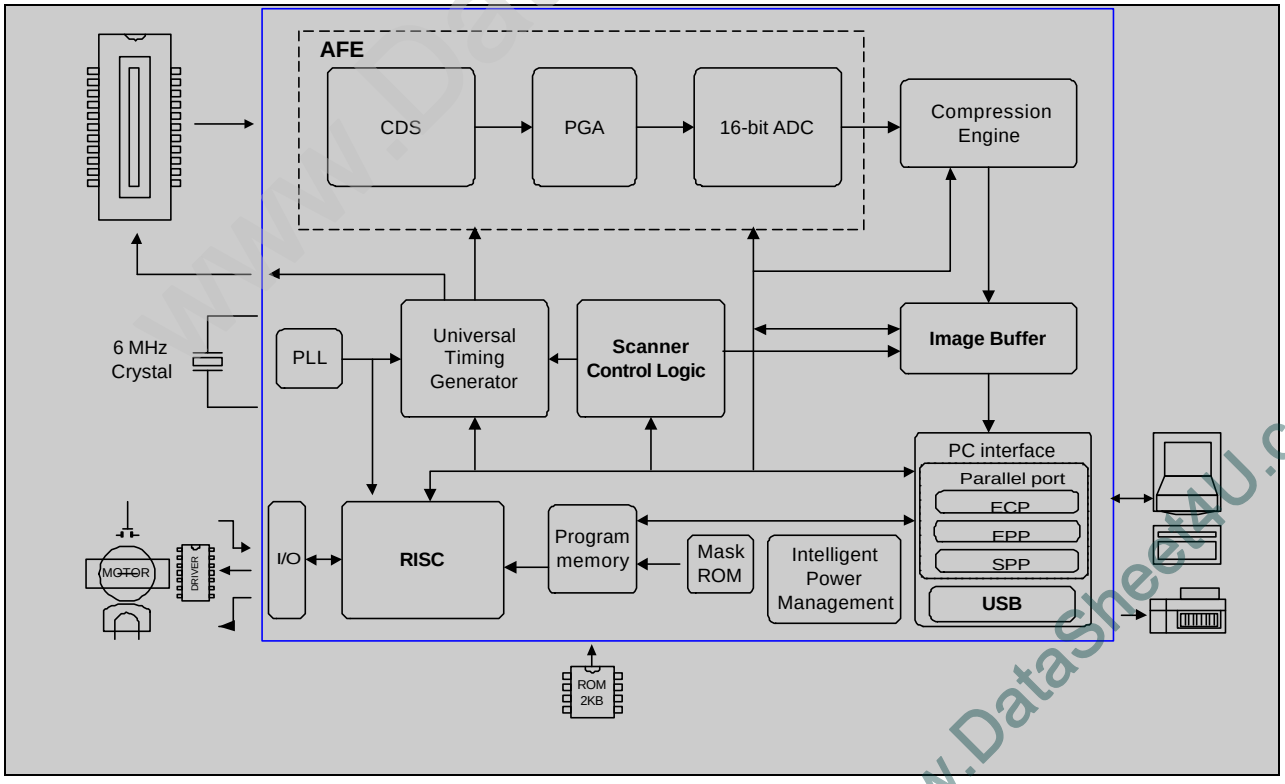
## I. General Descriptions-

The GT-6816 is an enhanced version of GT-6801, which provides highly integrated **System-On-Chip (SOC)** solution for high-performance color scanner. The GT-6816 is enhanced not only in the AFE (Analog Front End) from 12-bit to 16-bit but also built-in an intelligent power management circuit to meet both operating and suspend mode for USB bus-powered Scanner. The GT-6816 is also pin to pin backward compatible with the GT-6801, providing system designer easy way to upgrade the current applications without changing the hardware design.

## II. Features-

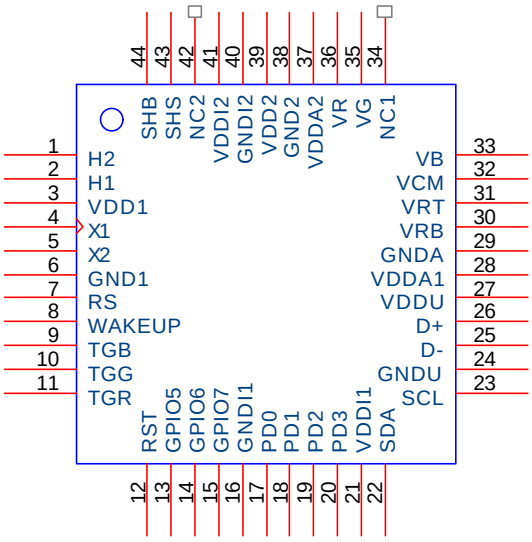
- Single-chip integration for high-performance color scanner application
- On-chip Analog Front End: CDS/AGC and 16-bit ADC Maximum 6MHz
- On-chip universal TG supports various types of CCD/CIS sensors
- Embedded high-performance RISC controller
- On-chip USB transceiver
- Built-in 16KB image line buffers
- PC interface supports : USB/EPP/ECP/BPP
- No external memory component required for typical application
- Firmware programmable frame size
- Intelligent power management meets both operating and suspend mode for USB bus power
- On-chip PLL circuits
- Operating clock :48 MHz with external crystal: 6 MHz
- Operating voltage: Core: 3.3V, I/O: 5V
- Operating current: Core 80mA, AFE 50mA
- Suspend current: 50 A
- Package: 128-QFP & 44-QFP



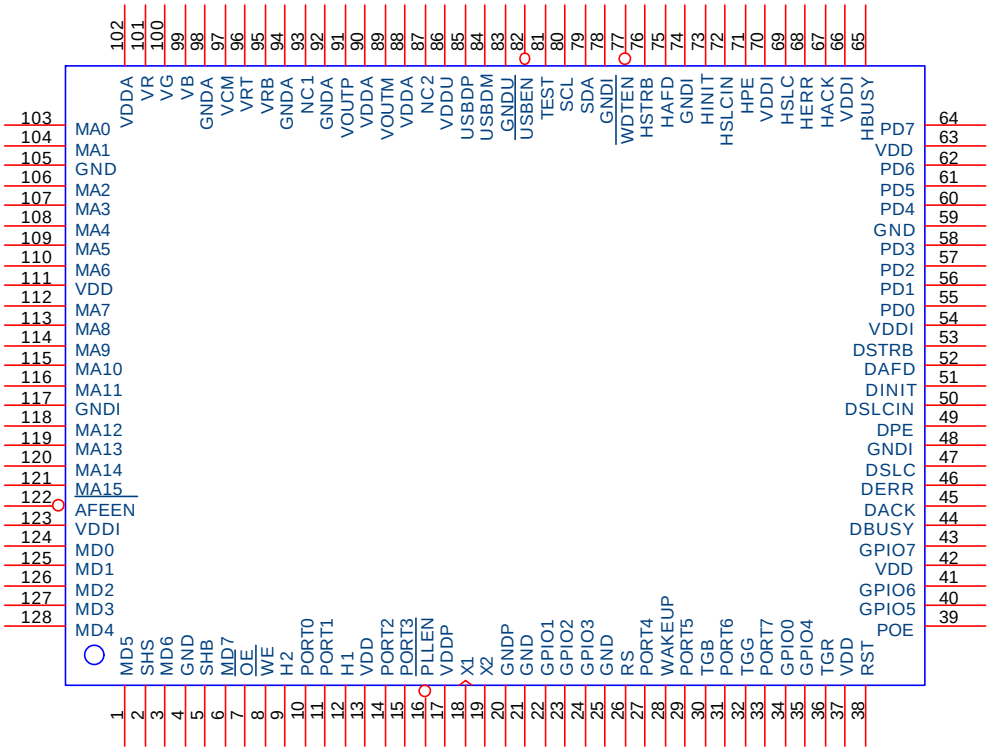
**III. Benefits-**

- ✧ Provides a total solution, fast time-to-market
- ✧ USB bus power without additional power line
- ✧ Supports a wide range of applications
- ✧ Customer differentiation via firmware
- ✧ Provides firmware update path via USB channel
- ✧ Easy to meet EMI standard
- ✧ Minimum external components
- ✧ Supports a full set of development kit
- ✧ Single chip solution, stock management becomes easier

**IV. Pin Configuration-**




Type(A)- 44-pin QFP



**V. Pin Descriptions-**

| 128 No. | 44 No. | Name | Reset State | Type | Driven | Description |
|---|---|---|---|---|---|---|
| 1 | | MD4 | Tri-state | I/O | 4mA | External memory data bus bit 4 |
| 2 | | MD5 | Tri-state | I/O | 4mA | External memory data bus bit 5 |
| 3 | 43 | SHS | Tri-state | O | 8mA | CCD sample hold signal control signal |
| 4 | | MD6 | Tri-state | I/O | 4mA | External memory data bus bit 6 |
| 5 | | GNDC1 | | P | | Core ground |
| 6 | 44 | SHB | Tri-state | O | 8mA | CCD sample hold reset control signal |
| 7 | | MD7 | Tri-state | I/O | 4mA | External memory data bus bit 7 |
| 8 | | MOE# | High | O | 4mA | External memory output enable |
| 9 | | MWE# | High | O | 4mA | External memory write enable |
| 10 | 1 | H2/CLK | Tri-state | O | 8mA | CCD shift clock/CIS clock control signal |
| 11 | | PORT1_0 | Tri-state | I/O | 4mA | uP port 1 bit 0 |
| 12 | | PORT1_1 | Tri-state | I/O | 4mA | uP port 1 bit 1 |
| 13 | 2 | H1/SP | Tri-state | O | 8mA | CCD shift clock/CIS SP control signal |
| 14 | | VDDC1 | | P | | Core power |
| 15 | | PORT1_2 | Tri-state | I/O | 4mA | uP port 1 bit 2 |
| 16 | | PORT1_3 | Tri-state | I/O | 4mA | uP port 1 bit 3 |
| 17 | | PLL_EN# | | I | | PLL enable control signal, with pull-down, not |
| 18 | 3 | VDDP | | P | | PLL power |
| 19 | 4 | X1 | | I | | Crystal input |
| 20 | 5 | X2 | | O | | Crystal output |
| 21 | 6 | GNDP | | P | | PLL ground |
| 22 | | GNDC2 | | P | | Core ground |
| 23 | | GPIO1 | Tri-state | I/O | 4mA | GPIO bit 1 |
| 24 | | GPIO2 | Tri-state | I/O | 4mA | GPIO bit 2 |
| 25 | | GPIO3 | Tri-state | I/O | 4mA | GPIO bit 3 |
| 26 | | GNDC3 | | P | | Core ground |
| 27 | 7 | RS | Tri-state | O | 8mA | CCD reset signal |
| 28 | | PORT1_4 | Tri-state | I/O | 4mA | uP port 1 bit 4 |
| 29 | 8 | WAKEUP | | I | | USB device remoter wakeup |
| 30 | | PORT1_5 | Tri-state | I/O | 4mA | uP port 1 bit 5 |
| 31 | 9 | TGB/LEDB | Tri-state | O | 8mA | CCD TG/CIS LED B channel control signal |
| 32 | | PORT1_6 | Tri-state | I/O | 4mA | uP port 1 bit 6 |
| 33 | 10 | TGG/LEDG | Tri-state | O | 8mA | CCD TG/CIS LED G channel control signal |
| 34 | | PORT71 | Tri-state | I/O | 4mA | uP port 1 bit 7 |
| 35 | | GPIO0 | Tri-state | I/O | 4mA | GPIO bit 0 |
| 36 | | GPIO4 | Tri-state | I/O | 4mA | GPIO bit 4 |
| 37 | 11 | TGR/LEDR | Tri-state | O | 8mA | CCD TG/CIS LED R channel control signal |
| 38 | | VDDC2 | | P | | Core power |
| 39 | 12 | RESET | | I | | Power on reset, high active |
| 40 | | GPIO5 | Tri-state | I/O | 4mA | GPIO bit 5 |
| 41 | 13 | GPIO6 | Tri-state | I/O | 4mA | GPIO bit 6 |
| 42 | 14 | VDDC3 | | P | | Core power |
| 43 | | GPIO7 | Tri-state | I/O | 4mA | GPIO bit 7 |
| 44 | 15 | DBUSY | | I | | Device parallel port (Busy) signal, with |

| | | | | | | |
|---|---|---|---|---|---|---|
| 45 | | DACK | | I | | Device parallel port (Ack) signal, with |
| 46 | | DERR | | I | | Device parallel port (Fault) signal |
| 47 | | DSLC | | I | | Device parallel port (select) signal, with |
| 48 | | GNDI1 | | P | | IO ground |
| 49 | 16 | GNDI2 | | P | | IO ground |
| 50 | | DPE | | I | | Device parallel port (Paper end) signal, |
| 51 | | DSLCIN | Low | O | 24mA | Device parallel port (Select in) signal |
| 52 | | DINIT | Low | O | 24mA | Device parallel port (Init) signal |
| 53 | | DAFD | Low | O | 24mA | Device parallel port (Auto feed) signal |
| 54 | | DSTRB | Low | O | 24mA | Device parallel port (Strobe) signal |
| 55 | | VDDI1 | | P | | IO power |
| 56 | 17 | PD0 | Tri-state | I/O | 24mA | Parallel port data bus bit 0, with |
| 57 | 18 | PD1 | Tri-state | I/O | 24mA | Parallel port data bus bit 1, with |
| 58 | 19 | PD2 | Tri-state | I/O | 24mA | Parallel port data bus bit 2, with |
| 59 | 20 | PD3 | Tri-state | I/O | 24mA | Parallel port data bus bit 3, with |
| 60 | | GNDC4 | | P | | Core ground |
| 61 | | PD4 | Tri-state | I/O | 24mA | Parallel port data bus bit 4, with |
| 62 | | PD5 | Tri-state | I/O | 24mA | Parallel port data bus bit 5, with |
| 63 | | PD6 | Tri-state | I/O | 24mA | Parallel port data bus bit 6, with |
| 64 | | VDDC4 | | P | | Core power, |
| 65 | | PD7 | Tri-state | I/O | 24mA | Parallel port data bus bit 7, with |
| 66 | | HBUSY | Low | O | 24mA | Host parallel port (Busy) signal |
| 67 | 21 | VDDI2 | | P | | I/O Power |
| 68 | | VDDI3 | | P | | I/O Power |
| 69 | | HACK | Low | O | 24mA | Host parallel port (Ack) signal |
| 70 | | HERR | Low | O | 24mA | Host parallel port (Fault) signal, with |
| 71 | | HSLC | Low | O | 24mA | Host parallel port (Select) signal |
| 72 | | VDDI4 | | P | | I/O power |
| 73 | | HPE | Low | O | 24mA | Host parallel port (Paper end) signal |
| 74 | | HSLCIN | | I | | Host parallel port (Select in) signal, with |
| 75 | | HINIT | | I | | Host parallel port (Init) signal, with |
| 76 | | GNDI3 | | P | | I/O ground |
| 77 | | HAFD | | I | | Host parallel port (Auto feed) signal, |
| 78 | | HSTRB | | I | | Host parallel port (Strobe) signal, with |
| 79 | | GNDI4 | | P | | I/O ground |
| 80 | 22 | SDA | Tri-state | I/O | 4mA | Serial EEPROM data line, PCB need |
| 81 | 23 | SCL | Tri-state | I/O | 4mA | Serial EEPROM clock line, PCB need |
| 82 | | TEST | | I | | For test only, with pull-down, not |
| 83 | | GNDU | | P | | USB transceiver ground |
| 84 | 24 | D- | | I/O | | USB transceiver D- |
| 85 | 25 | D+ | | I/O | | USB transceiver D+ |
| 86 | 26 | VDDU | | P | | USB transceiver power |

VI.Registers map-

| | | | |
|---|---|---|---|
| ***PPCR*** | R/W | ‘ hff | Parallel port control register |
| ***PPHAR*** | R/W | ‘ hfe | Parallel port high address register |
| ***PPLAR*** | R/W | ‘ hfd | Parallel port low address register |
| ***PPDR*** | R/W | ‘ hfc | Parallel port data register |
| ***IVDHPR*** | R/W | ‘ hfb | Image valid data high pointer register |
| ***IVDLPR*** | R/W | ‘ hfa | Image valid data low pointer register |
| ***IVIR*** | R/W | ‘ hf9 | Image valid indication register |
| ***HADFR*** | R/W | ‘ he4 | Host access device flag register |
| ***TGCR*** | R/W | ‘ hff00 | TG Control Register |
| ***MOD6DR*** | R/W | ‘ hff01 | Modulo 6 bit for dot clock |
| ***MOD6PR*** | R/W | ‘ hff02 | Modulo 6 bit for pixel clock |
| ***MOD16HR*** | R/W | ‘ hff03 | Modulo 16 bit high byte |
| ***MOD16LR*** | R/W | ‘ hff04 | Modulo 16 bit low byte |
| ***ADCLKRR*** | R/W | ‘ hff05 | ADCLK rise register |
| ***ADCLKFR*** | R/W | ‘ hff06 | ADCLK fall register |
| ***SHR6R*** | R/W | ‘ hff07 | SH(TG) setup register(reference to 6bit counter) |
| ***SHF6R*** | R/W | ‘ hff08 | SH(TG) setup register(reference to 6bit counter) |
| ***SHR8R*** | R/W | ‘ hff09 | SH(TG) setup register(reference to 16bit counter) |
| ***SHF8R*** | R/W | ‘ hff0a | SH(TG) setup register(reference to 16bit counter) |
| ***SHCR*** | R/W | ‘ hff0b | SH(TG) configuration register |
| ***LEDRHR*** | R/W | ‘ hff0c | LED rise high register(reference to 16bit counter) |
| ***LEDRLR*** | R/W | ‘ hff0d | LED rise low register(reference to 16bit counter) |
| ***LEDFHR*** | R/W | ‘ hff0e | LED fall high register(reference to 16bit counter) |
| ***LEDFLR*** | R/W | ‘ hff0f | LED fall low register(reference to 16bit counter) |
| ***LEDCR*** | R/W | ‘ hff10 | LED configuration register |
| ***TGMR*** | R/W | ‘ hff11 | TG Mask register |
| ***RS1RR*** | R/W | ‘ hff12 | RS1 rise register |
| ***RS1FR*** | R/W | ‘ hff13 | RS1 fall register |
| ***RS2RR*** | R/W | ‘ hff14 | RS2 rise register |
| ***RS2FR*** | R/W | ‘ hff15 | RS2 fall register |
| ***SHRRR*** | R/W | ‘ hff16 | SHR rise register |
| ***SHRFR*** | R/W | ‘ hff17 | SHR fall register |
| ***SHSRR*** | R/W | ‘ hff18 | SHS rise register |
| ***SHSFR*** | R/W | ‘ hff19 | SHS fall register |
| ***H1RR*** | R/W | ‘ hff1a | H1 rise register |
| ***H1FR*** | R/W | ‘ hff1b | H1 fall register |
| ***H2RR*** | R/W | ‘ hff1c | H2 rise register |
| ***H2FR*** | R/W | ‘ hff1d | H2 fall register |
| ***CLAMP0RR*** | R/W | ‘ hff1e | CLAMP 0 rise register |
| ***CLAMP0FR*** | R/W | ‘ hff1f | CLAMP 0 fall register |

| Name | R/W | Address | *Description* |
|---|---|---|---|
| ***CCLP0RR*** | R/W | ‘ hff20 | CCLP 0 rise register |
| ***CCLP0FR*** | R/W | ‘ hff21 | CCLP 0 fall register |
| ***CLAMP1RR*** | R/W | ‘ hff22 | CLAMP 1 rise register |
| ***CLAMP1FR*** | R/W | ‘ hff23 | CLAMP 1 fall register |
| ***CCLP1RR*** | R/W | ‘ hff24 | CCLP 1 rise register |
| ***CCLP1FR*** | R/W | ‘ hff25 | CCLP 1 fall register |
| ***CLAMP2RR*** | R/W | ‘ hff26 | CLAMP 2 rise register |
| ***CLAMP2FR*** | R/W | ‘ hff27 | CLAMP 2 fall register |
| ***CCLP2RR*** | R/W | ‘ hff28 | CCLP 2 rise register |
| ***CCLP2FR*** | R/W | ‘ hff29 | CCLP 2 fall register |
| ***CDSR0RR*** | R/W | ‘ hff2a | CDSR channel 0 rise register |
| ***CDSR0FR*** | R/W | ‘ hff2b | CDSR channel 0 fall register |
| ***CDSR1RR*** | R/W | ‘ hff2c | CDSR channel 1 rise register |
| ***CDSR1FR*** | R/W | ‘ hff2d | CDSR channel 1 fall register |
| ***CDSR2RR*** | R/W | ‘ hff2e | CDSR channel 2 rise register |
| ***CDSR2FR*** | R/W | ‘ hff2f | CDSR channel 2 fall register |
| ***CDSS0RR*** | R/W | ‘ hff30 | CDSS channel 0 rise register |
| ***CDSS0FR*** | R/W | ‘ hff31 | CDSS channel 0 fall register |
| ***CDSS1RR*** | R/W | ‘ hff32 | CDSS channel 1 rise register |
| ***CDSS1FR*** | R/W | ‘ hff33 | CDSS channel 1 fall register |
| ***CDSS2RR*** | R/W | ‘ hff34 | CDSS channel 2 rise register |
| ***CDSS2FR*** | R/W | ‘ hff35 | CDSS channel 2 fall register |
| ***CDSD0RR*** | R/W | ‘ hff36 | CDSD channel 0 rise register |
| ***CDSD0FR*** | R/W | ‘ hff37 | CDSD channel 0 fall register |
| ***CDSD1RR*** | R/W | ‘ hff38 | CDSD channel 1 rise register |
| ***CDSD1FR*** | R/W | ‘ hff39 | CDSD channel 1 fall register |
| ***CDSD2RR*** | R/W | ‘ hff3a | CDSD channel 2 rise register |
| ***CDSD2FR*** | R/W | ‘ hff3b | CDSD channel 2 fall register |
| ***MUXA0RR*** | R/W | ‘ hff3c | MUXA sel 0 rise register |
| ***MUXA0FR*** | R/W | ‘ hff3d | MUXA sel 0 fall register |
| ***MUXA1RR*** | R/W | ‘ hff3e | MUXA sel 1 rise register |
| ***MUXA1FR*** | R/W | ‘ hff3f | MUXA sel 1 fall register |
| ***MUXB0RR*** | R/W | ‘ hff40 | MUXB sel 0 rise register |
| ***MUXB0FR*** | R/W | ‘ hff41 | MUXB sel 0 fall register |
| ***MUXB1RR*** | R/W | ‘ hff42 | MUXB sel 1 rise register |
| ***MUXB1FR*** | R/W | ‘ hff43 | MUXB sel 1 fall register |
| ***OFFM0RR*** | R/W | ‘ hff44 | OFFM sel 0 rise register |
| ***OFFM0FR*** | R/W | ‘ hff45 | OFFM sel 0 fall register |
| ***OFFM1RR*** | R/W | ‘ hff46 | OFFM sel 1 rise register |
| ***OFFM1FR*** | R/W | ‘ hff47 | OFFM sel 1 fall register |
| ***PGAM0RR*** | R/W | ‘ hff48 | PGAM sel 0 rise register |
| ***PGAM0FR*** | R/W | ‘ hff49 | PGAM sel 0 fall register |
| ***PGAM1RR*** | R/W | ‘ hff4a | PGAM sel 1 rise register |
| ***PGAM1FR*** | R/W | ‘ hff4b | PGAM sel 1 fall register |

| Name | R/W | Address | *Description* |
|---|---|---|---|
| ***GAINM0RR*** | R/W | ‘ hff4c | GAINM sel 0 rise register |
| ***GAINM0FR*** | R/W | ‘ hff4d | GAINM sel 0 fall register |
| ***GAINM1RR*** | R/W | ‘ hff4e | GAINM sel 1 rise register |
| ***GAINM1FR*** | R/W | ‘ hff4f | GAINM sel 1 fall register |
| ***DCR*** | R/W | ‘ hff80 | DMA Control Register |
| ***D1TBHR*** | W | ‘ hff81 | DMA1 Transfer Byte Count High Byte Register |
| ***D1TBLR*** | W | ‘ hff82 | DMA1 Transfer Byte Count Low Byte Register |
| ***M1SAHR*** | W | ‘ hff83 | M1 Starting Address High Byte Register |
| ***M1SALR*** | W | ‘ hff84 | M1 Starting Address Low Byte Register |
| ***M2SAHR*** | R/W | ‘ hff85 | M2 Starting Address High Byte Register |
| ***M2SALR*** | R/W | ‘ hff86 | M2 Starting Address Low Byte Register |
| ***D2BCHR*** | R/W | ‘ hff88 | DMA2 byte count high register |
| ***D2BCLR*** | R/W | ‘ hff89 | DMA2 byte count low register |
| ***DMACR*** | R/W | ‘ hff8a | DMA configuration register |
| ***D2THR*** | R/W | ‘ hff8b | DMA2 Transfer Count High Byte Register |
| ***D2TLR*** | R/W | ‘ hff8c | DMA2 Transfer Count Low Byte Register |
| ***D2SHR*** | R/W | ‘ hff8d | DMA2 Sync Count High Byte Register |
| ***D2SLR*** | R/W | ‘ hff8e | DMA2 Sync Count Low Byte Register |
| ***RHR*** | R/W | ‘ hff8f | Resolution High Byte Register |
| ***RLR*** | R/W | ‘ hff90 | Resolution Low Byte Register |
| ***CCR*** | W | ‘ hff91 | Compression Configuration Register |
| ***MBLHR*** | R/W | ‘ hff92 | Minimum Buffer length High Register |
| ***MBLLR*** | R/W | ‘ hff93 | Minimum Buffer length Low Register |
| ***GPIOCR*** | R/W | ‘ hff94 | GPIO control register |
| ***GPIODR*** | R/W | ‘ hff95 | GPIO data register |
| ***WDTRR*** | W | ‘ hff96 | Watch Dog Timer Reset register |
| ***ASR*** | R/W | ‘ hff87 | AFE suspend register |
| ***ACR*** | R/W | ‘ hff97 | AFE configuration register |
| ***AC2R*** | R/W | ‘ hff98 | AFE configuration 2 register |
| ***COFF0R*** | R/W | ‘ hff99 | CDS OFFSET for channel 0 |
| ***CPGA0R*** | R/W | ‘ hff9a | CDS PGA for channel 0 |
| ***COFF1R*** | R/W | ‘ hff9b | CDS OFFSET for channel 1 |
| ***CPGA1R*** | R/W | ‘ hff9c | CDS PGA for channel 1 |
| ***COFF2R*** | R/W | ‘ hff9d | CDS OFFSET for channel 2 |
| ***CPGA2R*** | R/W | ‘ hff9e | CDS PGA for channel 2 |
| ***SCR*** | R/W | ‘ hff9f | Suspend Control |
| ***PPMCR*** | R/W | ‘ hffA0 | Parallel port mode control register |
| ***PPHSR*** | R/W | ‘ hffA1 | Parallel port Host Status register |
| ***PPHDR*** | R/W | ‘ hffA2 | Parallel port Host Data register |
| ***PPDSR*** | R/W | ‘ hffA3 | Parallel port Device Status register |
| ***EDHSAR*** | R/W | ‘ hffA5 | EPP/USB DMA3 high start address register |
| ***EDLSAR*** | R/W | ‘ hffA6 | EPP/USB DMA3 low start address register |
| ***EDHBCR*** | R/W | ‘ hffA7 | EPP DMA3 high byte count register |
| ***EDLBCR*** | R/W | ‘ hffA8 | EPP DMA3 low byte count register |

| NAME | R/W | Address | Description |
|---|---|---|---|
| ***GAINM0RR*** | R/W | ‘ hff4c | GAINM sel 0 rise register |
| ***GAINM0FR*** | R/W | ‘ hff4d | GAINM sel 0 fall register |
| ***GAINM1RR*** | R/W | ‘ hff4e | GAINM sel 1 rise register |
| ***GAINM1FR*** | R/W | ‘ hff4f | GAINM sel 1 fall register |
| ***DCR*** | R/W | ‘ hff80 | DMA Control Register |
| ***D1TBHR*** | W | ‘ hff81 | DMA1 Transfer Byte Count High Byte Register |
| ***D1TBLR*** | W | ‘ hff82 | DMA1 Transfer Byte Count Low Byte Register |
| ***M1SAHR*** | W | ‘ hff83 | M1 Starting Address High Byte Register |
| ***M1SALR*** | W | ‘ hff84 | M1 Starting Address Low Byte Register |
| ***M2SAHR*** | R/W | ‘ hff85 | M2 Starting Address High Byte Register |
| ***M2SALR*** | R/W | ‘ hff86 | M2 Starting Address Low Byte Register |
| ***D2BCHR*** | R/W | ‘ hff88 | DMA2 byte count high register |
| ***D2BCLR*** | R/W | ‘ hff89 | DMA2 byte count low register |
| ***DMACR*** | R/W | ‘ hff8a | DMA configuration register |
| ***D2THR*** | R/W | ‘ hff8b | DMA2 Transfer Count High Byte Register |
| ***D2TLR*** | R/W | ‘ hff8c | DMA2 Transfer Count Low Byte Register |
| ***D2SHR*** | R/W | ‘ hff8d | DMA2 Sync Count High Byte Register |
| ***D2SLR*** | R/W | ‘ hff8e | DMA2 Sync Count Low Byte Register |
| ***RHR*** | R/W | ‘ hff8f | Resolution High Byte Register |
| ***RLR*** | R/W | ‘ hff90 | Resolution Low Byte Register |
| ***CCR*** | W | ‘ hff91 | Compression Configuration Register |
| ***MBLHR*** | R/W | ‘ hff92 | Minimum Buffer length High Register |
| ***MBLLR*** | R/W | ‘ hff93 | Minimum Buffer length Low Register |
| ***GPIOCR*** | R/W | ‘ hff94 | GPIO control register |
| ***GPIODR*** | R/W | ‘ hff95 | GPIO data register |
| ***WDTRR*** | W | ‘ hff96 | Watch Dog Timer Reset register |
| ***ASR*** | R/W | ‘ hff87 | AFE suspend register |
| ***ACR*** | R/W | ‘ hff97 | AFE configuration register |
| ***AC2R*** | R/W | ‘ hff98 | AFE configuration 2 register |
| ***COFF0R*** | R/W | ‘ hff99 | CDS OFFSET for channel 0 |
| ***CPGA0R*** | R/W | ‘ hff9a | CDS PGA for channel 0 |
| ***COFF1R*** | R/W | ‘ hff9b | CDS OFFSET for channel 1 |
| ***CPGA1R*** | R/W | ‘ hff9c | CDS PGA for channel 1 |
| ***COFF2R*** | R/W | ‘ hff9d | CDS OFFSET for channel 2 |
| ***CPGA2R*** | R/W | ‘ hff9e | CDS PGA for channel 2 |
| ***SCR*** | R/W | ‘ hff9f | Suspend Control |
| ***PPMCR*** | R/W | ‘ hffA0 | Parallel port mode control register |
| ***PPHSR*** | R/W | ‘ hffA1 | Parallel port Host Status register |
| ***PPHDR*** | R/W | ‘ hffA2 | Parallel port Host Data register |
| ***PPDSR*** | R/W | ‘ hffA3 | Parallel port Device Status register |
| ***EDHSAR*** | R/W | ‘ hffA5 | EPP/USB DMA3 high start address register |
| ***EDLSAR*** | R/W | ‘ hffA6 | EPP/USB DMA3 low start address register |
| ***EDHBCR*** | R/W | ‘ hffA7 | EPP DMA3 high byte count register |
| ***EDLBCR*** | R/W | ‘ hffA8 | EPP DMA3 low byte count register |

| NAME | R/W | Address | Description |
|---|---|---|---|
| ***EDCR*** | R/W | ‘ hffA9 | EPP/USB DMA3 control register |
| ***TISR*** | R/W | ‘ hffAA | Top Interrupt status Register |
| ***UIS2R*** | R/W | ‘ hffAB | USB Interrupt status 2 Register |
| ***UIS1R*** | R/W | ‘ hffAC | USB Interrupt status 1 Register |
| ***TIER*** | R/W | ‘ hffAD | Top Interrupt Enable Register |
| ***UIE2R*** | R/W | ‘ hffAE | USB Interrupt Enable 2 Register |
| ***UIE1R*** | R/W | ‘ hffAF | USB Interrupt Enable 1 Register |
| ***ISR*** | R/W | ‘ hffB0 | Interrupt Select Register |
| ***DFR*** | R/W | ‘ hffB1 | Device Flag register |
| ***DBCARCR*** | R/W | ‘ hffB2 | DMA3 byte count auto-reload control register |
| ***URLBCR*** | R/W | ‘ hffB3 | USB Receive low byte count Register |
| ***URHBCR*** | R/W | ‘ hffB4 | USB Receive high byte count Register |
| ***URLBCGIR*** | R/W | ‘ hffB5 | USB Receive low byte count to generate IRQ |
| ***URLBCGIR*** | R/W | ‘ hffB6 | USB Receive high byte count to generate IRQ |
| ***URT*** | R | ‘ hffB8 | USB Request Type |
| ***URC*** | R | ‘ hffB9 | USB Request Code |
| ***URVLB*** | R | ‘ hffBA | USB Request Value Low Byte |
| ***URVHB*** | R | ‘ hffBB | USB Request Value High Byte |
| ***URILB*** | R | ‘ hffBC | USB Request Index Low Byte |
| ***URIHB*** | R | ‘ hffBD | USB Request Index High Byte |
| ***URLLB*** | R | ‘ hffBE | USB Request Length Low Byte |
| ***URLHB*** | R | ‘ hffBF | USB Request Length High Byte |
| ***UEIR*** | R/W | ‘ hffC0 | USB Endpoint index Register |
| ***DUSR*** | R | ‘ hffC1 | DMA3 USB Status Register |
| ***DUTBCI8R*** | R/W | ‘ hffC2 | DMA3 USB TX Byte Count for internal |
| ***DUTPS2R*** | R/W | ‘ hffC3 | DMA3 USB TX Packet Size 2 Register |
| ***DUTPS1R*** | R/W | ‘ hffC4 | DMA3 USB TX Packet Size 1 Register |
| ***DUTPC2R*** | R/W | ‘ hffC5 | DMA3 USB TX Packet Count 2 (# of packet) |
| ***DUTPC1R*** | R/W | ‘ hffC6 | DMA3 USB TX Packet Count 1 (# of packet) |
| ***DURPSR*** | R | ‘ hffC7 | DMA3 USB Receive packet size Register |
| ***UECR*** | R/W | ‘ hffC8 | USB Endpoint Control Register |
| ***UETSR*** | R/W | ‘ hffC9 | USB Endpoint Transmit Status Register |
| ***UERSR*** | R/W | ‘ hffCA | USB Endpoint Receive Status Register |
| ***UDAR*** | R/W | ‘ hffCB | USB Device Address Register |
| ***UDCR*** | R/W | ‘ hffCC | USB Device Control Register |
| ***UDSTR0*** | R/W | ‘ hffD0 | USB Device Status Transmit Register 0 |
| ***UDSTR1*** | R/W | ‘ hffD1 | USB Device Status Transmit Register 1 |
| ***UDSTR2*** | R/W | ‘ hffD2 | USB Device Status Transmit Register 2 |
| ***UDSTR3*** | R/W | ‘ hffD3 | USB Device Status Transmit Register 3 |
| ***UDSTR4*** | R/W | ‘ hffD4 | USB Device Status Transmit Register 4 |
| ***UDSTR5*** | R/W | ‘ hffD5 | USB Device Status Transmit Register 5 |
| ***UDSTR6*** | R/W | ‘ hffD6 | USB Device Status Transmit Register 6 |
| ***UDSTR7*** | R/W | ‘ hffD7 | USB Device Status Transmit Register 7 |
| ***UDCRR0*** | R | ‘ hffD8 | USB Device Command Receive Register 0 |

| *Name* | R/W | Address | Description |
|---|---|---|---|
| ***UDCRR1*** | R | ‘ hffD9 | USB Device Command Receive Register 1 |
| ***UDCRR2*** | R | ‘ hffDA | USB Device Command Receive Register 2 |
| ***UDCRR3*** | R | ‘ hffDB | USB Device Command Receive Register 3 |
| ***UDCRR4*** | R | ‘ hffDC | USB Device Command Receive Register 4 |
| ***UDCRR5*** | R | ‘ hffDD | USB Device Command Receive Register 5 |
| ***UDCRR6*** | R | ‘ hffDE | USB Device Command Receive Register 6 |
| ***UDCRR7*** | R | ‘ hffDF | USB Device Command Receive Register 7 |

- Registers Definitions
- Timing Generator Setup Register

**CPU Read/Write**

Address: FF00H

| Bit | Reset | Description |
|---|---|---|
| 7:4 | 4’ b0 | Reserved |
| 3 | 1’ b0 | Select LED control signal output to PEPP_AD0~PEPP_AD2 for the sake of driving issue.<br>When this bit is set, the mapping of pins are changed to:<br>**PTGR = TG_R,**<br>**PTGG = TG_G,**<br>**PTGB = TG_B,**<br>**PEPP_AD0 = LED_R,**<br>**PEPP_AD1 = LED_G,**<br>**PEPP_AD2 = LED_B;**<br>Note: TG_R, TG_G, TG_B are TG signal for R, G, B channel.<br>LED_R, LED_G, LED_B are LED signal for R, G, B channel.<br>These signals are programmable by register definitions. |
| 2 | 1’ b1 | TG rise event enable :<br>0 = disable TG rise event.<br>1 = enable TG rise event. TG event will occur when TG signal changes state from ‘ 0’ to ‘ 1’ |
| 1 | 1’ b0 | Sensor type :<br>0 = CIS<br>1 = CCD<br>As select in CIS mode, PH1 will output TG signal and controlled by H1_pol and H1_en. |
| 0 | 1’ b0 | Global Timing Generator enable control. |

Referenced period for 6-bits dot clock counter:

CPU Read/Write

Address: FF01H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Defines the referenced period (N+1) cycles of the 6 bits dot clock counter. The 6-bits dot clock counter is referenced by master clock (48MHz). Dot clock is defined by this N+1 cycles referenced to 6-bits dot clock counter. |

Referenced period for 6-bits pixel clock counter:

CPU Read/Write

Address: FF02H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Defines the referenced period (N+1) cycles of the 6 bits pixel clock counter. The 6-bits pixel clock counter is referenced by master clock (48MHz). Pixel clock is defined by this N+1 cycles referenced to 6-bits pixel clock counter.<br>The programmable timing control signal for sensor and AFE are referenced to pixel clock. |

Referenced period high byte for 16-bits counter of timing generator:

CPU Read/Write

Address: FF03H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Period for 16-bits counter high byte |

Referenced period low byte for 16-bits counter of timing generator:

Address: FF04H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Period for 16-bits counter low byte.<br>These two registers define the period (N+1) cycles of 16-bits counter referenced to pixel clock. All programmable timing control signals are referenced to this 16-bits counter. |

AFE ADCLK rising phase control

CPU Read/Write

Address: FF05H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define ADCLK rising phase referenced to 6-bits dot clock counter |

AFE ADCLK falling phase control

**CPU Read/Write**

Address: FF06H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define ADCLK falling phase referenced to 6-bits dot clock counter |

Sensor TG control signal rising phase referenced to pixel counter

**CPU Read/Write**

Address: FF07H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define TG rising phase reference to 6-bits pixel counter |

Sensor TG control signal falling phase referenced to pixel counter

Address: FF08H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define TG falling phase referenced to 6-bits pixel counter |

Sensor TG signal rising phase referenced to 16-bits counter:

**CPU Read/Write**

Address: FF09H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define TG rising phase referenced to 16-bit counter |

Sensor TG signal falling phase referenced to 16-bits counter:

**CPU Read/Write**

Address: FF0AH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define TG falling phase referenced to 16-bit counter |
| Note :<br>The conditions for rising and falling only available during first 256 cycles of 16-bits counter. For cycle exceeds 256 of 16-bits counter, TG output ‘ 0’ state. | | |

**Sensor TG control signal configuration**

**CPU Read/Write**

Address: FF0BH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 8’ b0 | Reserved |

| | | |
|---|---|---|
| 5 | 1’ b0 | TG_B polarity |
| 4 | 1’ b0 | TG_B enable |
| 3 | 1’ b0 | TG_G polarity |
| 2 | 1’ b0 | TG_G enable |
| 1 | 1’ b0 | TG_R polarity |
| 0 | 1’ b0 | TG_R enable |
| Note : all these bits are synchronous with TG rising edge.<br>Polarity = 0 : normal output as programming output<br>Polarity = 1 : invert output from programming output<br>Enable = 0 : disable control signal output<br>Enable = 1 : enable programming output. | | |

Sensor LED control signal rising phase high byte

**CPU Read/Write**

Address: FF0CH

| Bit | Reset | Description |
|---|---|---|
| 7:4 | 4’ b0 | Reserved |
| 3:0 | 4’ b0 | Define rising phase bit11~bit8 |

Sensor LED control signal rising phase low byte

**CPU Read/Write**

Address: FF0DH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define rising phase bit7~bit0 |

Sensor LED control signal falling phase high byte

**CPU Read/Write**

Address: FF0EH

| Bit | Reset | Description |
|---|---|---|
| 7:4 | 4’ b0 | Reserved |
| 3:0 | 4’ b0 | Define falling phase bit11~bit8 |

Sensor LED control signal falling phase low byte

**CPU Read/Write**

Address: FF0FH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define falling phase bit7~bit0 |
| Note:<br>The programming rising and falling phase are referenced to 16-bit counter with 12 MSB. | | |

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 8’ b0 | Reserved |
| 5 | 1’ b0 | LED_B polarity |

| | | |
|---|---|---|
| 4 | 1’ b0 | LED_B enable |
| 3 | 1’ b0 | LED_G polarity |
| 2 | 1’ b0 | LED_G enable |
| 1 | 1’ b0 | LED_R polarity |
| 0 | 1’ b0 | LED_R enable |
| Note : all these bits are synchronous with TG rising edge.<br>Polarity = 0 : normal output as programming output<br>Polarity = 1 : invert output from programming output<br>Enable = 0 : disable control signal output<br>Enable = 1 : enable programming output. | | |

Sensor TG mask period control register

**CPU Read/Write**

Address: FF11H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define TG mask period. This mask signal will go active when initial the 16-bits counter and clear after N cycles of pixel clock. During this period internal sensor control signal RS1, RS2, H1, and H2 will enter idle state, i.e. , output ‘ 0’ when this masking function is enable. |

Sensor RS1 control signal rising phase

**CPU Read/Write**

Address: FF12H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | RS1 polarity<br>0 = normal operation.<br>1 = invert RS1 output |
| 6 | 1’ b0 | RS1 enable<br>0 = disable RS1 output<br>1 = enable RS1 output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel clock counter |

Sensor RS1 control signal falling phase

**CPU Read/Write**

Address: FF13H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |

| | | |
|---|---|---|
| 6 | 1’ b0 | RS1 masking function enable<br>0 = disable masking function<br>1 = enable masking function |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel clock counter |

Sensor RS2 control signal rising phase

**CPU Read/Write**

Address: FF14H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | RS2 polarity<br>0 = normal operation.<br>1 = invert RS2 output |
| 6 | 1’ b0 | RS2 enable<br>0 = disable RS2 output<br>1 = enable RS2 output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel clock counter |

Sensor RS2 control signal falling phase

**CPU Read/Write**

Address: FF15H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | RS2 masking function enable<br>0 = disable masking function<br>1 = enable masking function |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel clock counter |

Sensor SHR control signal rising phase

**CPU Read/Write**

Address: FF16H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | SHR polarity<br>0 = normal operation<br>1 = invert SHR output |
| 6 | 1’ b0 | SHR enable<br>0 = disable SHR ouptut<br>1 = enable SHR output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel clock counter |

Sensor SHR control signal falling phase

CPU Read/Write

Address: FF17H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel clock counter |

Sensor SHS control signal rising phase

**CPU Read/Write**

Address: FF18H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | SHS polarity<br>0 = normal operation<br>1 = invert SHS output |
| 6 | 1’ b0 | SHS enable<br>0 = disable SHS ouptut<br>1 = enable SHS output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel clock counter |

Sensor SHS control signal falling phase

**CPU Read/Write**

Address: FF19H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel clock counter |

Sensor H1 control signal rising phase

**CPU Read/Write**

Address: FF1AH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | H1 polarity<br>0 = normal operation.<br>1 = invert H1 output |
| 6 | 1’ b0 | H1 enable<br>0 = disable H1 output<br>1 = enable H1 output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel clock counter |

Sensor H1 control signal falling phase

**CPU Read/Write**

Address: FF1BH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | H1 half enable |
| 6 | 1’ b0 | H1 masking function enable<br>0 = disable masking function<br>1 = enable masking function |

| | | |
|---|---|---|
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel clock counter |

Sensor H2 control signal rising phase

**CPU Read/Write**

Address: FF1CH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | H2 polarity<br>0 = normal operation.<br>1 = invert H2 output |
| 6 | 1’ b0 | H2 enable<br>0 = disable H2 output<br>1 = enable H2 output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel clock counter |

Sensor H2 control signal falling phase

**CPU Read/Write**

Address: FF1DH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | H2 half enable |
| 6 | 1’ b0 | H2 masking function enable<br>0 = disable masking function<br>1 = enable masking function |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel clock counter |

AFE CLAMP0 control signal rising phase

**CPU Read/Write**

Address: FF1EH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define rising phase referenced to 16-bits counter |

AFE CLAMP0 control signal falling phase

**CPU Read/Write**

Address: FF1FH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define falling phase referenced to 16-bits counter |
| Note:<br>These two registers define the rising and falling phase of CLAMP0 control signal, and only available on the first 256 cycles of 16-bits counter. CLAMP0 output ‘0’ when 16-bits counter exceeds 256 cycle. | | |

AFE CCLP0 control signal rising phase

**CPU Read/Write**

Address: FF20H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | CLAMP0 enable<br>0 = CLAMP0 output disable<br>1 = CLAMP0 output enable |
| 6 | 1’ b0 | CCLP0 enable<br>0 = CCLP0 output disable<br>1 = CCLP0 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel clock counter |

AFE CCLP0 control signal falling phase

**CPU Read/Write**

Address: FF21H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CLAMP1 control signal rising phase

**CPU Read/Write**

Address: FF22H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define rising phase referenced to 16-bits counter |

AFE CLAMP1 control signal falling phase

**CPU Read/Write**

Address: FF23H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define falling phase referenced to 16-bits counter |
| Note:<br>These two registers define the rising and falling phase of CLAMP1 control signal, and only available on the first 256 cycles of 16-bits counter. CLAMP1 output ‘0’ when 16-bits counter exceeds 256 cycle. | | |

AFE CCLP1 control signal rising phase

**CPU Read/Write**

Address: FF24H

| Bit | Reset | Description |
|---|---|---|
| | 1’ b0 | CLAMP1 enable<br>0 = CLAMP1 output disable<br>1 = CLAMP1 output enable |
| 6 | 1’ b0 | CCLP1 enable<br>0 = CCLP1 output disable<br>1 = CCLP1 output enable |

| | | |
|---|---|---|
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CCLP1 control signal falling phase

**CPU Read/Write**

Address: FF25H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CLAMP2 control signal rising phase

**CPU Read/Write**

Address: FF26H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define rising phase referenced to 16-bits counter |

AFE CLAMP2 control signal falling phase

**CPU Read/Write**

Address: FF27H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | Define falling phase referenced to 16-bits counter |
| Note:<br>These two registers define the rising and falling phase of CLAMP2 control signal, and only available on the first 256 cycles of 16-bits counter. CLAMP2 output ‘0’ when 16-bits counter exceeds 256 cycle. | | |

AFE CCLP2 control signal rising phase

**CPU Read/Write**

Address: FF28H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | CLAMP2 enable<br>0 = CLAMP2 output disable<br>1 = CLAMP2 output enable |
| 6 | 1’ b0 | CCLP2 enable<br>0 = CCLP2 output disable<br>1 = CCLP2 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CCLP2 control signal falling phase

CPU Read/Write

AFE CDSR0 control signal rising phase

**CPU Read/Write**

Address: FF2AH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSR0 enable<br>0 = CDSR0 output disable<br>1 = CDSR0 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSR0 control signal falling phase

**CPU Read/Write**

Address: FF2BH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CDSR1 control signal rising phase

**CPU Read/Write**

Address: FF2CH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSR1 enable<br>0 = CDSR1 output disable<br>1 = CDSR1 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSR1 control signal falling phase

**CPU Read/Write**

Address: FF2DH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CDSR2 control signal rising phase

**CPU Read/Write**

Address: FF2EH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |

| | | |
|---|---|---|
| 6 | 1’ b0 | CDSR2 enable<br>0 = CDSR2 output disable<br>1 = CDSR2 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSR2 control signal falling phase

**CPU Read/Write**

Address: FF2FH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CDSS0 control signal rising phase

**CPU Read/Write**

Address: FF30H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSS0 enable<br>0 = CDSS0 output disable<br>1 = CDSS0 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSS0 control signal falling phase

**CPU Read/Write**

Address: FF31H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CDSS1 control signal rising phase

**CPU Read/Write**

Address: FF32H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSS1 enable<br>0 = CDSS1 output disable<br>1 = CDSS1 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSS1 control signal falling phase

CPU Read/Write

Address: FF33H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CDSS2 control signal rising phase

**CPU Read/Write**

Address: FF34H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSS2 enable<br>0 = CDSS2 output disable<br>1 = CDSS2 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSS2 control signal falling phase

**CPU Read/Write**

Address: FF35H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CDSD0 control signal rising phase

CPU Read/Write

Address: FF36H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSD0 enable<br>0 = CDSD0 output disable<br>1 = CDSD0 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSD0 control signal falling phase

**CPU Read/Write**

Address: FF37H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE CDSD1 control signal rising phase

**CPU Read/Write**

Address: FF38H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSD1 enable<br>0 = CDSD1 output disable<br>1 = CDSD1 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSD1 control signal falling phase

**CPU Read/Write**

Address: FF39H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSD2 control signal rising phase

**CPU Read/Write**

Address: FF3AH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | CDSD2 enable<br>0 = CDSD2 output disable<br>1 = CDSD2 output enable |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE CDSD2 control signal falling phase

**CPU Read/Write**

Address: FF3BH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE MUXA0 control signal rising phase

**CPU Read/Write**

Address: FF3CH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert MUXA0 output signal |
| 6 | 1’ b0 | Enable<br>0 = disable MUXA0 output<br>1 = enable MUXA0 output |

| | | |
|---|---|---|
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE MUXA0 control signal falling phase

**CPU Read/Write**

Address: FF3DH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE MUXA1 control signal rising phase

**CPU Read/Write**

Address: FF3EH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert MUXA1 output signal |
| 6 | 1’ b0 | Enable<br>0 = disable MUXA1 output<br>1 = enable MUXA1 output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE MUXA1 control signal falling phase

**CPU Read/Write**

Address: FF3FH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE MUXB0 control signal rising phase

**CPU Read/Write**

Address: FF40H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert MUXB0 output signal |
| 6 | 1’ b0 | Enable<br>0 = disable MUXB0 output<br>1 = enable MUXB0 output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE MUXB0 control signal falling phase

**CPU Read/Write**

Address: FF41H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE MUXB1 control signal rising phase

**CPU Read/Write**

Address: FF42H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert MUXB1 output signal |
| 6 | 1’ b0 | Enable<br>0 = disable MUXB1 output<br>1 = enable MUXB1 output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE MUXB1 control signal falling phase

**CPU Read/Write**

Address: FF43H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE OFFSET Select 0 control signal rising phase

**CPU Read/Write**

Address: FF44H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert OFFSET Select 0 control signal output |
| 6 | 1’ b0 | Enable<br>0 = disable OFFSET Select 0 control signal output<br>1 = enable OFFSET Select 0 control signal output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE OFFSET Select 0 control signal falling phase

**CPU Read/Write**

Address: FF45H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE OFFSET Select 1 control signal rising

**CPU Read/Write**

Address: FF46H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert OFFSET Select 1 control signal output |
| 6 | 1’ b0 | Enable<br>0 = disable OFFSET Select 1 control signal output<br>1 = enable OFFSET Select 1 control signal output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE OFFSET Select 1 control signal falling

**CPU Read/Write**

Address: FF47H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE PGA Select 0 control signal rising phase

**CPU Read/Write**

Address: FF48H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert PGA Select 0 control signal output |
| 6 | 1’ b0 | Enable<br>0 = disable PGA Select 0 control signal output<br>1 = enable PGA Select 0 control signal output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE PGA Select 0 control signal falling phase

**CPU Read/Write**

Address: FF49H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE PGA Select 1 control signal rising phase

**CPU Read/Write**

Address: FF4AH

| Bit | Reset | Description |
|---|---|---|

| | | |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert PGA Select 1 control signal output |
| 6 | 1’ b0 | Enable<br>0 = disable PGA Select 1 control signal output<br>1 = enable PGA Select 1 control signal output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE PGA Select 1 control signal falling phase

**CPU Read/Write**

Address: FF4BH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE GAIN Select 0 control signal rising phase

**CPU Read/Write**

Address: FF4CH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert GAIN Select 0 control signal output |
| 6 | 1’ b0 | Enable<br>0 = disable GAIN Select 0 control signal output<br>1 = enable GAIN Select 0 control signal output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE GAIN Select 0 control signal falling phase

**CPU Read/Write**

Address: FF4DH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

AFE GAIN Select 1 control signal rising phase

**CPU Read/Write**

Address: FF4EH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Polarity<br>0 = normal operation<br>1 = invert GAIN Select 1 control signal output |

| | | |
|---|---|---|
| 6 | 1’ b0 | Enable<br>0 = disable GAIN Select 1 control signal output<br>1 = enable GAIN Select 1 control signal output |
| 5:0 | 6’ b0 | Define rising phase referenced to 6-bits pixel counter |

AFE GAIN Select 1 control signal falling phase

**CPU Read/Write**

Address: FF4FH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 2’ b0 | Reserved |
| 5:0 | 6’ b0 | Define falling phase referenced to 6-bits pixel counter |

DMA control register

**CPU Read/Write**

Address: FF80H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Last scan line:<br>0 = not last scan line<br>1 = current scan is last line<br>When do last line scan, CPU must set this bit to inform the controller the next DMA2 transfer is the last line scan. Clear this bit automatically when DMA2 is done. |
| 6 | 1’ b0 | Disable DMA2 write to M2<br>0 = DMA2 write data to M2 is enable<br>1 = skip current DMA2 writing data to M2<br>When this bit is set, the current DMA2 writing data to M2 will be disable. DMA2 transfer count will not increment during this bit set high. DMA2 event will be triggered. |
| 5:4 | 3’ b0 | Reserved |
| 3 | 1’ b0 | SROM download<br>0 = normal operation<br>1 = re-download code from external EEPROM<br>When this command is requested, CPU will enter reset state and re-download the program code stored in external EEPROM. The starting address must be defined before set this command. When download is completed, CPU is running with the download program code. |

| | | |
|---|---|---|
| 2 | 1’ b0 | DMA3 enable read image<br>DMA3 is defined as the transfer between M2 and PC interface (ECP, EPP, or USB).<br>When this bit is set to ‘ 1’ , the incoming DMA3 transfer indicates the transfer of image data to PC interface. The image valid data byte count will decrement by one automatically when DMA3 is a read transfer and this bit is set to one. If this bit is cleared to ‘ 0’ , the request for read/write M2 by PC interface is just the memory access.<br>If the last line scanning is present, this bit will also be cleared automatically when DMA2 is completed. |
| 1 | 1’ b0 | DMA2 enable<br>0 = DMA2 is disable<br>1 = DMA2 is enable<br>DMA2 is defined as the transfer from sensor input via AFE to M2.<br>When each line scan is complete, DMA2 event occurs and DMA2 enable bit is cleared automatically.<br>The address is started from the current M2 starting address and incremented automatically. CPU can poll this bit to determine the line scan is end or not. If overrun occurs, i.e., the image buffer is not enough to store the incoming image data, this bit will be cleared and disable the DMA2 transfer. In this condition, interrupt will be generated. |
| 0 | 1’ b0 | DMA1 enable<br>DMA1 is defined as the transfer from M2 to M1. The purpose of DMA1 is to download the updated program code to M1. The updated program code must be stored in M2 before DMA1 is active.<br>When this bit is set to ‘ 1’ , DMA1 is active. CPU will be halted and transfer data from M2 to M1. This bit will be clear automatically when DMA1 transfer is done. |

DMA1 transfer bytes Count high byte register

**CPU Read/Write**

Address: FF81H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | DMA1 transfer byte count high byte |

DMA1 transfer bytes count low byte register

**CPU Read/Write**

Address: FF82H

| Bit | Reset | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 8’ b0 | DMA1 transfer byte count low byte |
| Note:<br>DMA1 transfer bytes count indicates the total bytes that will be transferred from M2 to M1. Before DMA1 is active, set these two registers first. | | |

DMA1 M1 starting address high byte register

**CPU Read/Write**

Address: FF83H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | M1 starting address high byte |

DMA1 M1 starting address low byte register

**CPU Read/Write**

Address: FF84H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | M1 starting address low byte |
| Note:<br>DMA1 starting address indicates the initial address when DMA1 is on going. Address is automatically increment by one when one byte transfer is done. | | |

DMA M2 starting address high byte register

**CPU Read/Write**

Address: FF85H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | M2 starting address high byte |

**DMA M2 starting address high byte register**

**CPU Read/Write**

**Address: FF86H**

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | M2 starting address low byte |
| Note:<br>These two registers define the M2 starting address when DMA1 or DMA2 transfer are enable. Note that these two registers are changed on the fly during DMA1 and DMA2. Be careful to set the real initial address before DMA1 or DMA2 is enable. | | |

**DMA2 valid byte count high register**

**CPU Read/Write**

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | DMA2 valid byte count high byte |

DMA2 valid byte count low register

**CPU Read/Write**

Address: FF89H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | DMA2 valid byte count low byte |
| Note:<br>These two registers record the valid byte count left in M2. This counter will be increment when DMA2 transfer occurs, and decrement when DMA3 transfer has been issued. Before starting scan first line, CPU has to write ‘0’ to clear contents of these two registers. In normal scanning image process, CPU writes to these two registers are not recommended. | | |

DMA configuration register

**CPU Read/Write**

Address: FF8AH

| Bit | Reset | Description |
|---|---|---|
| 7:2 | 6’ b0 | AFE data valid phase<br>This phase defines the AFE data valid phase referenced to 6-bit dot clock counter of Timing Generator. This phase can be modified to fit the optimized quality of AFE output data . |
| 1 | 1’ b0 | Reset DMA2 FIFO<br>0 = normal operation<br>1 = reset DMA2 FIFO.<br>Always return ‘0’ when CPU read this bit. |
| 0 | 1’ b0 | Select external memory<br>The external memory is now supported to 64K bytes SRAM.<br>** Entering suspend mode:<br>1. Writing expected output data to M2. (CPU cycle)<br>2. Suspend M2<br>3. Pin PMDx will output the previous write data to avoid floating output. |

**DMA2 transfer count high byte register**

**CPU Read/Write**

**Address: FF8BH**

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | DMA2 transfer count high byte |

DMA2 transfer count low byte register

**CPU Read/Write**

Address: FF8CH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | DMA2 transfer count low byte |
| Note:<br><br>When DMA2 begins to transfer, these two registers defines the pixel count per line, i.e., when transfer reaches the count defined in these two register, transfer will stop and interrupt will occurs. Also DMA2 enable bit will be cleared. To enable the next line scan, set DMA2 enable again. | | |

DMA2 bypass count high byte register

**CPU Read/Write**

Address: FF8DH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | DMA2 bypass count high byte |

DMA2 bypass count low byte register

**CPU Read/Write**

Address: FF8EH

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ b0 | DMA2 bypass count low byte |
| Note:<br><br>These two registers define the dummy pixel count when DMA2 is enable. Note that DMA2 will be synchronized with TG signal (falling edge) and bypass the dummy pixels count defined here, the following data will be really image data and writes to M2. | | |

DPI setting high byte register

**CPU Read/Write**

Address: FF8FH

| Bit | Reset | Description |
|---|---|---|
| 7:2 | 8’ b0 | Reserved |
| 1:0 | 2’ h2 | Dpi setting high byte |

DPI setting low byte register

**CPU Read/Write**

Address: FF90H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ h58 | Dpi setting low byte |
| Note:<br>DPI = (optical sensor dpi) x (dpi setting)/600<br>For example, if optical sensor is 600 dpi, to get 300 dpi output we set:<br>DPI = (600) x (300/600) = 300 dpi<br>Dpi setting must be set to 300 (12c:heximal). | | |

Data type configuration register

**CPU Read/Write**

Address: FF91H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Reserved |
| 6 | 1’ b0 | Inverse AFE data<br>When this bit is set the AFE output data will be inverted. |
| 5 | 1’ b0 | CCR[5] |
| 4 | 1’ b0 | Scan mode<br>0 = line mode<br>1 = pixel mode |
| 3:0 | 4’ b01 | CCR[3:0] |

| TSTR.0 | CCR[5] | CCR[1:0] | CCR[3:2] | Description |
|---|---|---|---|---|
| 1’ bx | 1’ b1 | 2’ bxx | 2’ bx0 | 7 bits compression |
| 1’ bx | 1’ b1 | 2’ bxx | 2’ bx1 | 6 bits compression |
| 1’ b0 | 1’ b0 | 2’ bx1 | 2’ bxx | 8 bits optimize |
| 1’ b0 | 1’ b0 | 2’ b10 | 2’ bxx | 10 bits non-optimize |
| 1’ b0 | 1’ b0 | 2’ b00 | 2’ bxx | 12 bits non-optimize |
| 1’ b1 | 1’ b0 | 2’ b10 | 2’ bxx | 10 bits optimize |
| 1’ b1 | 1’ b0 | 2’ b00 | 2’ bxx | 12 bits optimize |
| 1’ b1 | 1’ b0 | 2’ b01 | 2’ bxx | 14 bits non-optimize |
| 1’ b1 | 1’ b0 | 2’ b11 | 2’ bxx | 16 bits optimize |

TSTR.0 = test register bit 0

DMA3 minimum buffer length high byte register

**CPU Read/Write**

Address: FF92H

| Bit | Reset | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 8’ h0 | DMA3 minimum buffer length high byte |

DMA3 minimum buffer length low byte register

**CPU Read/Write**

Address: FF93H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ h40 | DMA3 minimum buffer length low byte |

Note:

These registers define the minimum buffer length when DMA3 transfer is enabled. If the size of images stored in image buffer is less than defined values, controller will decode not ready signal to USB or EPP, or ECP controller. In the last line scan process, when DMA2 is completed, the not ready signal will be cleared no matter image buffer bytes count are larger or less than these registers.

General-purpose I/O pin control register

**CPU Read/Write**

Address: FF94H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ h00 | GPIO_en[7:0]<br>0 = set PGPIOx to input mode<br>1 = set PGPIOx to output mode |

GPIO Data Register

**CPU Read/Write**

Address: FF95H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ h00 | GPIO_data[7:0]<br>In input mode, read will return corresponding pin status. While in output mode, writing data to this register will simultaneously output to corresponding pin. |

Watch Dog Timer Reset port

**CPU Write**

Address: FF96H

| Bit | Reset | Description |
|---|---|---|
| 7:0 | 8’ hxx | Writing values to this register with 68h, then 01h will cause resetting of watch-dog-timer. If CPU exceeds 500ms not reset this register, hardware reset will occurs. |

AFE suspend register

**CPU Write**

Address: FF87H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | Status of DEV_WAKEUP (Read) |
| 6 | 1’ b0 | Reserved |
| 5 | 1’ b1 | BIAS_PD2 |
| 4 | 1’ b1 | CDS_PD2 |
| 3 | 1’ b1 | CDS_PD1 |
| 2 | 1’ b1 | CDS_PD0 |
| 1 | 1’ b1 | PGA_PD |
| 0 | 1’ b1 | ADC_PD |

AFE configuration register

**CPU Write**

Address: FF97H

| Bit | Reset | Description |
|---|---|---|
| 7:4 | 2’ b00 | Vref_sel2[3:0] |
| 3 | 1’ b0 | Reserved |
| 2 | 1’ b0 | AFE CDS_CTL signal |
| 1:0 | 2’ b0 | AFE Switch select |

AFE configuration 2 register

**CPU Write**

Address: FF98H

| Bit | Reset | Description |
|---|---|---|
| 7:4 | 4’ b0 | Vref_sel1[3:0] |
| 3:0 | 4’ b0 | Vref_sel0[3:0] |

AFE CDS Offset 0 register

**CPU Write**

Address: FF99H

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b00 | Reserved |
| 5:0 | 6’ b0 | AFE CDS OFFSET for channel 0 |

AFE CDS PGA 0 register

**CPU Write**

Address: FF9AH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b00 | Reserved |
| 5 | 1’ b0 | AFE CDS gain for channel 0 |

| | | |
|---|---|---|
| 4:0 | 6’ b0 | AFE CDS PGA for channel 0 |

AFE CDS Offset 1 register

**CPU Write**

Address: FF9BH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b00 | Reserved |
| 5:0 | 6’ b0 | AFE CDS OFFSET for channel 1 |

AFE CDS PGA 1 register

**CPU Write**

Address: FF9CH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b00 | Reserved |
| 5 | 1’ b0 | AFE CDS gain for channel 1 |
| 4:0 | 6’ b0 | AFE CDS PGA for channel 1 |

AFE CDS Offset 2 register

**CPU Write**

Address: FF9DH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b00 | Reserved |
| 5:0 | 6’ b0 | AFE CDS OFFSET for channel 2 |

AFE CDS PGA 2 register

**CPU Write**

Address: FF9EH

| Bit | Reset | Description |
|---|---|---|
| 7:6 | 2’ b00 | Reserved |
| 5 | 1’ b0 | AFE CDS gain for channel 2 |
| 4:0 | 6’ b0 | AFE CDS PGA for channel 2 |

Suspend control register

**CPU Read/Write**

Address: FF9FH

| Bit | Reset | Description |
|---|---|---|
| 7 | 1’ b0 | cpu_suspend |
| 6 | 1’ b0 | Suspend X1 |

| | | |
|---|---|---|
| 5 | 1’ b0 | Suspend 6803 |

| | | |
|---|---|---|
| 4 | 1’ b0 | Select low speed |
| 3 | 1’ b0 | Disable PLL reference clock input |
| 2 | 1’ b0 | Suspend PLL |
| 1 | 1’ b0 | Suspend 6802 |
| 0 | 1’ b0 | Suspend M2 |

TEST register

**CPU Read/Write**

Address: FFFFH

| Bit | Reset | Description |
|---|---|---|
| 7:1 | 1’ b0 | Always write ‘ 0’ to these bits |
| 0 | 1’ b0 | AFE select optimize mode |

Parallel Port Control Register

**EPP Host Read/Write (EPP access only)**

Address: FFH

| Bit | Reset | Description |
|---|---|---|
| 7 – 6 | 2' b00 | Load program time-out control. Time-out for loading program is controlled by the most significant two bits of this 29-bit counter.<br>Bit7 Bit6<br>0 0 : Don' t care bit 28/27<br>0 1 : Don' t care bit 28, care bit 27<br>1 0 : Don' t care bit 27, care bit 28<br>1 1 : Care bit 28/27 |
| 5 – 4 | 2' b00 | Reserved |
| 3 | 0 | Disable time-out counter to self-clear Parallel Port Interface Pass control.<br>0:enable counter ; 1: disable counter |
| 2 | 0 | Data Access Request: This bit is set when S/W starts to access data, and will be cleared automatically after finishing the data access. |
| 1 | 0 | Data Access Acknowledge: This bit is set whenever the data access request from bit 2 is finished. This bit will be cleared automatically when Data Access Request (bit 2) is set by software. |
| 0 | 1 | Read/Write: This bit indicates the direction of EPP host access program RAM.<br>1: Write 0: Read (for M1_enable) |

Parallel Port High Address Register

**EPP Host Read/Write (EPP access only)**

Address: FEH

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | - | Address bit 15-8 of program RAM. This register defines the high address of EPP host access program RAM. |

Parallel Port Low Address Register

**EPP Host Read/Write (EPP access only)**

Address: FDH

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | - | Address bit 7-0 of program RAM. This register defines the low address of EPP host access program RAM. |

Parallel Port Data Register

**EPP Host Read/Write (EPP access only)**

Address: FCH

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | - | 8-bit data for EPP host access GT6816. |

Image Data Valid High Pointer Register

**EPP Host Read/Write (EPP access only)**

Address: FBH

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 8’ h00 | This byte reflects the high address bit 15-8 of valid image data in buffer. |

Image Data Valid High Pointer Register

**EPP Host Read/Write (EPP access only)**

Address: FAH

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 8’ h00 | This byte reflects the low address bit 7-0 of valid image data in buffer. |

Image Valid Indication Register

**EPP Host Read/Write (EPP access only)**

Address: F9H

| Bit | Reset | Description |
|---|---|---|
| 7 – 1 | - | Reserved. |
| 0 | 0 | Indication of image data. 0/1: invalid/valid |

Host Access Device Flag Register

**EPP Host Read/Write (EPP access only)**

Address: E4H

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | When read, the data reflects Device Flag Register which has CPU address FFB1h.<br>When write, the data is used to handshake with device, default value is 8’ h00. |

Note: EPP host access this port must issue ASTROBE# with address E4h first.

During DMA3 transfer, EPP host must issue another ASTROBE# with new address.

Parallel Port Mode Control Register

**CPU Read/Write**

Address: FFA0H

| Bit | Reset | Description |
|---|---|---|
| 7 | 0 | This bit is used to select HOST or PRINTER bus to be read from address FFA1h and address FFA3h. (RD_PRINTER). 0/1 : HOST / |
| 6 | 0 | GT6816 will be forced to scanner mode if this bit is high "1". |
| 5 | - | This is read only bit to reflect scanner mode or pass-through mode.<br>0/1 : pass-through mode / scanner mode |
| 4 | 0 | Clear scanner mode and changes to pass-through mode. 1/0: clear / no action |
| 3 | 0 | Printer chain control selection. After loading the program, this bit must be set, in order to control the chain printer by bit 4. |
| 2 – 0 | 000 | Parallel Port Mode selection.<br>Bit2 Bit1 Bit0<br>1 0 0 Controlled by S/W<br>0 0 0 EPP mode , controlled by H/W automatically<br>0 0 1 ECP mode, controlled by H/W automatically |

Notes: 1. There are three methods to enter scanner mode:

(a) After protect / release sequence

(b) After scanner mode data sequence

(c) After setting bit 6 of address FFA0h

2. Only one way to return pass-through mode: Set bit 4 of address FFA0h

Parallel Port Host Status Register

**CPU Read/Write**

Address: FFA1H

| Bit | Reset | Description |
|---|---|---|
| 7 | 1 | IN / OUT selection of bit 3 of data bus of host parallel port. When output is selected, bit 7 of FFA3h must be set to high. 0 / 1 : Output / Input |
| 6 | 1 | IN / OUT selection of bit 2 of data bus of host parallel port. When output is selected, bit 7 of FFA3h must be set to high. 0 / 1 : Output / Input |
| 5 | 1 | IN / OUT selection of bit 1 of data bus of host parallel port. When output is selected, bit 7 of FFA3h must be set to high. 0 / 1 : Output / Input |
| 4 | 1 | IN / OUT selection of bit 0 of data bus of host parallel port. When output is selected, bit 7 of FFA3h must be set to high. 0 / 1 : Output / Input |
| 3 | - | Status bit for HostClk (ECP) / nWrite (EPP) / nStrobe (SPP) |
| 2 | - | Status bit for HostAck (ECP) / nDStrb (EPP) / nAutoFd (SPP) |
| 1 | - | Status bit for Active1284 (ECP) / Nastrb (EPP) / nSelectIn (SPP) |
| 0 | - | Status bit for nReverseRequest (ECP) / nInit (EPP,SPP) |

Notes: 1. Data written to bit 3-0 will be put on printer control bus if this device is in scanner mode.

2. When RD_PRINTER is low, reading bit 3-0 to reflect EPP HOST control signals.

When RD_PRINTER is high, reading bit 3-0 to reflect the written data.

3. RD_PRINTER is bit 7 of address FFA0h.

Parallel Port Host Data Register

**CPU Read/Write**

Address: FFA2H

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 8' h00 | When write, data will be put on parallel port data bus if DATA_OE = 1 (bit 7 of address FFA3h).<br>When read, it reflects the data status in parallel port. |

Parallel Port Device Status Register

**CPU Read/Write**

Address: FFA3H

| Bit | Reset | Description |
|---|---|---|
| 7 | 0 | Enable Parallel Port Reverse transfer. Data bus will be driven from Parallel Port Output Data register (address=FFA4h) to bus. (DATA_OE)<br>0: disable ; 1: enable<br>Note: It is a necessary condition to output data before setting this bit to high, however, bit 3-0 of parallel port is also controlled by bit 7-4 of address FFA1h. |
| 6 | 0 | Disable image_data_not_ready signal on UsrDf1 in hardware EPP mode.<br>0/1: enable / disable |
| 5 | 0 | Reserved |
| 4 | 0 | Device status : nPeripRqst (ECP) / UsrDf2 (EPP) / nFault (SPP) |
| 3 | 0 | Device status : Xflag (ECP) / UsrDf3 (EPP) / Select (SPP) |
| 2 | 0 | Device status : nAckReverse (ECP) / UsrDf1 (EPP) / Perror (SPP) |
| 1 | 0 | Device status : PeripAck (ECP) / nWait (EPP) / Busy (SPP) |
| 0 | 0 | Device status : PeripClk (ECP) / Intr (EPP) / nAck (SPP) |

Notes: 1. Data written to bit 4-0 will be put on host parallel port (Excluding hardware control signals)

2. When RD_PRINTER is low, reading bit 4-0 to reflect written data.

When RD_PRINTER is high, reading bit 4-0 to reflect PRINTER status signals.

3. RD_PRINTER is bit 7 of address FFA0h.

EPP/USB DMA3 Start High Address Register

**CPU Read/Write**

Address: FFA5H

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 00h | For write, it is bit 15-8 of EPP DMA3 starting address.<br>For read, it reflects the data of bit 15-8 of current counting address |

EPP/USB DMA3 Start Low Address Register

**CPU Read/Write**

Address: FFA6H

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 00h | For write, it is bit 7-0 of EPP DMA3 starting address.<br>For read, it reflects the data of bit 7-0 of counting address |

Note: It is up counter for DMA address.

EPP DMA3 Byte Count High Register

**CPU Read/Write**

Address: FFA7H

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 00h | The bit 15-8 of byte count for DMA3 transfer. (EPP only) |

EPP DMA3 Byte Count Low Register

**CPU Read/Write**

Address: FFA8H

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 00h | The bit 7-0 of byte count for DMA3 transfer. (EPP only) |

Note: There will transfer n byte if byte count is programmed n

USB/EPP DMA3 Control Register

**CPU Read/Write**

Address: FFA9H

| Bit | Reset | Description |
|---|---|---|
| 7 – 5 | 111 | Setup time to be added after receiving M22IP_ACK.<br>Tsetup = Tcpu_period * bit[7:5] + Tnwait_to_dstrobe |
| 4 | 0 | USB transmit/receive target selection. 0: SRAM(M2) , 1:internal 8 bytes<br>For internal 8 bytes RX/TX, index 0 register is first. |
| 3 | 0 | DMA3 read or write. 0: read; 1:write |
| 2 | 1 | EPP/USB selection for DMA3 transfer. 1: EPP mode; 0:USB mode |
| 1 | 0 | DMA3 byte-count load. This bit will be self-cleared after loading byte count.<br>For EPP mode, byte count will be loaded when writing “1” to this bit.<br>For USB mode, byte count and packet count will be loaded when |

| | | |
|---|---|---|
| | | bit. After loading byte count, USB TX is enabled automatically. |
| 0 | 0 | Enable EPP DMA3 transfer. 0: disable; 1:enable.<br>This bit will be cleared after finishing the DMA3 transfer. This bit is for EPP mode only. |

Top Interrupt Status Register (1A -- with bit 7 of address FFB0h= 0)

**CPU Read/Write**

Address: FFAAH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | ECP send command event. 0:clear. Writing “1”, the state is not changed. |
| 5 | 0 | EPP read address E4h event. 0:clear. Writing “1”, the state is not changed. |
| 4 | 0 | EPP write address E4h event. 0:clear. Writing “1”, the state is not changed. |
| 3 | 0 | DMA3 transmit done. 0: clear. Writing “1”, the state is not changed. |
| 2 | 0 | DMA2 transmit done. 0:clear; Writing “1”, the state is not changed. |
| 1 | 0 | Memory data overrun. 0:clear; Writing “1”, the state is not changed. |
| 0 | 0 | Global USB interrupt event. 0 : clear ; 1 : set (read only)<br>This bit reflects all the USB events. |

Note: These bits are set by event, and cleared by writing “0” to this bit

Top Interrupt Status Register (1B -- with bit 7 of address FFB0h= 1)

**CPU Read/Write**

Address: FFAAH

| Bits | Reset | Description |
|---|---|---|
| 7 - 1 | 0 | Reserved |
| 0 | 0 | TG event. 0:clear; Writing “1”, the state is not changed. |

Note: These bits are set by event, and cleared by writing “0” to this bit

USB Interrupt Status 2 Register

**CPU Read/Write**

Address: FFABH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | NAK flag after USB EP2 RX done. 0: clear; Writing “1”, the state is not changed. |
| 6 | 0 | NAK flag after USB EP1 RX done. 0: clear; Writing “1”, the state is not changed. |
| 5 | 0 | NAK flag after USB EP0 RX done. 0: clear; Writing “1”, the state is not changed. |
| 4 | 0 | Detect USB Bus reset event. 0: clear; Writing “1”, the state is not changed. |
| 3 | 0 | This read-only bit reflects USB remote wake-up event (from device). |
| 2 | 0 | Detect USB bus suspend event. 0: clear; Writing “1”, the state is not changed. |
| 1 | 0 | This read-only bit reflects USB bus resume event. |
| 0 | 0 | Detect USB Start of Frame (SOF). 0: clear; Writing “1”, the state is not changed. |

Note: These bits are set by event, and cleared by writing “0” to this bit

USB Interrupt Status Register (1A -- with bit 7 of address FFB0h= 0)

**CPU Read/Write**

Address: FFACH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | RX byte-count match event. 0: clear; Writing “1”, the state is not changed. |
| 5 | 0 | USB EP2 RX Done. 0: clear; Writing “1”, the state is not changed. |
| 4 | 0 | USB EP2 TX Done. 0: clear; Writing “1”, the state is not changed. |
| 3 | 0 | USB EP1 RX Done. 0: clear; Writing “1”, the state is not changed. |
| 2 | 0 | USB EP1 TX Done. 0: clear; Writing “1”, the state is not changed. |
| 1 | 0 | USB EP0 RX Done (SETUP & OUT). 0: clear; Writing “1”, the state is not changed. |
| 0 | 0 | USB EP0 TX Done. 0: clear; Writing “1”, the state is not changed. |

USB Interrupt Status Register (1B -- with bit 7 of address FFB0h= 1)

**CPU Read/Write**

Address: FFACH

| Bits | Reset | Description |
|---|---|---|
| 7 – 6 | - | Reserved |
| 5 | 0 | USB TX under-run error. 0: clear; Writing “1”, the state is not changed. |
| 4 | 0 | OUT data toggle bit error for EP2. 0: clear; Writing “1”, the state is not changed. (OUT-DATAx-ACK//OUT-DATAx-ACK, HUB doesn’ t receive ACK) |
| 3 | 0 | TX data toggle bit error for EP1. 0: clear; Writing “1”, the state is not changed. (IN-DATA-no ACK from HUB) |
| 2 | 0 | TX data toggle bit error for EP0. 0: clear; Writing “1”, the state is not changed. (IN-DATA-no ACK from HUB) |
| 1 | 0 | USB EP0 RX Done for OUT. 0: clear; Writing “1”, the state is not changed. |
| 0 | 0 | USB EP0 RX Done for SETUP. 0: clear; Writing “1”, the state is not changed. |

Top Interrupt Enable Register (1A -- with bit 7 of address FFB0h= 0)

**CPU Read/Write**

Address: FFADH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Global Enable Interrupt for peripheral module. 0: disable ; 1:enable |
| 6 | 0 | Enable interrupt for ECP host send-command. 0:disable: 1:enable |
| 5 | 0 | Enable interrupt for EPP read address E4h event. 0:disable. 1:enable |
| 4 | 0 | Enable interrupt for EPP write address E4h event. 0:disable. 1:enable |
| 3 | 0 | Enable Interrupt for detecting DMA3 TX done. 0: disable ; 1: enable |
| 2 | 0 | Enable Interrupt for detecting DMA2 TX done. 0:disable ; 1:enable |
| 1 | 0 | Enable Interrupt for detecting memory overrun. 0:disable ; 1:enable |
| 0 | 0 | Enable Interrupt for global USB event. 0 ; disable ; 1 ; enable |

Top Interrupt Enable Register (1B -- with bit 7 of address FFB0h= 1)

CPU Read/Write

Address: FFADH

| Bits | Reset | Description |
|---|---|---|
| 7 - 1 | 0 | Reserved |
| 1 | 0 | Select TG event. 0: /INT0 1: /INT1 |
| 0 | 0 | Enable TG event. 0 : disable ; 1 : enable |

USB Interrupt Enable 2 Register

**CPU Read/Write**

Address: FFAEH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Enable NAK for detecting USB EP2 RX done. 0:disable ; 1:enable |
| 6 | 0 | Enable NAK for detecting USB EP1 RX done. 0:disable ; 1:enable |
| 5 | 0 | Enable NAK for detecting USB EP0 RX done. 0:disable ; 1:enable |
| 4 | 0 | Enable Interrupt for detecting USB bus reset event. 0 : disable ; 1 : enable |
| 3 | 0 | Enable Interrupt for detecting USB remote wake-up event. 0 : disable ; 1 : enable |
| 2 | 0 | Enable Interrupt for detecting USB bus suspend event. 0 : disable ; 1 : enable |
| 1 | 0 | Enable Interrupt for detecting USB bus resume event. 0 : disable ; 1 : enable |
| 0 | 0 | Enable Interrupt for detecting USB Start of Frame (SOF). 0 : disable ; 1 : enable |

USB Interrupt Enable Register (1A -- with bit 7 of FFB0h= 0)

**CPU Read/Write**

Address: FFAFH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | Enable RX byte-count match event. 0:disable; 1: enable |
| 5 | 0 | Enable Interrupt for USB EP2 RX Done. 0 : disable ; 1 : enable |
| 4 | 0 | Enable Interrupt for USB EP2 TX Done. 0 : disable ; 1 : enable |
| 3 | 0 | Enable Interrupt for USB EP1 RX Done. 0 : disable ; 1 : enable |
| 2 | 0 | Enable Interrupt for USB EP1 TX Done. 0 : disable ; 1 : enable |
| 1 | 0 | Enable Interrupt for USB EP0 RX Done. 0 : disable ; 1 : enable |
| 0 | 0 | Enable Interrupt for USB EP0 TX Done. 0 : disable ; 1 : enable |

USB Interrupt Enable Register (1B -- with bit 7 of FFB0h= 1)

**CPU Read/Write**

Address: FFAFH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Enable NAK for USB EP2 RX data-toggle error. 0: disable; 1:enable |
| 6 | 0 | Enable NAK for USB EP1/EP0 TX data-toggle error. 0: disable; 1:enable |
| 5 | 0 | Enable Interrupt for USB TX under-run error. 0:disable; 1:enable |
| 4 | 0 | Enable Interrupt for USB EP2 OUT data-toggle error. 0: disable; 1:enable |
| 3 | 0 | Enable Interrupt for USB EP1 TX data-toggle error. 0: disable; 1:enable |
| 2 | 0 | Enable Interrupt for USB EP0 TX data-toggle error. 0: disable; 1:enable |
| 1 | 0 | Enable Interrupt for USB EP0 RX Done for OUT. 0: disable; 1:enable |
| 0 | 0 | Enable Interrupt for USB EP0 RX Done for SETUP. 0: disable; 1:enable |

Interrupt Select Register

**CPU Read/Write**

Address: FFB0H

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Select USB interrupt enable register or USB interrupt status register to read or write.<br>0: USB interrupt status register 1A and USB interrupt enable register 1A<br>0: USB interrupt status register 1B and USB interrupt enable register 1B |
| 6 | 0 | Select ECP-host send-command event. 0: /INT0 1: /INT1 |
| 5 | 0 | Select EPP read address E4h event. 0: /INT0. 1: /INT1 |
| 4 | 0 | Select EPP write address E4h event. 0: /INT0. 1: /INT1 |
| 3 | 0 | Select Interrupt for detecting DMA3 TX done. 0: /INT0 ; 1: /INT1 |
| 2 | 0 | Select Interrupt for detecting DMA2 TX done. 0: /INT0 ; 1: /INT1 |
| 1 | 0 | Select Interrupt for detecting memory overrun. 0: /INT0 ; 1: /INT1 |
| 0 | 0 | Select Interrupt for all USB events. 0: /INT0 ; 1: /INT1 |

Device Flag Register

**CPU Read/Write**

Address: FFB1H

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | - | When write: The CPU written data will be latched in buffer, and can be read from EPP Host with E4h address. The default value is 8' h00<br>When read: The data comes from EPP Host port E4h (Host Access Device Flag Register) |

DMA3 Byte Count Auto-Reload Control Register

**CPU Read/Write**

Address: FFB2H

| Bit | Reset | Description |
|---|---|---|
| 7 – 5 | - | Reserved |
| 4 | 0 | Enable DMA counting value to be read. 0/1: disable/enable |
| 3 | 0 | Flush USB TX prefetch-buffer. 0/1: disable/enable |
| 2 | 0 | Force USB TX to be stopped. 0/1 : normal / stop |
| 1 | 0 | Load TX packet count automatically when packet counter reaches zero (for USB)<br>When this bit is set, USB TX data is always enabled until this bit is cleared by CPU.<br>0/1:disable /enable |
| 0 | 0 | Load DMA3 byte count automatically when counter reaches zero (for EPP)<br>When this bit is enabled, DMA3 is not disabled by hardware, it must be cleared by firmware. 0/1:disable /enable |

USB Received Byte Count Low Register

**CPU Read/Write**

Address: FFB3H

| Bits | Reset | Description |
|---|---|---|

| | | |
|---|---|---|
| 7 – 0 | 8’ b0 | The bit 7-0 of byte count of received data |

USB Received Byte Count High Register

**CPU Read/Write**

Address: FFB4H

| Bits | Reset | Description |
|---|---|---|
| 7 – 5 | - | Reserved |
| 4 – 0 | 5’ b0 | The bit 12-8 of byte count of received data |

USB Received Byte Count Low Register to Generate Interrupt

**CPU Read/Write**

Address: FFB5H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 8’ b0 | The bit 7-0 of byte count of USB RX data to generate interrupt |

USB Received Byte Count High Register to Generate Interrupt

**CPU Read/Write**

Address: FFB6H

| Bits | Reset | Description |
|---|---|---|
| 7 – 5 | 3’ b0 | Reserved |
| 4 – 0 | 5’ b0 | The bit 12-8 of byte count of USB RX data to generate interrupt |

USB Request Type Register (USB setup command)

**CPU Read Only**

Address: FFB8H

| Bit | Reset | Description |
|---|---|---|
| 7 - 0 | 0 | BmRequestType (offset=0 in SETUP command). Write from USB, Read from CPU |

USB Request Code Register (USB setup command)

**CPU Read Only**

Address: FFB9H

| Bit | Reset | Description |
|---|---|---|
| 7 - 0 | 0 | Brequest (offset=1 in SETUP command). Write from USB, Read from CPU |

USB Request Value Low Byte Register (USB setup command)

**CPU Read Only**

Address: FFBAH

| Bit | Reset | Description |
|---|---|---|

| | | |
|---|---|---|
| 7 - 0 | 0 | Wvalue (low byte, offset=2 in SETUP command). Write from USB, Read from CPU |

USB Request Value High Byte Register (USB setup command)

**CPU Read Only**

Address: FFBBH

| Bit | Reset | Description |
|---|---|---|
| 7 - 0 | 0 | Wvalue (high byte, offset=3 in SETUP command). Write from USB, Read from CPU |

USB Request Index Low Byte Register (USB setup command)

**CPU Read Only**

Address: FFBCH

| Bit | Reset | Description |
|---|---|---|
| 7 - 0 | 0 | Windex (low byte, offset=4 in SETUP command). Write from USB, Read from CPU |

USB Request Index High Byte Register (USB setup command)

**CPU Read Only**

Address: FFBDH

| Bit | Reset | Description |
|---|---|---|
| 7 - 0 | 0 | Windex (high byte, offset=5 in SETUP command). Write from USB, Read from CPU |

USB Request Length Low Byte Register (USB setup command)

**CPU Read Only**

Address: FFBEH

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | Wlength (low byte, offset=6 in SETUP command). Write from USB, Read from CPU |

USB Request Length High Byte Register (USB setup command)

**CPU Read Only**

Address: FFBFH

| Bit | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | Wlength (high byte, offset=7 in SETUP command). Write from USB, Read from CPU |

USB Endpoint Index Register (USB setup command)

CPU Read Only

Address: FFC0H

| Bits | Reset | Description |
|---|---|---|
| 7 - 2 | 0 | Reserved |
| 1 - 0 | 2' b0 | Endpoint index (EPIDX)<br>Bit[1:0]<br>00 Endpoint 0<br>01 Endpoint 1<br>10 Endpoint 2<br>11 Reserved |

DMA3 USB Status Register

**CPU Read Only**

Address: FFC1H

| Bits | Reset | Description |
|---|---|---|
| 7 – 2 | - | Reserved. Read as "0" |
| 1 | - | This read-only bit reflects Image_not_rdy status |
| 0 | - | This read-only bit reflects TX_NOT_RDY status |

DMA3 USB TX Byte Count for Internal 8-byte Register

**CPU Read/Write**

Address: FFC2H

| Bits | Reset | Description |
|---|---|---|
| 7 | - | Reserved |
| 6 – 0 | 0 | Byte_Cnt8[6:0]. This byte count value is used when bit 4 of address FFA9h is high. |

DMA3 USB TX Packet Size 2 Register (1A -- with bit 7 of FFB0h= 0)

**CPU Read/Write**

Address: FFC3H

| Bits | Reset | Description |
|---|---|---|
| 7 – 2 | 0 | Reserved |
| 1 – 0 | 0 | USB_pkt_size[9:8]. Bit 9-8 of USB TX packet size |

DMA3 USB TX Packet Size 1 Register (1A -- with bit 7 of FFB0h= 0)

**CPU Read/Write**

Address: FFC4H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | USB_pkt_size[7:0]. Bit 7-0 of USB TX packet size |

DMA3 Disable Counting Register 2 (1B -- with bit 7 of FFB0h= 1)

**CPU Read/Write**

Address: FFC3H

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Enable to stop counting for DMA3 transfer. 0/1: disable/enable |
| 6 | 0 | Enable hardware to solve IN-DATA-time-out recover (packet size 64 only).<br>0/1: disable / enable |
| 5 | 0 | Enable 8-byte registers USB TX request. 0/1: disable / enable |
| 4 | - | Reserved |
| 3 | 0 | Disable M2 USB TX request. 0/1: enable / disable |
| 2 | - | Reserved |
| 1 – 0 | 0 | Stop_DMA3_Count[9:8]. Bit 9-8 of stop DMA3 transfer count. |

DMA3 Disable Counting Register 1 (1B -- with bit 7 of FFB0h= 1)

**CPU Read/Write**

Address: FFC4H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | Stop_DMA3_Count[7:0]. Bit 7-0 of stop DMA3 transfer count. |

DMA3 USB TX Packet Count 2 Register

**CPU Read/Write**

Address: FFC5H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | For write: USB_pkt_count[15:8]. Bit 15-8 of USB TX packet count<br>For read: When bit 4 of FFB2h is low, it reflects the written data of register<br>When bit 4 of FFB2h is high, it reflects bit 15-8 of counting byte |

DMA3 USB TX Packet Count 1 Register

**CPU Read/Write**

Address: FFC6H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | For write: USB_pkt_count[7:0]. Bit 7-0 of USB TX packet count<br>For read: When bit 4 of FFB2h is low, it reflects the written data of register<br>When bit 4 of FFB2h is high, it reflects bit 7-0 of counting byte |

Note: The total TX byte count is Usb_pkt_count * USB_pkt_size.

Interrupt will be generated after total TX Byte Count is reached.

DMA3 USB Receive Packet Size Register 1 (1A -- with bit 7 of FFB0h= 0)

CPU Read/Write

Address: FFC7H

DMA3 USB Receive packet size Register 1A (with bit 7 of FFB0h= 0): Address = FFC7

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 8’ b0 | Packet size of current received packet |

USB Debug Register 1 (1B -- with bit 7 of FFB0h= 1)

**CPU Read/Write**

Address: FFC7H

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Enable to detect OUT error protocol. 0/1: disable/enable |
| 6 | 0 | Enable to detect EP0 IN error protocol. 0/1: disable/enable |
| 5 | 0 | Enable to detect EP1 IN error protocol. 0/1: disable/enable |
| 4 - 0 | 0 | Reserved |

USB Endpoint Control Register

**CPU Read/Write**

Address: FFC8H

| EPIDX[1:0]=2`b00,EPCON0; EPIDX[1:0]=2`b01,EPCON1<br>EPIDX[1:0]=2`b10,EPCON2 (selected by FFC0h) | | |
|---|---|---|
| Bits | Reset | Description |
| 7 | 0 | Stall Receive Endpoint (SRE): If this bit is set, this endpoint will stall OUT token. |
| 6 | 0 | Stall Transmit Endpoint (STE): If this bit is set, this endpoint will stall IN token. |
| 5 | 1 | Control Endpoint (CE): This bit is set for EP0 only. |
| 4 | 0 | ISO: Isochronous transfer |
| 3 | 1 | Receive Input Enable (RIE):<br>When disabled, this endpoint returns a NAK to OUT token. |
| 2 | 1 | Endpoint Receive Enable (RE):<br>When disabled, the endpoint does not respond to OUT token. |
| 1 | 1 | Transmit Output Enable (TOE):<br>When disabled, the endpoint returns a NAK to IN token. |
| 0 | 1 | Endpoint Transmit Enable (TE):<br>When disabled, the endpoint does not respond to IN token. |

USB Endpoint Transmit Status Register

**CPU Read/Write**

Address: FFC9H

| EPIDX[1:0]=2`b00,EP_TXST0 ; EPIDX[1:0]=2`b01,EP_TXST1<br>EPIDX[1:0]=2`b10,EP_TXST2 (selected by FFC0h) | | |
|---|---|---|
| Bits | Reset | Description |
| 7 | 0 | Transmit Sequence (TSEQ):<br>This bit will be transmitted in the next PID and toggled on a valid ACK handshake. |
| 6 – 3 | 3’ b0 | Reserved. “0” when read |
| 2 | 0 | Transmit Time-out (TOUT):<br>This bit is set when time-out occurred. (no ACK after DATA phase) |
| 1 | 0 | Transmit Error (TERR):<br>This bit is set when error occurred, or this endpoint responds NAK to IN token. |
| 0 | 0 | Transmit Acknowledge (TACK): When this bit is set, means the data transmission is completed and acknowledge successfully. |

USB Endpoint Receive Status Register

**CPU Read/Write**

Address: FFCAH

| EPIDX[1:0]=2`b00,EP_RXST0; EPIDX[1:0]=2`b01,EP_RXST1<br>EPIDX[1:0]=2`b10,EP_RXST2 (selected by FFC0h) | | |
|---|---|---|
| Bits | Reset | Description |
| 7 | 0 | Receive Sequence Bit (RSEQ): When receiving endpoint sequence, this bit is set. |
| 6 | 0 | Receive Setup Token (RSETUP): When receiving the SETUP token, this bit is set. |
| 5 - 3 | 0 | Reserved |
| 2 | 0 | Receive Time-out (ROUT):<br>This bit is set when time-out occurred. (No DATA after OUT token) |
| 1 | 0 | Receive Error (RERR):<br>This bit is set when error occurred, or this endpoint responds NAK to OUT token. |
| 0 | 0 | Receive Acknowledge (RACK): When this bit is set, means the data reception is completed and acknowledge successfully. |

USB Device Address Register

**CPU Read/Write**

Address: FFCBH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 – 0 | 7’ b0 | Device Address[6:0]. Address is programmed through the command received from EP0 during the enumeration stage. |

USB Device Control Register

**CPU Read/Write**

Address: FFCCH

| Bits | Reset | Description |
|---|---|---|
| 7 | 0 | USB bus will be forced to SE0 state if this bit and bit 5 are high. |
| 6 | 0 | This state of this bit will be put on USB data bus when bit 5 is enabled. |
| 5 | 0 | USB transceiver output enable. When this bit is set, transceiver output is forced to enable and bit 6 will be put on D+. This function is used to enumerate device insertion or remove .0 / 1: disable / enable |
| 4 - 3 | 0 | Reserved |
| 2 | 0 | Enable suspend/resume timer to detect suspend/resume event in USB bus<br>0 : disable ; 1 : enable |
| 1 | 1 | Enable reset timer to detect reset event in USB bus. 0 : disable ; 1 : enable |
| 0 | 0 | Put transceiver in power down mode. 0 : normal mode ; 1 : power down mode |

USB Device Status Transmit Register 0 (DSTR0)

**CPU Read/Write**

Address: FFD0H

| Bits | Reset | Description |
|---|---|---|

| | | |
|---|---|---|
| 7 – 0 | 0 | DSTR0[7:0] |

USB Device Status Transmit Register 1 (DSTR1)

**CPU Read/Write**

Address: FFD1H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DSTR1[7:0] |

USB Device Status Transmit Register 2 (DSTR2)

**CPU Read/Write**

Address: FFD2H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DSTR2[7:0] |

USB Device Status Transmit Register 3 (DSTR3)

**CPU Read/Write**

Address: FFD3H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DSTR4[7:0] |

USB Device Status Transmit Register 5 (DSTR5)

**CPU Read/Write**

Address: FFD5H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DSTR5[7:0] |

USB Device Status Transmit Register 6 (DSTR6)

**CPU Read/Write**

Address: FFD6H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DSTR6[7:0] |

USB Device Status Transmit Register 7 (DSTR7)

**CPU Read/Write**

Address: FFD7H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DSTR7[7:0] |

Note: TX sequence is DSTR0 -> DSTR1 -> …-> DSTR7

USB Device Command Receive Register 0 (DCRR0)

**CPU Read Only**

Address: FFD8H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR0[7:0] |

USB Device Command Receive Register 1 (DCRR1)

**CPU Read Only**

Address: FFD9H

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR1[7:0] |

USB Device Command Receive Register 2 (DCRR2)

**CPU Read Only**

Address: FFDAH

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR2[7:0] |

USB Device Command Receive Register 3 (DCRR3)

**CPU Read Only**

Address: FFDBH

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR3[7:0] |

USB Device Command Receive Register 4 (DCRR4)

**CPU Read Only**

Address: FFDCH

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR4[7:0] |

USB Device Command Receive Register 5 (DCRR5)

**CPU Read Only**

Address: FFDDH

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR5[7:0] |

USB Device Command Receive Register 6 (DCRR6)

**CPU Read Only**

Address: FFDEH

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR6[7:0] |

USB Device Command Receive Register 7 (DCRR7)

**CPU Read Only**

Address: FFDFH

| Bits | Reset | Description |
|---|---|---|
| 7 – 0 | 0 | DCRR7[7:0] |

Note: Receive sequence is DCRR0 -> DCRR1 -> …-> DCRR7

The program memory can be read or written through EPP host, however, the procedure must be followed. The summary is listed as below.

1. Write Procedure:

(a) Writing high address index by software ( nAStrb, high address index)

(b) Writing high address data by software ( nDStrb, high address data)

(c) Writing low address index by software ( nAStrb , low address index)

(d) Writing low address data by software ( nDStrb , low address data)

(e) Writing data index by software ( nAStrb , data index)

(f) Writing data by software ( nDStrb , data)

(g) Writing control register to request transfer, and ACK is cleared by hardware

(h) Writing data to program memory by hardware

(i) ACK bit is set by hardware, REQUEST is cleared by hardware

2. Read Procedure:

(a) Writing high address index by software ( nAStrb , high address index)

(b) Writing high address data by software ( nDStrb , high address data)

(c) Writing low address index by software ( nAStrb , low address index)

(d) Writing low address data by software ( nDStrb , low address data)

(e) Writing control register to request transfer by software, and ACK is cleared by hardware

(f) Reading data from program memory by hardware

(g) ACK bit is set by hardware, REQUEST is cleared by hardware

(h) Writing data index by software ( nAStrb , data index)

(i) Reading data by software ( nDStrb , data)

There are three special modes defined in GT6816. The first one is protect mode, it defines the mode which program can be read or written by EPP host. The second one is release mode, it defines the end of program memory can be accessed by EPP host. The protect mode and release modes are used to start and end of accessing program memory. The last one is scanner mode, it defines the signals from EPP host will be passed to printer chain or not?

By sampling the data on the EPP host data bus, these three modes will be detected. These special sequences are defined in 32 bytes consecutive data. The detail codes for these three modes are listed as below.

| | PROTECT | RELEASE | SCANNER MODE |
|---|---|---|---|
| 1. | 01010101 (55H) | 01010101 (55H) | 01010101 (55H) |
| 2. | 01011101 (5DH) | 01011101 (5DH) | 01011101 (5DH) |
| 3. | 01111101 (7DH) | 01111101 (7DH) | 01111101 (7DH) |
| 4. | 01111111 (7FH) | 01111111 (7FH) | 01111111 (7FH) |
| 5. | 11111111 (FFH) | 11111111 (FFH) | 11111111 (FFH) |
| 6. | 11111110 (FEH) | 11111110 (FEH) | 11111110 (FEH) |
| 7. | 11101110 (EEH) | 11101110 (EEH) | 11101110 (EEH) |
| 8. | 11100110 (E6H) | 11100110 (E6H) | 11100110 (E6H) |
| 9. | 10100110 (A6H) | 10100110 (A6H) | 10100110 (A6H) |
| 10. | 10100010 (A2H) | 10100010 (A2H) | 10100010 (A2H) |
| 11. | 00100010 (22H) | 00100010 (22H) | 00100010 (22H) |
| 12. | 00100000 (20H) | 00100000 (20H) | 00100000 (20H) |
| 13. | 00000000 (00H) | 00000000 (00H) | 00000000 (00H) |
| 14. | 00000001 (01H) | 00000001 (01H) | 00000001 (01H) |
| 15. | 10000001 (81H) | 10000001 (81H) | 10000001 (81H) |
| 16. | 10000011 (83H) | 10000011 (83H) | 10000011 (83H) |
| 17. | 11000011 (C3H) | 11000011 (C3H) | 11000011 (C3H) |
| 18. | 11001011 (CBH) | 11001011 (CBH) | 11001011 (CBH) |
| 19. | 01001011 (4BH) | 01001011 (4BH) | 01001011 (4BH) |
| 20. | 01001010 (4AH) | 01001010 (4AH) | 01001010 (4AH) |
| 21. | 01101010 (6AH) | 01101010 (6AH) | 01101010 (6AH) |
| 22. | 01101000 (68H) | 01101000 (68H) | 01101000 (68H) |
| 23. | 01111000 (78H) | 01111000 (78H) | 01111000 (78H) |
| 24. | 01111001 (79H) | 01111001 (79H) | 01111001 (79H) |
| 25. | 11111001 (F9H) | 11111001 (F9H) | 11111001 (F9H) |
| 26. | 11111101 (FDH) | 11111101 (FDH) | 11111101 (FDH) |
| 27. | 11011101 (DDH) | 11011101 (DDH) | 11011101 (DDH) |
| 28. | 11010101 (D5H) | 11010101 (D5H) | 11010101 (D5H) |
| 29. | 10010101 (95H) | 10010101 (95H) | 10010101 (95H) |

30. 10010100 (94H) 10010100 (94H) 10010100 (94H)

31. 00010100 (14H) 00010100 (14H) 00010100 (14H)

32. 00011100 (1CH) 00010101 (15H) 00111100 (3CH)

1. USB Overview

The USB hardware includes a USB function with one upstream port. The USB port interfaces to the micro-controller through a high-speed serial interface engine (SIE). This micro-controller provides the functionality of standard USB commands and scanner commands.

1-1 USB Serial Interface Engine (SIE)

In order to allow the micro-controller to communicate with the USB host. The SIE handles the following USB bus activity:

(a) NRZI decode / encode

(b) Bit stuffing / un-stuffing

(c) Checksum generation and checking

(d) Address checking

(e) Token type identification

(f) Time-out check

Firmware is required to handle the rest of USB interface:

(a) Flow of USB bus enumeration

(b) Function of USB standard commands

(c) Function of scanner commands

(d) Data transfer from buffer to USB host

(e) Down-load program code from USB host

1-2 USB Enumeration

The enumeration sequence is the process that the USB host uses to identify and manage the state of device when a USB device is attached to or removed from the USB. When a USB device is attached, the following actions are undertaken from device side:

(a) USB host sends RESET command

(b) USB host sends SETUP command with address 0 to get device descriptor

(c) After receiving device descriptor, USB host sends SETUP command with new address to device

(d) Device must store the new address after receiving the SETUP command

(e) The host sends SETUP command with new address to get device descriptor

(f) The host sends SETUP command to get configuration descriptor or other descriptor

(g) Enumeration is complete after the host has received all the descriptor

1-3. Control pipe (Endpoint 0)

Endpoint 0 is a bi-direction USB control endpoint. The USB host sends SETUP command to device through endpoint 0, this 8-byte command will be stored at GT6816 internal buffers which can be read by micro-controller from address FFB8H to FFBFH. The micro-controller must read command and interpret

command, and then handle the request from USB host. Both the USB standard commands and vendor-specific commands are through endpoint 0.

1-4. Image Line Buffer Read Pipe (Endpoint 1)

Endpoint 1 is a USB BULK-IN pipe that transmits each packet of 64 bytes. This endpoint is used to send image data from buffer to USB host. The total packet count to be transmitted is depended on the DMA3 USB TX Packet Count 2 Register (FFC5H), DMA3 USB TX Packet Count 1 Register (FFC6H) and DMA3 Byte Count Auto-Reload Control Register (FFB2H).

1-5. Host Data Write Pipe (Endpoint 2)

Endpoint 2 is a USB BULK-OUT pipe that receives packet up to 64 bytes in length. The received data can be written to image line buffer, the beginning address is specified in the EPP/USB DMA3 Start High Address Register (FFA5H) and EPP/USB DMA3 Start Low Address Register (FFA6H). The interrupt can be generated for each OUT transaction or last OUT transaction, it is easier for firmware to handle large data transfer.

## VII. System Block Diagram

### 1. 16-bits Analog Front End(AFE)

The Gain through each channel can be set between 0.8x and 7.8x. The Offset provides up to +/- 1.5V of offset correction. The gain and offset stages should be adjusted during coarse calibration so that input signal is a maximum of ? at the ADC input.

The digital data comes from a 6 MHz 16-bit pipelined ADC. The output data is formatted as a 16 bit word.

### 2. Clock source

GT6816 uses 6MHz external crystal as the input clock. Typically when **PPLL_EN_** is pull to low, internal built-in PLL is enable and output 48MHz as the master clock to eliminate EMI effects. As PPLL_EN_ is pull to high, PLL is disabled and external crystal output is directly sent to internal as the master clock. In this mode, always use 48MHz crystal as the input clock source.

### 3. External Serial EEPROM interface

GT6816 support I2C serial EEPROM interface for updating program code on the power-on state. GT6816 searches the external serial EEPROM on the power-on state. If no serial EEPROM is present, embedded program code will be used. This is helpful when system is on the developing state. Note that always pull high on SCL and SDA pin.

### 4. DMA transfer

There are 3 DMA transfers on GT6816.

(i) DMA1:

DMA1 is defined for transferring data from M2 (image buffer) to M1 (program memory). When DMA1 is active, CPU enters the suspend-state and waits for the end of DMA1. The total bytes of DMA1 transfer and the target address of M1 can be programmed. After DMA1 is complete, CPU leaves suspend-state and operates with newly updated program code. It is helpful for updating firmware stored in M1 and implement different function. PC can load the new code to M2 via DMA3 write operation and request CPU to do DMA1 transfer. On this concept, any changes of firmware version can be easily updated via Internet.

(ii) DMA2

The sensor analog input is sent to 16-bits AFE and output digital pixel data to M2. This process is called DMA2. Each DMA2 is synchronized with TG signal and generates end of DMA2 event to CPU. While end of DMA event occurs, CPU checks the image buffer status to determine whether to continue DMA2 or not.

(iii) DMA3:

PC interface read/write M2 is the process of DMA3

**5. Built-in 16K image buffer**

GT6816 uses internal 16K SRAM as the image buffer to store the pixel data via DMA2 process and send it to PC interface via DMA3. The purpose of image buffer is used as the buffer between DMA2 and DMA3. For the fast DMA3 (PC interface read from buffer), image buffer is never full and has a best performance on the pixel data transfer. If DMA3 is slower, as the buffer is near full, DMA2 (pixel data write to buffer) will be halted and wait for enough buffer size to continue. Built-in image buffer ,therefore, can save the external memory requirement.

**6. Watch dog timer**

GT6816 has built-in watch-dog-timer to avoid uncertain conditions that make CPU reach unknown state. CPU has to reset the watch-dog-timer every 500ms. If CPU does not reset watch-dog-timer more than 500ms, GT6816 will return initial state and reset all the hardware settings.

**7.Remote-wake up function**

Change states on pin WAKEUP will force GT6816 leave suspend state and do the corresponding service.

**8. Programmable timing generator**

All sensor and AFE control signals can be fully programmable by timing generator module. This provides the highly compatibility to the different system requirements.

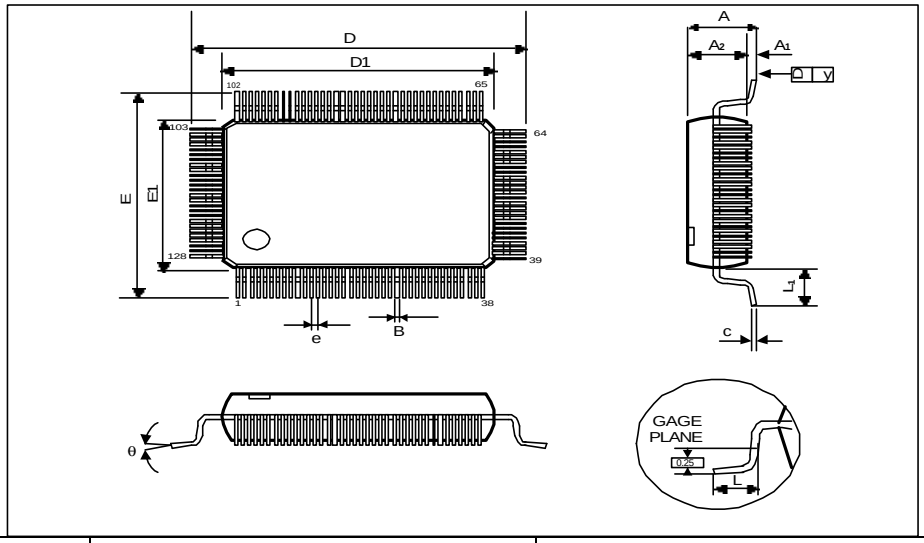
**9. Suspend management**

Each module include AFE, CPU, program memory, image buffer, DMA controller, etc, has its corresponding suspend control bit. Entering suspend state is determined by CPU setting these bits. That makes bus-powered of USB solution can be easily implemented.

**10. General-purpose I/O**

There are up to 42 pin GPIO in 128-pin package.

**VIII. APPENDIX: PACKAGE MECHANICAL DATA- 128-pin QFP**




| Symbol | Dimensions in inch | | | Dimensions in mm | | |
|---|---|---|---|---|---|---|
| | **Min.** | **Nom.** | **Max.** | **Min.** | **Nom.** | **Max.** |
| A | - | - | 0.134 | - | - | 3.40 |
| A**1** | 0.010 | - | - | 0.25 | - | - |
| A**2** | 0.107 | 0.112 | 0.117 | 2.73 | 2.85 | 2.97 |
| B | 0.007 | 0.009 | 0.011 | 0.17 | 0.22 | 0.27 |
| C | 0.004 | - | 0.008 | 0.09 | - | 0.20 |
| D | 0.906 | 0.913 | 0.921 | 23.00 | 23.20 | 23.40 |
| D1 | 0.783 | 0.787 | 0.791 | 19.90 | 20.20 | 20.10 |
| E | 0.669 | 0.677 | 0.685 | 17.00 | 17.20 | 17.40 |
| E1 | 0.547 | 0.551 | 0.555 | 13.90 | 14.00 | 14.10 |
| e | 0.020 BSC | | | 0.5 BSC | | |
| L | 0.029 | 0.035 | 0.041 | 0.73 | 0.88 | 1.03 |
| L**1** | 0.063 BSC | | | 1.60 BSC | | |
| y | - | - | 0.004 | - | - | 0.10 |
| θ | 0° | - | 7° | 0° | - | 7° |

**Notes:**

1. Dimensions D1 and E1 do not include mold protrusion. But mold mismatch is included. Allowable protrusion is .25mm/.010” per side.
2. Dimensions B does not include dambar protrusion. Allowable dambar protrusion is .08mm/.003”” per side. Total in excess of the B dimension at maximum material condition. Dambar cannot be located on the lower radius of the foot.

- This publication contains the design target or goal specifications for product development. Specifications and information herein are subject to change without prior notice.

- 

- All applications and circuit parameters herein are for illustrative purposes only. Grandtech Semiconductor makes no warranty that such applications will be suitable for volume production without further testing or modification.

- Grandtech products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support
  or sustain life.