Intel® Platform Controller Hub EG20T

Specification Update

July 2012

Revision 011US

Notice: The Intel® Platform Controller Hub EG20T may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Contents

Revision History

Date	Revision	Description
July 2012	011	Added Specification Clarifications 5 and 6. Added Documentation Only Change 8.
April 2012	010	Added Specification Clarification 4.
March 2012	009	Removed Specification Clarifications (update made in parent document). Removed Document Corrections (update made in parent document). Added Erratum 12. Added Specification Clarifications 1 through 3. Added Documentation Only Changes 1 through 7.
January 2012	008	Added Specification Clarifications 4 through 8. Added Documentation Changes 5 through 26.
November 2011	007	Removed Specification Clarifications 1 and 2 (update made in parent document). Removed Document Corrections 1 through 5 (update made in parent document). Updated Table 8, “Identification Information” Added Specification Clarifications 1, 2, 3 Added Document Corrections 1, 2, 3, 4
August 20011	006	Removed old Specification Clarifications 1 and 2 (update made in parent document). Removed old Document Corrections 1 through 18 (update made in parent document). Added errata 8 through 11. Added Document Corrections 1 through 3
June 2011	005	Removed old Specification Clarification 1 (update made in parent document). Removed old Document Corrections 1 through 15 (update made in parent document). Added Specification Clarifications 1, 2. Added Document Corrections 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18.
May 2011	004	Added Erratum 7. Added Specification Clarification 1. Added Document Corrections 10, 11, 12, 13, 14, 1516
February 2011	003	Added Erratum 6. Added Document Corrections 4, 5, 6, 7, 8, 9.
November 2010	002	Added Erratum 5. Added Document Corrections 1, 2, 3.
October 2010	001	Initial release

Introduction

Purpose/Scope/Audience

This document is an update to the specifications listed in the Parent Documents/Related Documents table that follows. This document is a compilation of device and documentation Errata, Specification Changes, Specification Clarifications, Document-Only Changes. It is intended for hardware and software system designers and manufacturers as well as developers of applications, operating systems, or tools.

Information types defined in Conventions and Terminology are consolidated into the Specification Update and are no longer published in other documents.

This document may also contain information that was not previously published.

Table 1. Parent Documents/Related Documents

Title	Number
Intel® Platform Controller Hub EG20T Datasheet	http://download.intel.com/embedded/processor/chipsets/324211.pdf

Notes:
1. Contact your Intel sales representative. Some documents may not be available at this time.

Conventions and Terminology

Note: Errata remain in the Specification Update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the Specification Update are archived and available upon request. Specification Changes, Specification Clarifications and Document-Only Changes are removed from the Specification Update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

Table 2. Conventions and Terminology (Sheet 1 of 2)

Term	Definition
Document-Only Changes	Document-Only Changes are changes to an Intel Parent Specification that result in changes only to an Intel customer document but no changes to a specification or to a parameter for an Intel product. An example of a document-only change is the correction of a typographical error.
Errata (plural) Erratum (singular)	Errata are design defects or errors. These may cause the Intel® Platform Controller Hub EG20T's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Table 2. Conventions and Terminology (Sheet 2 of 2)

Term	Definition
Parent Specification	A parent specification is a top-level specification from which other documents can be derived, depending on the product or platform. Typically, a parent specification includes a product's pinout, architectural overview, device operation, hardware interface, or electrical specifications. Examples of parent specifications include the following: Datasheet, External Design Specification (EDS), Developer's Manual, Technical Product Specification. The derived documents may be used for purposes other than that for which the parent specification is used.
Specification Changes	Specification Changes are the result of adding, removing, or changing a feature, after which an Intel product subsequently operates differently than specified in an Intel Parent Specification, but typically the customer does not have to do anything to achieve proper device functionality as a result of Intel adding, removing, or changing a feature.
Specification Clarifications	Specification Clarifications are changes to a document that arise when an Intel Parent Specification must be reworded so that the specification is either more clear or not in conflict with another specification.

Summary Tables of Current Product Issue Activity

Table 4 through Table 7 indicate the Errata, Specification Changes, Specification Clarifications, and Document-Only Changes that apply to the Intel® Platform Controller Hub EG20T product. Intel may fix some of the Errata in a future stepping of the component as noted in Table 3 or account for the other outstanding issues through Specification Changes, Specification Clarifications, or Document-Only Changes. Table 4 through Table 7 use the codes listed in Table 3.

Table 3. Codes Used in Summary Tables

Code	Column	Definition
X	Stepping	Indicates either that, for the stepping/revision listed, • an erratum exists and is not yet fixed • a specification change or specification clarification applies
No mark or blank	Stepping	Indicates either that, for the stepping/revision listed, • an erratum is fixed • a specification change or specification clarification does not apply
Plan Fix	Status	This erratum may be fixed in a future stepping/revision.
Fixed	Status	This erratum has been previously fixed.
No Fix	Status	There are no plans to fix this erratum.
A change bar to the left of a table row indicates an item that is either new or modified from the previous version of the Specification Update document.		

Table 4. Errata

No.	Stepping		Status	Errata Title
	A2	A3		
1	X	X	No Fix	PCIe* Port Does Not Respond with "Unsupported Request"
2	X	X	No Fix	PCIe* Port Does Not Support Memory Read Locked Access
3	X		Fixed	Gigabit Ethernet MAC, GPIO and UART0 Do Not Support Wake-up Events at Device State D3hot
4	X		Fixed	SATA PHY Not Compliant to SATA Specification on COMINIT/COMRESET Gap Detection Window
5	X		Fixed	Bit 8 of HcRevision Registers in OHCI Host Controller Incorrectly Indicates the Presence of Legacy Support Registers
6	X		Fixed	USB Host: EHCI Host Controller (D2:F3,D8:F3) PM_PTR is Incorrectly Hardwired with EHCI Next Capabilities Register
7	X		Fixed	USB Host Does Not Work Well at the Resume End Stage When HS Device is Attached
8	X	X	No Fix	PCIe* Bridge: Does Not Send Non-fatal Error Packet When Intel® PCH EG20T Receives Write Access Request to Undefined Address in IO or Memory Mapped IO Space
9	X	X	No Fix	PCIe* Bridge: Upstream Port of PCIe* Bridge Allows Forwarding Regardless of the PCI Command Register's Bus Master Enable Bit (PCICMD.BME) Value
10	X	X	No Fix	PCIe* Bridge: Reports Master Aborts Even Though Master-Abort Mode Bit of Bridge Control Register Is Hard Wired to 0
11	X	X	No Fix	PCI Configuration Register: PWR_CNTL_STS.POWERSTATE[1:0] Bits Allow Software to Write 10b or 01b, Despite Not Supporting D1 and D2 States
12	X	X	No Fix	USB Host: Memory Read Request Address Across DWord Boundary May Cause Incomplete Transfer

Table 5. Specification Changes

No.	Stepping		Specification Changes
	A2	A3	
			None for this revision of the Specification Update.

Table 6. Specification Clarifications

No.	Document Title	Rev.	Specification Clarifications
1	*Intel® Platform Controller Hub EG20T Datasheet*	008	Clarify the Method of USB Device Controller Hot Unplug/Plug
2	*Intel® Platform Controller Hub EG20T Datasheet*	008	Clarify the Endpoint Interrupt Register and Mask Register in USB Device Controller
3	*Intel® Platform Controller Hub EG20T Datasheet*	008	Updated IDCODE for A3 stepping
4	*Intel® Platform Controller Hub EG20T Datasheet*	008	Add in Section 9.5.4 USB Device DMA Operation
5	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Notes for Table 226 "54h: PORTSC - Port Status and Control Register"
6	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Error in Table 863 "Power and Ground Signals"

Table 7. Document-Only Changes

No.	Document Title	Rev.	Document-Only Changes
1	*Intel® Platform Controller Hub EG20T Datasheet*	008	Correct Address in Chapter 13 CAN Controller: "IF1 and IF2 Message Interface Register Sets" Table
2	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Title in "MSGLST" Table
3	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Typo in Figure 42 "Correlation between the IFm Register and the Message RAM"
4	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Typo in Figure 44 "Concept Flowchart of Transmit Message Handling"
5	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Typo in Figure 45 "Concept Flowchart of Receive Message Handling"
6	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Typo in Chapter 13 CAN Controller Table 525 "Structure of a Message Object in the Message RAM"
7	*Intel® Platform Controller Hub EG20T Datasheet*	008	Correct Typo in Chapter 13 CAN Controller Table 517 "24h: IFmCMASK: m = 1, 2 - IFm Command Mask Register"
8	*Intel® Platform Controller Hub EG20T Datasheet*	008	Corrected Error in Table 230 under Chapter 8.3.3.1 HcRevision Register

General Product Information

The Intel® Platform Controller Hub EG20T can be identified by the following register contents:

Table 8. Identification Information

Part Number	Stepping	Vendor ID[1]	Device ID[2]	Revision Number[3]
Intel® Platform Controller Hub EG20T	A2	8086	8800	01
Intel® Platform Controller Hub EG20T	A3	8086	8800	02

Notes:

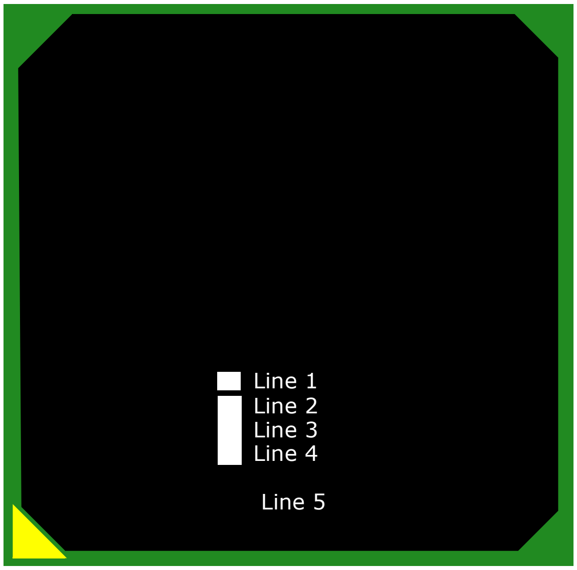
1. The Vendor ID corresponds to bits [15:0] of the VID - Vendor Identification Register located at Offset 00–01h in the PCI Bus 1 Device 0 Function 0 configuration space.
2. The Device ID corresponds to bits [15:0] of the DID - Device Identification Register located at Offset 02–03h in the PCI Bus 1 Device 0 Function 0 configuration space.
3. The Revision Number corresponds to bits [7:0] of the RID - Revision Identification Register located at Offset 08h in the PCI Bus 2 Device 0 Function 1 configuration space.

The Intel® Platform Controller Hub EG20T stepping can be identified by the following component markings:

Stepping	MM#	Identifier	SPEC Code	Notes
A2	908138	82TPCF	SLH7G	Intel® Platform Controller Hub EG20T
A3	914208	82TPCF	SLJ42	Intel® Platform Controller Hub EG20T

Legend for Figure 1	Mark Text
Line 1	Intel identifier
Line 2	Lot number
Line 3	Assembly date code; QDF number (ES=Engineering Sample)
Line 4	Copyright date; e1=ROHS marking
Line 5	Country of Origin

Figure 1. Top-Side Marking Example



Errata

1. PCIe* Port Does Not Respond with “Unsupported Request”

Problem: The PCIe* port of the Intel® PCH EG20T does not respond with “Unsupported Request” (UR) when the processor accesses the particular part of the PCI space secured by the configuration through PCIe*. In the case where a UR of the following scenarios should be responded, the Intel® PCH EG20T does not return any completion transaction.

- The request does not reference address space mapped within the device.
- A Poisoned Write Request addresses an I/O or Memory mapped control space in the Completer. Such transactions must be discarded by the completer and reported as a UR.

Implication: If the PCIe* root port of a system which does not support “Completion Timeout” accesses the Intel® Platform Controller Hub EG20T, the system may hang.

Workaround: Only use the Intel® Platform Controller Hub EG20T with a PCIe* root port that supports Completion Timeout such as the Intel® Atom™ Processor E6xx Series’s root ports.

Status: No Fix

2. PCIe* Port Does Not Support Memory Read Locked Access

Problem: The PCIe* port does not support “Memory Read Locked Access” (MRdLk) and does not respond “Locked Normal Completion with Data” (CplDLk) or “Locked Error Completion without Data” (CplLk) to the system when it receives an MRdLk from the system. It recognizes MRdLk as “Memory Read” (MRd).

Implication: None

Workaround: Not available

Status: No Fix

3. Gigabit Ethernet MAC, GPIO and UART0 Do Not Support Wake-up Events at Device State D3hot

Problem: Internal clocks to Gigabit Ethernet MAC, GPIO and UART0 functions are gated in the device state D3hot, and these functions cannot assert wake_out_n signal in D3hot.

Implication: Gigabit Ethernet MAC, GPIO and UART0 functions cannot wake up the system from system sleep states.

Workaround: The BIOS is required to place these functions in the D0 device state and restore their context prior to entering a system sleep state. For details, refer to the Firmware Writer’s Guide (FWG), reference number 28833.

Status: Fixed

4. SATA PHY Not Compliant to SATA Specification on COMINIT/ COMRESET Gap Detection Window

Problem: The SATA PHY in the Intel® Platform Controller Hub EG20T does not pass the SATA interoperability test specification OOB-07 as it does not meet the COMINIT/COMRESET gap value of 306 ns while meeting the gap value of 334 ns. The measured minimal gap value for the SATA PHY is 306.6 ns.

Implication: The SATA PHY is not compliant to the SATA specification on the COMINIT/COMRESET gap detection window. The compliance failure will not affect connectivity with a SATA device that passes the SATA interoperability test specification OOB-04.

Workaround: Not available

Status: Fixed

5. Bit 8 of HcRevision Registers in OHCI Host Controller Incorrectly Indicates the Presence of Legacy Support Registers

Problem: Bit 8 of HcRevision registers (Device 2 Function 0-2 Offset 00h, Device 8 Function 0-2 Offset 00h) is set indicating the presence of legacy support registers in the Intel® PCH EG20T OHCI host controller although the Intel® PCH EG20T does not support legacy support registers as defined in OpenHCI Open Host Controller Interface Specification for USB Release rev1.0a.

Implication: When software accesses these legacy support registers, the Intel® PCH EG20T does not respond with “Unsupported Request” (UR). If the PCIe* root port of a system that does not support “Completion Timeout” accesses these registers, the system may hang due to errata #1.

Workaround: Software should ignore the indication of legacy register support in bit 8 of HcRevision registers and never access the non-existent legacy support registers.

Status: Fixed

6. USB Host: EHCI Host Controller (D2:F3,D8:F3) PM_PTR is Incorrectly Hardwired with EHCI Next Capabilities Register

Problem: Intel® PCH EG20T has the wrong capability pointer 0xA0h in PCI Configuration registers of EHCI Host controller (D2:F3,D8:F3) PM_PTR (offset 51h) and hence the capability list does not comply to PCI specification.

PM_PTR (offset 51h) should have hardwired to 0x00h, which indicates it is the last item in the capabilities list.

Implication: Application software may show the unexpected registers, e.g., EHCI Debug Port registers. When software accesses the unexpected registers, Intel® PCH EG20T does not respond with “Unsupported Request” (UR). If the PCIe* root port of a system which does not support “Completion Timeout” accesses these registers, the system may hang due to errata #1.

Workaround: No workaround is available. Application software would not access illegal register if it uses EHCI class driver and implements registers referring to the *Intel® Platform Controller Hub EG20T Datasheet*.

Status: Fixed

7. USB Host Does Not Work Well at the Resume End Stage When HS Device is Attached

Problem: The Intel® Platform Controller Hub EG20T USB Host has an interface inconsistency between built-in PHY and the controller during HS (High Speed) resume.

Implication: When a HS device is attached to the USB Host port, USB bus does not return to HS-Idle at the end stage of resume. So the USB Host port is not available.

Workaround: Available workarounds are as below, but with limitations.

1. For S0 state: selective suspend mode for USB Host ports should be disabled.

<Windows OS>

Delete the SelectiveSuspendDisable REG-BINARY value from the Windows* registry.

Example:

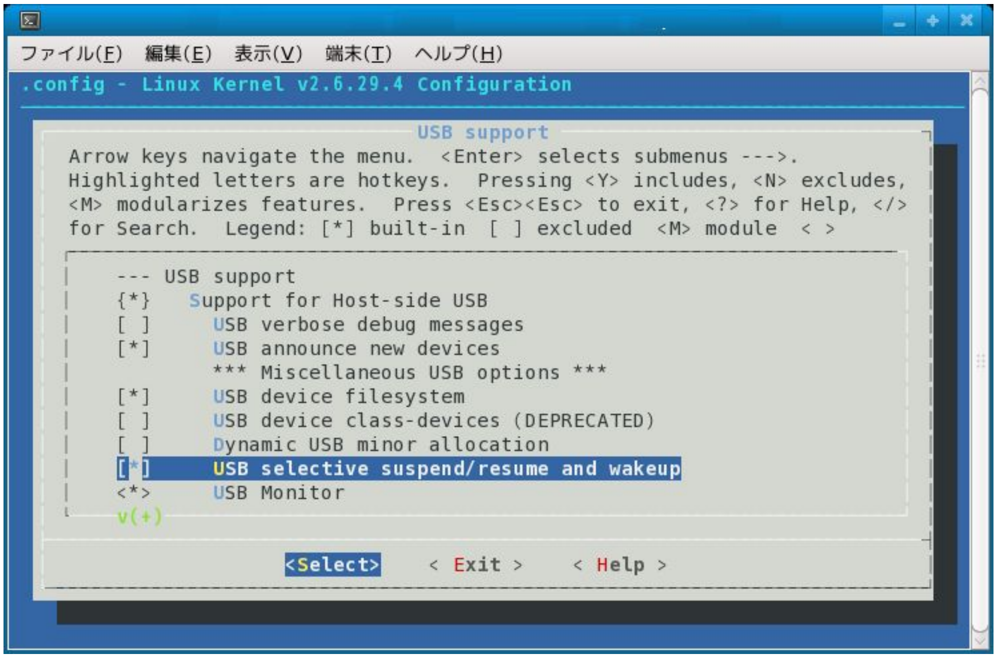
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\usb]

“DisableSelectiveSuspend”=dword:00000001

<Linux OS>

Exclude the feature of USB selective suspend/resume at kernel build.

Limitation of this workaround: battery power is still consumed even when the device becomes idle



```
.config - Linux Kernel v2.6.29.4 Configuration

                          USB support
  Arrow keys navigate the menu.  <Enter> selects submenus --->.
  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
  for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < >

      --- USB support
      {*}   Support for Host-side USB
      [ ]     USB verbose debug messages
      [*]     USB announce new devices
              *** Miscellaneous USB options ***
      [*]     USB device filesystem
      [ ]     USB device class-devices (DEPRECATED)
      [ ]     Dynamic USB minor allocation
      [*]     USB selective suspend/resume and wakeup
      <*>     USB Monitor
      v(+)

               <Select>    < Exit >    < Help >
```

2. For S3 state: BIOS to control USB bus reset sequence while transition to S0 state

Limitations of this workaround: USB device will be reinitialized after return to S0 state.

- the edited content before transition to S3 state will not be preserved
- this BIOS workaround should not be applied if using USB boot device

Status: Fixed

8. PCIe* Bridge: Does Not Send Non-fatal Error Packet When Intel® PCH EG20T Receives Write Access Request to Undefined Address in IO or Memory Mapped IO Space

Problem: The PCIe* bridge does not send a Non-fatal error packet when the Intel® PCH EG20T receives a write access request to an undefined address in IO or memory mapped IO space.

Implication: This issue is seen when the SERR bit in PCICMD-PCI Command Register has been set to 1 and the Intel® PCH EG20T receives a write access request to an undefined address in IO or memory mapped IO space.

Workaround: No workaround is available.

Status: No Fix

9. PCIe* Bridge: Upstream Port of PCIe* Bridge Allows Forwarding Regardless of the PCI Command Register's Bus Master Enable Bit (PCICMD.BME) Value

Problem: The upstream port of the PCIe* bridge allows forwarding regardless of the value of Bus Master Enable bit (PCICMD.BME) of the PCI Command Register.

Implication: When PCICMD.BME of the PCIe* bridge is disabled, operation of the PCI devices connected to the bridge is not guaranteed.

Workaround: No workaround is available.

Status: No Fix

10. PCIe* Bridge: Reports Master Aborts Even Though Master-Abort Mode Bit of Bridge Control Register Is Hard Wired to 0

Problem: When a UR response is received from the Intel® Atom™ Processor E6xx Series for non-posted transactions, the PCIe* bridge reports a Master Abort even though the Master Abort Mode bit of the Bridge Control Register is hard wired to 0.

Implication: The implication for this issue is minor.

Workaround: No workaround is available.

Status: No Fix

11. PCI Configuration Register: PWR_CNTL_STS.POWERSTATE[1:0] Bits Allow Software to Write 10b or 01b, Despite Not Supporting D1 and D2 States

Problem: The PWR_CNTL_STS.POWERSTATE[1:0] bits in the PCI configuration register allow software to write values of 10b and 01b, despite not supporting D1 and D2 states.

Implication: There is no implication for this issue, because no state change occurs.

Workaround: No workaround is available.

Status: No Fix

12. USB Host: Memory Read Request Address Across DWord Boundary May Cause Incomplete Transfer

Problem: When the Device Read Pre-Fetch Control Register (Packet Hub MEM_BASE+014h) for the USB host is enabled, transfer incompletion may occur if the USB host issues byte or word (16 bit) memory read requests to PCIe and this address is not aligned to a DWord boundary (memory read request address is xxxxxxx0/4/8/Ch).

Implication: When transfer incompletion occurs and the invalid transfer continues, the QoS buffer memory of the Intel® PCH EG20T overflows and the system may hang.

Workaround: Do not issue memory read requests across a DWord boundary. Since memory read requests across the DWord boundary may be issued by the Windows* 7 standard USB driver, use the Intel® PCH EG20T driver for Windows 7 version 1.2 or later. It may reduce the probability of an incomplete transfer.

Status: No Fix

Specification Changes

None for this revision of this Specification Update.

Specification Clarifications

1. Clarify the Method of USB Device Controller Hot Unplug/Plug

Issue: Clarified the method of USB Device Hot Unplug/Plug.

Old Text:

9.5.2 Hot-Unplug Is Not Supported

The USB Device controller should be reset properly before Hot-unplug to avoid system hang. An operator should have stopped the driver through an application interface before Hot-unplugging the device. Example, operate the "Safe to remove the device" before Hot-unplugging the device.

New Text:

9.5.2 Hot Unplug/Plug

For users who choose to use VBUS detection method to handle USB Device Controller hot plug/unplug, the following software procedure is needed.

Since the USB Device controller doesn't have a function to detect VBUS directly, it needs a 5V tolerant GPIO pin to detect the loss of VBUS.

After detecting that the VBUS was lost (device is unplugged), software must disable all interrupts of the USB device controller and abort the DMA operation without accessing UDC registers. This operation will terminate DMA operation correctly and prevent unexpected condition.

After detecting that the VBUS was lost (device is unplugged), software should reset and initialize the USB device controller as needed.

Affected Docs: *Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

2. Clarify the Endpoint Interrupt Register and Mask Register in USB Device Controller

Issue: Clarified the relation of Endpoint Interrupt Register and Mask Register.

New Text:

9.5.3 Operation of Endpoint Interrupt Register and Mask Register

When multiple interrupts are triggered, even if software clears the Endpoint (EP) Interrupt register one time, the interrupt output will be asserted again due to loading from Hold register (invisible from software) to EP Interrupt register. The device interrupt Register does not have hold register.

EP Interrupt MASK register does not mask the interrupt output. It masks the interrupt trigger.

Software should mask the EP interrupt based on the following procedure.

1. Before setting the EP interrupt mask, read the Interrupt register (and endpoint status) and clear it to make sure there are no pending interrupts at that point.
2. In case the interrupt event happened during the mask process, then the interrupt will be generated. In this case, read the Interrupt register (and endpoint status) and clear it to make sure there are no pending interrupts after the EP interrupt mask register set also.

Figure 29. USB Device Controller Interrupts

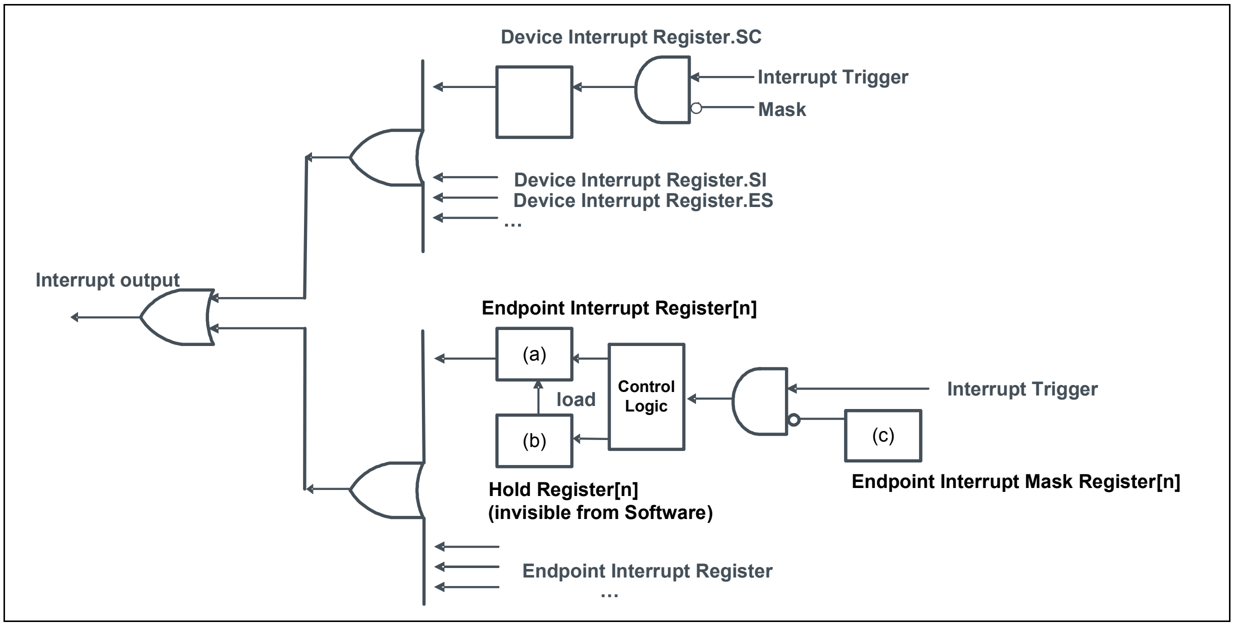
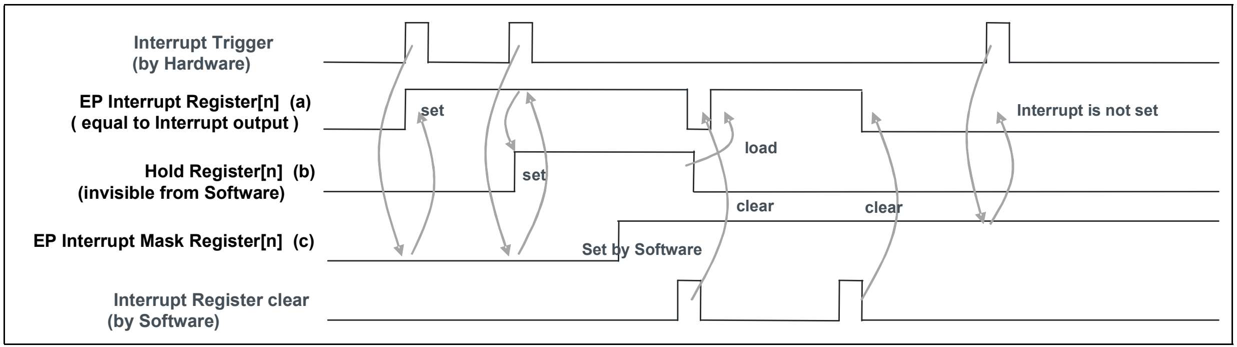

Device Interrupt Register.SC
Interrupt Trigger
Mask
Device Interrupt Register.SI
Device Interrupt Register.ES
...
Interrupt output
Endpoint Interrupt Register[n]
(a)
load
(b)
Control Logic
Interrupt Trigger
(c)
Endpoint Interrupt Mask Register[n]
Hold Register[n]
(invisible from Software)
Endpoint Interrupt Register
...

Figure 30. Endpoint Interrupt Register and Mask Operation



Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

3. Updated IDCODE for A3 stepping

Issue: Clarified IDCODE version for A3 stepping.

Old Text:

19.1.5.1 IDCODE

The IDCODE instruction is used to select an ID register, where the device information including serial numbers and parts numbers is stored. The instruction can be used to verify the part that is mounted on the board is matched.

The IDCODE of Intel Intel® Platform Controller Hub EG20T is "0000 1010 0101 0000 0000 0000 0001 0011"

From MSB, version (0000), parts No. (1010 0101 0000 0000), Manufacturer ID (0000 0001 001), 1(fixed value)

New Text:

19.1.5.1 IDCODE

The IDCODE instruction is used to select an ID register, where the device information including serial numbers and parts numbers is stored. The instruction can be used to verify the part that is mounted on the board is correct.

The IDCODE of Intel Intel® Platform Controller Hub EG20T is:
0011 1010 0101 0000 0000 0000 0001 0011 (A3 stepping)
0010 1010 0101 0000 0000 0000 0001 0011 (A2 stepping)

Key (from MSB): version (001x), part No. (1010 0101 0000 0000), Manufacturer ID (0000 0001 001), 1(fixed value)

Note: The version number for the A3 stepping is 0011b, and for the A2 stepping is 0010b.

Affected Docs: *Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211.

4. Add in Section 9.5.4 USB Device DMA Operation

Issue: DMA Operation is not included in Datasheet.

New Text:

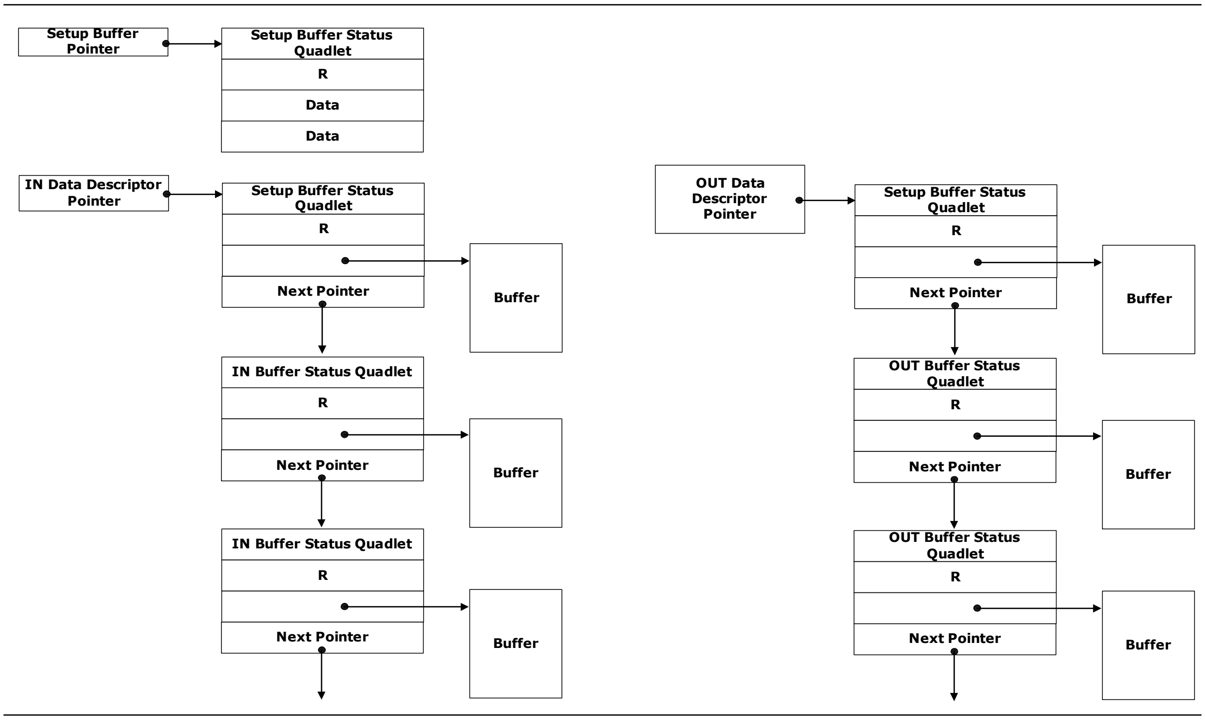
9.5.4 USB_Device DMA Operation

This chapter explains the functionality of the USB_Device Subsystem in DMA mode. A DMA-based implementation of the USB_Device is best suited to designs requiring a high degree of configurability and software control. A major advantage of DMA-based implementations is that they spare the main processor’s computing power from involvement in data movement tasks. Use of a scatter-gather DMA helps applications make efficient and optimal use of system memory, a major design constraint on portable systems. The DMA mode of implementation demands that data be presented in predefined memory structure formats. These memory structures are explained in detail in the following subtopics.

In DMA mode, USB_Device implements a true scatter-gather memory distribution, in which memory structures are scattered over the system memory. Each endpoint memory structure is implemented as a linked list, where each element of the list is a data buffer of a predefined size. In addition to data, each buffer also has a status quadlet and a pointer to the next buffer. The last element of such a linked list can point either to a null pointer, or, if the list is implemented as a ring buffer, to the first element of the list. All the endpoints implement these structures in the memory, and apart from these structures, all control endpoints implement an additional 16-byte buffer to store SETUP data. The SETUP data structure does not implement a linked-list structure.

The descriptor memory structures are displayed in Figure 27.

Figure 27 Descriptor Memory Structures



9.5.4.1 SETUP Data Memory Structure

Figure 28 shows the memory structure for SETUP data. Table 297 describes the SETUP Data Memory Structure values. The subsystem receives SETUP data only from the USB host. The buffer size for SETUP data is always 16 bytes. The SETUP buffer applies only to control endpoints. A "Host Busy" buffer status indicates that the application is accessing the location. DMA does not check the buffer status before transferring the 8-byte SETUP packet. The application must always give highest priority to, and have the buffer ready for, a SETUP packet.

Figure 28 SETUP Data Memory Structure

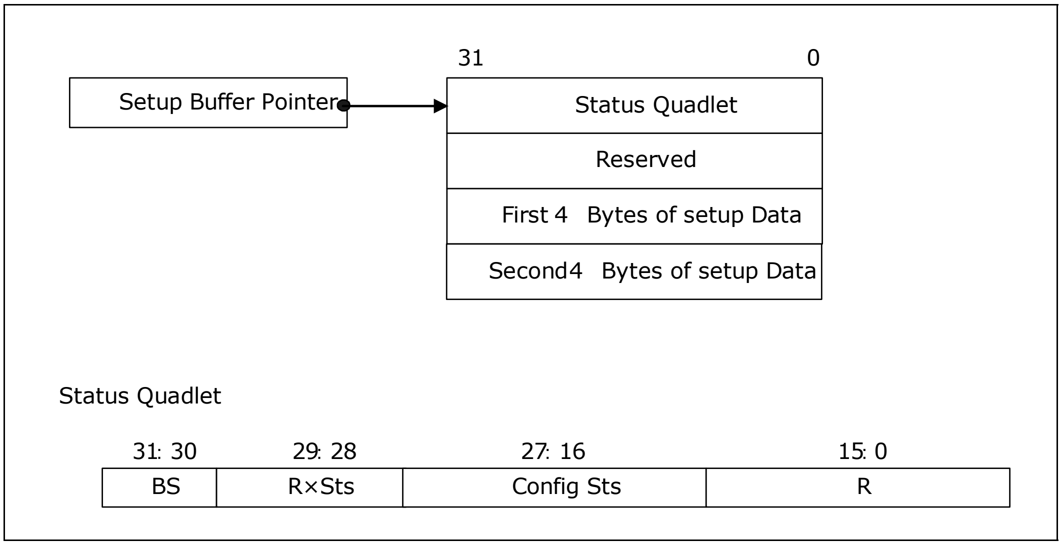


Table 297 SETUP Data Memory Structure Values

Bit	Bit ID	Description
BS [1:0]	Buffer Status	Status of the data buffer. The possible options are: • 2'b00: Host ready • 2'b01: DMA busy • 2'b10: DMA done • 2'b11: Host busy
Rx Sts [1:0]	Receive Status	Status of received SETUP data. This reflects whether the SETUP data is received correctly or with errors. The possible combinations are: • 2'b00: Success • 2'b01: DESERR • 2'b10: Reserved • 2'b11: BUFERR
Config Sts [27:16]	Configuration Status	Status of the current configuration associated with the SETUP packet for a control endpoint (endpoint type "00", which is different from the default endpoint 0). • Configuration number [27:24] • Interface number [23:20] •Alternate setting number [19:16]
R [15:0]	Reserved	Reserved for future use.

9.5.4.2. OUT Data Memory Structure

All endpoints that support OUT direction transactions (endpoints that receive data from the USB host), must implement a memory structure with the following characteristics:

- Each data buffer must have a descriptor associated with it that provides the status of the buffer. The buffer itself contains only raw data.
- Each buffer descriptor is 4 quadlets in length.

If the buffer status of the first descriptor is "Host Ready," the DMA fetches and processes its data buffer; otherwise the DMA skips to the next descriptor until it reaches the end of the descriptor chain. The buffers the descriptor points to hold packet data for non-isochronous endpoints and frame data for isochronous endpoints. Buffer fill mode does not support Isochronous OUT endpoints. Use packet-perbuffer mode instead.

"Host Ready" indicates that the descriptor is available for the DMA to process. "DMA Busy" indicates that the DMA is still processing the descriptor. "DMA Done" indicates that the buffer data transfer is complete. "Host Busy" indicates that the application is processing the descriptor. A DESERR receive status indicates that the buffer status is something other than host ready (2'b00) during descriptor fetch.

The OUT data memory structure is shown in Figure 29.

Figure 29. OUT Data Memory Structure

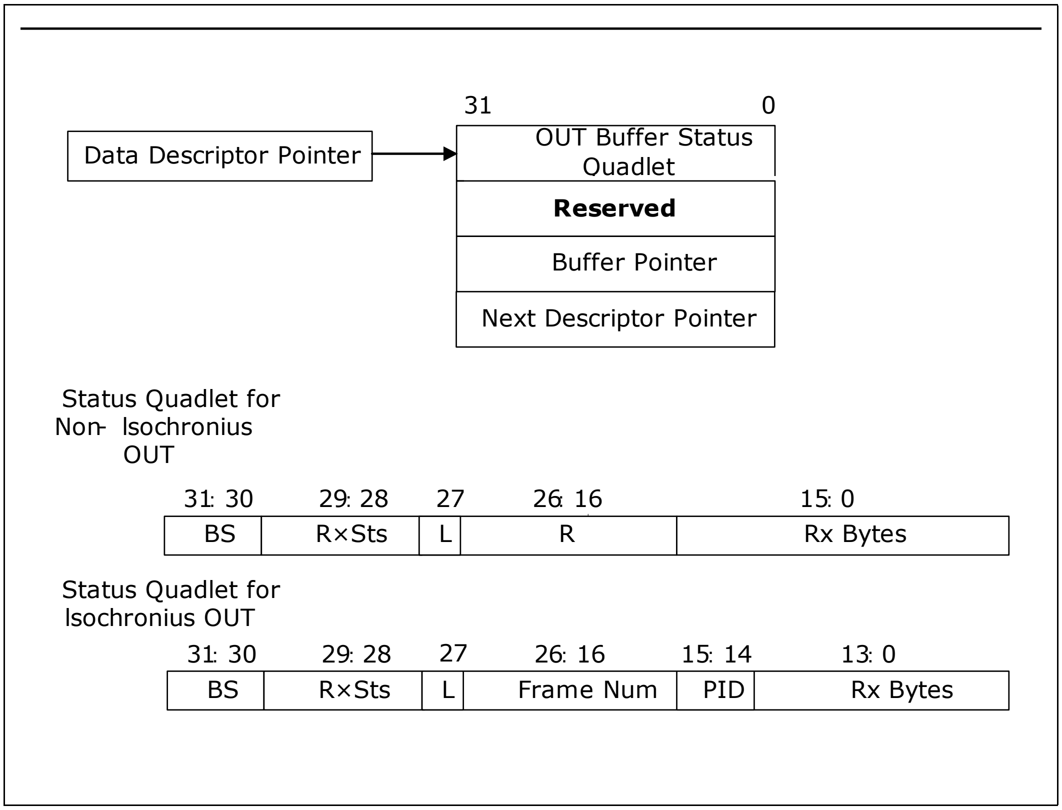


Table 298. OUT Data Memory Structure Values

Bit	Bit ID	Description
BS [31:30]	Buffer Status	This 2-bit value describes data buffer status. Possible options are: • 2'b00: Host Ready • 2'b01: DMA Busy • 2'b10: DMA Done • 2'b11: Host Busy
Rx Sts [29:28]	Receive Status	This 2-bit value describes the status of the received data. This reflects whether OUT data has been received correctly or with errors. The possible combinations are: • 2'b00: Success • 2'b01: DESERR • 2'b10: Reserved • 2'b11: BUFERR

Table 298. OUT Data Memory Structure Values

Bit	Bit ID	Description
L [27]	Last	When set, this bit indicates that this descriptor is the last one of the chain.
Frame Number [26:16] (ISO OUT)	Frame number	The 11-bit frame number in which the current ISO-OUT packet is received.
R [26:16] (non-ISO OUT)	Reserved	Reserved.
PID [15:14] (ISO OUT)	ISO Received Data PID	This field is for only high-speed isochronous transactions, and indicates the data PID for an isochronous receive packet. • 2'b00: The packet contained in this descriptor is received with a data PID of DATA0. • 2'b01: The packet contained in this descriptor is received with a data PID of DATA1. • 2'b10: The packet contained in this descriptor is received with a data PID of DATA2. • 2'b11: The packet contained in this descriptor is received with a data PID of MDATA.
Rx Bytes [15:0] (non-ISO OUT)	Received number of bytes	This 16-bit value can take values from 0 to 64K bytes*, depending on the packet size of data received from the USB host. *Note: Actual value is (64K – 1) bytes. Equivalent to ‘hFFFF.
Rx Bytes [13:0] (ISO OUT)	Received number of bytes	This 14-bit value can take values from 0 to 16K bytes*, depending on the packet size of data received from the USB host. *Note: Actual value is (16K – 1) bytes. Equivalent to `h3FFF.

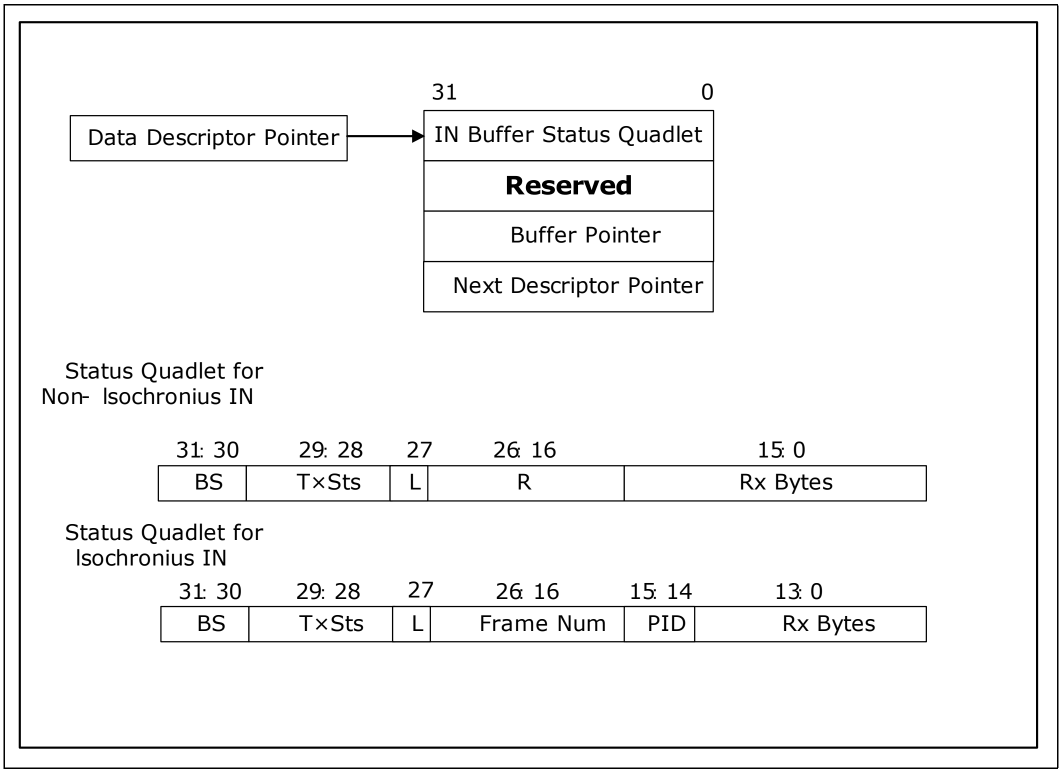
9.5.4.3 IN Data Memory Structure

All endpoints that support IN direction transactions (transmitting data to the USB host) must implement the following memory structure. Each buffer must have a descriptor associated with it. The application fills the data buffer, updates its status in the descriptor, and sets the Poll Demand bit. The DMA fetches this descriptor and processes it, moving on in this fashion until it reaches the end of the descriptor chain.

The buffers the descriptor points to hold packet data for non-isochronous endpoints and frame data for isochronous endpoints.

The IN data memory structure is shown in Figure 30.

Figure 30. IN Data Memory Structure



Status Quadlet for Non- Isochronius IN

31: 30	29: 28	27	26: 16	15: 0
BS	TxSts	L	R	Rx Bytes

Status Quadlet for Isochronius IN

31: 30	29: 28	27	26: 16	15: 14	13: 0
BS	TxSts	L	Frame Num	PID	Rx Bytes

Table 299. IN Data Memory Structure Values

Bit	Bit ID	Description
BS [31:30]	Buffer Status	This 2-bit value describes the status of the data buffer. The possible options are: • 2'b00: Host ready • 2'b01: DMA busy • 2'b10: DMA done • 2'b11: Host busy
Tx Sts [29:28]	Transmit Status	The status of the transmitted data. This reflects if the IN data has been transmitted correctly or with errors. The possible combinations are: • 2'b00: Success • 2'b01: DESERR • 2'b10: Reserved • 2'b11: BUFERR
L [27]	Last	When set, this bit indicates that this descriptor is the last one of the chain.
Frame Number [26:16] (ISO IN)	Frame Number	Frame number in which the current packet must be transmitted. • [26:19]: millisecond frame number • [18:16]: microframe number
R[26:16] (non-ISO IN)	Reserved	Reserved.
PID [15:14] (ISO IN)	Number of packets per frame	This 2-bit value indicates the number of packets per μSOF (microframe) for isochronous IN transfers during high-speed operation. The application must program these bits in the descriptor (these bits must be the same for all descriptors of the same μSOF) such that the subsystem core returns an isochronous packet with an appropriate data PID per frame. These bits are reserved for full-speed operation. • 2'b00 and 2'b01: 1 packet per microframe • 2'b10: 2 packets per microframe • 2'b11: 3 packets per microframe
Tx bytes [15:0] (non-ISO IN)	Number of bytes to be transmitted	This 16-bit value can take values from 0 to 64K bytes*, indicating the number of bytes of data to be transmitted to the USB host. *Note: Actual value is (64K – 1) bytes. Equivalent to ‘hFFFF.
Tx bytes [13:0] (ISO IN)	Number of bytes to be transmitted	This 14-bit value can take values from 0 to 16K bytes*, indicating the number of bytes of data to be transmitted to the USB host. *Note: Actual value is (16K – 1) bytes. Equivalent to `h3FFF.

Affected Docs: *Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211.

5. Corrected Notes for Table 226 “54h: PORTSC - Port Status and Control Register”

Issue: The Intel® EG20T does not support USB Debug Port.

Old Text:

Table 226. 54h: PORTSC - Port Status and Control Register

Notes:

1. When a device is attached, the port state transitions to the connected state and the system software processes this register as with any status change notification. Refer to EHCI Specification Section 4.3 for operational requirements for how the change events interact with port suspend mode.
2. If a port is being used as the Debug Port, the port may report device connected and enabled when the Configured Flag is 0.
3. A host controller must implement one or more port registers. The number of port registers implemented by a particular instantiation of a host controller is documented in the HCSPARAMs register (EHCI Specification Section 2.2.3). The software uses this information as an input parameter to determine how many ports need to be serviced. All ports have the structure defined below.
4. This register is in the auxiliary power well. It is only reset by hardware when the auxiliary power is initially applied or in response to a host controller reset. The initial conditions of a port are:
   - No device connected
   - Port disabled
5. If the port has port power control, the software cannot change the state of the port until after it applies power to the port by setting port power to 1. The software must not attempt to change the state of the port until after power is stable on the port. The host is required to have power stable to the port within 20 milliseconds of the zero to one transition.

New Text: (Deleted Note 2)

Table 226. 54h: PORTSC - Port Status and Control Register

Notes:

1. When a device is attached, the port state transitions to the connected state and the system software processes this register as with any status change notification. Refer to EHCI Specification Section 4.3 for operational requirements for how the change events interact with port suspend mode.
2. A host controller must implement one or more port registers. The number of port registers implemented by a particular instantiation of a host controller is documented in the HCSPARAMs register (EHCI Specification Section 2.2.3). The software uses this information as an input parameter to determine how many ports need to be serviced. All ports have the structure defined below.
3. This register is in the auxiliary power well. It is only reset by hardware when the auxiliary power is initially applied or in response to a host controller reset. The initial conditions of a port are:
   - No device connected
   - Port disabled
4. If the port has port power control, the software cannot change the state of the port until after it applies power to the port by setting port power to 1. The software must not attempt to change the state of the port until after power is stable on the port. The host is required to have power stable to the port within 20 milliseconds of the zero to one transition.

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211.

6. Corrected Error in Table 863 “Power and Ground Signals”

Issue: Corrected ball assignment in Table 863 to clarify ball K1 in a separate row.

Old Text:

Table 863. Power and Ground Signals

Power Plane	Power Pin	Function	Ball #
Plane C	VSUS1	SATA1 Analog VSS	P1, K1

New Text:

Table 863. Power and Ground Signals

Power Plane	Power Pin	Function	Ball #
Plane C	VSUS1	SATA1 Analog VSS	P1
Plane C	VSUSA	SATA0/1 Analog VSS	K1

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211.

Document-Only Changes

1. Correct Address in Chapter 13 CAN Controller: “IF1 and IF2 Message Interface Register Sets” Table

Issue: Corrected errors in table content

Old Text:

Table 515. IF1 and IF2 Message Interface Register Sets

Address	IF1 register set	Address	IF2 register set
BASE + 020h	IF1 command request register	BASE080h	IF2 command request register
BASE + 024h	IF1 command mask register	BASE084h	IF2 command mask register
BASE + 028h	IF1 mask 1 register	BASE088h	IF2 mask 1 register
BASE + 02Ch	IF1 mask 2 register	BASE08Ch	IF2 mask 2 register
BASE + 030h	IF1 identifier 1 register	BASE090h	IF2 identifier 1 register
BASE + 034h	IF1 identifier 2 register	BASE094h	IF2 identifier 2 register
BASE + 038h	IF1 message control register	BASE098h	IF2 message control register
BASE + 03Ch	IF1 data A1 register	BASE09Ch	IF2 data A1 register
BASE + 040h	IF1 data A2 register	BASE0A0h	IF2 data A2 register
BASE + 044h	IF1 data B1 register	BASE0A4h	IF2 data B1 register
BASE + 048h	IF1 data B2 register	BASE0A8h	IF2 data B2 register

Notes:

1. Reserved bits are read as 0. Always write 0 to the reserved bits. Operation cannot be guaranteed if 1 is written.
2. The reserved bits of the IF1 mask 2 register and IF2 mask 2 register are read “1”. Always write 1 to these reserved bits. Operation cannot be guaranteed, if 0 is written.

New Text:

Table 515. IF1 and IF2 Message Interface Register Sets (Sheet 1 of 2)

Address	IF1 register set	Address	IF2 register set
BASE + 020h	IF1 command request register	BASE + 080h	IF2 command request register
BASE + 024h	IF1 command mask register	BASE + 084h	IF2 command mask register
BASE + 028h	IF1 mask 1 register	BASE + 088h	IF2 mask 1 register
BASE + 02Ch	IF1 mask 2 register	BASE + 08Ch	IF2 mask 2 register
BASE + 030h	IF1 identifier 1 register	BASE + 090h	IF2 identifier 1 register
BASE + 034h	IF1 identifier 2 register	BASE + 094h	IF2 identifier 2 register

Table 515. IF1 and IF2 Message Interface Register Sets (Sheet 2 of 2)

BASE + 038h	IF1 message control register	BASE + 098h	IF2 message control register
BASE + 03Ch	IF1 data A1 register	BASE + 09Ch	IF2 data A1 register
BASE + 040h	IF1 data A2 register	BASE + 0A0h	IF2 data A2 register
BASE + 044h	IF1 data B1 register	BASE + 0A4h	IF2 data B1 register
BASE + 048h	IF1 data B2 register	BASE + 0A8h	IF2 data B2 register

Notes:
1. Reserved bits are read as 0. Always write 0 to the reserved bits. Operation cannot be guaranteed if 1 is written.
2. The reserved bits of the IF1 mask 2 register and IF2 mask 2 register are read "1". Always write 1 to these reserved bits. Operation cannot be guaranteed, if 0 is written.

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

2. Corrected Title in "MSGLST" Table

Issue: Corrected typo in table title.

Old Text:

Table 536. MsGLST (Only valid for receive message objects with direction = receive)

New Text:

Table 536. MSGLST (Only valid for receive message objects with direction = receive)

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

3. Corrected Typo in Figure 42 "Correlation between the IFm Register and the Message RAM"

Issue: Corrected typo in the content of Figure 42.

Old Text:

MASK[23:0]

Message Number[7:0]

New Text:

MASK[28:0]

Message Number[5:0]

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

4. Corrected Typo in Figure 44 "Concept Flowchart of Transmit Message Handling"

Issue: Corrected typo in the content of Figure 44.

Old Text:

CLRINTPND ⟵ 1

New Text:

CLRINTPND ⟵ 0

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

5. Corrected Typo in Figure 45 “Concept Flowchart of Receive Message Handling”

Issue: Corrected typo in the content of Figure 45.

Old Text:

Read CANNDATAn 1 and CNNDATAn 2 and verify updated message number

...

MASK ⟵ 0

ARB ⟵ 0

New Text:

Read CANNDATAn 1 and CANNDATAn 2 and verify updated message number

...

MASK ⟵ 1

ARB ⟵ 1

Affected Docs: *Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

6. Corrected Typo in Chapter 13 CAN Controller Table 525 “Structure of a Message Object in the Message RAM”

Issue: Corrected typo in table content.

Old Text:

Table 525. Structure of a Message Object in the Message RAM

One message object												
MXTD	MDIR	MSK[28: 0]	XTD	DIR	ID[28: 0]	MSGLST	UMASK	TXIE	RXIE	RMTER	EoB	DLC[3: 0]
DATA0 [7: 0]	DATA1 [7: 0]	DATA2 [7: 0]	DATA3 [7: 0]	DATA4 [7: 0]	DATA5 [7: 0]	DATA6 [7: 0]	DATA7 [7: 0]	TXRQST	NEWDAT	INTPND	MSGVALI	

Note: Above table simply shows the structure of a message object, and there is no correlation between the higher and lower bits.

New Text:

Table 525. Structure of a Message Object in the Message RAM

One message object												
MXTD	MDIR	MSK[28: 0]	XTD	DIR	ID[28: 0]	MSGLST	UMASK	TXIE	RXIE	RMTER	EoB	DLC[3: 0]
DATA0 [7: 0]	DATA1 [7: 0]	DATA2 [7: 0]	DATA3 [7: 0]	DATA4 [7: 0]	DATA5 [7: 0]	DATA6 [7: 0]	DATA7 [7: 0]	TXRQST	NEWDAT	INTPND	MSGVAL	

Note: Above table simply shows the structure of a message object, and there is no correlation between the higher and lower bits.

Affected Docs: *Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

7. Correct Typo in Chapter 13 CAN Controller Table 517 “24h: IFmCMASK: m = 1, 2 - IFm Command Mask Register”

Issue: Deleted redundant text in bit 1 description.

Old Text:

01	0b	RW	DATA_A	Access data bytes 0-3. At WR/RD bit write: 0 = Data bytes 0-3 (IF data A) remain unchanged. 1 = Transfers the data bytes 0-3 to a message object. At WR/RD bit read: 0 = Data bytes 0-3 (IF data A) remain unchanged. 1 = Transfers the data bytes 0-3 to the IFm message buffer register. Transfers the data bytes 0-3 to the IFm message buffer register.

New Text:

01	0b	RW	DATA_A	Access data bytes 0-3. At WR/RD bit write: 0 = Data bytes 0-3 (IF data A) remain unchanged. 1 = Transfers the data bytes 0-3 to a message object. At WR/RD bit read: 0 = Data bytes 0-3 (IF data A) remain unchanged. 1 = Transfers the data bytes 0-3 to the IFm message buffer register.

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

Affected Docs:

8. Corrected Error in Table 230 under Chapter 8.3.3.1 HcRevision Register

Issue: Corrected error in table content.

Old Text:

Table 230. 00h: HcRevision Register

08	0b	RO	L	**Legacy:** This read-only field is set to 1 to indicate that the legacy support registers are present in this HC.

New Text:

Table 230. 00h: HcRevision Register

08	0b	RO	L	**Legacy:** This read-only field is set to 0 to indicate that the legacy support registers are not present in this HC.

Affected Docs:*Intel® Platform Controller Hub EG20T Datasheet*, revision 008, order number 324211

§ §