# INIC-1511
# USB to PATA Bridge
# Specification

**Version 0.5**
**June 12, 2006**
**Initio Corporation**

Change History:

Table of Contents

# 1. Introduction:

The INIC-1511 provides an advanced solution to connect ATAPI or EIDE devices to USB interface with integrated CPU and embedded ROM. To provide a high performance and cost effective solution, the INIC-1511 integrates USB-PHY, Mass Storage Class Bulk-Only USB function, ATA control block and microprocessor into a single ASIC. The INIC-1511 provides the data transfer rate of up to 60 MB/sec; its ATA interface supports ultra DMA modes up to 100 MB/sec.

## 1.1 Feature Summary

- USB Mass Storage Class Bulk-Only specification-compliant (version 1.0)
- Two Bulk endpoints(IN and OUT) and one Interrupt endpoint(IN)
- T13 1410D ATA/ATAPI-6 Compliant(3.3 V with 5V tolerance)
- Support DMA mode 0-2, and UDMA mode 2, 3, 4 and 5 (up to 100MB/s)
- Integrated internal CPU with embedded ROM
- Implement the NVRAM download mechanism
- Low power CMOS with 3.3Volts
- 48-pin LQFP package

## 1.2 Firmware/Software Support

- USB Mass Storage Class Bulk-Only Transport support
- Provide software utilities for downloading the upgraded NVRAM

## 1.3 Devices Support

- Hard disk drives
- CD-RW devices
- DVDs
- Removable media devices.

## 2. USB to ATA Block Diagram:

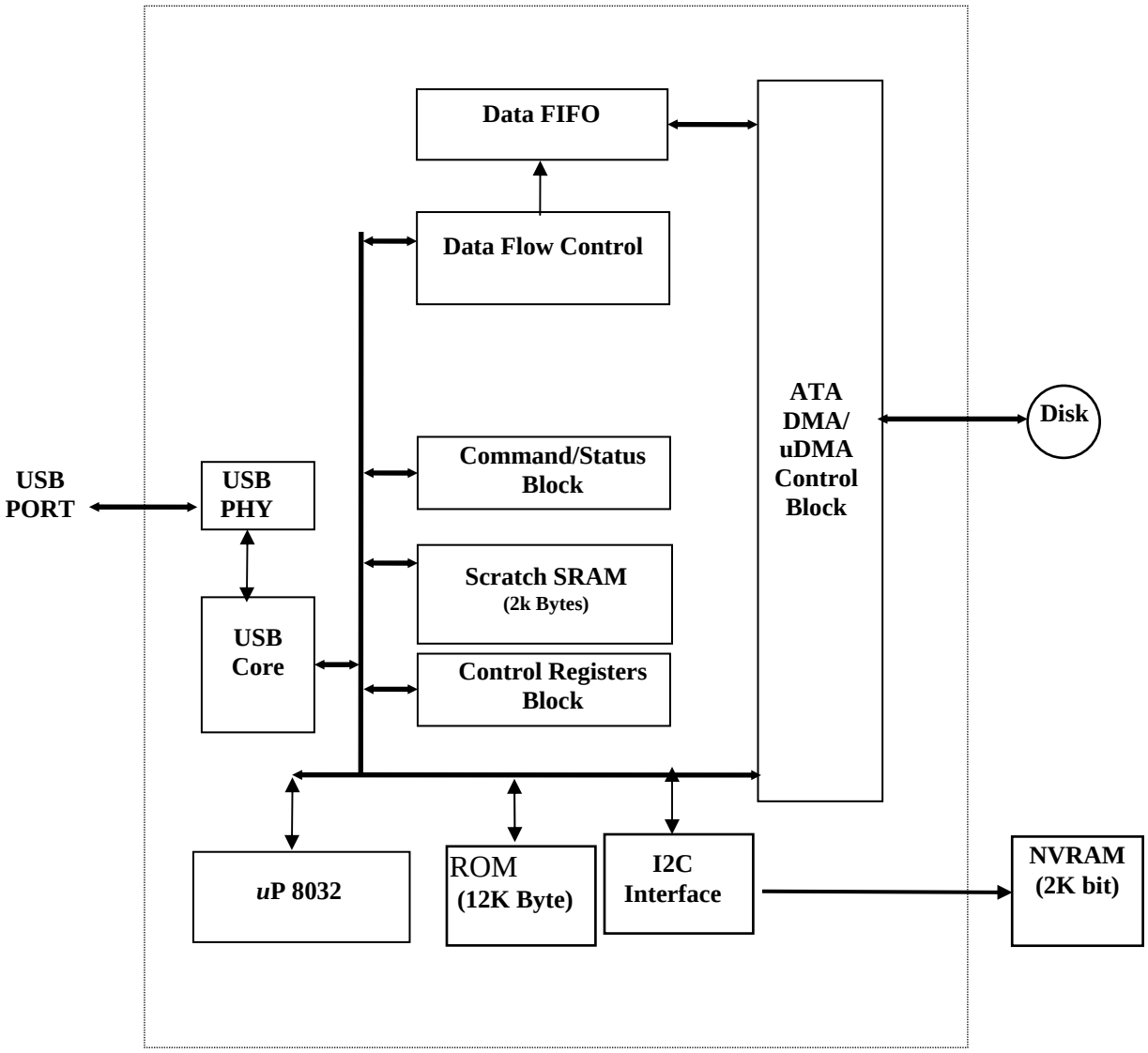



**Figure 1: INIC-1511 Block Diagram**

## 3. Pin Signal Description: (48-pin package)

### 3.1 USB Interface

| Signal Name | Pin Number | I/O | Driver Type | Description |
|---|---|---|---|---|
| DP | 16 | I/O | USB high /full speed buffer (D+) | High/Full speed D+ signal |
| DM | 17 | I/O | USB high/full speed buffer (D-) | High/Full speed D- signal |
| RREF | 14 | A | Power | PLL voltage reference. Current source for 330R(1%) resistor connected to AVSS |
| VBUS | 13 | I | PISW | Active HIGH. Indicates that VBUS is present. |
| XIN | 19 | I | A | Crystal oscillator input (12MHz) |
| XOUT | 20 | O | A | Crystal oscillator output (12MHz) |

### 3.2 NVRAM Interface

| Signal Name | Pin Number | I/O | Driver Type | Description |
|---|---|---|---|---|
| I2C_SCL | 7 | I/O | PBU4W | NVRAM Clock, or P3_3<br>Internal pullup |
| I2C_SDA | 8 | I/O | PBU4W | NVRAM Data., or P1_4<br>Internal pullup |

### 3.3 ATA Interface (driving current 8mA-16mA controlled by Register 0x40A8[1:0])

| Signal Name | Pin Number | I/O | Driver Type | Description |
|---|---|---|---|---|
| ataD[15:0] | 35, 37, 39, 41,<br>43, 45, 47, 1,<br>48, 46, 44, 42,<br>40, 38, 36, 34 | I/O | PBSCUDSL | ATA Data Bus |
| ataDA[2:0] | 27, 26, 25 | I/O | PBSCUDSL | ATA Device Address |
| ataRST# | 5 | I/O | PBSCUDSL | ATA Reset (Out) |
| ataCS[1:0]# | 3, 2 | I/O | PBSCUDSL | ATA Device Chip Select (Out) |
| ataDMARQ | 33 | I/O | PBSCUDSL | ATA DMA Request (In) |
| ataDMACK# | 28 | I/O | PBSCUDSL | ATA DMA Acknowledge (Out) |

| | | | | |
|---|---|---|---|---|
| ataDIOW# | 32 | I/O | PBSCUDSL | ATA I/O Write (Out) |
| ataDIOR# | 30 | I/O | PBSCUDSL | ATA I/O Read (Out) |
| ataIORDY | 29 | I/O | PBSCUDSL | ATA I/O ready (In) |
| ataINTR | 4 | I/O | PBSCUDSL | ATA Interrupt Request |

## 3.4 System Interface

| Signal Name | Pin Number | I/O | Driver Type | Description |
|---|---|---|---|---|
| PORST# | 12 | I | PISW | Power On Reset. When this signal is active, all of pins on ATA interface should be tri-stated. |
| TEST0 | 11 | I | PIDW | Test Mode Select. Default 0.<br>0-> Normal Operation Mode<br>1-> Test Mode |

## 3.5 GPIO Interface

| Signal Name | Pin Number | I/O | Driver Type | Description |
|---|---|---|---|---|
| P1_0 | 10 | I/O | PBU4W | uP8032 I/O port . Internal pullup |

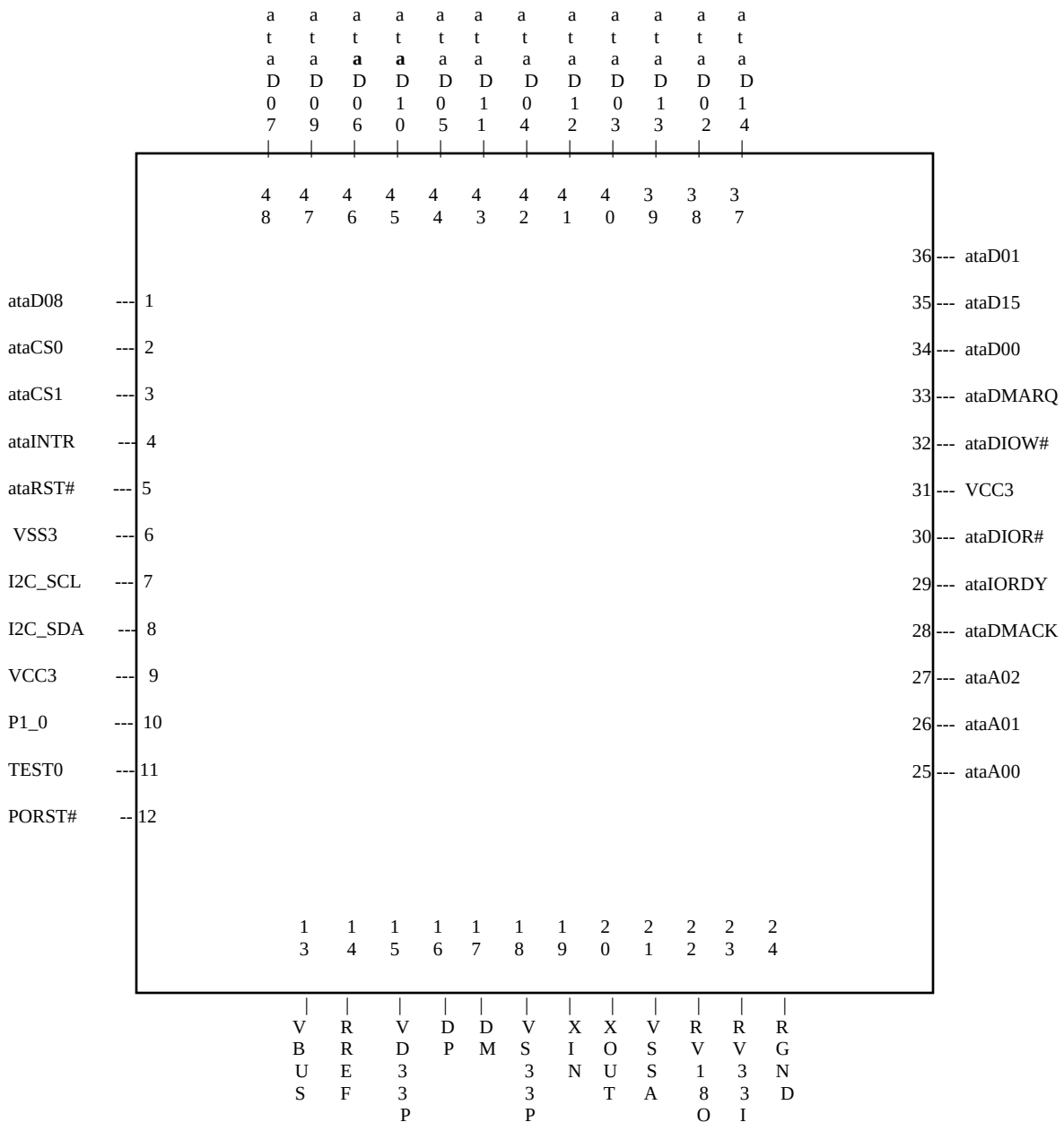
## 3.6 Power Regulator pins

| Signal Name | Pin Number | I/O | Driver Type | Description |
|---|---|---|---|---|
| RV33I | 23 | I | | Internal regulator 3.3V input |
| RGND | 24 | I | | Regulator ground |
| RV18O | 22 | O | | Internal regulator 1.8V output |

## 3.7 Power/GND

| Signal Name | Pin Number | I/O | Driver Type | Description |
|---|---|---|---|---|
| VCC3 | 9,31 | | | IO 3.3V |
| VSS3 | 6 | | | IO GND |
| VD33P | 15 | | | 3.3V, for VDD33P and VD33 |
| VS33P | 18 | | | USB IO Cell GND |
| VSSA | 21 | | | VSSA for PLL |

## 3.8 48-Pin Diagram



# 4. NVRAM Program/Download procedure

## 4.1 CPU Write NVRAM.

1. CPU write access address at register I2C_Addr(0x40E0).
2. CPU write data at register I2C_Data(0x40E1).
3. CPU write Control Code at register I2C_Ctrl(0x40E2).
4. CPU write Run and Read_Write direction(0) at register I2C_Comm(0x40E3).
5. CPU poll I2C_Comm bit 7, wait until cleared.
6. CPU may read register I2C_Status(0x40E4) to check write success or not.

## 4.2 CPU Read NVRAM.

1. CPU write access address at register I2C_Addr(0x40E0).
2. CPU write Control Code at register I2C_Ctrl(0x40E2).
3. CPU write Run and Read_Write direction(1) at register I2C_Comm(0x40E3).
4. CPU poll I2C_Comm bit 7, wait until cleared.
5. CPU read register I2C_Data(0x40E1).
6. CPU may read register I2C_Status(0x40E4) to check read success or not.

## 4.3 CPU Poll NVRAM.

After write data to NVRAM, NVRAM need certain time before next write operation can be accepted. CPU may poll NVRAM to decide NVRAM ready or not. It is done same as a NVRAM read. If I2C_Status(0x40E4) bit 0 is 1, the NVRAM not ready.

## 4.4 Host Read/Write 8051 data space from USB

1. Host send READ_CHIP_ID packet through control channel to read chip-ID, which is 0x29C5_1511 here. Default hardware report PID is 0x1518.
2. Host send HOLD_CPU packet through control channel to set HOLD_CPU bit.
3. Host may send DATA_WRITE/DATA_READ packet through control channel to WRITE/READ 8051 data space.

**DATA_WRITE setup packet format is,**

| offset | field | size | value | Description |
|---|---|---|---|---|
| 0 | bmReqType | 1 | 0x40 | Vendor write |
| 1 | bReq | 1 | 0x81 | Data space write |
| 2 | wValue | 2 | Addr[7:0] | Address to be writen |
| 3 | | | Addr[15:8] | |
| 4 | wIndex | 2 | Data[7:0] | Data space data |
| 5 | | | 0x00 | Don't care |
| 6 | wLength | 2 | 0x00 | |
| 7 | | | 0x00 | |

**DATA_READ setup packet format is,**

| offset | field | size | value | data | Description |
|---|---|---|---|---|---|
| 0 | bmReqType | 1 | 0xc0 | Data from Data space | Vendor read |
| 1 | bReq | 1 | 0x82 | | Data read |
| 2 | wValue | 2 | Addr[7:0] | | Address to be writen |
| 3 | | | Addr[15:8] | | |

| | | | | | |
|---|---|---|---|---|---|
| 4 | wIndex | 2 | 0x00 | | Don't care |
| 5 | | | 0x00 | | Don't care |
| 6 | wLength | 2 | 0x01 | | |
| 7 | | | 0x00 | | |

**READ_CHIP_ID setup packet format is:**

| Offset | Field | Size | Value | Data | Description |
|---|---|---|---|---|---|
| 0 | bmReqType | 1 | 0xc0 | Chip-ID 0x11, 0x15, 0xc9, 0x25 | Vendor read |
| 1 | bReq | 1 | 0x03 | | |
| 2 | wValue | 2 | 0x00 | | |
| 3 | | | 0x00 | | |
| 4 | wIndex | 2 | 0x00 | | Don't care |
| 5 | | | 0x00 | | Don't care |
| 6 | wLength | 2 | 0x04 | | |
| 7 | | | 0x00 | | |

**HOLD_CPU setup packet format is:**

| Offset | Field | Size | Value | Description |
|---|---|---|---|---|
| 0 | bmReqType | 1 | 0x40 | Vendor write |
| 1 | bReq | 1 | 0x04 | HOLD_CPU |
| 2 | wValue | 2 | 0x00 | Don't care |
| 3 | | | 0x00 | |
| 4 | wIndex | 2 | 0x00 | Don't care |
| 5 | | | 0x00 | Don't care |
| 6 | wLength | 2 | 0x00 | Don't care |
| 7 | | | 0x00 | Don't care |

## 4.5 NVRAM Download from USB cable

The download utility may read/write NVRAM through access I2C registers similar as CPU does.

1. Host send READ_CHIP_ID packet through control channel to read chip-ID, which is 0x29C5_1511 here. Default hardware report PID is 0x1518.
2. Host send HOLD_CPU packet through control channel to set HOLD_CPU bit.

### 4.5.1 NVRAM Write.

3. Host send DATA_WRITE packet with wValue I2C_Addr(0x40E0) and wIndex[15:8] the NVRAM address to be accessed
4. Host send DATA_WRITE packet with wValue I2C_Data(0x40E1) and wIndex[15:8] the value to be write to NVRAM
5. Host send DATA_WRITE packet with wValue I2C_Ctrl(0x40E2) and wIndex[15:8] the Control Code to be write to NVRAM
6. Host send DATA_WRITE packet with wValue I2C_Comm(0x40E3) and wIndex[15:8] the Run bit[b7] and read/write direction 0 [b0]

7. Host poll bit 7 of I2C_Comm(0x40E3) until this bit is cleared by sending DATA_READ packet with wValue I2C_Comm(0x40BF)
8. Host send DATA_READ packet with wValue I2C_Status(0x40E4) to check write success or not

#### 4.5.2 NVRAM Read.

9. Host send DATA_WRITE packet with wValue I2C_Addr(0x40E0) and wIndex[15:8] the NVRAM address to be accessed
10. Host send DATA_WRITE packet with wValue I2C_Ctrl(0x40E2) and wIndex[15:8] the Control Code to be write to NVRAM
11. Host send DATA_WRITE packet with wValue I2C_Comm(0x40E3) and wIndex[15:8] the Run bit[b7] and read/write direction 1 [b0]
12. Host poll bit 7 of I2C_Comm(0x40E3) until this bit is cleared by sending DATA_READ packet with wValue I2C_Comm(0x40E3)
13. Host send DATA_READ packet with wValue I2C_Data(0x40E1) to get the data from NVRAM.
14. Host send DATA_READ packet with wValue I2C_Status(0x40E4) to check write success or not

#### 4.5.3 NVRAM Polling.

Same as NVRAM Read. If I2C_Status(0x40E4) bit 0 is 1, NVRAM not ready for next write.

## 5. Register Address Mapping:

### 5.1 USB Block (address area: 60xx)

| Address | Read Value | Write Value |
|---|---|---|
| 20h | Dev_Status | Dev_Status |
| 21h | Funct_Adr | Funct_Adr |
| 22h | Test_mode | Test_mode |
| 25h | EpTxLength[7:0] | EpTxLength[7:0] |
| 26h | EpTxLength[15:8] | EpTxLength[15:8] |
| 30h | EP0_Status | EP0_ Control (Set) |
| 31h | EP0_Status | EP0_ Control (Clear) |
| 32h | EP0_Status2 | EP0_ Control 2(Set) |
| 33h | EP0_Status2 | EP0_ Control 2(Clear) |
| 34h | EP0TxLength | EP0TxLength |
| 38-3F | Hdr0-7 | - |
| 40h | EP1_Status | EP1_ Control (Set) |
| 41h | EP1_Status | EP1_ Control (Clear) |
| 50h | EP2_Status | EP2_ Control (Set) |
| 51h | EP2_Status | EP2_ Control (Clear) |
| 52h | Usb_rxLength[7:0] | - |
| 53h | Usb_rxlength[15:8] | - |
| 60h | EP3_Status | EP3_ Control (Set) |
| 61h | EP3_Status | EP3_ Control (Clear) |
| 70h | TotalCnt0 | TotalCnt0 |
| 71h | TotalCnt1 | TotalCnt1 |
| 72h | TotalCnt2 | TotalCnt2 |
| 73h | TotalCnt3 | TotalCnt3 |
| 74h | - | LoadTotalCnt |
| 80h | GTotalCnt0 | GTotalCnt0 |
| 81h | GTotalCnt1 | GTotalCnt1 |
| 82h | GTotalCnt2 | GTotalCnt2 |
| 83h | GTotalCnt3 | GTotalCnt3 |

## 5.2 ATA Block: (address area: 40xx)

| Address | Read Value | Write Value |
|---|---|---|
| 90h | Data[7:0] | Data[7:0] |
| 91h | Error | Features |
| 92h | SectorCount | SectorCount |
| 93h | SectorNumber | SectorNumber |
| 94h | CyclinderLow | CylinderLow |
| 95h | CylinderHigh | CylinderHigh |
| 96h | Device/Head | Device/Head |
| 97h | Status | Command |
| 98h | Reserved | Reserved |
| 99h | Reserved | reserved |
| 9Ah | Reserved | reserved |
| 9Bh | Reserved | reserved |
| 9Ch | Reserved | reserved |
| 9Dh | Reserved | reserved |
| 9Eh | AlternateStatus | DeviceControl |
| 9Fh | Reserved | reserved |
| A0h | FIFO0D[7:0] | FIFO0D[7:0] |
| A2h | sgPCtl | sgPCtl |
| A3h | FifoSt | FifoSt |
| A4h | gpioData | gpioData |
| A5h | gpioCtl | gpioCtl |
| A6h | TestCtl0 | TestCtl0 |
| A8h | DrvCtl | DrvCtl |
| ACh | upCtl | upCtl |
| AFh | MiscCtl | MiscCtl (internal testing) |
| B0h | LinkCtl | LinkCtl |
| B1h | DmaCtl | DmaCtl |
| B4h | sgDCtl | sgDCtl_Set |
| B5h | sgDCtl | sgDCtl_Clr |
| B6h | AtaCtl | AtaCtl |
| B7h | AtaStatus | AtaStatus |
| C0h,C2h-CEh | RData[7:0] | WData[7:0] |
| C1h,C3h-CFh | RData[15:8] | WData[15:8] |
| D6h | ATA_Status_Hi | ATA_Status_Hi |
| D7h | ATA_Status_Lo | ATA_Status_Lo |
| D8h | UsbINT_En | UsbINT_En |
| D9h | UsbINT_Status | UsbINT_Clr |
| E0h | I2C_Addr[7:0] | I2C_Addr[7:0] |
| E1h | I2C_Data | I2C_Data |
| E2h | I2C_Ctrl | I2C_Ctrl |
| E3h | I2C_Comm | I2C_Comm |
| E4h | I2C_Status | - |
| E8h | OTB_Counter[7:0] | OTB_Counter[7:0] |
| E9h | OTB_Ctrl[7:0] | OTB_Ctrl[7:0] |
| EAh | OTB_INT_En | OTB_INT_En |

**Note**: 1. Every Read operation from any of 9Xh registers needs to be followed by another Read operation on C0h.
2. Register C1h is used for accessing the high byte of 16-bit PIO data.

### 5.3 CMD/DATA Block: (address area: 40xx)

| Address | Read Value | Write Value |
|---|---|---|
| 100h-13Fh (64 bytes) | CmdRx0Buffer | Can not be written by CPU |
| 140h-17Fh (64 bytes) | CmdRx1Buffer | Can not be written by CPU |
| 1C0h-1EFh (64 bytes) | CmdTx1Buffer | CmdTx1Buffer |
| 240h-26Fh (64 bytes) | CmdTx3Buffer | CmdTx3Buffer |
| 280h-2AFh (64 bytes) | CmdTx4Buffer | CmdTx4Buffer |

### 5.4 DMA Block: (address area: 40xx)

| Address | Read Value | Write Value |
|---|---|---|
| 500h-53Fh | sgList[3:0] | sgList[3:0] |

### 5.5 Data Space Mapping

| Mapping Address | Type | Access Type | Mapping Block |
|---|---|---|---|
| 0000h-07FFh | Data | Read/Write | Internal SRAM (2KB) |
| 4000h-47FFh | Data | Read/Write | Internal Register/Buffers |
| 5000h-57FFh | Data | Read/Write | SgBuffer (2KB) |
| 6000h-60FFh | Data | Read/Write | USB registers |

## 6. Register Descriptions:

*The following are USB registers, based on 0x6000*

### 6.1.1 Device Status (Dev_Status[7:0], 0x20)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RSVD | r | 7 | 1'b0 | Reserved |
| Test_mode | rsu | 6 | 1'b0 | Set when SET_FEATURE (TEST_MODE). Exit by cycle VBUS. |
| Attach | ru | 5 | 1'b1 | Hardware reset default state. Clear if detect VBUS valid. Then set Power bit |
| Powered | ru | 4 | 1'b0 | Set if VBUS=1 & previous state is Attach. Or, power interruption. |
| Suspend | ru | 3 | 1'b0 | After bus IDLE for sometime, hardware set this bit. When RESUME detected, hardware reset this bit and return to previous state |
| Default | ru | 2 | 1'b0 | After bus reset, hardware set this bit. |
| Addressed | rscu | 1 | 1'b0 | Set_Address or Set_Configuration(0) |
| Configured | rscu | 0 | 1'b0 | Set_configuration |

### 6.1.2 Function Address (Funct_Adr[7:0], 0x21)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RSVD | ru | 7 | 1'b0 | Reserved |
| Adr | ru | 6:0 | 7'b0 | Set_Address |

### 6.1.3 Test Mode (Test_mode[7:0], 0x22)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RSVD | ru | 7:4 | 4'b0 | Reserved |
| Test_mode | rwu | 3:0 | 4'b0 | Test Mode Selectors(Table 9-7, USB2.0 spec)<br>4'h1: Test_J<br>4'h2: Test_K<br>4'h4: Test_SE0_NAK<br>4'h8: Test_Packet<br>others: RSVD |

### 6.1.4 End Point TX Data Length Low Bytes (Ep_TxLength[7:0], 0x25)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Ep_TxLength | rwu | 7:0 | 8'b0 | For EP1 (Bulk_IN): For ATA-Command-no-DMA-involved, this field indicates how many bytes sent back to host. Maximum 512-bytes |

### 6.1.5 End Point TX Data Length High Bytes (Ep_TxLength[15:8], 0x26)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RSVD | r | 7:2 | 6'b0 | Reserved |
| Ep_TxLength | rwu | 1:0 | 2'b0 | High bytes |

### 6.1.6 End Point 0 Status/Control (EP0_Status[7:0], 0x30: Set, 0x31: Clear)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Suspend_gnt | rsc | 7 | 1'b0 | Suspend-request granted |
| Usb_busRst | rcu | 6 | 1'b0 | Set by hardware after an usb bus reset detected. Clear by firmware. |
| Bulk_only_Rst | rcu | 5 | 1'b0 | Set by hardware, read and cleared by firmware after firmware responds bulk-only-reset command done. |
| RSVD | ru | 4:3 | 2'b0 | Reserved |
| EP0_speed | ru | 2 | 1'b0 | 1—HS, 0--FS |
| Remote_wakeup | rscu | 1 | 1'b0 | Set/Clr by firmware. Remote wakeup request. |
| Halt | rscu | 0 | 1'b0 | 1-EP0 halt. Function STALL. Device reset is require to clear this bit |

### 6.1.7 End Point 0 Status/Control2 (EP1_Status2[7:0], 0x32: Set, 0x33: Clear, Bulk-IN)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| FW_RDY | rsc | 7 | 1'b0 | 0: Default value as no firmware installed. Hardware response all control packet for firmware download in most case.<br>1: Firmware controls some setup packet response. |
| RSVD | r | 6:4 | 3'b0 | Reserved |
| EP0_StatRun | rsu | 3 | 1'b0 | Set by firmware if device ready to go to control status stage. |
| EP0_OUT | rcu | 2 | 1'b0 | Set by hardware if a control command-data is received. Clear by firmware after processing. |
| EP0_Run | rsu | 1 | 1'b0 | Set by firmware. When firmware set this bit, the data will be transferred from data buffer to USB. How many bytes transferred is based on the data transfer length in the Ep_TxLength( 0x25, 0x26) |
| EP0_Setup | rcu | 0 | 1'b0 | Set by hardware if a control command is received. Clear by firmware after processing. |

### 6.1.8 End Point TX Data Length Low Bytes (Ep0TxLength[7:0], 0x34)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RSVD | r | 7 | 1'b0 | Reserved |
| Ep0TxLength | rwu | 6:0 | 7'b0 | For EP0 (Control): This field is filled by firmware. When firmware taking control setup packet response, firmware write this field to inform hardware the data length to be send back to host. Maximum 64-bytes. |

### 6.1.9 Setup Packet (Hdr0—Hdr7[7:0], 0x38—0x3F)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Hdr | ru | 7:0 | 8'bx | 8 bytes setup packet. |

### 6.1.10 End Point 1 Status/Control (EP1_Status[7:0], 0x40: Set, 0x41: Clear, Bulk-IN)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| GTotalCntEq0 | r | 7 | 0 | 1-> Ata Global TotalCnt equ 0<br>0-> else |
| TotalCntEq0 | r | 7 | 0 | 1-> Ata TotalCnt equ 0<br>0-> else |
| Short_IN | r | 5 | 1'b0 | 1-> the last sent packet is a short packet. For bulk-in packet only |
| RSVD | r | 4 | 1'b0 | Reserved |
| CSW_Run | rscu | 3 | 1'b0 | Set by firmware when firmware ready to send CSW. Clear by hardware after CSW is sent successfully. |
| RSVD | r | 2 | 1'b0 | Reserved |

| | | | | |
|---|---|---|---|---|
| EP1_Run | rscu | 1 | 1'b0 | Set by firmware. When firmware set this bit, the data will be transferred from data buffer to USB. How many bytes transferred is based on the data transfer length in the Ep_TxLength( 0x25, 0x26) |
| Halt | rscu | 0 | 1'b0 | 1-EP1 halt. |

### 6.1.11 End Point 2 Status/Control (EP2_Status[7:0], 0x50: Set, 0x51 Clear, Bulk-OUT)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| FS_En | rw | 7 | 1'b0 | 1-> force device to Full Speed mode only |
| RSVD | r | 6 | 1'b0 | Reserved |
| Short_OUT | r | 5 | 1'b0 | 1-> the last received packet is a short packet. For bulk-out packet only |
| RX_DONE | rsc | 4 | 2'b0 | 1->brdge received all data from host. Ready for CSW transmit. Auto-clear after Rxed CBW |
| RX_2K | rsc | 3 | 1'b0 | 1-> bridge Rxed 2K bytes. Auto-clear after Rxed CBW |
| EP2_Rx | rcu | 2 | 1'b0 | Set by hardware after the bulk out packet received. The number of total data length received will be shown in Usb_rxLength register. This bit is used by firmware to monitor the data transfer between USB and internal data buffer. This bit is cleared by firmware or automatically cleared by hardware after the next CBW received or sg0Run bit set by firmware. |
| EP2_CBW | rcu | 1 | 1'b0 | Set by hardware if a valid CBW received. Clear after processing by firmware. |
| Halt | rscu | 0 | 1'b0 | 1-EP2 halt. |

### 6.1.12 Usb_rxLength(usb_rxLength[7:0], 0x52, Bulk-OUT)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| rxLength | rwu | 7:0 | 8'b0 | The low byte of data length received. This register is used to show how many date received from USB to internal data buffer. |

### 6.1.13 Usb_rxLength(usb_rxLength[15:8], 0x53, Bulk-OUT)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| rxLength | rwu | 7:0 | 8'b0 | The high byte of data length received. This register is used by firmware to show how many data received from USB to internal data buffer. |

### 6.1.14 End Point 3 Status/Control (EP3_Status[7:0], 0x60: Set, 0x61: Clear, INTR-IN)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RSVD | r | 7:3 | 5'b0 | Reserved |
| EP3_run | rsu | 2 | 1'b0 | 1—packet ready. Cleared by hardware after Tx completed |
| RSVD | r | 1 | 1'b0 | Reserved |

| | | | | |
|---|---|---|---|---|
| Halt | rscu | 0 | 1'b0 | 1-EP3 halt. |

### 6.1.15 End Point TX Data Length Low Bytes (Ep3TxLength[7:0], 0x62)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RSVD | r | 7 | 1'b0 | Reserved |
| Ep3TxLength | rwu | 6:0 | 7'b0 | For EP3 (INT_IN): This field is filled by firmware. Firmware writes this field to inform hardware the data length to be sent back to host. Maximum 64-bytes. |

### 6.1.16 Total Count0 (TotalCnt[7:0], 0x70 TotalCnt0)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| TotalCnt0 | rwu | 7-0 | 8'b0 | TotalCnt[7:0] |

### 6.1.17 Total Count1 (TotalCnt[15:8], 0x71 TotalCnt1)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| TotalCnt1 | rwu | 7-0 | 8'b0 | TotalCnt[15:8] |

### 6.1.18 Total Count2 (TotalCnt[23:16], 0x72 TotalCnt2)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| TotalCnt2 | rwu | 7-0 | 8'b0 | TotalCnt[23:16] |

### *6.1.19* Total Count3 (TotalCnt[31:24], 0x73 TotalCnt3*)*

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| TotalCnt3 | rwu | 7-0 | 8'b0 | TotalCnt[31:24] |

### *6.1.20* Load Total Count (Load TotalCnt, 0x74*)*

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7-1 | 7'b0 | Reserved |
| LoadTotalCnt | w | 0 | 1'b0 | Write an 1 to this bit will re-load the value from register 0x73-0x70's TotalCnt[31:0] to internal counter. |

### 6.1.21 Global Total Count0 (GTotalCnt[7:0], 0x80 GTotalCnt0)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| GTotalCnt0 | rwu | 7-0 | 8'b0 | GTotalCnt[7:0] |

### 6.1.22 Global Total Count1 (GTotalCnt[15:8], 0x81 GTotalCnt1)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| GTotalCnt1 | rwu | 7-0 | 8'b0 | GTotalCnt[15:8] |

### 6.1.23 Global Total Count2 (GTotalCnt[23:16], 0x82 GTotalCnt2)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| GTotalCnt2 | rwu | 7-0 | 8'b0 | GTotalCnt[23:16] |

### *6.1.24* Global Total Count3 (GTotalCnt[31:24], 0x83 GTotalCnt3*)*

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| GTotalCnt3 | rwu | 7-0 | 8'b0 | GTotalCnt[31:24] |

## *The following are general registers, and are in address area: 40XX*

### 6.2.1 FIFO 0 Data Register (FIFO0D[7:0], 0x0A0)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Fifo0Data | rw | 7:0 | 8'h0 | DMA FIFO 0 Data register. Software can access DMA FIFO 0 through this register. |

### 6.2.2 sgPioCmd Control Register (sgPCtl, 0x0A2)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7-1 | 6'b0 | Reserved. |
| Pio0Run | rw | 0 | 1'b0 | After set. Before set this bit, the firmware needs to disable AtaDMAEn bit on DMA Control register, write the package header into the segment of sgCMD buffer, set RUN bit on sgCMD Control register, and fill data into FIFO data register. |

### 6.2.3 FIFO Status Register (FifoST, 0x0A3)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7-1 | 6'b0 | Reserved. |
| Fifo0Rst | rw | 0 | 1'b0 | DMA FIFO 0 Reset. This bit is used to reset DMA FIFO 0. This bit is self-cleared by hardware after set. |

### 6.2.4 GPIO Data Register (gpioDATA, 0x0A4)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| VBUS | r | 7 | 1'b0 | Read: USB's VBUS status |
| AtaRST0# | w | 6 | 1'b1 | ATA channel 0 reset signal. When clear, it will reset the ATA device. |
| GPIOD[5:0] | rw | 5-0 | 6'h0 | Write: GPIOOut[5:0], Read: GPIOIn[5:0] |

### 6.2.5 GPIO Control Register (gpioCtl, 0x0A5)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| GPIOEn | rw | 7 | 1'b0 | GPIO mode enable. |
| Reserved | r | 6 | 1'b0 | Reserved. |
| GPIOCtl[5:0] | rw | 5-0 | 6'h0 | General Purpose I/O Control 5-0. When set, the GPIO data is output. |

### 6.2.6 Test Control 0 Register (TestCtl0, 0x0A6)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| RstAtareq_ | rw | 7 | 1'b0 | Ata request reset. When set, it will reset the current Ata request. |
| Reserved | rw | 6:4 | 3'b0 | Reserved. |
| RevID | r | 3:0 | 4'h0 | Revision ID. These 4 bits are read only. |

### 6.2.7 ATA I/O Cell Driving Control Register (DrvCtl, 0x0A8)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| CFEn | rw | 7 | 1'b0 | Compact Flash Mode Enable. When set, the ATA engine will run in pseudo DMA mode enabling fast compact flash access. |
| CFReq | rw | 6 | 1'b0 | Compact Flash Request. When set, it indicates to the ATA engine that compact flash has requested data transfer. |
| Reserved | r | 5-3 | 4'h0 | Reserved. |
| AtaTSEn | rw | 2 | 1'b0 | ATA Bus Tristate Enable. Clear this bit to 0 will put the INI1430 ATA bus in tristate mode. |
| DrvSel | rw | 1-0 | 2'h3 | ATA Output Driving:<br>00: 8 mA<br>01: 10 mA<br>10: 12 mA<br>11: 16 mA |

### 6.2.8 uP Control Register (upCtl, 0x0AC)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7-2 | 6'b0 | Reserved. |
| usbWakeupEn | rw | 1 | 1'b0 | 1-> enable external interrupt wake up host |
| Reserved | r | 0 | 1'b0 | Reserved. |

### 6.2.9 MiscCtl register (0x0AF)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | rw | 7 | 1'b0 | Reserved. |
| NewMode | rw | 6 | 1'b0 | 0: inic1530 mode<br>1: enhance mode |
| Set_usbClkEn | rw | 5 | 1'b0 | 1: usb clock free run |
| Enum | rw | 4 | 1'b1 | 0: disconnect device from usb host |
| HidEn | rw | 3 | 1'b0 | 0: endp will be IN(1)OUT(2)INTR(3)<br>1: endp will be IN(8)OUT(2)INTR(1) |
| Reserved | rw | 2 | 3'b0 | Reserved. |

| | | | | |
|---|---|---|---|---|
| OUT_ABORT_En | rw | 1 | 1'b0 | 1->wait all data sent from host done. |
| CSW_ACK_En | rw | 0 | 1'b0 | 1-> Fw don't have to do the following things after CSW sent,<br>* write totalCnt to 0<br>*set Flush |

### 6.2.10 sgCmd Definition

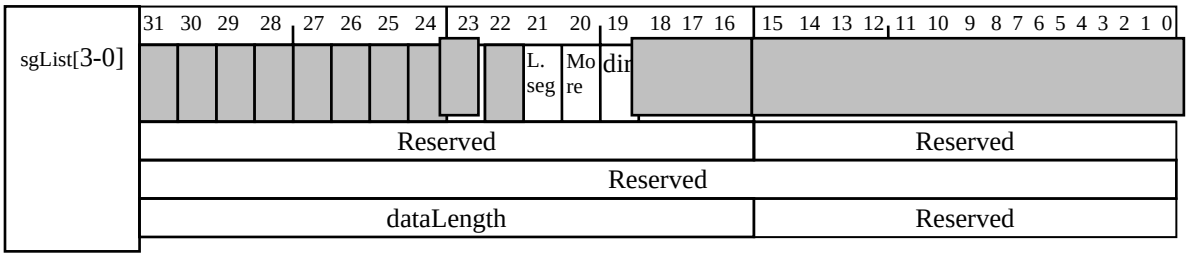



*Figure 2: sgCmd format*

| Field name | width | quadlet | Bit # | Description |
|---|---|---|---|---|
| L. seg | 1 | 1 | 21 | If the command is the last one of S/G segments, this bit should be set by firmware. |
| More | 1 | 1 | 20 | If the number of commands in SgCmd buffer is more than one, this bit should be set by firmware. |
| Dir | 1 | 1 | 19 | When 0, the DMA data are transferred from P1394 bus to ATA device.<br>When 1, the DMA data are transferred from ATA device to P1394 bus. |

### 6.2.11 Link Control Register (linkCtl, 0x0B0)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7-2 | 6'b0 | Reserved. |
| HW_RESET | w | 1 | 1'b0 | 1->Reset whole chip. Function same as RC_RESET pin. Auto clear by hardware. |
| softReset | rwu | 0 | 1'b0 | When set to 1, all Host Controller state is reset, all FIFO's are flushed, and Host Controller registers is reset.<br>The read value of this bit is 1 while a soft reset or hard reset is in progress.<br>The read value of this bit is 1 when neither soft reset nor hard reset is in progress. Software can use the value of his bit to determine when a reset has completed and the Host Controller is safe to operate. |

### 6.2.12 DMA Control register (DmaCtl, 0x0B1)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7-4 | 1'h0 | Reserved. |
| DIOW# | rw | 3 | 1'b1 | When DMA FIFO is underrun, this bit is used by firmware to toggle DIOW# signal. |
| DIOR# | rw | 2 | 1'b1 | When DMA FIFO is underrun, this bit is used by firmware to toggle DIOR# signal. |
| DMACK# | rw | 1 | 1'b1 | When DMA FIFO is underrun, this bit is used by firmware to toggle DMACK# signal. |
| Flush/Abort | rw | 0 | 1'b0 | When DMA FIFO is overrun, this bit is used by firmware to flush data |

| | | | | |
|---|---|---|---|---|
| | | | | out for outgoing data or abort the DMA operation for incoming data. This bit is self-cleared by hardware. |

### 6.2.13 Channel Clear Register (cmdCtl_Clr, 0x0B3)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7 | 7'b0 | Reserved |
| CmdTx4Run (for HID_out) | rwu | 6 | 1'b0 | When the Clear register is set by software, the corresponding channel is cleared. |
| CmdTx3Run (for CSW_out) | rwu | 5 | 1'b0 | When the Clear register is set by software, the corresponding channel is cleared. |
| Reserved | r | 4 | 1'b0 | Reserved |
| CmdTx1Run (for Control_out) | rwu | 3 | 1'b0 | When the Clear register is set by software, the corresponding channel is cleared. |
| Reserved | r | 2 | 1'b0 | Reserved |
| CmdTx1Run (for CBW) | rwu | 1 | 1'b0 | When the Clear register is set by software, the corresponding channel is cleared. |
| CmdTx1Run (for Setup-Packet) | rwu | 0 | 1'b0 | When the Clear register is set by software, the corresponding channel is cleared. |

### 6.2.14 sgDma Control Register (sgCtl_Set, 0x0B4) (sgCtl_Clr, 0x0B5)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7:1 | 7'b0 | Reserved. |
| Sg0Run | rwu | 0 | 1'b0 | When Set register is set by software, the corresponding channel is ready to be transmitted. The hardware clears these bits when the transfer is completed. Software can set bit[3:0] on Clear register to clear the corresponding bit. When Clear register is written by software, the DMA channels will be reset to idle. (USB bulk transfer will only use sg0Run) |

### 6.2.15 ATA Control register (AtaCtl, 0x0B6)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| AtaDMAEn | rw | 7 | 1'b1 | When 1, ataDMA is enabled. |
| pioReq/pioGnt | rw | 6 | 1'b0 | Write 1 for PIO request. PIO grant status when read. |
| dmaMode | rw | 5 | 1'b0 | 0-2: DMA mode 0 - 2 |
| | rw | 4 | 1'b0 | 4-7: UDMA mode 2, 3, 4 and 5 (up to uDMA100) |
| | rw | 3 | 1'b0 | |
| pioMode | rw | 2 | 1'b0 | 0-4: PIO mode 0-4 |
| | rw | 1 | 1'b0 | |
| | rwu | 0 | 1'b0 | |

### 6.2.16 ATA Control/Status register (ataStatus, 0x0B7)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | r | 7 | 1'b0 | Reserved |
| AtaCh0En | rw | 6 | 1'b0 | Ata Channel Enable. |
| PioXEn | rw | 5 | 1'b0 | PIO transfer engine enable. When set, the PIO engine will transfer data to/from the ATA bus using PIO transfer mechanism. |

| | | | | |
|---|---|---|---|---|
| Reserved | r | 4:3 | 2'b0 | Reserved. |
| AtaIORDY0 | ru | 2 | 1'bx | Ata channel IORDY line. |
| Reserved | r | 1 | 1'b0 | Reserved |
| AtaINTRQ0 | ru | 0 | 1'bx | Ata channel INTRQ line. |

### 6.2.17 ATA_Status_Low Register (0x0D6)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| ATA_Status[7:0] | rw | 7-0 | 8'h0 | CPU writes the ATA status to this register |

### 6.2.18 ATA_Status_High Register (0x0D7)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| ATA_Status[15:8] | rw | 7-0 | 8'h0 | CPU writes the ATA status to this register |

### 6.2.19 usb_INT_Enable Register (0x0D8)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Usb_busRst_INT_En | rw | 7 | 1'b0 | 1: Enable Usb_busRst to trigger sysINT. |
| Usb_bulkOnlyRst_IN | rw | 6 | 1'b0 | 1: Enable Usb_bulkOnlyRst to trigger sysINT. |
| Usb_Ep0Req_INT_E | rw | 5 | 1'b0 | 1: Enable Usb_ Ep0Req to trigger sysINT. |
| Usb_CBW_INT_En | rw | 4 | 1'b0 | 1: Enable Usb_CBW to trigger sysINT. |
| Usb_wakeup_INT_E | rw | 3 | 1'b0 | 1: Enable Usb_wakeup to trigger sysINT. |
| Usb_suspendINT_En | rw | 2 | 1'b0 | 1: Enable Usb_suspend to trigger sysINT. |
| VBUS_P__INT_En | rw | 1 | 1'b0 | 1: Enable positive of VBUS to trigger sysINT. |
| VBUS_N__INT_En | rw | 0 | 1'b0 | 1: Enable negative of VBUS to trigger sysINT. |

### 6.2.20 usb_INT_Status/Clear Register (0x0D9)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Usb_busRst | rw | 7 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit |
| Usb_bulkOnlyRst | rw | 6 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit. |
| Usb_Ep0Req | rw | 5 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit |
| Usb_CBW | rw | 4 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit |
| Usb_wakeup | rw | 3 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit |
| Usb_suspend | rw | 2 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit |
| VBUS_P__INT | rw | 1 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit |
| VBUS_N__INT | rw | 0 | 1'b0 | Read: 1: indicates this interrupt event occurred<br>Write 1: will clear this status bit |

### 6.2.21 I2C_Addr Register (0x0E0)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| I2C_Addr | rw | 7-0 | 8'h0 | CPU writes the to-be executed NVRAM address to this register. |

### 6.2.22 I2C_Data Register (0x0E1) (I2C Data port)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| I2C_Data | rw | 7-0 | 8'h0 | This is the data port for CPU to access NVRAM<br><br>A: To Write to NVRAM:<br>CPU Writes a 8-bit data to this port, Hardware will send this data to NVRAM<br><br>B: To Read from NVRAM:<br>CPU reads this port to get data from NVRAM |

### 6.2.23 I2C_Ctrl Register (0x0E2) (I2C Control Code)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| Reserved | w | 7 | 1'b0 | reserved |
| Control_Code | rw | 6:3 | 4'b0 | Control Code |
| Block_Select | rw | 2:0 | 3'b0 | I2C device block select bits |

### 6.2.24 I2C_COMM Register (0x0E3)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| I2C_TX_START | rw | 7 | 1'b0 | 1-> Hardware start to Read/Write NVRAM. Clear by hardware when finished. |
| reserved | rw | 6:1 | 6'b0 | reserved |
| Rd_nWr | wr | 0 | 1'b0 | 0-> write to NVRAM<br>1-> read data from NVRAM |

### 6.2.25 I2C_Status Register (0x0E4)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| reserved | rw | 7-3 | 5'b0 | reserved |
| Data_Ack | r | 0 | 1'b0 | 0-> NVRAM ACK with Data write phase<br>1->NVRAM NACK with Data write phase |
| Addr_Ack | r | 0 | 1'b0 | 0-> NVRAM ACK with Address write phase<br>1->NVRAM NACK with Address write phase |
| Ctrl_Ack | r | 0 | 1'b0 | 0-> NVRAM ACK with Contrl Code write phase<br>1->NVRAM NACK with Contrl Code write phase |

### 6.2.26 OTB_Counter Register (0x0E8)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| OTB_Counter | rw | 7:0 | 8'b0 | How many push-release done. Clear after FW read. |

### 6.2.27 OTB_Ctrl Register (0x0E9)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| P1_4_OTB_en | rw | 7:6 | 2'b0 | 00->I2C mode<br>01->OTB mode<br>1x->I2C_SCL as P3_3, I2C_SDA as P1_4 |
| reserved | rw | 6:3 | 5'b0 | reserved |
| Debouncing time | rw | 1:0 | 2'b0 | 2'b00-> 36ms<br>2'b01->72ms<br>2'b10->108ms<br>2'b11->144ms |

### 6.2.28 OTB_INT_Enable Register (0x0EA)

| Field name | rscu | bit # | reset | Description |
|---|---|---|---|---|
| OTB_Counter_INTE | rw | 7 | 1'b0 | 1: Enable OTB_Counter<>0 trigger sysINT. |
| OTB_pos_INTE | rw | 6 | 1'b0 | 1: Enable button RELEASE trigger sysINT |
| OTB_neg_INTE | rw | 5 | 1'b0 | 1: Enable button PUSH trigger sysINT |
| reserved | rw | 4:0 | 5'b0 | reserved |

## 7. Electrical Information:

### *7.1* Absolute Maximum Ratings

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| Vcc | Power Supply | -0.3 | 3.6 | V |
| Vin | Input Voltage | -0.3 | Vcc+0.3 | V |
| Vout | Output Voltage | -0.3 | Vcc+0.3 | V |
| Tstg | Storage Temperature | -55 | 150 | °C |

### *7.2* Recommended Operating Conditions

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Vcc | Power Supply | 3.0 | 3.3 | 3.6 | V |
| Vin | Input Voltage | 0 | - | Vcc | V |
| Tj | Commercial Junction Operating Temperature | 0 | 25 | 115 | °C |
| | Industrial Junction Operation Temperature | -40 | 25 | 125 | °C |

## *7.3* General DC Characteristics

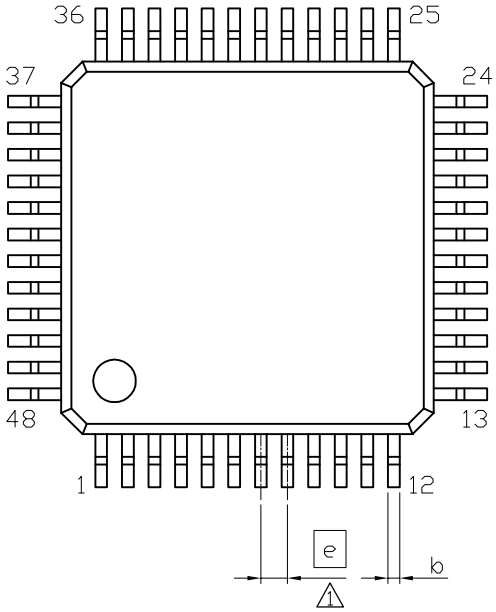
| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Iil | Input Leakage Current | -1 | | 1 | μA |
| Ioz | Tristate Leakage Current | -1 | | 1 | μA |
| Cin | Input Capacitance | | 2.8 | | pF |
| Cout | Output Capacitance | 2.7 | | 4.9 | pF |
| Cbid | Bi-directional Buffer Capacitance | 2.7 | | 4.9 | pF |

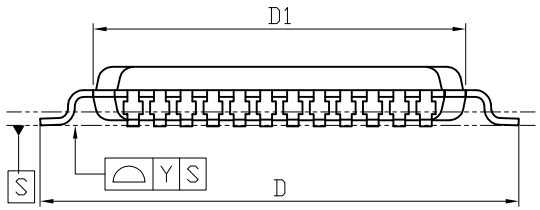
## *7.4* DC Electrical Characteristics for 3.3V Operation

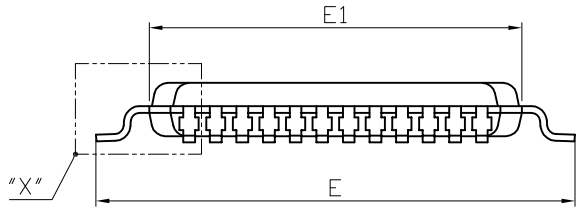
(Under Vcc=3.0-3.6V, Tj=0-115C)

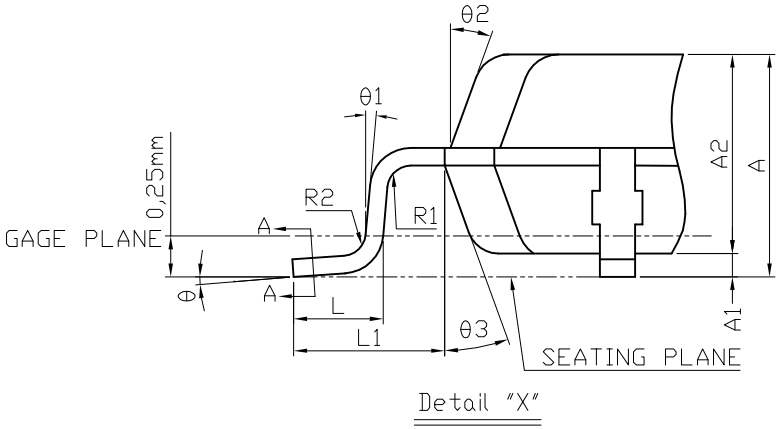
| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| Vil | Input Low Voltage | CMOS | | | 0.3*Vcc | V |
| | | CMOS Schmitt Trigger | | 1.20 | 0.3*Vcc | V |
| Vih | Input High Voltage | CMOS | 0.7*Vcc | | | V |
| | | CMOS Schmitt Trigger | | 2.10 | | V |
| Vol | Output Low Voltage | Ioh-2-24mA | | | 0.4 | V |
| Voh | Output High Voltage | Ioh=2-24mA | 2.4 | | | V |
| Ri | Input Pullup/pulldown Resistance | Vil=0/Vih=Vcc | | 75 | | kΩ |
| Icc | Operating Supply Current | Vcc=3.3V | | 40 | 60 | mA |

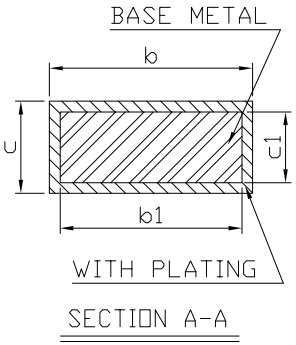
# 8. Packaging Specification




| SYMBOL | DIMENSION (MM) | | | DIMENSION (MIL) | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | | | 1.60 | | | 63 |
| A1 | 0.05 | | 0.15 | 2 | | 6 |
| A2 | 1.35 | 1.40 | 1.45 | 53 | 55 | 57 |
| b | 0.17 | 0.22 | 0.27 | 7 | 9 | 11 |
| b1 | 0.17 | 0.20 | 0.23 | 7 | 8 | 12 |
| c | 0.09 | | 0.20 | 4 | | 8 |
| c1 | 0.09 | | 0.16 | 4 | | 6 |
| D | 9.00 BSC | | | 354 BSC | | |
| D1 | 7.00 BSC | | | 276 BSC | | |
| E | 9.00 BSC | | | 354 BSC | | |
| E1 | 7.00 BSC | | | 276 BSC | | |
| ⚠1 e | 0.35 | 0.50 | 0.65 | 14 | 20 | 26 |
| L | 0.45 | 0.60 | 0.75 | 18 | 24 | 30 |
| L1 | 1.00 REF | | | 39 REF | | |
| R1 | 0.08 | | | 3 | | |
| R2 | 0.08 | | 0.20 | 3 | | 8 |
| Y | | | 0,075 | | | 3 |
| θ | 0° | 3.5° | 7° | 0° | 3.5° | 7° |
| θ1 | 0° | | | 0° | | |
| θ2 | 11° | 12° | 13° | 11° | 12° | 13° |
| θ3 | 11° | 12° | 13° | 11° | 12° | 13° |

NOTE:
1.REFER TO JEDEC MS-026/BBC
2.DIMENSION D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
ALLOWABLE PROTRUSION IS 0.25mm PER SIDE D1 AND E1 ARE
MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.
3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE
DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED
THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.
4.ALL DIMENSIONS IN MILLIMETERS.
⚠1 5.Remark: Modify [e] .

| | | | | PKG. CODE | DRAWING NUMBER | REV. |
|---|---|---|---|---|---|---|
| | | | | | | 4 |
| SIZE | A3 | BY | DATE | TITLE | LQFP48 (7x7mm) | |
| DRAWN | | | 2005.09.06 | | PACKAGE OUTLINE | |
| DESIGNED | | | | | Footprint 2.0mm | |
| CHECKED | | | | SCALE | 10 : 1 | PROJ. |
| APPROVED | | | | SHEET | 1 OF 1 | |