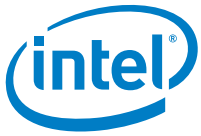


Field	Bit(s)	Init Val	Description
GPI_SDP2	26	0b	General Purpose Interrupt on SDP2. If GPI interrupt detection is enabled on this pin (via GPIE), this interrupt cause is set when the SDP2 is sampled high.
GPI_SDP3	27	0b	General Purpose Interrupt on SDP3. If GPI interrupt detection is enabled on this pin (via GPIE), this interrupt cause is set when the SDP3 is sampled high.
ECC	28	0b	Unrecoverable ECC Error. This bit is set when an unrecoverable error is detected in one of the device memories. Software should issue a software reset following this error.
Reserved	29	0b	Reserved.
TCP Timer	30	0b	TCP Timer Expired. This bit is set when the timer expires.
Other Cause Interrupt	31	0b	Activated when any bit (29:20) in the Extended Interrupt Cause Register (EICR) is set and its relevant mask bit in the EIMS is enabled.

8.2.3.5.2 Extended Interrupt Cause Set Register- EICS (0x00808; WO)

Field	Bit(s)	Init Val	Description
Interrupt Cause Set	30:0	0x0	Setting any bit in this field, sets its corresponding bit in the EICR register and generates an interrupt if enabled by the EIMS register.
Reserved	31	0b	Reserved.

8.2.3.5.3 Extended Interrupt Mask Set/Read Register- EIMS (0x00880; RWS)

Field	Bit(s)	Init Val	Description
Interrupt Enable	30:0	0x0	Each bit set to 1b enables its corresponding interrupt in the EICR. Writing 1b to any bit sets it. Writing 0b has no impact. Reading this register provides a map of those interrupts that are enabled.
Reserved	31	0b	Reserved.

8.2.3.5.4 Extended Interrupt Mask Clear Register- EIMC (0x00888; WO)

Field	Bit(s)	Init Val	Description
Interrupt Mask	30:0	0x0	Writing a 1b to any bit clears its corresponding bit in the EIMS register disabling the corresponding interrupt in the EICR register. Writing 0b has no impact. Reading this register provides no meaningful data.
Reserved	31	0b	Reserved.

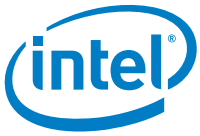
8.2.3.5.5 Extended Interrupt Auto Clear Register — EIAC (0x00810; RW)

Field	Bit(s)	Init Val	Description
RTxQ Auto Clear	15:0	0x0	At 1b, each bit enables auto clear of the corresponding RTxQ bits in the EICR register following interrupt assertion. At 0b, the corresponding bits in the EICR register are not auto cleared.
Reserved	29:16	0x0	Reserved.
TCP Timer Auto Clear	30	0b	At 1b, this bit enables auto clear of the TCP timer interrupt cause in the EICR register following interrupt assertion. At 0b auto clear is not enabled.
Reserved	31	0b	Reserved.

Note: Bits 29:20 should never set auto clear since they share the same MSI-X vector.

8.2.3.5.6 Extended Interrupt Auto Mask Enable Register — EIAM (0x00890; RW)

Field	Bit(s)	Init Val	Description
Auto Mask	30:0	0x0	At 1b, each bit enables auto set and clear of its corresponding bits in the EIMS register. In MSI-X mode, if any of the Auto Mask enable bits is set, the GPIE.EIAME bit must be set as well.
Reserved	31	0b	Reserved.



8.2.3.5.7 Extended Interrupt Cause Set Registers — EICS[n] (0x00A90 + 4*(n-1), n=1...2; WO)

Field	Bit(s)	Init Val	Description
Interrupt Cause Set	31:0	0x0	Setting any bit in these registers sets its corresponding bit in the EICR[n] register and generates an interrupt if enabled by EIMS[n] register. Reading this register provides no meaningful data.

8.2.3.5.8 Extended Interrupt Mask Set/Read Registers — EIMS[n] (0x00AA0 + 4*(n-1), n=1...2; RWS)

Field	Bit(s)	Init Val	Description
Interrupt Enable	31:0	0	Each bit set at 1b enables its corresponding interrupt in the EICR[n] register. Writing 1b to any bit sets it. Writing 0b has no impact. Reading this register provides a map of those interrupts that are enabled. Bits 15:0 of EIMS1 are mirrored in EIMS bits 15:0.

8.2.3.5.9 Extended Interrupt Mask Clear Registers — EIMC[n] (0x00AB0 + 4*(n-1), n=1...2; WO)

Field	Bit(s)	Init Val	Description
Interrupt Mask	31:0	0x0	Writing a 1b to any bit clears its corresponding bit in the EIMS[n] register disabling the corresponding interrupt in the EICR[n] register. Writing 0b has no impact. Reading this register provides no meaningful data.

8.2.3.5.10 Extended Interrupt Auto Mask Enable registers — EIAM[n] (0x00AD0 + 4*(n-1), n=1...2; RW)

Field	Bit(s)	Init Val	Description
Auto Mask	31:0	0x0	At 1b, each bit enables auto set and clear of its corresponding bits in the EIMS[n] register. Bits 15:0 of EIAM1 are mirrored in EIAM bits 15:0. In MSI-X mode, if any of the Auto Mask enable bits is set, the GPIE.EIAME bit must be set as well.

8.2.3.5.11 MSIX to EITR Select — EITRSEL (0x00894; RW)

Field	Bit(s)	Init Val	Description
VFSelect	31:0	0x0	Each bit 'n' in this register selects the VF index (32+'n') or PF interrupt source for the EITR registers (VF 0-31 are not multiplexed as described in Section 7.3.4.3.3). At 0b, it selects the PF and at 1b it selects the VF.

8.2.3.5.12 Extended Interrupt Throttle Registers — EITR[n] (0x00820 + 4*n, n=0...23 and 0x012300 + 4*(n-24), n=24...128; RW)

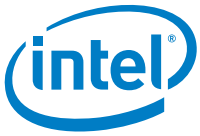
Mapping of the EITR registers to the MSI-X vectors is described in Section 7.3.4.3.3.

Field	Bit(s)	Init Val	Description
Reserved	2:0	000b	Reserved.
ITR Interval	11:3	0x0	Minimum inter-interrupt interval specified in 2 µs units at 1 Gb/2 and 10 Gb/s link. At 100 Mb/s link, the interval is specified in 20 µs units. At 0x0 interrupt throttling is disabled while any "event" causes an immediate interrupt.
Reserved	14:12	000b	Reserved.
LLI Moderation	15	0b	When set, LLI moderation is enabled. Otherwise, any LLI packet generates an immediate interrupt. LLI moderation might be set only if interrupt throttling is enabled by the *ITR Interval* field in this register and LLI moderation is enabled by the *LL Interval* field in the GPIE register.
LLI Credit	20:16	0x0	Reflects the current credits for associated interrupt. When CNT_WDIS is not set on a write cycle, this field must be set to 0x0.
ITR Counter	27:21	0x0	This field represents the seven MS bits (out of nine bits) of the ITR counter. It is a down counter that is loaded with an ITR interval value each time the associated interrupt is asserted. When the ITR counter reaches zero it stops counting and triggers an interrupt. On a write cycle, software must set this field to 0 if CNT_WDIS in this register is cleared (write enable to the ITR counter).
Reserved	30:28	000b	Reserved.
CNT_WDIS	31	0b	Write disable to the LLI credit and ITR counter. When set, the LLI credit and ITR counter are not overwritten by the write access. When cleared, software must set the LLI credit and ITR counter to zero, which enables an immediate interrupt on packet reception. This bit is write only. Always read as 0b.

8.2.3.5.13 L3 L4 Tuples Immediate Interrupt Rx — L34TIMIR[n] (0x0E800 + 4*n, n=0...127; RW)

This register must be initialized by software.

Field	Bit(s)	Init Val	Description
Reserved	11:0	X	Reserved.
Size_BP	12	X	Size Bypass. 0b = Size check is performed. 1b = Size check is bypassed.



Field	Bit(s)	Init Val	Description
Reserved	19:13	X	Reserved. Must be set to 1000000b on any programmed filter.
Low Latency Interrupt	20	X	Enables issuing a LLI when the following conditions are met: • The 5-tuple filter associated with this register matches. • If enabled by the Size_BP bit, the packet length is smaller than the length defined by LLITHRESH.SizeThresh.
Rx Queue	27:21	X	Identifies the Rx queue associated with this 5-tuple filter.
Reserved	31:28	X	Reserved.

8.2.3.5.14 LLI Size Threshold — LLITHRESH (0x0EC90; RW)

Field	Bit(s)	Init Val	Description
SizeThresh	11:0	0x000	Size Threshold. A packet with length below this threshold that matches one of the 5-tuple filters with an active *Low Latency Interrupt* flag in the L34TIMIR[n] registers might trigger an LLI.
Reserved	25:12	0x0	Reserved.
Reserved	31:26	000101b	Reserved.

8.2.3.5.15 Immediate Interrupt Rx VLAN Priority Register- IMIRVP (0x0EC60 / 0x05AC0; RW)

IMIRVP is also mapped to address 0x05AC0 to maintain compatibility with the 82598.

Field	Bit(s)	Init Val	Description
Vlan_Pri	2:0	000b	VLAN Priority. This field includes the VLAN priority threshold. When Vlan_pri_en is set to 1b, then an incoming packet with VLAN tag with a priority equal or higher to VlanPri must trigger a LLI, regardless of the ITR moderation.
Vlan_pri_en	3	0	VLAN Priority Enable. When 1b, an incoming packet with VLAN tag with a priority equal or higher to Vlan_Pri must trigger a LLI, regardless of the ITR moderation. When 0b, the interrupt is moderated by ITR.
Reserved	31:4		Reserved.

8.2.3.5.16 Interrupt Vector Allocation Registers — IVAR[n] (0x00900 + 4*n, n=0...63; RW)

These registers map interrupt causes into EICR entries (legacy/MSI modes) or into MSI-X vectors (MSI-X modes). See Section 7.3.4 for mapping and use of these registers.

Transmit and receive queues mapping to IVAR registers is shown in Figure 8-1:

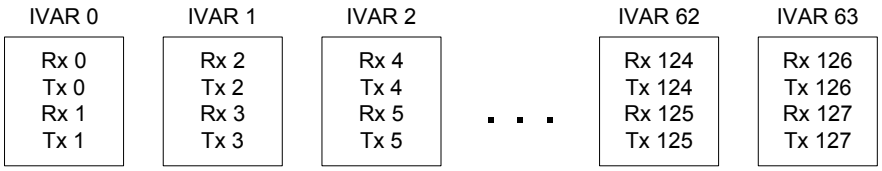
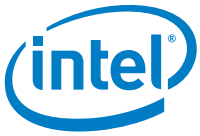


Figure 8-1 Transmit and Receive Queues Mapping to IVAR Registers

Fields of the IVAR registers are described in Table 8-5.

Table 8-5 Fields of IVAR Register

Field	Bit(s)	Init Val	Description
INT_Alloc[0]	5:0	X	The interrupt allocation for Rx queue (‘2xN’ for IVAR register ‘N’).
Reserved	6	0b	Reserved.
INT_Alloc_val[0]	7	0b	Interrupt allocation valid indication for INT_Alloc[0].
INT_Alloc[1]	13:8	X	The interrupt allocation for Tx queue (‘2xN’ for IVAR register ‘N’).
Reserved	14	0b	Reserved.
INT_Alloc_val[1]	15	0b	Interrupt allocation valid indication for INT_Alloc[1].
INT_Alloc[2]	21:16	X	The interrupt allocation for Rx queue (‘2xN’+1 for IVAR register ‘N’).
Reserved	22	0b	Reserved.
INT_Alloc_val[2]	23	0b	Interrupt allocation valid indication for INT_Alloc[2].
INT_Alloc[3]	29:24	X	The interrupt allocation for Tx queue (‘2xN’+1 for IVAR register ‘N’).
Reserved	30	0b	Reserved.
INT_Alloc_val[3]	31	0b	Interrupt allocation valid indication for INT_Alloc[3].



8.2.3.5.17 Miscellaneous Interrupt Vector Allocation — IVAR_MISC (0x00A00; RW)

These register maps interrupt causes into MSI-X vectors (MSI-X modes). See Section 7.3.4 for mapping and use of these registers.

Field	Bit(s)	Init Val	Description
INT_Alloc[0]	6:0	X	Defines the MSI-X vector assigned to the TCP timer interrupt cause.
INT_Alloc_val[0]	7	0b	Valid bit for INT_Alloc[0].
INT_Alloc[1]	14:8	X	Defines the MSI-X vector assigned to the other interrupt cause.
INT_Alloc_val[1]	15	1b	Valid bit for INT_Alloc[1].
Reserved	31:16	0b	Reserved.

Note: The INT_ALLOC_VAL[1] bit default value is one — to enable legacy driver functionality.

8.2.3.5.18 General Purpose Interrupt Enable — GPIE (0x00898; RW)

Field	Bit(s)	Init Val	Description
SDP0_GPIEN	0	0b	General Purpose Interrupt Detection Enable for SDP0. If software-controllable I/O pin SDP0 is configured as an input, this bit (when 1b) enables use for GPI interrupt detection.
SDP1_GPIEN	1	0b	General Purpose Interrupt Detection Enable for SDP1. If software-controllable I/O pin SDP1 is configured as an input, this bit (when 1b) enables use for GPI interrupt detection.
SDP2_GPIEN	2	0b	General Purpose Interrupt Detection Enable for SDP2. If software-controllable I/O pin SDP2 is configured as an input, this bit (when 1b) enables use for GPI interrupt detection.
SDP3_GPIEN	3	0b	General Purpose Interrupt Detection Enable for SDP3. If software-controllable I/O pin SDP3 is configured as an input, this bit (when 1b) enables use for GPI interrupt detection.
Multiple_MSIX	4	0b	MSI-X Mode. Selects between MSI-X interrupts and other interrupt modes. 0b = Legacy and MSI mode (non-MSI-X mode). 1b = MSI-X mode.
OCD	5	0b	Other Clear Disable. 0b = The entire EICR is cleared on read. 1b = Indicates that only bits 29:16 of the EICR register are cleared on read.
EIMEN	6	0b	EICS Immediate Interrupt Enable. When set, setting this bit in the EICS register causes a LLI. If not set, the EICS interrupt waits for EITR expiration.

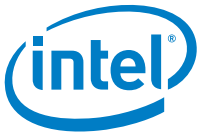
Field	Bit(s)	Init Val	Description
LL Interval	10:7	0x0	Low latency Credits Increment Rate. The interval is specified in 4 µs increments at 1 Gb/s and 10 Gb/s link. It is defined as 40 µs at 100 Mb/s link. A value of 0x0 disables moderation of LLI for all interrupt vectors. When LLI is disabled by the *LL Interval* bit, the *LLI Moderation* bit in all EITR registers must not be set.
RSC Delay	13:11	000b	Delay from RSC completion triggered by ITR and interrupt assertion. The delay = (RSC Delay + 1) x 4 µs = 4, 8, 12... 32 µs.
VT_Mode	15:14	00b	Specify the number of active VFs. Software must set this field the same as GCR_Ext.VT_Mode. 00b = Non-IOV mode. 10b = 32 VF mode. 01b = 16 VF mode. 11b = 64 VF mode.
Reserved	29:16	0x0	Reserved.
EIAME	30	0b	Extended Interrupt Auto Mask Enable. When set, the EIMS register can be auto-cleared (depending on EIAM setting) upon interrupt assertion. In any case, the EIMS register can be auto-cleared (depending on EIAM setting) following a write-to-clear (or read) to the EICR register. Software may set the EIAME only in MSI-X mode.
PBA_support	31	0b	PBA Support. When set, setting one of the extended interrupts masks via EIMS causes the *PBA* bit of the associated MSI-X vector to be cleared. Otherwise, the 82599 behaves in a way supporting legacy INT-x interrupts. *Note:* Should be cleared when working in INT-x or MSI mode and set in MSI-X mode.

The 82599 allows for up to four externally controlled interrupts. The lower four software-definable pins, SDP[3:0], can be mapped for use as GPI interrupt bits. These mappings are enabled by the SDPx_GPIEN bits only when these signals are also configured as inputs via SDPx_IODIR. When configured to function as external interrupt pins, a GPI interrupt is generated when the corresponding pin is sampled in an active-high state.

The bit mappings are listed in Table 8-6 for clarity.

Table 8-6 GPI-to-SDP Bit Mappings

SDP (pin to be used as GPI)	ESDP Field Settings		Resulting EICR Bit (GPI)
	Directionality	Enable as GPI interrupt	
3	SDP3_IODIR	SDP3_GPIEN	27
2	SDP2_IODIR	SDP2_GPIEN	26
1	SDP1_IODIR	SDP1_GPIEN	25
0	SDP0_IODIR	SDP0_GPIEN	24



8.2.3.6 MSI-X Table Registers

MSI-X capability is described in section Section 9.3.8. The MSI-X table is described in Section 9.3.8.2 and the Pending Bit Array (PBA) is described in Section 9.3.8.3. These registers are located in the MSI-X BAR.

8.2.3.6.1 MSI-X PBA Clear — PBACL[n] (0x110C0 + 4*n, n=0...7 / 0x11068 [n=0]; RW)

PBACL[0] is also mapped to address 0x11068 to maintain compatibility with the 82598.

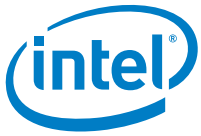
Field	Bit(s)	Init Val	Description
PENBITCLR	31:0	0x0	MSI-X Pending Bits Clear. Writing 1b to any bit clears it's content; writing 0b has no effect. Reading this register returns the MSIPBA.PENBIT value.

8.2.3.7 Receive Registers

8.2.3.7.1 Filter Control Register — FCTRL (0x05080; RW)

Field	Bit(s)	Init Val	Description
Reserved	0	0b	Reserved.
SBP	1	0b	Store Bad Packets. 0b = Do not store. 1b = Store. *Note:* CRC errors before the SFD are ignored. Any packet must have a valid SFD (RX_DV with no RX_ER in the XGMII/GMII i/f) in order to be recognized by the device (even bad packets). *Note:* Packets with errors are not routed to manageability even if this bit is set. *Note:* Erroneous packets can be routed to the default queue rather than the originally intended queue. *Note:* In packets shorter than 64 bytes, the checksum errors can be hidden while MAC errors are reported. *Note:* A packet with a valid error (caused by byte error or illegal error) might have data contamination in the last eight bytes when stored in the host memory if the *Store Bad Packet* bit is set.
Reserved	7:2	0x0	Reserved.
MPE	8	0b	Multicast Promiscuous Enable. 0b = Disabled. 1b = Enabled. When set, all received multicast and broadcast packets pass L2 filtering and can be directed to manageability or the host by a one of the decision filters.
UPE	9	0b	Unicast Promiscuous Enable. 0b = Disabled. 1b = Enabled.
BAM	10	0b	Broadcast Accept Mode. 0b = Ignore broadcast packets to host. 1b = Accept broadcast packets to host.
Reserved	31:11	0x0	Reserved.

Note: Before receive filters are updated/modified the RXCTRL.RXEN bit should be set to 0b. After the proper filters have been set the RXCTRL.RXEN bit can be set to 1b to re-enable the receiver.



8.2.3.7.2 VLAN Control Register — VLNCTRL (0x05088; RW)

Field	Bit(s)	Init Val	Description
VET	15:0	0x8100	VLAN Ether Type (the VLAN Tag Protocol Identifier — TPID). This register contains the type field hardware matches against to recognize an 802.1Q (VLAN) Ethernet packet. For proper operation, software must not change the default setting of this field (802.3ac standard defines it as 0x8100). This field must be set to the same value as the VT field in the DMATXCTL register. *Note:* This field appears in little endian order (the upper byte is first on the wire (VLNCTRL.VET[15:8]).
Reserved	27:16		Reserved.
CFI	28	0b	Canonical Form Indicator Bit Value. If CFIEN is set to 1b, then 802.1q packets with CFI equal to this field are accepted; otherwise, the 802.1q packet is discarded.
CFIEN	29	0b	Canonical Form Indicator Enable. 0b = Disabled (CFI bit not compared to decide packet acceptance). 1b = Enabled (CFI from packet must match next CFI field to accept 802.1q packet).
VFE	30	0b	VLAN Filter Enable. 0b = Disabled (filter table does not decide packet acceptance). 1b = Enabled (filter table decides packet acceptance for 802.1q packets).
Reserved	31	0b	Reserved.

8.2.3.7.3 Multicast Control Register — MCSTCTRL (0x05090; RW)

Field	Bit(s)	Init Val	Description
MO	1:0	00b	Multicast Offset. This determines which bits of the incoming multicast address are used in looking up the bit vector. 00b = [47:36]. 01b = [46:35]. 10b = [45:34]. 11b = [43:32].
MFE	2	0b	Multicast Filter Enable. 0b = The multicast table array filter (MTA[n]) is disabled. 1b = The multicast table array (MTA[n]) is enabled.
Reserved	31:3	0x0	Reserved.

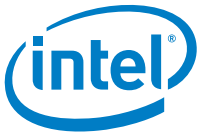
8.2.3.7.4 Packet Split Receive Type Register — PSRTYPE[n] (0x0EA00 + 4*n, n=0...63 / 0x05480 + 4*n, n=0...15; RW)

Registers 0...15 are also mapped to 0x05480 to maintain compatibility with the 82598.

Note:

- This register must be initialized by software.
- Packets are split according to the lowest-indexed entry that applies to the packet and that is enabled. For example, if bits 4 and 8 are set, then an IPv4 packet that is not TCP is split after the IPv4 header.
- This bit mask table enables or disables each type of header to be split. A value of 1b enables an entry.
- In virtualization mode, a separate PSRTYPE register is provided per pool up to the number of pools enabled. In non-virtualization mode, only PSRTYPE[0] is used.
- PSR_type4 should be set to enable RSC, regardless header split mode.

Field	Bit(s)	Init Val	Description
PSR_type0	0	x	Reserved.
PSR_type1	1	x	Split received NFS packets after NFS header.
PSR_type2	2	x	Reserved.
PSR_type3	3	x	Reserved.
PSR_type4	4	x	Split received TCP packets after TCP header.
PSR_type5	5	x	Split received UDP packets after UDP header.
PSR_type6	6	x	Reserved.
PSR_type7	7	x	Reserved.
PSR_type8	8	x	Split received IPv4 packets after IPv4 header.
PSR_type9	9	x	Split received IPv6 packets after IPv6 header.
PSR_type10	10	x	Reserved.
PSR_type11	11	x	Reserved.
PSR_type12	12	x	Split received L2 packets after L2 header.
PSR_type13	13	x	Reserved.
PSR_type14	14	x	Reserved.
PSR_type15	15	x	Reserved.
PSR_type16	16	x	Reserved.



Field	Bit(s)	Init Val	Description
PSR_type17	17	x	Reserved.
PSR_type18	18	x	Reserved.
Reserved	28:19	x	Reserved.
RQPL	31:29	x	Defines the number of bits to use for RSS redirection of packets dedicated to this pool. Valid values are zero, 0001b and 0010b. The default value should be 0010b, meaning that up to 4 queues can be enabled for this pool. A value of 0001b means that up to 2 queues can be enabled for this pool. A value of zero means that all the traffic of the pool is sent to queue 0 of the pool. This field is used only if MRQC.MRQE equals 1010b or 1011b.

8.2.3.7.5 Receive Checksum Control — RXCSUM (0x05000; RW)

Field	Bit(s)	Init Val	Description
Reserved	11:0	0x0	Reserved.
IPPCSE	12	0b	IP Payload Checksum Enable.
PCSD	13	0b	RSS/Fragment Checksum Status Selection. When set to 1b, the extended descriptor write back has the *RSS* field. When set to 0b, it contains the fragment checksum.
Reserved	31:14	0x0	Reserved.

The Receive Checksum Control register controls the receive checksum offloading features of the 82599. The 82599 supports the offloading of three receive checksum calculations: the fragment checksum, the IP header checksum, and the TCP/UDP checksum.

PCSD: The Fragment 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 Table 8-7.

Table 8-7 Checksum Enable/Disable

RXCSUM.PCSD	0b (Checksum Enable)	1b (Checksum Disable)
	Fragment checksum and IP identification are reported in the Rx descriptor.	RSS hash value is reported in the Rx descriptor.

IPPCSE: IPPCSE controls the fragment checksum calculation. As previously noted, the fragment checksum shares the same location as the *RSS* field. The fragment checksum is reported in the receive descriptor when the RXCSUM.PCSD bit is cleared.

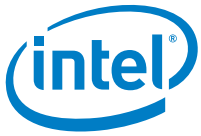
If RXCSUM.IPPCSE is cleared (the default value), the checksum calculation is not done and the value that is reported in the Rx fragment checksum field is 0b.

If RXCSUM.IPPCSE is set, the fragment checksum is aimed to accelerate checksum calculation of fragmented UDP packets. See Section 7.1.13 for a detailed explanation.

This register should only be initialized (written) when the receiver is not enabled (for example, only write this register when RXCTRL.RXEN = 0b).

8.2.3.7.6 Receive Filter Control Register — RFCTL (0x05008; RW)

Field	Bit(s)	Init Val	Description
Reserved	5:0	0x0	Reserved.
RSC_DIS	5	0	RSC Disable. When set, disable RSC for the port by the Rx filter unit. The default value is 0b (RSC feature is enabled).
NFSW_DIS	6	0b	NFS Write disable. Disable filtering of NFS write request headers.
NFSR_DIS	7	0b	NFS Read disable. Disable filtering of NFS read reply headers.
NFS_VER	9:8	00b	NFS version recognized by the hardware. 00b = NFS version 2 01b = NFS version 3 10b = NFS version 4 11b = Reserved for future use
IPv6_dis	10	0b	IPv6 Disable. Disable IPv6 packet filtering Internal use only – should not be set to 1b.
Reserved	11	0b	Reserved, always set to 0b.
Reserved	13:12	00b	Reserved.
IPFRSP_DIS	14	0b	IP Fragment Split Disable When this bit is set the header of IP fragmented packets are not set. Internal use only. Should not be set to 1b.
Reserved	15	0b	Reserved.
Reserved	17:16	00b	Reserved.
Reserved	31:18	0x0	Reserved. Should be written with 0x0 to ensure future compatibility.



8.2.3.7.7 Multicast Table Array — MTA[n] (0x05200 + 4*n, n=0...127; RW)

This table should be initialized by software before transmit and receive are enabled.

Field	Bit(s)	Init Val	Description
Bit Vector	31:0	X	Word wide bit vector specifying 32 bits in the multicast address filter table. The 82599 provides multicast filtering for 4096 multicast addresses by providing single-bit entry per multicast address. Those 4096 address locations are organized in the Multicast Table Array (MTA); 128 registers of 32 bits for each one. Only 12 bits out of the 48-bit destination address are considered as a multicast address. Those 12 bits can be selected by the *MO* field of MCSTCTRL register. The 7 MS bits of the Ethernet MAC address (out of the 12 bits) selects the register index while the 5 LS bits (out of the 12 bits) selects the bit within a register.

8.2.3.7.8 Receive Address Low — RAL[n] (0x0A200 + 8*n, n=0...127; RW)

While "n" is the exact unicast/multicast address entry and it is equals to 0,1,...127.

Field	Bit(s)	Init Val	Description
RAL	31:0	X	Receive Address Low. The lower 32 bits of the 48-bit Ethernet MAC address. *Note:* Field is defined in big endian (LS byte of RAL is first on the wire).

These registers contain the lower bits of the 48-bit Ethernet MAC address. All 32 bits are valid.

If the EEPROM is present, the first register (RAL0) is loaded from the EEPROM.

8.2.3.7.9 Receive Address High — RAH[n] (0x0A204 + 8*n, n=0...127; RW)

While "n" is the exact unicast/multicast address entry and it is equals to 0,1,...127.

Field	Bit(s)	Init Val	Description
RAH	15:0	X	Receive Address High. The upper 16 bits of the 48 bit Ethernet MAC Address. *Note:* Field is defined in Big Endian (MS byte of RAH is Last on the wire).
Reserved	21:16	0x0	Reserved.

Field	Bit(s)	Init Val	Description
Reserved	30:22	0x0	Reserved. Reads as 0. Ignored on write.
AV	31	X (see desc.)	Address Valid. All receive addresses are not initialized by hardware and software should initialize them before receive is enabled. If the EEPROM is present, Receive Address[0] is loaded from the EEPROM and its *Address Valid* field is set to 1b after a software, PCI reset or EEPROM read.

These registers contain the upper bits of the 48-bit Ethernet MAC address. The complete address is {RAH, RAL}. AV determines whether this address is compared against the incoming packet. AV is cleared by a master reset.

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

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

8.2.3.7.10 MAC Pool Select Array — MPSAR[n] (0x0A600 + 4*n, n=0...255; RW)

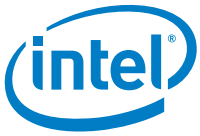
Software should initialize these registers before transmit and receive are enabled.

Field	Bit(s)	Init Val	Description
POOL_ENA	31:0	X	Pool Enable Bit Array. Each couple of registers ‘2*n’ and ‘2*n+1’ are associated with Ethernet MAC address filter ‘n’ as defined by RAL[n] and RAH[n]. Each bit when set, enables packet reception with the associated pools as follows: Bit ‘i’ in register ‘2*n’ is associated with POOL ‘i’. Bit ‘i’ in register ‘2*n+1’ is associated with POOL ‘32+i’.

8.2.3.7.11 VLAN Filter Table Array — VFTA[n] (0x0A000 + 4*n, n=0...127; RW)

This table should be initialized by software before transmit and receive are enabled.

Field	Bit(s)	Init Val	Description
VLAN_FLT	31:0	X	VLAN Filter. Each bit ‘i’ in register ‘n’ affects packets with VLAN tags equal to 32*n+i. 128 VLAN Filter registers compose a table of 4096 bits that cover all possible VLAN tags. Each bit when set, enables packets with the associated VLAN tags to pass. Each bit when cleared, blocks packets with this VLAN tag.



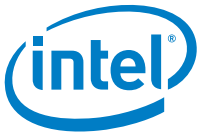
8.2.3.7.12 Multiple Receive Queues Command Register- MRQC (0x0EC80 / 0x05818; RW)

MRQC is also mapped to address 0x05818 to maintain compatibility with the 82598.

Field	Bit(s)	Init Val	Description
MRQE	3:0	0x0	Multiple Receive Queues Enable. Defines the allocation of the Rx queues per RSS, virtualization and DCB. 0000b = RSS disabled. 0001b = RSS only — Single set of RSS 16 queues. 0010b = DCB enabled and RSS disabled — 8 TCs, each allocated 1 queue. 0011b = DCB enabled and RSS disabled — 4 TCs, each allocated 1 queue. 0100b = DCB and RSS — 8 TCs, each allocated 16 RSS queues. 0101b = DCB and RSS — 4 TCs, each allocated 16 RSS queues. 0110b = Reserved 0111b = Reserved 1000b = Virtualization only — 64 pools, no RSS, each pool allocated 2 queues. 1001b = Reserved 1010b = Virtualization and RSS — 32 pools, each allocated 4 RSS queues. 1011b = Virtualization and RSS — 64 pools, each allocated 2 RSS queues. 1100b = Virtualization and DCB — 16 pools, each allocated 8 TCs. 1101b = Virtualization and DCB — 32 pools, each allocated 4 TCs. 1110b = Reserved 1111b = Reserved
Reserved	14:4	0x0	Reserved.
Reserved	15	0x0	Reserved.
RSS Field Enable	31:16	0x0	Each bit, when set, enables a specific field selection to be used by the hash function. Several bits can be set at the same time. Bit[16] = Enable TcpIPv4 hash function. Bit[17] = Enable IPv4 hash function. Bit[19:18] = Reserved Bit[20] = Enable IPv6 hash function. Bit[21] = Enable TcpIPv6 hash function. Bit[22] = Enable UdpIPV4. Bit[23] = Enable UdpIPV6. Bits[31:24] = Reserved *Note:* On Tunnel packets IPv4-IPv6 only the IPv4 header can be used for the RSS filtering.

8.2.3.7.13 RSS Queues Per Traffic Class Register — RQTC (0x0EC70; RW)

Field	Bit(s)	Init Val	Description
RQTC0	2:0	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to Traffic Class (TC) 0. A value of zero means that all the traffic of TC0 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	3	0b	Reserved.
RQTC1	6:4	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to TC 1. A value of zero means that all the traffic of TC1 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	7	0b	Reserved.
RQTC2	10:8	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to TC 2. A value of zero means that all the traffic of TC2 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	11	0b	Reserved.
RQTC3	14:12	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to TC 3. A value of zero means that all the traffic of TC3 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	15	0b	Reserved.
RQTC4	18:16	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to TC 4. A value of zero means that all the traffic of TC4 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	19	0b	Reserved.
RQTC5	22:20	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to TC 5. A value of zero means that all the traffic of TC5 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	23	0b	Reserved.
RQTC6	26:24	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to TC 6. A value of zero means that all the traffic of TC6 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	27	0b	Reserved.
RQTC7	30:28	0x4	Defines the number of bits to use for RSS redirection of packets dedicated to TC 7. A value of zero means that all the traffic of TC7 is sent to queue 0 of the TC. This field is used only if MRQC.MRQE equals 0100b or 0101b.
Reserved	31	0b	Reserved.



8.2.3.7.14 RSS Random Key Register — RSSRK (0x0EB80 + 4*n, n=0...9 / 0x05C80 + 4*n, n=0...9; RW)

RSSRK is also mapped to addresses 0x05C80-0x05CA4 to maintain compatibility with the 82598. The RSS Random Key is 40 bytes wide (see RSS hash in Section 7.1.2.8.1).

Field	Bit(s)	Init Val	Description
K0	7:0	0x0	RSS Key Byte '4*n+0' of the RSS random key, for each register 'n'.
K1	15:8	0x0	RSS Key Byte '4*n+1' of the RSS random key, for each register 'n'.
K2	23:16	0x0	RSS Key Byte '4*n+2' of the RSS random key, for each register 'n'.
K3	31:24	0x0	RSS Key Byte '4*n+3' of the RSS random key, for each register 'n'.

8.2.3.7.15 Redirection Table — RETA[n] (0x0EB00 + 4*n, n=0...31 / 0x05C00 + 4*n, n=0...31; RW)

RETA is also mapped to addresses 0x05C00-0x05C7C to maintain compatibility with the 82598. The redirection table has 128-entries in 32 registers.

Field	Bit(s)	Init Val	Description
Entry0	3:0	X	Entry0 defines the RSS output index for hash value '4*n+0'. While 'n' is the register index, equals to 0...31.
Reserved	7:4	0x0	Reserved.
Entry1	11:8	X	Entry1 defines the RSS output index for hash value '4*n+1'. While 'n' is the register index, equals to 0...31.
Reserved	15:12	0x0	Reserved.
Entry2	19:16	X	Entry2 defines the RSS output index for hash value '4*n+2'. While 'n' is the register index, equals to 0...31.
Reserved	23:20	0x0	Reserved.
Entry3	27:24	X	Entry3 defines the RSS output index for hash value '4*n+3'. While 'n' is the register index, equals to 0...31.
Reserved	31:28	0x0	Reserved.

The contents of the redirection table are not defined following reset of the Memory Configuration registers. System software must initialize the table prior to enabling multiple receive queues. It can also update the redirection table during run time. Such updates of the table are not synchronized with the arrival time of received packets. Therefore, it is not guaranteed that a table update takes effect on a specific packet boundary.

8.2.3.7.16 Source Address Queue Filter — SAQF[n] (0x0E000 + 4*n, n=0...127; RW)

This register must be initialized by software

Field	Bit(s)	Init Val	Description
Source Address	31:0	X	IP Source Address. Part of the 5-tuple queue filters. *Note:* Field is defined in big endian (LS byte is first on the wire).

8.2.3.7.17 Destination Address Queue Filter — DAQF[n] (0x0E200 + 4*n, n=0...127; RW)

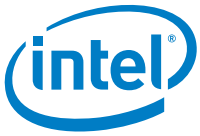
This register must be initialized by software.

Field	Bit(s)	Init Val	Description
Destination Address	31:0	X	IP Destination Address. Part of the 5-tuple queue filters. *Note:* Field is defined in big endian (LS byte is first on the wire).

8.2.3.7.18 Source Destination Port Queue Filter — SDPQF[n] (0x0E400 + 4*n, n=0...127; RW)

This register must be initialized by software.

Field	Bit(s)	Init Val	Description
Source Port	15:0	X	TCP/UDP Source Port. Part of the 5-tuple queue filters. *Note:* Field is defined in Big Endian (LS byte is first on the wire).
Destination Port	31:16	X	TCP/UDP Destination Port. Part of the 5-tuple queue filters.



8.2.3.7.19 Five tuple Queue Filter — FTQF[n] (0x0E600 + 4*n, n=0...127; RW)

Field	Bit(s)	Init Val	Description
Protocol	1:0	X	IP L4 protocol, part of the 5-tuple queue filters. 00b = TCP. 01b = UDP. 10b = SCTP. 11b = Other. *Note:* Encoding of the protocol type for the 128 x 5-tuple filters is defined differently than the L4TYPE encoding for the flow director filters.
Priority	4:2	X	Priority value in case more than one 5-tuple filter matches. 000b = Reserved 001b = Lowest priority. ... 111b = Highest priority.
Reserved	7:5	X	Reserved.
Pool	13:8	X	The pool Index of the pool associated with this filter.
Reserved	24:14	X	Reserved for extension of the *Pool* field.
Mask	29:25	X	Mask bits for the 5-tuple fields (1b = don't compare). The corresponding field participates in the match if the following bit is cleared: Bit 25 = Mask source address comparison. Bit 26 = Mask destination address comparison. Bit 27 = Mask source port comparison. Bit 28 = Mask destination port comparison. Bit 29 = Mask protocol comparison.
Pool Mask	30	X	Mask bit for the *Pool* field. When set to 1b, the *Pool* field is not compared as part of the 5-tuple filter. Software can clear (activate) the *Pool Mask* bit only when operating in IOV mode.
Queue Enable	31	X	When set, enables filtering of Rx packets by the 5-tuple defined in this filter to the queue indicated in register L34TIMIR. *Note:* There are 128 different 5-tuple filter configuration registers sets, with indexes [0] to [127]. The mapping to a specific Rx queue is done by the Rx Queue field in the L34TIMIR register, and not by the index of the register set.

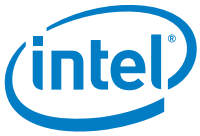
8.2.3.7.20 SYN Packet Queue Filter — SYNQF (0x0EC30; RW)

Field	Bit(s)	Init Val	Description
Queue Enable	0	0b	When set, enables routing of Rx packets to the queue indicated in this register.
Rx Queue	7:1	0x0	Identifies an Rx queue associated with SYN packets.
Reserved	9:8	00b	Reserved for extension of the *Rx Queue* field.
Reserved	30:10	0x0	Reserved.
SYNQFP	31	0b	Defines the priority between SYNQF and 5-tuples filter. 0b = 5-tuple filter priority 1b = SYN filter priority.

8.2.3.7.21 EType Queue Filter — ETQF[n] (0x05128 + 4*n, n=0...7; RW)

See Section 7.1.2.3 for more details on the use of this register.

Field	Bit(s)	Init Val	Description
EType	15:0	0x0	Identifies the protocol running on top of IEEE 802. Used to route Rx packets containing this EtherType to a specific Rx queue. *Note:* Field is defined in little endian (MS byte is first on the wire).
UP	18:16	0x0	User Priority. A 802.1Q UP value to be compared against the *User Priority* field in the Rx packet. Enabled by the *UP Enable* bit.
UP Enable	19	0b	User Priority Enable. Enables comparison of the *User Priority* field in the received packet.
Pool	25:20	0x0	In virtualization modes, determines the target pool for the packet.
Pool Enable	26	0b	In virtualization modes, enables the *Pool* field.
FCoE	27	0b	When set, packets that match this filter are identified as FCoE packets.
Reserved	28	0b	Reserved.
Reserved	29	0b	Reserved.
1588 time stamp	30	0b	When set, packets with this EType are time stamped according to the IEEE 1588 specification.
Filter Enable	31	0b	0b = The filter is disabled for any functionality. 1b = The filter is enabled. Exact actions are determined by separate bits.



8.2.3.7.22 EType Queue Select — ETQS[n] (0x0EC00 + 4*n, n=0...7; RW)

See Section 7.1.2.3 for more details on the use of this register.

Field	Bit(s)	Init Val	Description
Reserved	15:0	0x0	Reserved.
Rx Queue	22:16	0x0	Identifies the Rx queue associated with this EType.
Reserved	24:23	0x0	Reserved for future extension of the *Rx Queue* field.
Reserved	28:25	0x0	Reserved.
Low Latency Interrupt	29	0b	When set, packets that match this filter generate a LLI.
Reserved	30	0x0	Reserved.
Queue Enable	31	0b	When set, enables filtering of Rx packets by the EType defined in this register to the queue indicated in this register.

8.2.3.7.23 Rx Filter ECC Err Insertion 0 — RXFECCERR0 (0x051B8; RW)

Field	Bit(s)	Init Val	Description
Reserved	8:0	0x1FF	Reserved.
ECCFLT_EN	9	0b	Filter ECC Error indication Enablement. When set to 1b, enables the ECC-INT + the RXF-blocking during ECC-ERR in one of the Rx filter memories. At 0b, the ECC logic can still function overcoming only single errors while dual or multiple errors can be ignored silently.
Reserved	31:10	0x0	Reserved.

8.2.3.8 Receive DMA Registers

8.2.3.8.1 Receive Descriptor Base Address Low — RDBAL[n] (0x01000 + 0x40*n, n=0...63 and 0x0D000 + 0x40*(n-64), n=64...127; RW)

Field	Bit(s)	Init Val	Description
0	6:0	0x0	Ignored on writes. Returns 0x0 on reads.
RDBAL	31:7	X	Receive Descriptor Base Address Low.

This register contains the lower bits of the 64-bit descriptor base address. The lower 7 bits are always ignored. The receive descriptor base address must point to a 128 byte-aligned block of data.

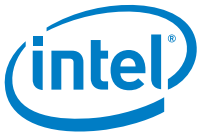
8.2.3.8.2 Receive Descriptor Base Address High — RDBAH[n] (0x01004 + 0x40*n, n=0...63 and 0x0D004 + 0x40*(n-64), n=64...127; RW)

Field	Bit(s)	Init Val	Description
RDBAH	31:0	X	Receive Descriptor Base Address [63:32].

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

8.2.3.8.3 Receive Descriptor Length — RDLEN[n] (0x01008 + 0x40*n, n=0...63 and 0x0D008 + 0x40*(n-64), n=64...127; RW)

Field	Bit(s)	Init Val	Description
LEN	19:0	0x0	Descriptor Ring Length. This register sets the number of bytes allocated for descriptors in the circular descriptor buffer. It must be 128-byte aligned (7 LS bit must be set to zero). *Note:* Validated lengths up to 128 K (8 K descriptors).
Reserved	31:20	0x0	Reads as 0x0. Should be written to 0x0 for future compatibility.



8.2.3.8.4 Receive Descriptor Head — RDH[n] (0x01010 + 0x40*n, n=0...63 and 0x0D010 + 0x40*(n-64), n=64...127; RO)

Field	Bit(s)	Init Val	Description
RDH	15:0	0x0	Receive Descriptor Head. This register holds the head pointer for the receive descriptor buffer in descriptor units (16-byte datum). The RDH is controlled by hardware.
Reserved	31:16	0x0	Reserved. Should be written with 0x0.

8.2.3.8.5 Receive Descriptor Tail — RDT[n] (0x01018 + 0x40*n, n=0...63 and 0x0D018 + 0x40*(n-64), n=64...127; RW)

Field	Bit(s)	Init Val	Description
RDT	15:0	0x0	Receive Descriptor Tail.
Reserved	31:16	0x0	Reads as 0x0. Should be written to 0x0 for future compatibility.

This register contains the tail pointer for the receive descriptor buffer. The register points to a 16-byte datum. Software writes the tail register to add receive descriptors to the hardware free list for the ring.

Note: The tail pointer should be set to one descriptor beyond the last empty descriptor in host descriptor ring.

8.2.3.8.6 Receive Descriptor Control — RXDCTL[n] (0x01028 + 0x40*n, n=0...63 and 0x0D028 + 0x40*(n-64), n=64...127; RW)

Field	Bit(s)	Init Val	Description
Reserved	13:0	0x0	Reserved.
Reserved	14	0b	Reserved (softwaremight read and write in order to maintain backward compatibility.)
Reserved	15	0b	Reserved.
Reserved	22:16	0x0	Reserved (software might read and write in order to maintain backward compatibility).
Reserved	24:23	00b	Reserved.
ENABLE	25	0b	Receive Queue Enable. When set, the *ENABLE* bit enables the operation of the specific receive queue. Upon read it gets the actual status of the queue (internal indication that the queue is actually enabled/disabled).

Field	Bit(s)	Init Val	Description
Reserved	26	0b	Reserved (software can read and write in order to maintain backward compatibility).
Reserved	29:27	0x0	Reserved.
VME	30	0b	VLAN Mode Enable. 0b = Do not strip VLAN tag. 1b = Strip VLAN tag from received 802.1Q packets destined to this queue.
Reserved	31	0b	Reserved.

8.2.3.8.7 Split Receive Control Registers — SRRCTL[n] (0x01014 + 0x40*n, n=0...63 and 0x0D014 + 0x40*(n-64), n=64...127 / 0x02100 + 4*n, [n=0...15]; RW)

SRRCTL[0...15] are also mapped to address 0x02100... to maintain compatibility with the 82598.

Field	Bit(s)	Init Val	Description
BSIZEPACKET	4:0	0x2	Receive Buffer Size for Packet Buffer. The value is in 1 KB resolution. Value can be from 1 KB to 16 KB. Default buffer size is 2 KB. This field should not be set to 0x0. This field should be greater or equal to 0x2 in queues where RSC is enabled.
Rsv	7:5	000b	Reserved. Should be written with 000b to ensure future compatibility.
BSIZEHEADER	13:8	0x4	Receive Buffer Size for Header Buffer. The value is in 64 bytes resolution. Value can be from 64 bytes to 1024 bytes. *Note:* The maximum supported header size is limited to 1023. Default buffer size is 256 bytes. This field must be greater than zero if the value of DESCTYPE is greater or equal to two. Values above 1024 bytes are reserved for internal use only.
Reserved	21:14	0x0	Reserved.
RDMTS	24:22	000b	Receive Descriptor Minimum Threshold Size. A LLI associated with this queue is asserted each time the number of free descriptors is decreased to RDMTS * 64 (this event is considered as Rx ring buffer almost empty).
DESCTYPE	27:25	000b	Define the descriptor type in Rx: 000b = Legacy. 001b = Advanced descriptor one buffer. 010b = Advanced descriptor header splitting. 011b = Reserved. 100b = Reserved. 101b = Advanced descriptor header splitting always use header buffer. 110b = Reserved. 111b = Reserved.