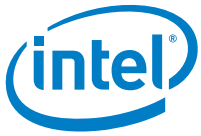


  - Split Receive Control (SRRCTL[0-127]): The *Drop_En* bit should be set per receive queue according to the required drop / no-drop policy of the TC of the queue.
- Tx descriptor plane control and status (RTTDCS) bits:
  - TDPAC=1b, VMPAC=1b, TDRM=1b, BDPM=0b if Tx rate limiting is not enabled and 1b if Tx rate limiting is enabled, BPBFSM=0b.
- Disable VM arbitration layer:
  - Clear RTTDT1C register, per each queue, via setting RTTDQSEL first
  - RTTDCS.VMPAC=0b

4.6.11.3.3 DCB-Off, VT-On

Set the configuration bits as specified in Section 4.6.11.3.1 with the following exceptions:

- Disable multiple packet buffers and allocate all queues to PB0:
  - RXPBSIZE[0].SIZE=0x200, RXPBSIZE[1-7].SIZE=0x0
  - TXPBSIZE[0].SIZE=0xA0, TXPBSIZE[1-7].SIZE=0x0
  - TXPBTHRESH.THRESH[0]=0xA0 — Maximum expected Tx packet length in this TC TXPBTHRESH.THRESH[1-7]=0x0
  - MRQC and MTQC
    - Set MRQE to 1xxxb, with the three least significant bits set according to the number of VFs and RSS mode
    - Clear *RT_Ena* bit and set the *VT_Ena* bit in the MTQC register.
    - Set MTQC.NUM_TC_OR_Q according to the number of VFs enabled
  - Set PFVTCTL.VT_Ena (as the MRQC.VT_Ena)
  - Rx UP to TC (RTRUP2TC), UPnMAP=0b, n=0,...,7
  - Tx UP to TC (RTTUP2TC), UPnMAP=0b, n=0,...,7
  - DMA TX TCP Maximum Allowed Size Requests (DTXMXSZRQ) — set Max_byte_num_req = 0xFFF = 1 MB
- Disable PFC and enabled legacy flow control:
  - Disable receive PFC via: MFLCN.RPFCE=0b
  - Enable transmit legacy flow control via: FCCFG.TFCE=01b
  - Enable receive legacy flow control via: MFLCN.RFCE=1b
- Configure VM arbiters only, reset others:
  - Tx Descriptor Plane T1 Config (RTTDT1C) *per pool*, via setting RTTDQSEL first for the pool index. Clear RTTDT1C for other queues. Note that the RTTDT1C for queue zero must always be initialized.
  - Clear RTTDT2C[0-7] registers
  - Clear RTTPT2C[0-7] registers

  - Clear RTRPT4C[0-7] registers

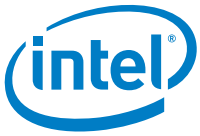
- Disable TC arbitrations while enabling the packet buffer free space monitor:
  - Tx Descriptor Plane Control and Status (RTTDCS), bits:

    TDPAC=0b, VMPAC=1b, TDRM=0b, BDPM=1b, BPBFSM=0b
  - Tx Packet Plane Control and Status (RTTPCS): TPPAC=0b, TPRM=0b, ARBD=0x224
  - Rx Packet Plane Control and Status (RTRPCS): RAC=0b, RRM=0b

4.6.11.3.4 DCB-Off, VT-Off

Set the configuration bits as specified in Section 4.6.11.3.1 with the following exceptions:

- Disable multiple packet buffers and allocate all queues and traffic to PB0:
  - RXPBSIZE[0].SIZE=0x200, RXPBSIZE[1-7].SIZE=0x0
  - TXPBSIZE[0].SIZE=0xA0, TXPBSIZE[1-7].SIZE=0x0
  - TXPBTHRESH.THRESH[0]=0xA0 — Maximum expected Tx packet length in this TC TXPBTHRESH.THRESH[1-7]=0x0
  - MRQC and MTQC
    - Set MRQE to 0xxxb, with the three least significant bits set according to the RSS mode
    - Clear both *RT_Ena* and *VT_Ena* bits in the MTQC register.
    - Set MTQC.*NUM_TC_OR_Q* to 00b.
  - Clear *PFVTCTL.VT_Ena* (as the MRQC.VT_Ena)
  - Rx UP to TC (RTRUP2TC), UPnMAP=0b, n=0,...,7
  - Tx UP to TC (RTTUP2TC), UPnMAP=0b, n=0,...,7
  - DMA TX TCP Maximum Allowed Size Requests (DTXMXSZRQ) — set Max_byte_num_req = 0xFFF = 1 MB
- Allow no-drop policy in Rx:
  - PFQDE: The *QDE* bit should be set to 0b in the PFQDE register for all queues enabling per queue policy by the SRRCTL[n] setting.
  - Split Receive Control (SRRCTL[0-127]): The *Drop_En* bit should be set per receive queue according to the required drop / no-drop policy of the TC of the queue.
- Disable PFC and enable legacy flow control:
  - Disable receive PFC via: MFLCN.RPFCE=0b
  - Enable receive legacy flow control via: MFLCN.RFCE=1b
  - Enable transmit legacy flow control via: FCCFG.TFCE=01b
- Reset all arbiters:
  - Clear RTTDT1C register, per each queue, via setting RTTDQSEL first



  - Clear RTTDT2C[0-7] registers
  - Clear RTTPT2C[0-7] registers
  - Clear RTRPT4C[0-7] registers
- Disable TC and VM arbitration layers:
  - Tx Descriptor Plane Control and Status (RTTDCS), bits:

    TDPAC=0b, VMPAC=0b, TDRM=0b, BDPM=1b, BPBFSM=1b
  - Tx Packet Plane Control and Status (RTTPCS): TPPAC=0b, TPRM=0b, ARBD=0x224
  - Rx Packet Plane Control and Status (RTRPCS): RAC=0b, RRM=0b

4.6.11.4 Transmit Rate Scheduler

In some applications it might be useful to setup rate limiters on Tx queues for other usage models (rate-limiting VF traffic for instance). In all cases, setting a rate limiter on Tx queue N to a TargetRate requires the following settings:

Global Setting

- The Transmit Rate-scheduler memory for all transmit queues must be cleared before rate limiting is enabled on any queue. This memory is accessed by the RTTBCNRC register mapped by the RTTDQSEL.TXDQ_IDX.
- Set global transmit compensation time to the MMW_SIZE in RTTBCNRM register. Typically MMW_SIZE=0x014 if 9.5 KB (9728-byte) jumbo is supported and 0x004 otherwise.

Per Queue Setting

- Select the requested queue by programming the queue index - RTTDQSEL.TXQ_IDX
- Program the desired rate as follow:
  - Compute the Rate_Factor which equals Link_Speed / Target_Rate. Link_Speed could be either 10 Gb/s or 1 Gb/s. Note that the Rate_Factor is composed of an integer number plus a fraction. The integer part is a 10 bit number field and the fraction part is a 14 bit binary fraction number.
  - Integer (Rate_Factor) is programmed by the RTTBCNRC.RF_INT[9:0] field
  - Fraction (Rate_Factor) is programmed by the RTTBCNRC.RF_DEC[13:0] field. It equals RF_DEC[13] * 2^{-1} + RF_DEC[12] * 2^{-2} + ... + RF_DEC[0] * 2^{-14}
- Enable Rate Scheduler by setting the RTTBCNRC. RS_ENA

Numerical Example

- Target_Rate = 240 Mb/s; Link_Speed = 10 Gb/s
- Rate_Factor = 10 / 0.24 = 41.6666... = 101001.10101010101011b
- RF_DEC = 10101010101011b; RF_INT = 0000101001b
- Therefore, set RTTBCNRC to 0x800A6AAB

Note: The IPG pacing feature is a parallel feature to the Tx rate scheduler where IPG pacing is applied to the entire Tx data flow while the Tx rate scheduler is applied separately to each Tx queue. Therefore, if a single queue is used, either feature can be used to limit the Tx data rate; however, if multiple queues are used, the IPG pacing feature is a better choice for a homogeneous Tx data rate limitation.

4.6.11.5 Configuration Rules

4.6.11.5.1 TC Parameters

Traffic Class

Per 802.1p, priority #7 is the highest priority.

A specific TC can be configured to receive or transmit a specific amount of the total bandwidth available per port.

Bandwidth allocation is defined as a fraction of the total available bandwidth, which can be less than the full Ethernet link bandwidth (if it is bounded by the PCIe bandwidth or by flow control).

Low latency TC should be configured to use the highest priority TC possible (TC 6, 7). The lowest latency is achieved using TC7.

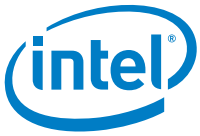
Bandwidth Group (BWGs)

The main reason for having BWGs is to represent different traffic types. A traffic type (such as storage, IPC LAN or manageability) can have more than one TC (for example, one for control traffic and one for the raw data), by grouping these two TC to a BWG the user can allocate bandwidth to the storage traffic so that unused bandwidth by the control could be used by the data and vise versa. This BWG concept supports the converged fabric as each traffic type, that is used to run on a different fabric, can be configured as a BWG and gets its resources as if it was on a different fabric.

1. To configure DCB not to share bandwidth between TCs, each TC should be configured as a separate BWG.
2. There are no limits on the TCs that can be bundled together as a BWG. All TCs can be configured as a single BWG.
3. BWG numbers should be sequential starting from zero until the total number of BWGs minus one.
4. BWG numbers do not imply priority, priority is only set according to TCs.

Refill Credits

Refill credits regulate the bandwidth allocated to BWG and TC. The ratio between the credits of the BWG's represents the relative bandwidth percentage allocated to each BWG. The ratio between the credits of the TC's represents the relative bandwidth percentage allocated to each TC within a BWG.



Credits are configured and calculated using 64 bytes granularity.

1. In any case, the number of refill credits assigned per TC should be as small as possible but must be larger than the maximum frame size used and larger than 1.5 KB. Using a lower refill value causes more refill cycles before a packet can be sent. These extra cycles unnecessarily increase the latency.
2. Refill credits ratio between TCs should be equal to the desired ratio of bandwidth allocation between the different TCs. Applying rule #1, means bandwidth shares are sorted from the smaller to the bigger, and just one maximum sized frame is allocated to the smallest.
3. The ratio between the refill credits of any two TCs should not be greater than 100.
4. Exception to rule #2 — TCs that require low latency should be configured so that they are under subscribed. For example, credit refill value should provide these TCs somewhat more bandwidth than what they actually need. Low latency TCs should always have credits so they can be next in line for the WSP arbitration.

   This exception causes the low latency TC to always have maximum credits (as it starts with maximum credits and on average cycle uses less than the refill credits).

The end point that is sending/receiving packets of 127 bytes eventually gets double the bandwidth it was configured to, as we do all the credit calculation by rounding the values down to the next 64 byte aligned value.

Maximum Credit Limit

The maximum credit limit value establishes a limit for the number of credits that a TC or BWG can own at any given time. This value prevents stacking up stale credits that can be added up over a relatively long period of time and then used by TCs all at once, altering fairness and latency.

Maximum credits limits are configured and calculated using 64 bytes granularity.

1. Maximum credit limit should be bigger than the refill credits allocated to the TC.
2. Maximum credit limit should be set to be as low as possible while still meeting other rules to minimize the latency impact on low latency TCs.
3. If a low latency TC generates a burst that is larger than its maximum credit limit this TC might experience higher latency since the TC needs to wait for allocation of additional credits because it finished all its credits for this cycle. Therefore maximum credit limit for a low latency TC must be set bigger than the maximum burst length of traffic expected on that TC (for all the VMs at once). If TC7 and TC6 are for low latency traffic, it leads to:

   ```
   Max(TC7,6) >= MaxBurst(TC7,6) served with low latency
   ```

4. An arbitration cycle can extend when one or more TCs accumulate credits more than their refill values (up to their maximum credit limit). For such a case, a low latency TC should be provided with enough credits to cover for the extended cycle duration. Since the low latency TC operates at maximum credits (see rule #3) its maximum credit limit should meet the following formula:

   ```
   {Max(TCx)/SUMi=0..7[Max(TCi)]} >= {BW(TCx)/Full BW}
   ```

   The formula applies to both descriptor arbiter and data arbiter.

5. When in a virtualized environment, the low latency TC condition checked by the VM WRR arbiter (see Section 7.7.2.3.2) induces the following relation between the maximum credits of a low latency TC and the refill credits of its attached VM arbiter:

```
Max(TCx) >= 2 x {SUMi=0...15[Refill(VMi)]}
```

6. To ensure bandwidth for low priority TC (when those are allocated with most of the bandwidth) the maximum credit value of the low priority TC in the data arbiter needs to be high enough to ensure sync between the two arbiters. In the equation that follows the bandwidth numbers are from the descriptor arbiter while the maximum values are of the data arbiter.

```
{Max(TCx)/SUMi=x+1..7[Max(TCi)]} >= {BW(TCx)/Full_PCIE_BW}
```

Note that the previous equation is worst case and covers the assumption that all higher TCs have the full maximum to transmit.

Tip: A simplified maximum credits allocation scheme would be to find the minimum number N >= 2 such that rules #3 and #5 are respected, and allocate

```
Max(TCi) = N x Refill(TCi), for i=0...7
```

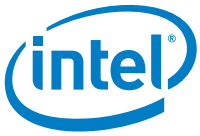
By maintaining the same ratios between the maximum credits and the bandwidth shares, the bandwidth allocation scheme is made more immune to disturbing events such as reception of priority pause frames with short timer values.

GSP and LSP

TC Link Strict Priority (TC.LSP): This bit specifies that the configured TC can transmit without any restriction of credits. This effectively means that the TC can take up entire link bandwidth, unless preempted by higher priority traffic. The Tx queues associated with LSP TC must be set as *Strict Low Latency* in the TXLLQ[n] registers.

TC Strict Priority within group (TC.GSP): This bit defines whether strict priority is enabled or disabled for this TC within its BWG. If TC.GSP is set to 1b, the TC is scheduled for transmission using strict priority. It does not check for availability of credits in the TC. It does check whether the BWG of this TC has credits. For example, the amount of traffic generated from this TC is still limited by the BWG allocated for the BWG.

1. TC's with the *LSP* bit set should be the first to be considered by the scheduler. This implies that *LSP* should be configured to the highest priority TC's. For example, starting from priority 7 and down. The other TC's should be used for groups with bandwidth allocation. It is recommended to use LSP only for one TC (TC7) as the first LSP TC takes its bandwidth and there are no guarantees to the lower priority LSPs.
2. GSP can be set to more than one TC in a BWG, always from the highest priority TC within that BWG downward. For the LAN scenario, all TCs could be configured to be GSP as their bandwidth needs are not known.
3. To a low latency TC for which the *GSP* bit is set, non-null refill credits must be set for at least one maximum sized frame. It ensures that even after having been quiet for a while, some BWG credits are left available to the GSP TC, for serving it with minimum latency (without waiting for replenishing). Bigger refill credits values ensure longer burst of GSP traffic served with minimum latency.



4.6.11.5.2 VM Parameters

Refill Credits

Refill credits regulate the fraction of the TC's bandwidth that is allocated to a VM. The ratio between the credits of the VMs represents the relative TC bandwidth percentage allocated to each VM.

Credits are configured and calculated using 64 bytes granularity.

1. The number of refill credits assigned per VM should be as small as possible but still larger than the maximum frame size used and larger than 1.5 KB in any case. Using a lower refill value causes more refill cycles before a packet can be sent. These extra cycles increase the latency unnecessarily.
2. Refill credits ratio between VMs should be equal to the desired ratio of bandwidth allocation between the different TCs. Applying rule #1, means bandwidth shares are sorted from the smaller to the bigger, and just one maximum sized frame is allocated to the smallest.
3. The ratio between the refill credits of any two VMs within the TC should not be greater than 10.

VMs that are sending/receiving packets of 127 bytes eventually gets double the bandwidth it was configured to as we do all the credit calculation by rounding the values down to the next 64 byte aligned value.

4. In a low latency TC, non-null refill credits must be set to a VSP VM, for at least one maximum sized frame. It ensures that even after having been quiet for a while, some TC credits are left available to the VSP VM, for serving it with minimum latency (without waiting for TC to replenish). Bigger refill credits values ensure longer burst of VSP traffic served with minimum latency.

Example 4-1 Refill and MaxCredits Setting Example

This example assumes a system with only four TCs and three VMs present, and with the following bandwidth allocation scheme. Also, full PCIe bandwidth is evaluated to 15 G.

Table 4-9 Bandwidth Share Example

TCs and VMs		Bandwidth Share%	Notes
TC0	Total	40	9.5 KB (9728-byte) jumbo allowed.
	VM0	60	
	VM1	30	
	VM2	10	
TC1	Total	20	No jumbo.
	VM0	34	
	VM1	33	
	VM2	33	

Table 4-9 Bandwidth Share Example (Continued)

TCs and VMs		Bandwidth Share%	Notes
TC2	Total	30	Low latency TC. No jumbo. Bandwidth share already increased. MaxBurstTC2=120 KB
	VM0	80	
	VM1	10	
	VM2	10	
TC3	Total	10	Low latency LSP TC. No jumbo. MaxBurstTC3=36 KB
	VM0	20	
	VM1	60	
	VM2	20	

The ratios between TC refills were driven by TC0, which was set as 152 for supporting 9.5 KB jumbos.

The ratio between MaxCredits and Refill were taken as 17 for all the TCs, as driven by TC2 relation between MaxCredits and MaxBurstTC2.

Table 4-10 Refill and MaxCredits Setting

TCs and VMs		Refill (64-Byte Units)	MaxCredits (64-Byte Units)
TC0	Total	152	2584
	VM0	912	
	VM1	456	
	VM2	152	
TC1	Total	76	1292
	VM0	25	
	VM1	24	
	VM2	24	

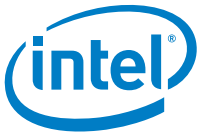


Table 4-10 Refill and MaxCredits Setting (Continued)

TCs and VMs		Refill (64-Byte Units)	MaxCredits (64-Byte Units)
TC2	Total	114	1938
	VM0	192	
	VM1	24	
	VM2	24	
TC3	Total	38	646
	VM0	24	
	VM1	72	
	VM2	24	

4.6.12 Security Initialization

After power up or device reset, security offload is disabled by default (both LinkSec and IPsec), and the content of SA tables must be cleared by software.

Security offload cannot be enabled if internal security fuses are not enabled or the SDP0_4 pin is set to 0b. In this case, both IPsec and LinkSec are disabled and the following security related fields are not writable:

- SECTXCTRL.SECTX_DIS is set to 1b and read as 1b.
- SECRXCTRL.SECRX_DIS is set to 1b and read as 1b.
- IPSTXIDX.IPS_TX_EN is cleared to 0b and read as 0b.
- IPSRXIDX.IPS_RX_EN is cleared to 0b and read as 0b.
- LSECTXCTRL bits 1:0 are cleared to 00b and read as 00b.
- LSECRXCTRL bits 3:2 are cleared to 00b and read as 00b.

Security offload can be used when enabled by internal security fuses and when the SDP0_4 pin is set to 1b. In this case, the security offload can be enabled/disabled via the flows described as follows.

4.6.12.1 Security Enablement Flow

To enable one of the security modes perform the following steps:

1. Stop the data paths by setting the SECTXCTRL.TX_DIS and SECRXCTRL.RX_DIS bits.
2. Wait for hardware to empty the data paths. Poll the SECTXSTAT.SECTX_RDY and SECRXSTAT.SECRX_RDY bits until they are both asserted by hardware.
3. Clear the SECTXCTRL.SECTX_DIS and SECRXCTRL.SECRX_DIS bits to enable the Tx and Rx crypto engines.

When enabling IPsec or LinkSec offload, set SECTXMINIFG.MINSECIFG to 0x3 extending back-to-back gap to the security block required for its functionality.

When enabling IPsec, set the SECTXCTRL.STORE_FORWARD bit, since a store and forward IPsec buffer is required for the processing of AH packets (*ICV* field insertion is done at the beginning of the frame). Otherwise, clear this bit.

When enabling IPsec, write the SEC buffer almost full threshold, register SECTXBUFFAF.buff_af_thresh, with the value of 0x15.

4. Enable SA lookup:

   For IPsec, set the IPSTXIDX.IPS_TX_EN and the IPSRXIDX.IPS_RX_EN bits.

   For LinkSec, set the enable bits in the LSECTXCTRL and LSECRXCTRL registers.

5. Restart the data paths by clearing the SECTXCTRL.TX_DIS and SECRXCTRL.RX_DIS bits.

4.6.12.2 Security Disable Flow

To disable one of the security modes perform the following steps:

1. Stop the data paths by setting the SECTXCTRL.TX_DIS and SECRXCTRL.RX_DIS bits.
2. Wait for hardware to empty the data paths. Poll the SECTXSTAT.SECTX_RDY and SECRXSTAT.SECRX_RDY bits until they are both asserted by hardware.
3. Disable SA lookup:

   For IPsec, clear the IPSTXIDX.IPS_TX_EN and the IPSRXIDX.IPS_RX_EN bits.

   For LinkSec, clear the enable bits in the LSECTXCTRL and LSECRXCTRL registers.

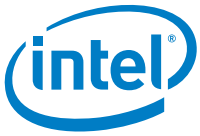
4. Set the SECTXCTRL.SECTX_DIS and SECRXCTRL.SECRX_DIS bits to disable the Tx and Rx crypto engines.

   When disabling IPsec, clear the SECTXCTRL.STORE_FORWARD bit, to avoid using the IPsec buffer and thus reducing Tx internal latency.

   When disabling IPsec, write the SEC buffer almost full threshold, register SECTXBUFFAF.buff_af_thresh, with the value of 0x250.

5. Restart the data paths by clearing the SECTXCTRL.TX_DIS and SECRXCTRL.RX_DIS bits.

Note: Disabling the crypto engine reduces the 82599's power consumption.



4.6.13 Alternate MAC Address Support

In some systems, the MAC address used by a port needs to be replaced with a temporary MAC address in a way that is transparent to the software layer. One possible usage is in blade systems, to allow a standby blade to use the MAC address of another blade that failed, so that the network image of the entire blade system does not change.

In order to allow this mode, a management console might change the MAC address in the NVM image. It is important in this case to be able to keep the original MAC address of the device as programmed at the factory.

In order to support this mode, the 82599 provides the Alternate Ethernet MAC Address structure in the NVM to store the original MAC addresses. This structure is described in Section 6.2.7. In some systems, it might be advantageous to restore the original MAC address at power on reset, to avoid conflicts where two network controllers would have the same MAC address.

NOTE: ***This page intentionally left blank.***

5.0 Power Management and Delivery

This section defines how power management is implemented in the 82599.

5.1 Power Targets and Power Delivery

See Section 11.2.1 for the current consumption and see Section 11.4.1 for the power supply specification.

5.2 Power Management

5.2.1 Introduction to the 82599 Power States

The 82599 supports the D0 and D3 power states defined in the PCI Power Management and PCIe specifications. D0 is divided into two sub-states: D0u (D0 un-initialized), and D0a (D0 active). In addition, the 82599 supports a Dr state that is entered when PE_RST_N is asserted (including the D3cold state).

Figure 5-1 shows the power states and transitions between them.

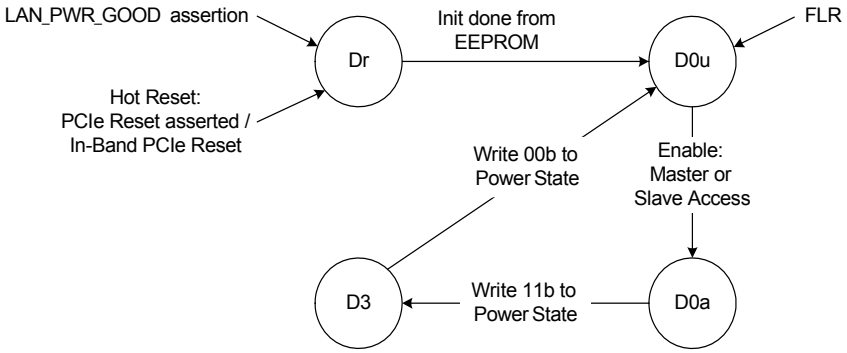


Figure 5-1 Power Management State Diagram

5.2.2 Auxiliary Power Usage

The 82599 uses the AUX_PWR indication that auxiliary power is available to it, and therefore advertises D3cold wake up support. The amount of power required for the function, which includes the entire Network Interface Card (NIC) is advertised in the Power Management Data register, which is loaded from the EEPROM.

If D3cold is supported, the *PME_En* and *PME_Status* bits of the Power Management Control/Status Register (PMCSR), as well as their shadow bits in the Wake Up Control (WUC) register are reset only by the power up reset (detection of power rising).

The only effect of setting AUX_PWR to 1b is advertising D3cold wake up support and changing the reset function of PME_En and PME_Status. AUX_PWR is a strapping option in the 82599.

The 82599 tracks the PME_En bit of the PMCSR and the *Auxiliary (AUX) Power PM Enable* bit of the PCIe Device Control register to determine the power it might consume (and therefore its power state) in the D3cold state (internal Dr state). Note that the actual amount of power differs between form factors.

5.2.3 Power Limits by Certain Form Factors

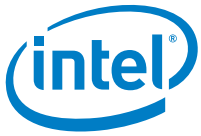
Table 5-1 lists the power limitations introduced by different form factors.

Table 5-1 Power Limits by Form-Factor

	Form Factor	
	LOM	**PCIe Card**
Main	N/A	25 W
Auxiliary (aux enabled)	375 mA @ 3.3V	375 mA @ 3.3V
Auxiliary (aux disabled)	20 mA @ 3.3V	20 mA @ 3.3V

Note: Auxiliary current limit only applies when the primary 3.3V voltage source is not available (the card is in a low power D3 state).

The 82599 exceeds the allocated auxiliary power in some configuration (such as both ports running at GbE or 10 GbE speed). The 82599 must be configured such that it meets the previously described requirements. To do so, link speed can be restricted to GbE and one of the LAN ports can be disabled when operating on auxiliary power. See Section 5.2.5.4.



5.2.4 Interconnects Power Management

This section describes the power reduction techniques used by the 82599's main interconnects.

5.2.4.1 PCIe Link Power Management

The PCIe link state follows the power management state of the device. Since the 82599 incorporates multiple PCI functions, the device power management state is defined as the power management state of the most awake function:

- If any function is in D0 state (either D0a or D0u), the PCIe link assumes the device is in D0 state.

  Else,

- If the functions are in D3 state, the PCIe link assumes the device is in D3 state.

  Else,

- The device is in Dr state (PE_RST_N is asserted to all functions).

The 82599 supports all PCIe power management link states:

- L0 state is used in D0u and D0a states.
- The L0s state is used in D0a and D0u states each time link conditions apply.
- The L1 state is used in D0a and D0u states each time link conditions apply, as well as in the D3 state.
- The L2 state is used in the Dr state following a transition from a D3 state if PCI-PM PME is enabled.
- The L3 state is used in the Dr state following power up, on transition from D0a and also if PME is not enabled in other Dr transitions.

The 82599 support for Active State Link Power Management (ASLPM) is reported via the PCIe Active State Link PM Support register loaded from EEPROM.

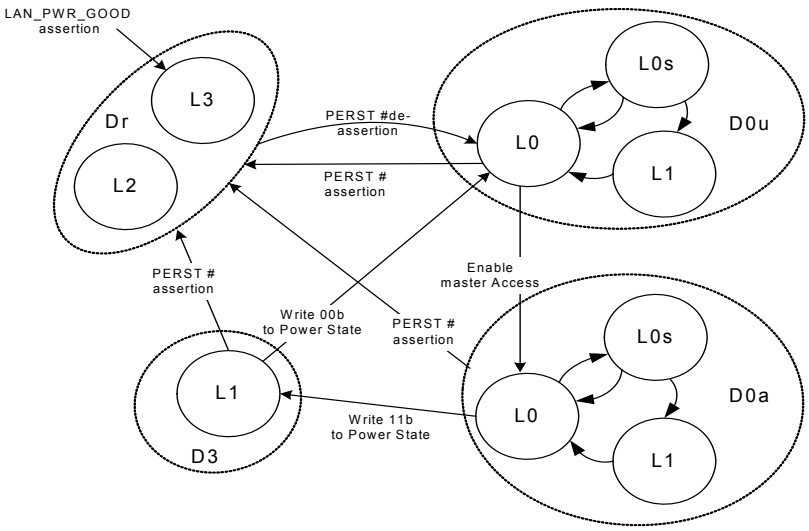


Figure 5-2 Link Power Management State Diagram

While in L0 state, the 82599 transitions the transmit lane(s) into L0s state once the idle conditions are met for a period of time defined as follows.

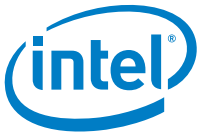
L0s configuration fields are:

- L0s enable — The default value of the *Active State Link PM Control* field in the PCIe Link Control register is set to 00b (both L0s and L1 disabled). System software can later write a different value into the Link Control register. The default value is loaded on any reset of the PCI configuration registers.
- The *L0S_ENTRY_LAT* bit in the PCIe Control Register (GCR), determines L0s entry latency. When set to 0b, L0s entry latency is the same as L0s exit latency of the device at the other end of the link. When set to 1b, L0s entry latency is (L0s exit Latency of the device at the other end of the link /4). Default value is 0b (entry latency is the same as L0s exit latency of the device at the other end of the link).
- L0s exit latency (as published in the *L0s Exit Latency* field of the Link Capabilities register) is loaded from the EEPROM. Separate values are loaded when the 82599 shares the same reference PCIe clock with its partner across the link, and when the 82599 uses a different reference clock than its partner across the link. The 82599 reports whether it uses the slot clock configuration through the *PCIe Slot Clock Configuration* bit loaded from the *Slot_Clock_Cfg* EEPROM bit.
- L0s acceptable latency (as published in the *Endpoint L0s Acceptable Latency* field of the Device Capabilities register) is loaded from the EEPROM.

While in L0s state, the 82599 transitions the link into L1 state once the transmit lanes or both directions of the link have been in L0s state for a period of time defined in PCI configuration space loaded from the PCIe Init Configuration 1 word in the EEPROM.

The following EEPROM fields control L1 behavior:

- Act_Stat_PM_Sup — Indicates support for ASPM L1 in the PCIe configuration space (loaded into the *Active State Link PM Support* field)
- PCIe PLL Gate Disable — Controls PCIe PLL gating while in L1 or L2 states



- L1_Act_Ext_Latency — Defines L1 active exit latency
- L1_Act_Acc_Latency — Defines L1 active acceptable exit latency
- Latency_To_Enter_L1 — Defines the period (in the L0s state) before transitioning into an L1 state

5.2.4.2 Network Interfaces Power Management

The 82599 transitions any of the XAUI interfaces into a low-power state in the following cases:

- The respective LAN function is in LAN disable mode using LANx_DIS_N pin.
- The 82599 is in Dr State, APM WoL is disabled for the port, ACPI wake is disabled for the port and pass-through manageability is disabled for the port.

Use of the LAN ports for pass-through manageability follows this behavior:

- If manageability is disabled (*MNG Enable* bit in the EEPROM is cleared), then LAN ports are not allocated for manageability.
- If manageability is enabled:
  - Power up — Following EEPROM read, a single port is enabled for manageability, running at the lowest speed supported by the interface. If APM WoL is enabled on a single port, the same port is used for manageability. Otherwise, manageability protocols (like teaming) determine which port is used.
  - D0 state — Both LAN ports are enabled for manageability.
  - D3 and Dr states — A single port is enabled for manageability, running at the lowest speed supported by the interface. If WoL is enabled on a single port, the same port is used for manageability. Otherwise, manageability protocols (like teaming) determine which port is used.

Enabling a port as a result of the previous causes an internal reset of the port.

When a XAUI interface is in low-power state, the 82599 asserts the respective SDP pin to enable an external PHY device to power down as well.

5.2.5 Power States

5.2.5.1 D0uninitialized State

The D0u state is a low-power state used after PE_RST_N is de-asserted following power up (cold or warm), on hot reset (in-band reset through PCIe physical layer message) or on D3 exit.

When entering D0u, the 82599 disables wake ups. If the *APM Mode* bit in the EEPROM's Control Word 3 is set, then APM wake up is enabled.

5.2.5.1.1 Entry to a D0u State

D0u is reached from either the Dr state (on de-assertion of internal PE_RST_N) or the D3hot state (by configuration software writing a value of 00b to the *Power State* field of the PCI PM registers).

De-assertion of internal PE_RST_N means that the entire state of the device is cleared, other than sticky bits. State is loaded from the EEPROM, followed by establishment of the PCIe link. Once this is done, configuration software can access the device.

On a transition from D3 to D0u state, the 82599 requires that software perform a full re-initialization of the function including its PCI configuration space.

5.2.5.2 D0active State

Once memory space is enabled, the 82599 enters an active state. It can transmit and receive packets if properly configured by the driver. Any APM wake up previously active remains active. The driver can deactivate APM wake up by writing to the Wake Up Control (WUC) register, or activate other wake up filters by writing to the Wake Up Filter Control (WUFC) register.

5.2.5.2.1 Entry to D0a State

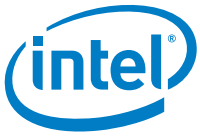
D0a is entered from the D0u state by writing a 1b to the *Memory Access Enable* or the *I/O Access Enable* bit of the PCI Command register. The DMA, MAC, and PHY of the appropriate LAN function are enabled.

5.2.5.3 D3 State (PCI-PM D3hot)

The 82599 transitions to D3 when the system writes a 11b to the *Power State* field of the PMCSR. Any wake-up filter settings that were enabled before entering this reset state are maintained. Upon transitioning to D3 state, the 82599 clears the *Memory Access Enable* and *I/O Access Enable* bits of the PCI Command register, which disables memory access decode. In D3, the 82599 only responds to PCI configuration accesses and does not generate master cycles.

Configuration and message requests are the only PCIe TLPs accepted by a function in the D3hot state. All other received requests must be handled as unsupported requests, and all received completions can optionally be handled as unexpected completions. If an error caused by a received TLP (such as an unsupported request) is detected while in D3hot, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. See section 5.3.1.4.1 in the PCIe Base Specification.

A D3 state is followed by either a D0u state (in preparation for a D0a state) or by a transition to Dr state (PCI-PM D3cold state). To transition back to D0u, the system writes a 00b to the *Power State* field of the PMCSR. Transition to Dr state is through PE_RST_N assertion.



5.2.5.3.1 Entry to D3 State

Transition to D3 state is through a configuration write to the *Power State* field of the PCI-PM registers.

Prior to transition from D0 to the D3 state, the device driver disables scheduling of further tasks to the 82599; it masks all interrupts, it does not write to the Transmit Descriptor Tail register or to the Receive Descriptor Tail register and operates the master disable algorithm as defined in Section 5.2.5.3.2. If wake-up capability is needed, the driver should set up the appropriate wake-up registers and the system should write a 1b to the *PME_En* bit of the PMCSR or to the *Auxiliary (AUX) Power PM Enable* bit of the PCIe Device Control register prior to the transition to D3.

If all PCI functions are programmed into D3 state, the 82599 brings its PCIe link into the L1 link state. As part of the transition into L1 state, the 82599 suspends scheduling of new TLPs and waits for the completion of all previous TLPs it has sent. The 82599 clears the *Memory Access Enable* and *I/O Access Enable* bits of the PCI Command register, which disables memory access decode. Any receive packets that have not been transferred into system memory is kept in the device (and discarded later on D3 exit). Any transmit packets that have not be sent can still be transmitted (assuming the Ethernet link is up).

In preparation to a possible transition to D3cold state, the device driver might disable one of the LAN ports (LAN disable) and/or transition the link(s) to GbE speed (if supported by the network interface). See Section 5.2.4.2 for a description of network interface behavior in this case.

5.2.5.3.2 Master Disable

System software can disable master accesses on the PCIe link by either clearing the *PCI Bus Master* bit or by bringing the function into a D3 state. From that time on, the device must not issue master accesses for this function. Due to the full-duplex nature of PCIe, and the pipelined design in the 82599, it might happen that multiple requests from several functions are pending when the master disable request arrives. The protocol described in this section insures that a function does not issue master requests to the PCIe link after its master enable bit is cleared (or after entry to D3 state).

Two configuration bits are provided for the handshake between the device function and its driver:

- *PCIe Master Disable* bit in the Device Control (CTRL) register — When the *PCIe Master Disable* bit is set, the 82599 blocks new master requests by this function. The 82599 then proceeds to issue any pending requests by this function. This bit is cleared on master reset (LAN Power Good all the way to software reset) to enable master accesses.
- *PCIe Master Enable Status* bits in the Device Status register — Cleared by the 82599 when the *PCIe Master Disable* bit is set and no master requests are pending by the relevant function (set otherwise). Indicates that no master requests are issued by this function as long as the *PCIe Master Disable* bit is set. The following activities must end before the 82599 clears the *PCIe Master Enable Status* bit:
  - Master requests by the transmit and receive engines.
  - All pending completions to the 82599 are received.

The device driver disables any reception to the Rx queues as described in Section 4.6.7.1. Then the device driver sets the *PCIe Master Disable* bit when notified of a pending master disable (or D3 entry).

The 82599 then blocks new requests and proceeds to issue any pending requests by this function. The driver then reads the change made to the PCIe Master Disable bit and then polls the PCIe Master Enable Status bit. Once the bit is cleared, it is guaranteed that no requests are pending from this function.

The driver might time out if the *PCIe Master Enable Status* bit is not cleared within a given time. Examples for cases that the device might not clear the *PCIe Master Enable Status* bit for a long time are cases of flow control, link down, or DMA completions not making it back to the DMA block.

In these cases, the driver should check that the *Transaction Pending* bit (bit 5) in the Device Status register in the PCI config space is clear before proceeding. In such cases the driver might need to initiate two consecutive software resets with a larger delay than 1 µs between the two of them.

In the above situation, the data path must be flushed before the software resets the 82599. The recommended method to flush the transmit data path is as follows:

1. Inhibit data transmission by setting the HLREG0.LPBK bit and clearing the RXCTRL.RXEN bit. This configuration avoids transmission even if flow control or link down events are resumed.
2. Set the GCR_EXT.Buffers_Clear_Func bit for 20 microseconds to flush internal buffers.
3. Clear the HLREG0.LPBK bit and the GCR_EXT.Buffers_Clear_Func
4. It is now safe to issue a software reset.

5.2.5.4 Dr State

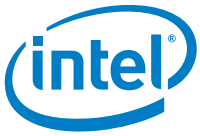
Transition to Dr state is initiated on several occasions:

- On system power up — Dr state begins with the assertion of the internal power detection circuit (LAN_PWR_GOOD) and ends with de-assertion of PE_RST_N.
- On transition from a D0a state — During operation the system can assert PE_RST_N at any time. In an ACPI system, a system transition to the G2/S5 state causes a transition from D0a to Dr state.
- On transition from a D3 state — The system transitions the device into the Dr state by asserting PCIe PE_RST_N.

Any wake-up filter settings that were enabled before entering this reset state are maintained.

The system can maintain PE_RST_N asserted for an arbitrary time. The de-assertion (rising edge) of PE_RST_N causes a transition to D0u state.

While in Dr state, the 82599 can maintain functionality (for WoL or manageability) or can enter a Dr Disable state (if no WoL and no manageability) for minimal device power. The Dr Disable mode is described in the sections that follow.



5.2.5.4.1 Dr Disable Mode

The 82599 enters a Dr disable mode on transition to D3cold state when it does not need to maintain any functionality. The conditions to enter either state are:

- The device (all PCI functions) is in Dr state
- APM WOL is inactive for both LAN functions
- Pass-through manageability is disabled
- ACPI PME is disabled for all PCI functions

Entry into Dr disable is usually done on assertion of PCIe PE_RST_N. It can also be possible to enter Dr disable mode by reading the EEPROM while already in Dr state. The usage model for this later case is on system power up, assuming that manageability and wake up are not required. Once the device enters Dr state on power up, the EEPROM is read. If the EEPROM contents determine that the conditions to enter Dr disable are met, the device then enters this mode (assuming that PCIe PE_RST_N is still asserted).

Exit from Dr disable is through de-assertion of PCIe PE_RST_N.

If Dr disable mode is entered from D3 state, the platform can remove the 82599 power. If the platform removes the 82599 power, it must remove all power rails from the device if it needs to use this capability. Exit from this state is through power-up cycle to the 82599. Note that the state of the SDP pins is undefined once power is removed from the device.

5.2.5.4.2 Entry to Dr State

Dr-entry on platform power up is as follows:

- Assertion of the internal power detection circuit (LAN_PWR_GOOD). Device power is kept to a minimum by keeping the XAUI interfaces in low power.
- The EEPROM is then read and determines device configuration.
- If the *APM Enable* bit in the EEPROM's Control Word 3 is set, then APM wake up is enabled (for each port independently).
- If the *MNG Enable* bit in the EEPROM word is set, pass-through manageability is not enabled.
- Each of the LAN ports can be enabled if required for WoL or manageability. See Section 5.2.4.2 for exact condition to enable a port.
- The PCIe link is not enabled in Dr state following system power up (since PE_RST_N is asserted).

Entry to Dr state from D0a state is through assertion of the PE_RST_N signal. An ACPI transition to the G2/S5 state is reflected in a device transition from D0a to Dr state. The transition can be orderly (such as a user selected a shut down operating system option), in which case the device driver can have a chance to intervene. Or, it can be an emergency transition (like power button override), in which case, the device driver is not notified.

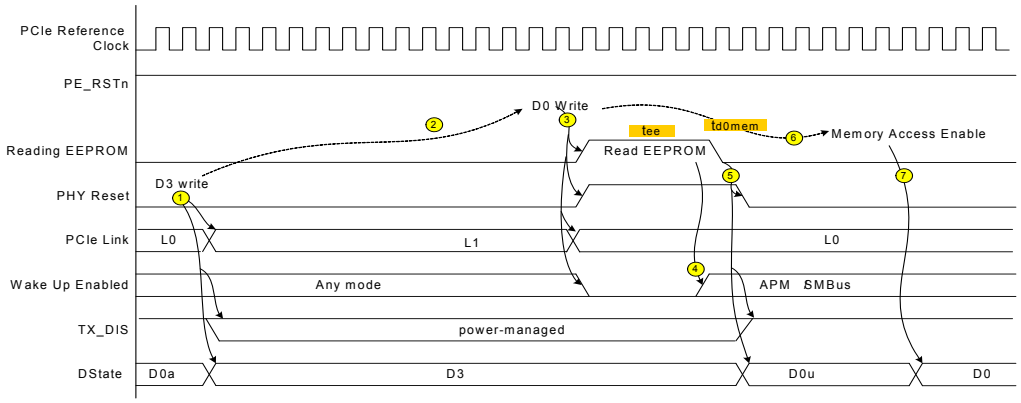
Transition from D3 state to Dr state is done by assertion of PE_RST_N signal. Prior to that, the system initiates a transition of the PCIe link from L1 state to either the L2 or L3 state (assuming all functions were already in D3 state). The link enters L2 state if PCI-PM PME is enabled.

5.2.6 Timing of Power-State Transitions

The following sections give detailed timing for the state transitions. In the diagrams the dotted connecting lines represent the 82599 requirements, while the solid connecting lines represent the 82599 guarantees.

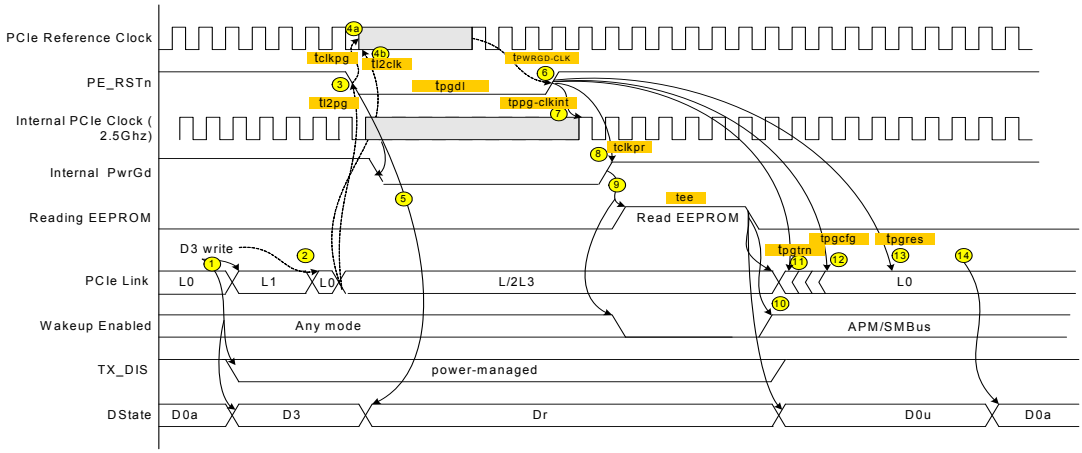
Note: The timing diagrams are not to scale. The clocks edges are shown to indicate running clocks only and are not used to indicate the actual number of cycles for any operation.

5.2.6.1 Transition from D0a to D3 and Back without PE_RST_N



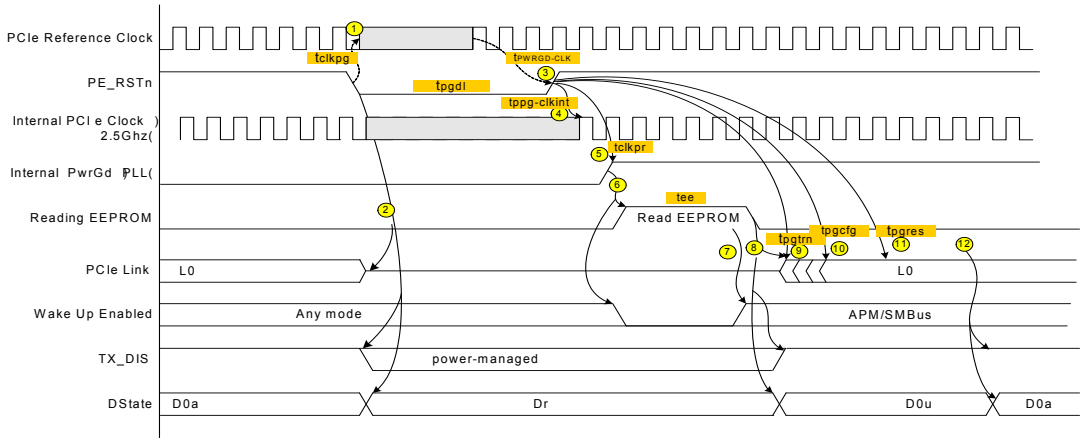
Note	
1	Writing 11b to the *Power State* field of the PMCSR transitions the 82599 to D3.
2	The system can keep the 82599 in D3 state for an arbitrary amount of time.
3	To exit D3 state the system writes 00b to the *Power State* field of the PMCSR.
4	APM wake up or manageability can be enabled based on what is read in the EEPROM.
5	After reading the EEPROM, the LAN ports are enabled and the 82599 transitions to D0u state.
6	The system can delay an arbitrary time before enabling memory access.
7	Writing a 1b to the *Memory Access Enable* bit or to the *I/O Access Enable* bit in the PCI Command register transitions the 82599 from D0u to D0 state.

5.2.6.2 Transition from D0a to D3 and Back with PE_RST_N

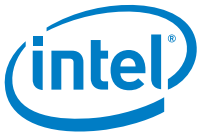


Note	
1	Writing 11b to the *Power State* field of the PMCSR transitions the 82599 to D3. PCIe link transitions to L1 state. Possible indication to external PHYs to enter low-power mode.
2	The system can delay an arbitrary amount of time between setting D3 mode and transitioning the link to an L2 or L3 state.
3	Following link transition, PE_RST_N is asserted.
4	The system must assert PE_RST_N before stopping the PCIe reference clock. It must also wait tl2clk after link transition to L2/L3 before stopping the reference clock.
5	On assertion of PE_RST_N, the 82599 transitions to Dr state.
6	The system starts the PCIe reference clock $t_{PWRGD-CLK}$ before de-assertion PE_RST_N.
7	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
8	The PCIe internal PWRGD signal is asserted tclkpr after the external PE_RST_N signal.
9	Assertion of internal PCIe PWRGD causes the EEPROM to be re-read and disables wake up.
10	APM wake-up mode can be enabled based on what is read from the EEPROM. External PHYs are enabled.
11	Link training starts after tpgtrn from PE_RST_N de-assertion.
12	A first PCIe configuration access can arrive after t_{pgcfg} from PE_RST_N de-assertion.
13	A first PCI configuration response can be sent after tpgres from PE_RST_N de-assertion.
14	Writing a 1b to the *Memory Access Enable* bit in the PCI Command register transitions the device from D0u to D0 state.

5.2.6.3 Transition from D0a to Dr and Back without Transition to D3



Note	
1	The system must assert PE_RST_N before stopping the PCIe reference clock. It must also wait tl2clk after link transition to L2/L3 before stopping the reference clock.
2	On assertion of PE_RST_N, the 82599 transitions to Dr state and the PCIe link transition to electrical idle. Possible indication to external PHYs to enter low-power mode.
3	The system starts the PCIe reference clock $t_{PWRGD-CLK}$ before de-assertion PE_RST_N.
4	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
5	The PCIe internal PWRGD signal is asserted tclkpr after the external PE_RST_N signal.
6	Assertion of internal PCIe PWRGD causes the EEPROM to be re-read and disables wake up.
7	APM wake-up mode can be enabled based on what is read from the EEPROM.
8	After reading the EEPROM, external PHYs are enabled.
9	Link training starts after tpgtrn from PE_RST_N de-assertion.
10	A first PCIe configuration access can arrive after t_{pgcfg} from PE_RST_N de-assertion.
11	A first PCI configuration response can be sent after tpgres from PE_RST_N de-assertion
12	Writing a 1b to the *Memory Access Enable* bit in the PCI Command register transitions the device from D0u to D0 state.



5.2.6.4 Timing Requirements

The 82599 requires the following start up and power state transitions.

Table 5-2 Start Up and Power State Transitions

Parameter	Description	Min	Max.	Notes
t_{xog}	Xosc stable from power stable		10 ms	
$t_{PWRGD-CLK}$	PCIe clock valid to PCIe power good	100 µs	-	According to PCIe specification.
t_{PVPGL}	Power rails stable to PCIe PE_RST_N inactive	100 ms	-	According to PCIe specification.
t_{pgcfg}	External PE_RST_N signal to first configuration cycle	100 ms		According to PCIe specification.
td0mem	Device programmed from D3h to D0 state to next device access	10 ms		According to PCI power management specification.
tl2pg	L2 link transition to PE_RST_N assertion	0 ns		According to PCIe specification.
tl2clk	L2 link transition to removal of PCIe reference clock	100 ns		According to PCIe specification.
tclkpg	PE_RST_N assertion to removal of PCIe reference clock	0 ns		According to PCIe specification.
tpgdl	PE_RST_N assertion time	100 µs		According to PCIe specification.

5.2.6.5 Timing Guarantees

The 82599 guarantees the following start up and power state transition related timing parameters.

Table 5-3 Start-up and Power State Transition Timing Parameters[1]

Parameter	Description	Min	Max.	Notes
t_{ee}	EEPROM read duration		20 ms	
$t_{ppg-clkint}$	PCIe PE_RST_N to internal PLL lock	-	50 µs	
t_{clkpr}	Internal PCIe PWGD from external PCIe PE_RST_N		50 µs	
t_{pgtrn}	PCIe PE_RST_N to start of link training		20 ms	According to PCIe specification.
t_{pgres}	External PE_RST_N to response to first configuration cycle	100 ms	1 sec	According to PCIe specification.

1. See also: Table 4-3, Power-Up Timing Guarantees.

5.3 Wake Up

5.3.1 Advanced Power Management Wake Up

Advanced Power Management Wake Up, or APM Wake Up, was previously known as Wake on LAN (WoL). It is a feature that has existed in the 10/100 Mb/s NICs for several generations. The basic premise is to receive a broadcast or unicast packet with an explicit data pattern, and then to assert a signal to wake up the system. In the earlier generations, this was accomplished by using a special signal that ran across a cable to a defined connector on the motherboard. The NIC would assert the signal for approximately 50 ms to signal a wake up. The 82599 uses (if configured to) an in-band PM_PME message for this.

At power up, the 82599 reads the *APM Enable* bit from the EEPROM into the *APM Enable (APME)* bits of the GRC register. This bit control the enabling of APM wake up.

When APM wake up is enabled, the 82599 checks all incoming packets for Magic Packets.

Once the 82599 receives a matching Magic Packet, it:

- Sets the *PME_Status* bit in the PMCSR.
- Asserts PE_WAKE_N.
- Issues a PM_PME message.

APM wake up is supported in all power states and only disabled if a subsequent EEPROM read results in the *APM Wake Up* bit being cleared.

5.3.2 ACPI Power Management Wake Up

The 82599 supports ACPI power management-based wake up. It can generate system wake-up events from three sources:

- Reception of a Magic Packet.
- Reception of a network wake-up packet.
- Detection of a link change of state.
- Activating ACPI power management wake up requires the following steps:
  - The operating system (at configuration time) writes a 1b to the *PME_En* bit of the PMCSR (bit 8).
  - The driver programs the Wake Up Filter Control (WUFC) register to indicate the packets it needs to wake up and supplies the necessary data to the IPv4/v6 Address Table (IP4AT, IP6AT), Flexible Host Filter Table (FHFT) registers. It can also set the *Link Status Change Wake Up Enable (LNKC)* bit in the WUFC register to cause wake up when the link changes state. If the SW driver enables any of the wakeup options above it should also set the WUC.PME_En bit as well.
  - Once the 82599 wakes the system, the driver needs to clear WUFC until the next time the system goes to a low power state with wake up.