad Rx Packet CNT.** Number of packets received having an invalid tag.

12.0.3.14.5 MACsec RX Packet No SCI - LSECRXNOSCI (0x04328; RC)

Bits	Type	Default	Description
31:0	RC	0b	**No SCI Rx Packet CNT.** Number of packets received having unrecognized SCI and dropped due to that condition.

12.0.3.14.6 MACsec RX Packet Unknown SCI count - LSECRXUNSCI (0x432C; RC)

Bits	Type	Default	Description
31:0	RC	0b	**Unknown SCI Rx Packet CNT.** Number of packets received with an unrecognized SCI but still forwarded to the host.

12.0.3.15 MACsec Rx SC Statistic Counters

12.0.3.15.1 MACsec RX Unchecked Packets - LSECRXUNCH (0x04330; RC)

SW/FW needs to maintain the full sized register.

Bits	Type	Default	Description
31:0	RC	0b	**Unchecked Rx Packet CNT.** Rx Packet CNT. Number of packets received with MACsec encapsulation (SecTag) while ValidateFrames is disabled (LSECRXCTRL bits 3:2 equal 00b).”

12.0.3.15.2 MACsec RX Delayed Packets - LSECRXDELAY (0x04340 + 4*n (n=0…3); RC)

SW/FW needs to maintain the full sized register.

Bits	Type	Default	Description
31:0	RC	0b	**Delayed Rx Packet CNT.** Number of packets received and accepted for validation having failed replay-protection and ReplayProtect is false (LSECRXCTRL bit 7 is zero).

12.0.3.15.3 MACsec RX Late Packets - LSECRXLATE (0x04350 + 4*n (n=0…3); RC)

SW/FW needs to maintain the full sized register.

Bits	Type	Default	Description
31:0	RC	0b	**Late Rx Packet CNT.** Number of packets received and accepted for validation having failed replay-protection and ReplayProtect is true (LSECRXCTRL bit 7 is '1').

12.0.3.16 MACsec Rx SA Statistic Counters

12.0.3.16.1 MACsec RX Packet OK - LSECRXOK[n] (0x04360 + 4*n (n=0…7); RC)

Bits	Type	Default	Description
31:0	RC	0b	**OK Rx Packet CNT.** Number of packets received that were valid (authenticated) and passed replay protection.

12.0.3.16.2 MACsec Check RX Invalid - LSECRXINV[n] (0x43A0 + 4*n (n=0…7); RC)

Bits	Type	Default	Description
31:0	RC	0b	**Invalid Rx Packet CNT.** Number of packets received that were not valid (authentication failed) and were forwarded to host.

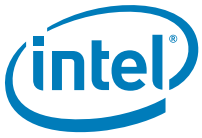
12.0.3.16.3 MACsec RX Not valid count - LSECRXNV[n] (0x04380 + 4*n [n=0...7]; RC)

Bits	Type	Default	Description
31:0	RC	0b	**Invalid Rx Packet CNT.** Number of packets received that were not valid (authentication failed) and were dropped.

12.0.3.16.4 MACsec RX Not using SA - LSECRXNUSA[n] (0x043C0 + 4*n (n=0…3); RC)

Bits	Type	Default	Description
31:0	RC	0b	**Invalid SA Rx Packet CNT.** Number of packets received that were associated with an SA that is not "inUse" (No match on AN or not valid or retired) and were dropped.[1]

1. The implementation maintains one such counter per SC.



12.0.3.16.5 MACsec RX Unused SA - LSECRXUNSA[n] (0x043D0 + 4*n (n=0…3); RC)

Bits	Type	Default	Description
31:0	RC	0b	**Invalid SA Rx Packet CNT.** Number of packets received that were associated with an SA that is not "inUse" (No match on AN or not valid or retired) and where forwarded to host.[1]

1. The implementation maintains one such counter per SC.