ENE TECHNOLOGY INC.

KB3930 for OLPC

Keyboard Controller

Data Sheet

V 0.2

May. 2010

Headquarters
4F-1, No.9, Prosperity Rd.,
Science-based Industrial Park,
Hsinchu City, Taiwan, R.O.C
TEL: 886-3-6662888
FAX: 886-3-6662999
http://www.ene.com.tw

Taipei Office
4F, No.88, Bauchiau Rd.
Shindian City, Taipei,
Taiwan, R.O.C.
TEL: 886-2-89111525
FAX: 886-2-89111523

Revision

Revision	Description	Date
0.1	1. 1st release as KB3930 OLPC datasheet	2010/04 (0.7)
0.2	1. Remove Watermark	2010/5

1. GENERAL DESCRIPTION 0
1.1 OVERVIEW 0
1.2 FEATURES 1
1.3 COMPARISON (KB3926D VS. KB3930) 6
1.4 BLOCK DIAGRAM 7
2. PIN ASSIGNMENT AND DESCRIPTION 8
2.1 KB3930 128-PIN LQFP DIAGRAM TOP VIEW 8
2.2 KB3930 128 LFBGA BALL MAP 9
2.3 KB3930 PIN ASSIGNMENT SIDE A 10
2.4 KB3930 PIN ASSIGNMENT SIDE B 11
2.5 KB3930 PIN ASSIGNMENT SIDE C 12
2.6 KB3930 PIN ASSIGNMENT SIDE D 13
2.7 I/O CELL DESCRIPTIONS 14
2.7.1 I/O Buffer Table 14
2.7.2 I/O Buffer Characteristic Table 14
3. PIN DESCRIPTIONS 15
3.1 HARDWARE TRAP 15
3.2 PIN DESCRIPTIONS BY FUNCTIONS 16
3.2.1 Low Pin Count I/F Descriptions. 16
3.2.2 SPI Flash I/F Descriptions 16
3.2.3 PS/2 I/F Descriptions 16
3.2.4 Internal Keyboard Encoder (IKB) Descriptions 17
3.2.5 SMBus Descriptions 17
3.2.6 FAN Descriptions 17
3.2.7 Pulse Width Modulation (PWM) Descriptions 17
3.2.8 Analog-to-Digital Converter Descriptions 17
3.2.9 Digital-to-Analog Converter Descriptions 18
3.2.10 8051 External I/F Descriptions 18
3.2.11 External Clock Descriptions 18
3.2.12 Miscellaneous Signals Descriptions 18
3.2.13 Voltage Comparator Pins Descriptions 19
3.2.14 Power Pins Descriptions 19
4. MODULE DESCRIPTIONS 20
4.1 CHIP ARCHITECTURE 20
4.1.1 Power Planes 20
4.1.2 Clock Domains 21

4.1.4 Internal Memory Map 24
4.2 GPIO 25
4.2.1 GPIO Function Description 25
4.2.2 GPIO Structures 28
4.2.3 GPIO Attribution Table 29
4.2.3 GPIO Registers Descriptions 32
4.2.4 GPIO Programming Sample 43
4.3 KEYBOARD AND MOUSE CONTROL INTERFACE (KBC) 44
4.3.1 KBC I/F Function Description 44
4.4 ENE SERIAL BUS CONTROLLER (ESB) 49
4.4.1 ESB Function Description 49
4.4.2 ESB Registers Description 50
4.5 RESERVED 56
4.6 PECI 57
4.6.1 PECI Functional Description 57
4.6.2 PECI Register Description (Base address = FCD0h, 16 bytes) 58
4.7 OWM 61
4.7.1 OWM Functional Description 61
4.8 PULSE WIDTH MODULATION (PWM) 65
4.8.1 PWM Function Description 65
4.8.2 PWM Registers Description 66
4.8.3 PWM Programming Sample 68
4.9 FAN CONTROLLER 69
4.9.1 Fan Function Description 69
4.9.2 Fan Registers Description 70
4.9.3 Fan Programming Sample 76
4.10 GENERAL PURPOSE TIMER (GPT) 77
4.10.1 GPT Function Description 77
4.10.2 GPT Registers Description 77
4.10.3 GPT Programming Sample 79
4.11 SDI HOST/DEVICE INTERFACE CONTROLLER 80
4.11.1 SDI Host/Device Interface Description 80
4.11.2 SDI Host Interface Description 80
4.11.2 SDI Device Interface Description 81
4.11.3 SDI Programming Sample 84
4.12 WATCHDOG TIMER (WDT) 85

4.12.1 WDT Function Description 85
4.12.2 WDT Registers Description 85
4.12.3 WDT Programming Sample 88
4.13 LOW PIN COUNT INTERFACE (LPC) 89
4.13.1 LPC Function Description 89
4.13.2 LPC I/O Decode Range 89
4.13.3 LPC Memory Decode Range 89
4.13.4 FWH Memory Decode Range 90
4.13.5 Index-I/O Port 90
4.13.6 Extended I/O Port (Debug Port, Port80) 91
4.13.7 LPC Registers Description 92
4.14 X-BUS INTERFACE (XBI) 100
4.14.1 XBI Function Description 100
4.14.2 XBI SPI Enhancement 100
4.14.3 XBI Registers Description 103
4.15 CONSUMER IR CONTROLLER (CIR) 109
4.15.1 CIR Function Description 109
4.15.2 CIR Block Diagram 111
4.15.3 CIR Remote Protocol 112
4.15.3.1 Philips RC5 Protocol 112
4.15.3.2 Philips RC6 Protocol 113
4.15.3.3 NEC Protocol 113
4.15.4 CIR Automatic Carrier Frequency Detection and Modulation 114
4.15.5 CIR Registers Description 116
4.15.3 CIR Programming Sample 120
4.16 PS/2 INTERFACE (PS/2) 121
4.17 EMBEDDED CONTROLLER (EC) 122
4.17.1 EC Function Description 122
4.17.2 EC Command Program Sequence 123
4.17.3 EC SCI Generation 124
4.17.4 EC/KBC Clock Configuration 125
4.17.5 A/D Converter Control 125
4.17.6 D/A Converter Control 127
4.17.7 Power Management Control 128
4.17.8 EC Registers Description 129
4.18 GENERAL PURPOSE WAKE-UP CONTROLLER (GPWU) 140
4.18.1 GPWU Function Description 140

4.18.2 GPWU Registers Description 141
4.18.3 GPWU Programming Sample 146
4.19 SYSTEM MANAGEMENT BUS CONTROLLER (SMBUS) 147
4.19.1 SMBus Function Description 147
4.19.2 SMBus Register Description 149
4.20 8051 MICROPROCESSOR 154
4.20.1 8051 Microprocessor Function Description 154
4.20.2 8051 Microprocessor Instruction 155
4.20.3 8051 Interrupt Controller 159
4.20.4 Interrupt Enable/Flag Table 160
4.20.5 8051 Special Function Register (SFR) 162
4.20.6 8051 Microprocessor Register Description 163
5. ELECTRICAL CHARACTERISTICS 170
5.1 ABSOLUTE MAXIMUM RATING 170
5.2 DC ELECTRICAL CHARACTERISTICS 170
BQCZ16HIV 170
BQC04HIV 170
BQCW16HIV 171
BCC16HI 171
BQC04HI 172
IQTHI (ADC cell) 172
OCT04H (DAC cell) 172
BQC08HIV 173
BQC04HIVPECI 173
5.3 A/D & D/A CHARACTERISTICS 174
5.5 OPERATING CURRENT 175
5.6 PACKAGE THERMAL INFORMATION 175
5.7 AC ELECTRICAL CHARACTERISTICS 176
5.7.1 SPI Flash Timing 176
5.7.2 LPC interface Timing 177
5.7.3 PS/2 interface Timing 179
5.7.4 SMBus interface Timing 180
2. SMBUS frequencry dependant 180
5.7.5 PECI interface Timing 181
5.7.6 OWM interface Timing 182
6. PACKAGE INFORMATION 183

6.1 LQFP 128-PIN OUTLINE DIAGRAM 183
6.1.1 Top View.....183
6.1.2 Side View 184
6.1.3 Lead View.....185
6.1.4 LQFP Outline Dimensions 186
6.2 LFBGA 128-PIN OUTLINE DIAGRAM.....187
6.2.1 Top View.....187
6.2.2 Side View 188
6.2.3 Bottom View.....189
6.2.4 LFBGA Outline Dimensions 190
6.3 PART NUMBER DESCRIPTION 191

1. General Description

1.1 Overview

The ENE KB3930 is a highly customized embedded controller (EC) for notebook platforms. The embedded controller contains industrial standard 8051 microprocessor and provides function of i8042 keyboard controller basically. KB3930 is embedded LPC interface used to communicate with Host. KB3930 is designed with Shared-ROM architecture. The EC firmware and system BIOS will co-exist in single SPI flash. The embedded controller also features rich interfaces for general applications, such as PS/2 interface, Keyboard matrix encoder, PWM controller, A/D converter, D/A converter, Fan controller, SMBus controller, GPIO controller, PECI controller, one wire master, SPI controller, voltage comparator and extended interface (ENE Serial Bus) for more applications, like capacitive touch button application and GPIO extender.

Compared with last generation of KB3926 series, KB3930 added PECI/OWM, another 2 SMBus, another 2 Fan tachometers, enhanced SPI host/slave controller, voltage comparator, internal oscillator for newest application. KB3930 also improves structure of other modules including 8051, XBI, LPC, IKB, FAN, WDT, GPIO, ESB, EDI. For detail improvement, please refer the related section.

1.2 Features

LPC Low Pin Count Interface

- SIRQ supporting IRQ1, IRQ12, SCI or SMI# interrupt and one programmable IRQ provided.
- I/O Address Decoding:
  - Legacy KBC I/O port 60h/64h
  - Programmable EC I/O port, 62h/66h(recommend)
  - I/O port 68h/6Ch (sideband)
  - **2** Programmable 4-byte Index-I/O ports to access internal EC registers.
  - **1** Programmable extended (debug) port I/O.
- Memory Decoding:
  - Firmware Hub decode
  - LPC memory decode
- Compatible with LPC specification v1.1
- Support LPC interface re-direction to IKB for debugging

X-bus Bus Interface (XBI) : Flash Interface

- SPI flash is supported, size up to 4MB.
- SPI frequency supports 33/45/66MHz.
- New SPI command (dual read) to enhance the performance.
- The 64KB code memory can be mapped into system memory by one 16KB and one 48KB programmable pages independently.
- Support SPI flash in-system-programming via IKB pins.
- Enhanced pre-fetch mechanism.

8051 Microprocessor

- Compatible with industrial 8051 instructions with 3 cycles.
- 8051 runs at 8/16/22 MHz, programmable.
- 256 bytes internal RAM and 4KB tight-coupled SRAM.
- 24 extended interrupt sources.
- Two 16-bit timers.
- Full duplex UART integrated.
- Supports idle and stop mode.
- Enhanced ENE debug interface.
- Support Tx/Rx re-direction to IKB for debugging

8042 Keyboard Controller

- 8 standard 8042 commands processed by hardware.
- Each hardware command can be optionally processed by firmware.
- Pointing device multiplex mode support.
- Fast GA20 and KB reset support.

PS/2 Controller

- Support at most 3 external PS/2 devices.
- External PS/2 device operation in firmware mode.

Internal Keyboard Matrix (IKB)

- 18x8 keyboard scan matrix.
- Support W2K Internet and multimedia keys.
- Support hotkey events defined.
- Ghost key cancellation mechanism provided.
- Enhanced de-bounce feature added

Embedded Controller (EC)

- ACPI Spec 2.0 compliant.
- 5 standard EC command supported directly by hardware.
- Each hardware command can be processed by firmware optionally.
- Programmable EC I/O ports, 62h/66h by default.

SMBus Host Controller

- 4 SMBus interfaces with 2 SMBus controllers
- SMBus Spec 2.0 compliant.
- Byte mode support.
- Slave function support.

Digital-to-Analog Converter (DAC)

- 4 DAC channels with 8-bit resolution.
- All DAC pins can be alternatively configured as GPO (general purpose output) function.

Analog-to-Digital Converter (ADC)

- 6 ADC channels with 10-bit resolution.
- All ADC pins can be alternatively configured as GPI (general purpose input) function.

Pulse Width Modulator (PWM)

- 6 PWM channels are provided. (8-bit *2, 14-bit *2 and FANPWM(12-bit) *2)
- Clock source selectable:.
  - 1MHz/64KHz/4KHz/256Hz (for 8-bit PWM)
  - Peripheral clock or 1MHz (for 14-bit PWM)
  - Peripheral clock (for FANPWM)
- Duty cycle programmable and cycle time up to 1 sec(for 8-bit PWM)

Watch Dog Timer (WDT)

- 32.768KHz input clock.
- 10-bit counter with 32ms unit for watchdog reset.
- Three watchdog reset mechanisms.
  - Reset 8051 only
  - Reset whole chip, except GPIO module.
  - Reset whole chip including GPIO module.

Real Time Clock

- 32.768KHz input clock.
- 24-bit timer support.

General Purpose Timer (GPT)

- Two 16-bit and two 8-bit general purpose timer with 32.768KHz clock source.

General Purpose Wakeup (GPWU)

- Those I/O with GPI (general purpose input) configuration can generate interrupts or wakeup events, except those pins named in **GPXIOAxx**.

General Purpose Input/Output (GPIO)

- All general purpose I/O can be programmed as input or output.
- All output pins can be configured to be tri-state optionally.
- All input pins are equipped with pull-up, high/low active and edge/level trigger selection.
- For the pins of DAC can be configured as GPO function only.
- For the pins of ADC can be configured as GPI function only.
- A specific pair of GPIO pins with signal pass-through feature.

FAN Controller

- Two fan controllers with tachometer inputs.
- Automatic fan control support.
- 12-bit FANPWM support.
- Enhanced FAN tracking resolution added

Consumer IR (CIR)

- Several protocols decoded/encoded by hardware.
- Interrupt for CIR application.
- Support wide/narrow band receiver.
- Transmit/Receive simultaneously.
- Remote power-on support.

ENE Serial Bus Interface (ESB)

- A proprietary and flexible interface for extension with ENE KBC.
- Firmware accesses ESB devices via internal memory address directly.
- Interrupt capability.

ENE Debug Interface (EDI)

- Flexible debug interface with SPI pins.
- Keil-C development tool compatible

SPI Device Interface (SDI)

- An enhanced SPI host/device controller is embedded.
- Flexible design for SPI applications.

One Wire Master Interface

- Embedded One Wire controller used to control one wire devices.

PECI

- Support Intel PECI
- Support wide speed range from 2Kbps to 2Mbps.

Power Management

- Sleep mode: 8051 program counter (PC) stops and enters idle mode.
- Deep sleep mode: All clocks stop except external 32.768KHz OSC. 8051 enters stop mode.

Misc.

- Support two hardware voltage comparator (initialed by F/W, operated by H/W), two voltage input sources and one digital output, used to detect abnormal situation (like over temperature).
- Support two output pins to report KB3930 power fail status.

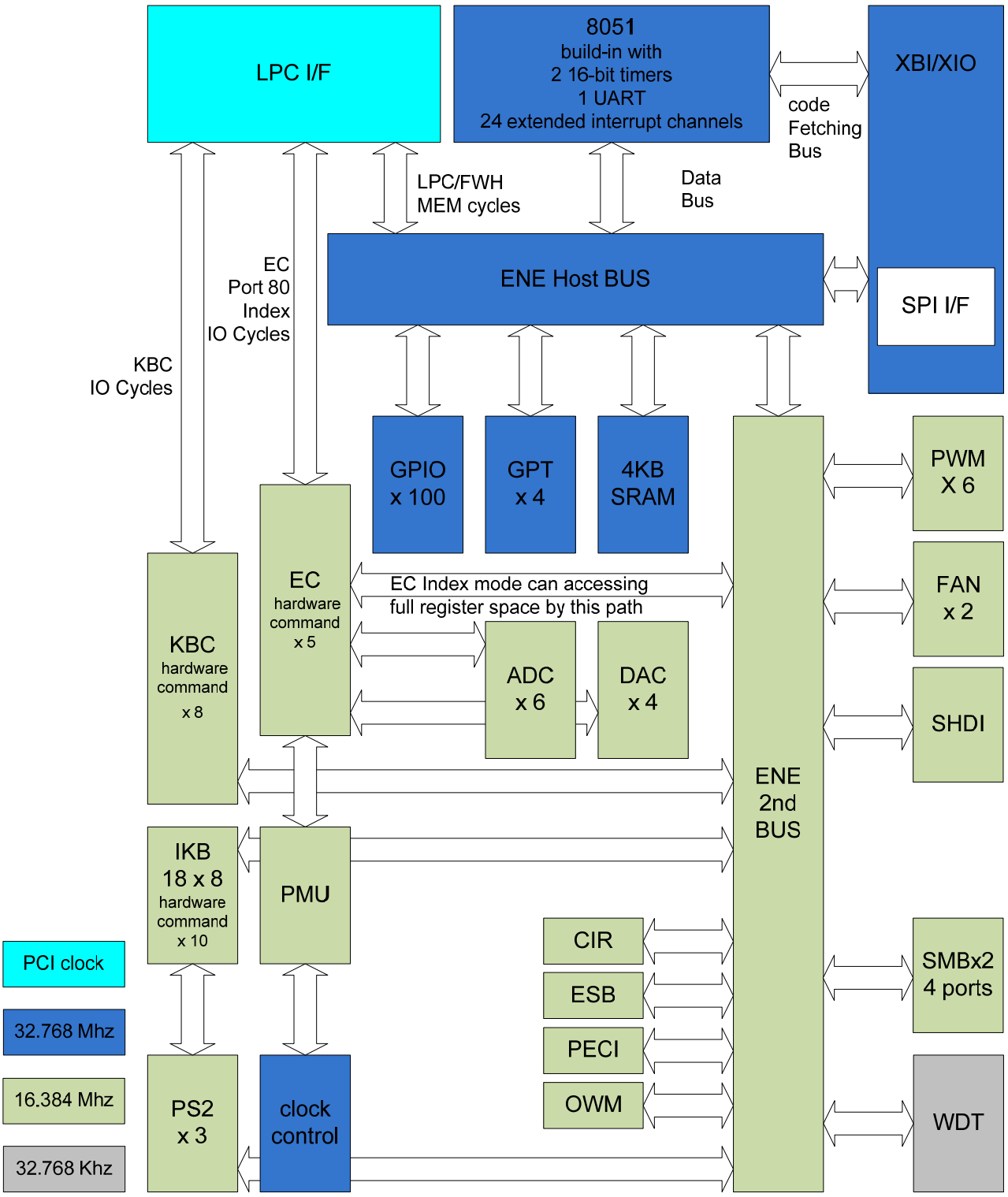
Package

- 128-pin LQFP package, Lead Free (RoHS).

1.3 Comparison (KB3926D vs. KB3930)

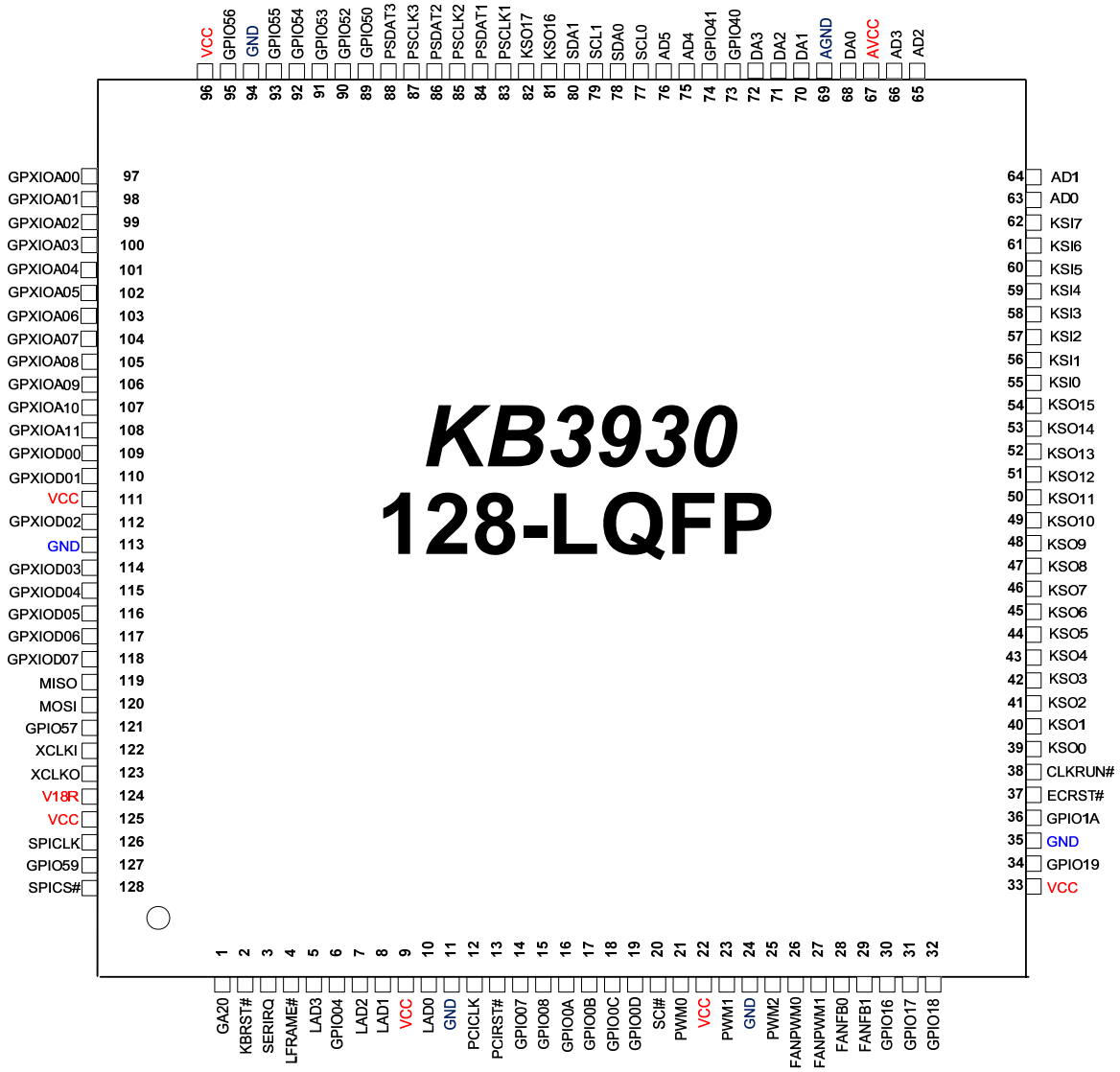
	KB3926D	KB3930
Microprocessor	8051	8051
Built-in SRAM	2KB	4KB
LPC	2 index-I/O sets	2 index I/O sets
X-Bus	SPI ROM: 4MB Enhanced pre-fetch mechanism.	SPI ROM: 4MB Enhanced pre-fetch mechanism.
Real Time Clock	Support	Support
ADC	Six 10-bit ADC channels	Six 10-bit ADC channels
DAC	Four 8-bit DAC channels	Four 8-bit DAC channels
WDT	128ms timer unit with 8bits control	32ms timer unit with 10bits control
PWM	6 sets PWM0/1 – 8 bit PWM2/3 – 14 bit FANPWM0/1 – 12 bit	6 sets PWM0/1 – 8 bit PWM2/3 – 14 bit FANPWM0/1 – 12 bit
External PS/2 I/F	3	3
GPIO	Programmable Bi-direction I/O GPIO pass through : 1 pair Max GPIO: 100	Enhanced Bi-direction I/O cell GPIO pass through : 1 pair Max GPIO: 100
IKB Matrix	18x8	18x8
FAN controller	2	2 (Enhanced precision and 2 additional Tachometer Monitors)
GPT	4	4
SMBus	2 Byte mode support	4 Byte mode support
CIR	Hardware encode/decode IRQ and I/O port support Carrier frequency calculation TX with carrier modulation Learning mode support TX/RX simultaneously	Hardware encode/decode IRQ and I/O port support Carrier frequency calculation TX with carrier modulation Learning mode support TX/RX simultaneously
EDI	Enhanced	Enhanced (Support break point)
ESB	Support	Support
SDI	Support	Support both SPI host/device
Package	128 LQFP	128 LQFP
Dimension	14mmx14mm	14mm x 14 mm
New-added Function		PECI One Wire Master POFR signals Voltage Comparator

1.4 Block Diagram



2. Pin Assignment and Description

2.1 KB3930 128-pin LQFP Diagram Top View



2.2 KB3930 128 LFBGA Ball Map

This page is leaved blank intentionally.

2.3 KB3930 Pin Assignment Side A

KB3930 Pin No.	KB3930 BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
1		GA20	GPIO00	GA20		GPIO00	HiZ / HiZ	BQC04HIV
2		KBRST#	GPIO01	KBRST#		GPIO01	HiZ / HiZ	BQC04HIV
3		SERIRQ					HiZ / HiZ	BCC16HI
4		LFRAME#					HiZ / HiZ	BCC16HI
5		LAD3					HiZ / HiZ	BCC16HI
6		GPIO04	GPIO04			GPIO04	HiZ / HiZ	BQC04HIV
7		LAD2					HiZ / HiZ	BCC16HI
8		LAD1					HiZ / HiZ	BCC16HI
9		VCC						VCC
10		LAD0					HiZ / HiZ	BCC16HI
11		GND						GND
12		PCICLK					IE/IE	BCC16HI
13		PCIRST#	GPIO05		PCIRST#	GPIO05	HiZ / IE	BCC16HI
14		GPIO07	GPIO07	i_clk_8051		GPIO07	HiZ / HiZ	BQC04HIV
15		GPIO08	GPIO08	i_clk_peri		GPIO08	HiZ / HiZ	BQC04HIV
16		GPIO0A	GPIO0A	OWM	RLC_RX2 OWM	GPIO0A	HiZ / HiZ	BQC04HIV
17		GPIO0B	GPIO0B	ESB_CLK		GPIO0B	PU / PU	BQCW16HIV
18		GPIO0C	GPIO0C	ESB_DAT	ESB_DAT	GPIO0C	HiZ / HiZ	BQC08HIV
19		GPIO0D	GPIO0D	RLC_TX2		GPIO0D	HiZ / HiZ	BQC04HIV
20		SCI#	GPIO0E	SCI#		GPIO0E	HiZ / HiZ	BQC04HIV
21		PWM0	GPIO0F	PWM0		GPIO0F	HiZ / HiZ	BQC16HIV
22		VCC	VCC					VCC
23		PWM1	GPIO10	PWM1		GPIO10	HiZ / HiZ	BQC04HIV
24		GND	GND					GND
25		GPIO11	GPIO11	PWM2		GPIO11	HiZ / HiZ	BQC04HIV
26		FANPWM0	GPIO12	FANPWM0		GPIO12	HiZ / HiZ	BQC04HIV
27		FANPWM1	GPIO13	FANPWM1		GPIO13	HiZ / HiZ	BQC04HIV
28		FANFB0	GPIO14		FANFB0	GPIO14	HiZ / HiZ	BQC04HIV
29		FANFB1	GPIO15		FANFB1	GPIO15	HiZ / HiZ	BQC04HIV
30		GPIO16	GPIO16	E51TXD		GPIO16	HiZ / HiZ	BQC04HIV
31		GPIO17	GPIO17	E51CLK	E51RXD	GPIO17	HiZ / HiZ	BQC04HIV
32		GPIO18	GPIO18			GPIO18	HiZ / HiZ	BQC04HIV

2.4 KB3930 Pin Assignment Side B

KB3930 Pin No.	KB3930 BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
33		VCC						VCC
34		GPIO19	GPIO19	PWM3		GPIO19	HiZ / HiZ	BCC16HI
35		GND						GND
36		GPIO1A	GPIO1A	NUMLED#		GPIO1A	HiZ / HiZ	BCC16HI
37		ECRST#					IE / IE	BQC04HIV
38		CLKRUN#	GPIO1D	CLKRUN#	CLKRUN#	GPIO1D	HiZ / HiZ	BCC16HI
39		KSO0	GPIO20	KSO0	TP_TEST	GPIO20	IE(PU)/IE(PU)	BQC04HIV
40		KSO1	GPIO21	KSO1	TP_PLL	GPIO21	IE(PU)/IE(PU)	BQC04HIV
41		KSO2	GPIO22	KSO2	TP_TMUX	GPIO22	IE(PU)/IE(PU)	BQC04HIV
42		KSO3	GPIO23	KSO3	TP_ISP	GPIO23	IE(PU)/IE(PU)	BQC04HIV
43		KSO4	GPIO24	KSO4		GPIO24	HiZ / HiZ	BQC04HIV
44		KSO5	GPIO25	KSO5	PCICLK (LPC)	GPIO25	HiZ / HiZ	BQCZ16HIV
45		KSO6	GPIO26	KSO6	PCIRST# (LPC)	GPIO26	HiZ / HiZ	BQC04HIV
46		KSO7	GPIO27	KSO7 SERIRQ(LPC)	SERIRQ(LPC)	GPIO27	HiZ / HiZ	BQC04HIV
47		KSO8	GPIO28	KSO8	LFRAME# (LPC)	GPIO28	HiZ / HiZ	BQC04HIV
48		KSO9	GPIO29	KSO9		GPIO29	HiZ / HiZ	BQCZ16HIV
49		KSO10	GPIO2A	KSO10		GPIO2A	HiZ / HiZ	BQCZ16HIV
50		KSO11	GPIO2B	KSO11 LAD3(LPC)	LAD3(LPC)	GPIO2B	HiZ / HiZ	BQC04HIV
51		KSO12	GPIO2C	KSO12 LAD2(LPC)	LAD2(LPC)	GPIO2C	HiZ / HiZ	BQC04HIV
52		KSO13	GPIO2D	KSO13 LAD1(LPC)	LAD1(LPC)	GPIO2D	HiZ / HiZ	BQC04HIV
53		KSO14	GPIO2E	KSO14 LAD0(LPC)	LAD0(LPC)	GPIO2E	HiZ / HiZ	BQC04HIV
54		KSO15	GPIO2F	KSO15	E51_RXD	GPIO2F	HiZ / HiZ	BQC04HIV
55		KSI0	GPIO30	E51_TXD	KSI0	GPIO30	IE(PU)/IE(PU)	BQC04HIV
56		KSI1	GPIO31		KSI1	GPIO31	IE(PU)/IE(PU)	BQC04HIV
57		KSI2	GPIO32		KSI2	GPIO32	IE(PU)/IE(PU)	BQC04HIV
58		KSI3	GPIO33		KSI3	GPIO33	IE(PU)/IE(PU)	BQC04HIV
59		KSI4	GPIO34		KSI4/EDI_CS	GPIO34	IE(PU)/IE(PU)	BQC04HIV
60		KSI5	GPIO35		KSI5/EDI_CLK	GPIO35	IE(PU)/IE(PU)	BQC04HIV
61		KSI6	GPIO36		KSI6/EDI_DIN	GPIO36	IE(PU)/IE(PU)	BQC04HIV
62		KSI7	GPIO37	EDI_DO	KSI7	GPIO37	IE(PU)/IE(PU)	BQC04HIV
63		AD0	**GPI**38		AD0	GPI38	HiZ / HiZ	IQTHI
64		AD1	**GPI**39		AD1	GPI39	HiZ / HiZ	IQTHI

2.5 KB3930 Pin Assignment Side C

KB3930 Pin No.	KB3930 BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
65		AD2	**GPI**3A		AD2	**GPI**3A	HiZ / HiZ	IQTHI
66		AD3	**GPI**3B		AD3	**GPI**3B	HiZ / HiZ	IQTHI
67		AVCC						AVCC
68		DA0	**GPO**3C	DA0		**GPO**3C	HiZ / HiZ	OCT04H
69		AGND						AGND
70		DA1	**GPO**3D	DA1		**GPO**3D	HiZ / HiZ	OCT04H
71		DA2	**GPO**3E	DA2		**GPO**3E	HiZ / HiZ	OCT04H
72		DA3	**GPO**3F	DA3		**GPO**3F	HiZ / HiZ	OCT04H
73		GPIO40	GPIO40		CIR_RX	GPIO40	HiZ / HiZ	BQC04HI
74		GPIO41	GPIO41	CIR_RLC_TX / PECI	PECI	GPIO41	HiZ / HiZ	BQC04HIVPECI
75		AD4	**GPI**42		AD4	**GPI**42	HiZ / HiZ	IQTHI
76		AD5	**GPI**43		AD5	**GPI**43	HiZ / HiZ	IQTHI
77		SCL0	GPIO44	SCL0		GPIO44	HiZ / HiZ	BQC04HI
78		SDA0	GPIO45	SDA0		GPIO45	HiZ / HiZ	BQC04HI
79		SCL1	GPIO46	SCL1		GPIO46	HiZ / HiZ	BQC04HI
80		SDA1	GPIO47	SDA1		GPIO47	HiZ / HiZ	BQC04HI
81		KSO16	GPIO48	KSO16		GPIO48	HiZ / HiZ	BQC04HIV
82		KSO17	GPIO49	KSO17		GPIO49	HiZ / HiZ	BQC04HIV
83		PSCLK1	GPIO4A	PSCLK1 / SCL2		GPIO4A	HiZ / HiZ	BQC04HI
84		PSDAT1	GPIO4B	PSDAT1 / SDA2		GPIO4B	HiZ / HiZ	BQC04HI
85		PSCLK2	GPIO4C	PSCLK2 / SCL3		GPIO4C	HiZ / HiZ	BCC16HI
86		PSDAT2	GPIO4D	PSDAT2 / SDA3		GPIO4D	HiZ / HiZ	BCC16HI
87		PSCLK3	GPIO4E	PSCLK3		GPIO4E	HiZ / HiZ	BQC04HI
88		PSDAT3	GPIO4F	PSDAT3		GPIO4F	HiZ / HiZ	BQC04HI
89		GPIO50	GPIO50			GPIO50	HiZ / HiZ	BQC04HI
90		GPIO52	GPIO52	E51CS#		GPIO52	HiZ / HiZ	BCC16HI
91		GPIO53	GPIO53	CAPSLED#	E51TMR1	GPIO53	HiZ / HiZ	BCC16HI
92		GPIO54	GPIO54	WDT_LED#	E51TMR0	GPIO54	HiZ / HiZ	BCC16HI
93		GPIO55	GPIO55	SCROLED#	E51INT0	GPIO55	HiZ / HiZ	BCC16HI
94		GND						GND
95		GPIO56	GPIO56		E51INT1	GPIO56	HiZ / HiZ	BQC04HIV
96		VCC						VCC

2.6 KB3930 Pin Assignment Side D

KB3930 Pin No.	KB3930 BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
97		GPXIOA00	GPXIOA00	SDICS#			HiZ / HiZ	BQC04HIV
98		GPXIOA01	GPXIOA01	SDICLK			HiZ / HiZ	BQC04HIV
99		GPXIOA02	GPXIOA02	SDIMOSI			HiZ / HiZ	BQC04HIV
100		GPXIOA03	GPXIOA03		FANFB2		HiZ / HiZ	BQC04HIV
101		GPXIOA04	GPXIOA04		FANFB3		HiZ / HiZ	BQC04HIV
102		GPXIOA05	GPXIOA05				HiZ / HiZ	BQC04HIV
103		GPXIOA06	GPXIOA06	VCOUT			HiZ / HiZ	BQC04HIV
104		GPXIOA07	GPXIOA07				HiZ / HiZ	BQC04HIV
105		GPXIOA08	GPXIOA08				HiZ / HiZ	BQCZ16HIV
106		GPXIOA09	GPXIOA09				HiZ / HiZ	BQCZ16HIV
107		GPXIOA10	GPXIOA10				HiZ / HiZ	BQCZ16HIV
108		GPXIOA11	GPXIOA11				HiZ / HiZ	BQCZ16HIV
109		GPXIOD00	GPXIOD00		SDIMISO VCIN0		HiZ / HiZ	BQC04HIV
110		GPXIOD01	GPXIOD01				HiZ / HiZ	BQC04HIV
111		VCC					HiZ / HiZ	VCC
112		GPXIOD02	GPXIOD02				HiZ / HiZ	BQC04HIV
113		GND					HiZ / HiZ	GND
114		GPXIOD03	GPXIOD03		VCIN1		HiZ / HiZ	BQC04HIV
115		GPXIOD04	GPXIOD04				HiZ / HiZ	BQC04HIV
116		GPXIOD05	GPXIOD05				HiZ / HiZ	BQC04HIV
117		GPXIOD06	GPXIOD06				HiZ / HiZ	BQC04HIV
118		GPXIOD07	GPXIOD07				HiZ / HiZ	BQC04HIV
119		MISO			MISO	MISO	HiZ / IE	BQCZ16HIV
120		MOSI		MOSI		MOSI	HiZ / Ox	BQCZ16HIV
121		GPIO57	GPIO57	XCLK32K		GPIO57	HiZ / HiZ	BQC04HIV
122		XCLKI						
123		XCLKO						
124		V18R						
125		VCC						VCC
126		SPICLK	GPIO58	SPICLK		SPICLK	HiZ / Ox	BQCW16HIV
127		GPIO59	GPIO59		TEST_CLK SPICLKI	GPIO59	IE / IE	BQC04HIV
128		SPICS#		SPICS#		SPICS#	HiZ / Ox	BQCZ16HIV

2.7 I/O Cell Descriptions

2.7.1 I/O Buffer Table

Cell	Description	Application
BQCZ16HIV	Schmitt trigger, 16mA Output / Sink Current, Input / Output / Pull Up Enable(40KΩ), 5 V Tolerance.	GPIO
BQC04HIV	Schmitt trigger, 4mA Output / Sink Current, Input / Output / Pull Up Enable(40KΩ), 5 V Tolerance	GPIO
BQCW16HIV	Schmitt trigger, 16mA Output / Sink Current, 5 V Tolerance, Input / Output / Pull Up Enable	ESB_CLK/ SPI_CLK
BCC16HI	16mA Output / Sink Current , 5 V Tolerance, Input / Output Enable	LPC I/F
BQC04HI	Schmitt trigger, 4mA Output / Sink Current, 5 V Tolerance, Input / Output Enable	GPIO
IQTHI	Mixed mode IO, ADC Enable, with GPI, Input Enable	ADC, GPI
OCT04H	Mixed mode IO, DAC Enable, with GPO, 4mA Output Current, Output Enable **(For GPO function, it is not recommended to control the device powered before KBC chip.)**	DAC, GPO
BQC08HIV	Schmitt trigger, 8mA Output / Sink Current, 5V Tolerance, Input / Output / Pull Up Enable	ESB_DAT
BQC04HIVPECI	Mixed Mode IO, PECI enable, with GPIO GPIO: Schmitt trigger, 4mA Output / Sink Current, PECI: 0.9V~1.2V	PECI, GPIO

*** 5V Tolerance, only if pull-high disable and output disable.**
**** Please note, the total current in each side on VCC or VSS of chip can not exceed over 48mA.**

2.7.2 I/O Buffer Characteristic Table

Cell	Output	Input	Analog Signal	Pull-High Enable(40k)	5V Tolerance	Current (mA)	Application
BQCZ16HIV	V	V		V	V	8~16	GPIO
BQC04HIV	V	V		V	V	2~4	GPIO
BQCW16HIV	V	V		V	V	8~16	ESB_CLK/ SPI_CLK
BCC16HI	V	V			V	8~16	LPC I/F
BQC04HI	V	V			V	2~4	GPIO
IQTHI		V	V				ADC, GPI
OCT04H	V		V			2~4	DAC, GPO
BQC08HIV	V	V		V	V	4~8	ESB_DAT
BQC04HIVPECI	V	V		V	V	2~4	PECI, GPIO

Application Notice: The Pads with I/O cells of IQTHI, OCT04H should be designed carefully. Under specific environment when: KBC is power-off, external application circuit is power-on. Signals must not be connected with pads of IQTHI/OCT04H (ADCs/DACs). It would cause unexpected voltage level on these pad if KBC is still power-off.

3. Pin Descriptions

3.1 Hardware Trap

Hardware trap pins are used to latch external signal at rising edge of **ECRST#**. The hardware trap pins are for some special purpose which should be defined while boot-up. The following table gives the collection of hardware trap pins. Please note, all the following hardware trap pins are **pull-high** internally after reset.

Trap Name	Pin No.	Description
TP_TEST (GPIO20,KSO0)	39	While this trap is asserted to be low, the internal DPLL circuit uses other clock source for reference, instead of 32KHz oscillator. **Low:** test clock mode enable **High:** normal mode using 32KHz oscillator.
TP_PLL (GPIO21,KSO1)	40	While this trap is asserted to be low, some DPLL related signals can be output for test. **Low:** DPLL test mode enable. **High:** DPLL test mode disable
TP_TMUX (GPIO22,KSO2)	41	TestMux Mode Trap **Low:** Test mode **High:** Normal operation
TP_ISP (GPIO23,KSO3)	42	While this trap is asserted to be low, SPI Flash can be programmed with ISP mode **Low:** SPI flash programming in ISP mode enable ＊ **High**: SPI flash programming in ISP mode disable

＊ Please note while TP_TMUX and TP_ISP keep low at the same time, a mechanism called **FlashDirectAccess** will enable. That is, users can flush and program a SPI flash via specific IKB pins with external tool.

FlashDirectAccess:

The KBC provides a new interface to program SPI flash via IKB interface. With this feature, users can easily utilize 4 pins from keyboard matrix (IKB) without disassembly whole machine. These 4 pins are connected directly to external SPI-Flash interface. The following table shows the mapped pins while entering FlashDirectAccess mode.

Pin No.	Normal Mode	FlashDirectAccess Mode
59	KSI4 (I)	(Input) EDI_CS, Transfer signal from terminal into KBC and though SPICS# to SPI_Flash
60	KSI5 (I)	(Input) EDI_CLK, Transfer signal from terminal into KBC and though SPICLK to SPI_Flash
61	KSI6 (I)	(Input) EDI_DIN, Transfer signal from terminal into KBC and though MOSI to SPI_Flash
62	KSI7 (I)	(Output) EDI_DO, Transfer signal from terminal into KBC and though MISO to SPI_Flash

3.2 Pin Descriptions by Functions

3.2.1 Low Pin Count I/F Descriptions.

Pin Name	Pin No.	Direction	Description
LAD[3:0]	5, 7,8,10	I/O	LPC address bus.
LFARAME#	4	I	LPC frame control signal.
PCIRST#	13	I	LPC module reset by this signal.
PCICLK	12	I	33MHz PCI clock input.
SERIRQ	3	I/O	Serial IRQ
CLKRUN#	38	I/OD	Clock run control

3.2.2 SPI Flash I/F Descriptions

Pin Name	Pin No.	Direction	Description
MISO	119	I	SPI read control signal
MOSI	120	O	SPI write control signal
SPICLK	126	O	SPI clock output
SPICS#	128	O	SPI chip select signal
These pins are input/output disable during reset phase.			

3.2.3 PS/2 I/F Descriptions

Pin Name	Pin No.	Direction	Description
PSCLK1	83	I/OD	PS/2 port 1 clock Muxed with SMBus port 2 clock
PSDAT1	84	I/OD	PS/2 port 1 data Muxed with SMBus port 2 data
PSCLK2	85	I/OD	PS/2 port 2 clock Muxed with SMBus port 3 clock
PSDAT2	86	I/OD	PS/2 port 2 data Muxed with SMBus port 3 data
PSCLK3	87	I/OD	PS/2 port 3 clock
PSDAT3	88	I/OD	PS/2 port 3 data

3.2.4 Internal Keyboard Encoder (IKB) Descriptions

Pin Name	Pin No.	Direction	Description
KSO[17:0]	82,81,54-39	O	Keyboard Scan Out
KSI[7:0]	62-55	I	Keyboard Scan In

3.2.5 SMBus Descriptions

Pin Name	Pin No.	Direction	Description
SCL0	77	I/OD	SMBus clock (interface 0)
SDA0	78	I/OD	SMBus data (interface 0)
SCL1	79	I/OD	SMBus clock (interface 1)
SDA1	80	I/OD	SMBus data (i`nterface 1)
SCL2	83	I/OD	SMBus clock (interface 2) Muxed with PS/2 port 1 clock
SDA2	84	I/OD	SMBus data (interface 2) Muxed with PS/2 port 1 data
SCL3	85	I/OD	SMBus clock (interface 3) Muxed with PS/2 port 2 clock
SDA3	86	I/OD	SMBus data (interface 3) Muxed with PS/2 port 2 data

3.2.6 FAN Descriptions

Pin Name	Pin No.	Direction	Description
FANPWM0	26	O	FANPWM0 output
FANPWM1	27	O	FANPWM1 output
FANFB0	28	I	FAN0 tachometer input
FANFB1	29	I	FAN1 tachometer input
FANFB2	100	I	FAN2 tachometer input
FANFB3	101	I	FAN3 tachometer input

3.2.7 Pulse Width Modulation (PWM) Descriptions

Pin Name	Pin No.	Direction	Description
PWM0	21	O	PWM pulse output
PWM1	23	O	PWM pulse output
PWM2	25	O	PWM pulse output
PWM3	34	O	PWM pulse output

3.2.8 Analog-to-Digital Converter Descriptions

Pin Name	Pin No.	Direction	Description
AD[3:0]	66-63	I	10bit A/D converter input
AD[5:4]	76,75	I	10bit A/D converter input

3.2.9 Digital-to-Analog Converter Descriptions

Pin Name	Pin No.	Direction	Description
DA[3:0]	72-70,68	O	8bit D/A converter output

3.2.10 8051 External I/F Descriptions

Pin Name	Pin No.	Direction	Description
E51TXD	30	O	8051 serial port, transmit port.
E51RXD	31	I	8051 serial port, receive port.
E51CLK	31	O	For different serial scheme, E51CLK will shift out clock.
E51CS#	90	O	
E51TMR0	92	I	
E51INT0	93	I	
E51TMR1	91	I	
E51INT1	95	I	

3.2.11 External Clock Descriptions

Pin Name	Pin No.	Direction	Description
XCLKI	122	I	32.768KHz input
XCLKO	123	O	32.768KHz output

3.2.12 Miscellaneous Signals Descriptions

Pin Name	Pin No.	Direction	Description
GA20	1	O	KBC will gate A20 address line
KBRST#	2	O	KBRST# is used to generate system reset.
SCI#	20	O	SCI# asserts to the system for requesting service while related events occur.
ECRST#	37	I	While ECRST# asserted, the KBC will reset globally.
OWM	16	I/O	One Wire Master input and output signal
PECI	74	I/O	PECI input and output signal

3.2.13 Voltage Comparator Pins Descriptions

Pin Name	Pin No.	Direction	Description
VCIN0	109	I	Voltage comparator input port0
VCIN1	114	I	Voltage comparator input port1
VCOUT	103	O	Voltage comparator output

3.2.14 Power Pins Descriptions

Pin Name	Pin No.	Direction	Description
VCC	9,22,33,96,111,125		Power supply for digital plane.
GND	11,24,35,94,113		Power ground for digital plane.
AVCC	67		Power supply for analog plane.
AGND	69		Power ground for analog plane.
V18R	124		Connected to external Capacitor for internal 1.8V

4. Module Descriptions

4.1 Chip Architecture

4.1.1 Power Planes

Two power planes are in the KBC. One is for digital logic and the other is for analog circuit. Both power planes are ±10% tolerance for recommend operation condition, The KBC provides V1.8 power plane for different generation.

Power Plane	Description	Power	Ground
Digital Plane	This power provides power for all digital logic no matter what power mode is.	VCC	GND
Analog Plane	This power provides power for all analog logic, such as A/D and D/A converter.	AVCC	AGND
Digital V1.8	The system inputs 3.3V power and the internal regulator outputs 1.8V voltage. The 1.8V output should connect a **capacitor** for stable purpose.	V1.8	GND

4.1.2 Clock Domains

Three clock sources, PCICLK, DPLL_CLK and XCLKI will be discussed in this section. A summary is list in the following table.

Clock	Description
PCICLK	PCI clock 33MHz for LPC I/F.
DPLL_CLK	Main clock for 8051/peripheral. DPLL clock can be generated with or without XCLK for reference. DPLL clock can be divided for different applications. Fig. 4-1 gives an example for illustration.
XCLKI	External 32.768KHz for reference.

The following figure shows more detail about the operation in the KBC. The external 32.768KHz is provided for two purposes. One is to provide an accurate reference for internal DPLL module, and the other one is to provide another clock source for watchdog timer.

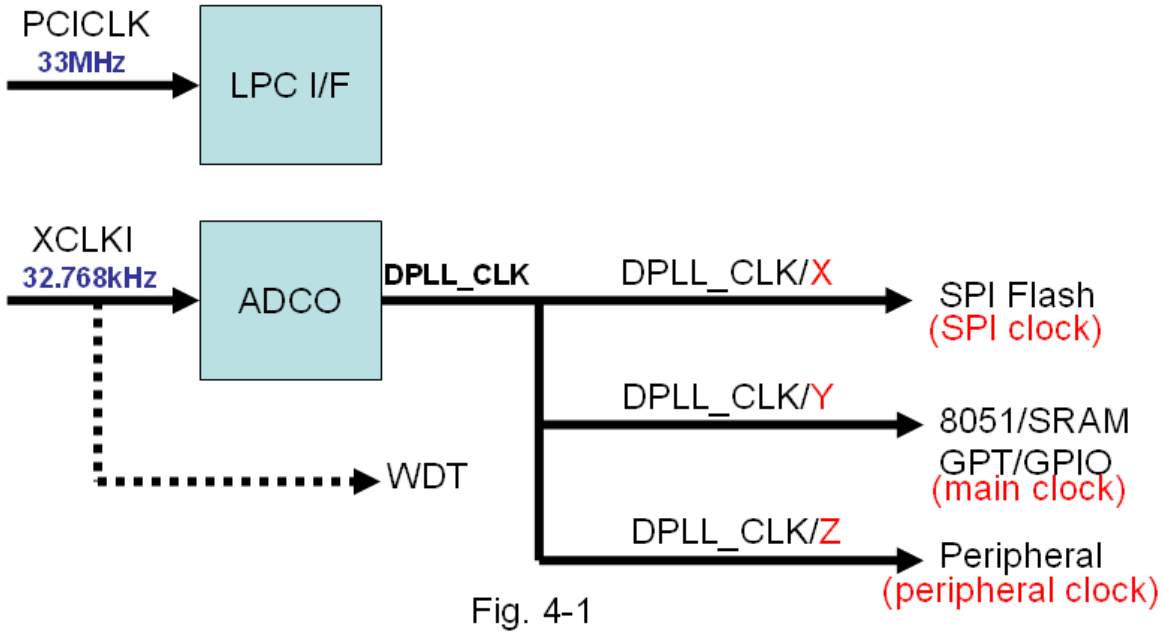


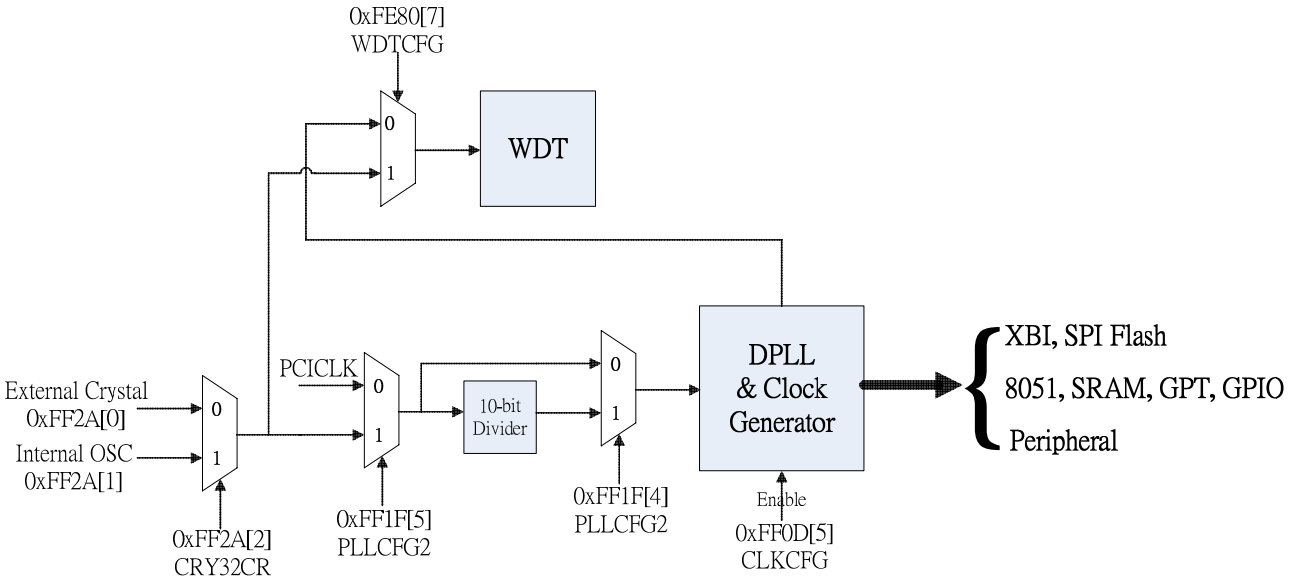
Fig. 4-1

The possible (X,Y,Z) combination with exact clock value is summarized as the following table.

	SPI Clock (X)		Main Clock (Y)		Peripheral Clock (Z)	
	CLKCFG[6]=0 (default)	CLKCFG[6]=1	CLKCFG[6]=0 (default)	CLKCFG[6]=1	CLKCFG[6]=0 (default)	CLKCFG[6]=1
CLKCFG[3:2]=0 (default)	16*	66	8*	8	4*	4
CLKCFG[3:2]=1	32	66	16	16	8	8
CLKCFG[3:2]=2	32	66	22	22	11	11
CLKCFG[3:2]=3	32	66	32	32	16	16
* While power on default, no matter what value CLKCFG[3:2], CLKCFG[6] are, the dividend (X,Y,Z) is always (4, 8, 16). The PCI clock is 66MHz, X= 66/4 = 16MHz, Y= 66/8 = 8Mhz , Z= 66/16 = 4MHz						
Be noted that, these clock frequency is only valid after KBC correctly referring clock.						

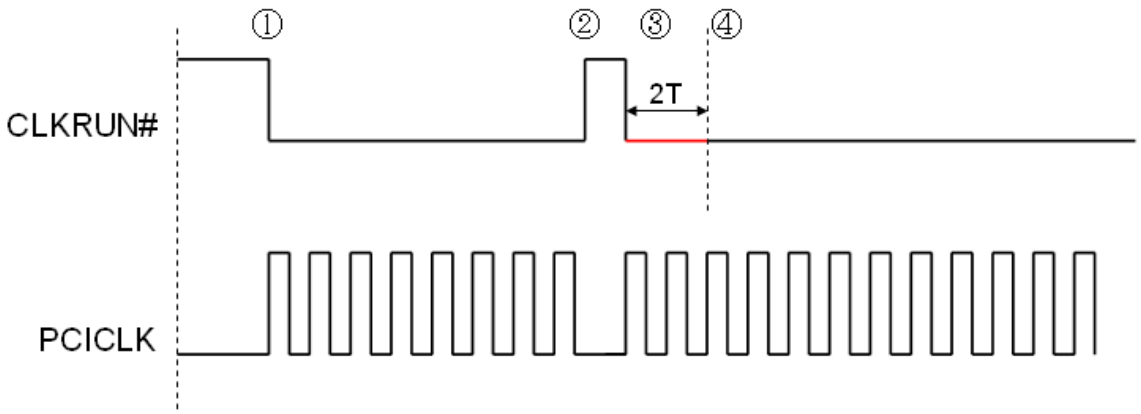
Note: Internal OSC of KBx930 application

Since KBx930 provide internal OSC, the clock source selection are different from KBx926D series. Developer could chose clock source from internal-OSC, external crystal, or host LPCLCK depending on different application and system status. As following is simplified clocking distribution tree for setting.



4.1.3 PCICLK and CLKRUN#

While system power-on, the host starts to drive CLKRUN# low for a while to inform the slaves that a 33MHz PCICLK will be given. At this moment, CLKRUN# of KBC is in input mode. If the host tries to stop the PCICLK for some purpose, the CLKRUN# will be de-asserted. In the current design, the KBC needs PCICLK for normal operation. Therefore the KBC keeps CLKRUN# for 2 clock cycles and releases it. This forces the host to start driving PCICLK. The following figure gives the explanation. For more detail please refer to *PCI Mobile Design Guide version 1.1*.



① Host asserts CLKRUN# and PCICLK is driven
② Host de-asserts CLKRUN# for some considerations
③ KBC monitors CLKRUN# de-asserting and then KBC keeps asserting CLKRUN#. This forces PCICLK keeping driving.
④ Host monitors CLKRUN# for 3T and sees the request from device. And then Host keeps CLKRUN# asserting.

4.1.4 Internal Memory Map

No	Module	Descriptions	Address Range	Size (Byte)	
1	Flash	Space mapped to system BIOS	0x0000~0xEBFF	59K	
2	XRAM	Embedded SRAM	0xEC00~0xFBFF	4K	
3	GPIO	General purpose I/O	0xFC00~0xFC7F	128	1K
4	KBC	Keyboard controller	0xFC80~0xFC8F	16	
5	ESB	ENE serial bus controller	0xFC90~0xFC9F	16	
6	IKB	Internal keyboard matrix	0xFCA0~0xFCAF	16	
7	RSV	Reserved	0xFCB0~0xFCBF	16	
8	RSV	Reserved	0xFCC0~0xFCCF	16	
9	PECI	PECI controller	0xFCD0~0xFCDF	16	
10	RSV	Reserved	0xFCE0~0xFCEF	16	
11	OWM	One Wire Master	0xFCF0~0xFCFF	16	
12	RSV	Reserved	0xFD00~0xFDFF	256	
13	PWM	Pulse width modulation	0xFE00~0xFE1F	32	
14	FAN	Fan controller	0xFE20~0xFE4F	48	
15	GPT	General purpose timer	0xFE50~0xFE6F	32	
16	SDIH/ SDID	SPI host interface/ SPI device interface	0xFE70~0xFE7F	16	
17	WDT	Watchdog timer	0xFE80~0xFE8F	16	
18	LPC	Low pin count interface	0xFE90~0xFE9F	16	
19	XBI	X-bus interface	0xFEA0~0xFEBF	32	
20	CIR	Consumer IR controller	0xFEC0~0xFECF	16	
21	RSV	Reserved	0xFED0~0xFEDFh	16	
22	PS2	PS/2 interface	0xFEE0~0xFEFF	32	
23	EC	Embedded controller	0xFF00~0xFF2F	48	
24	GPWU	General purpose wakeup event	0xFF30~0xFF7F	80	
25	SMBus	System management bus controller	0xFF80~0xFFBF	64	
26	RSV	Reserved	0xFFC0~0xFFCF	16	
27	RSV	Reserved	0xFFD0~0xFFFF	48	

4.2 GPIO

GPIOFSx is only for **Output Function Selection**, not for **Input Function**.
Example1 – GPIO14 is used as FANFB1, then
GPIO(GPIOFS10) 0xFC02 b'4 must be 0,
GPIO(GPIOIE10) 0xFC62 b'4 must be 1.
Example2 – PS/2 clock/data lines and SMBus clock/data are bi-directional.
They must be programmed as **Output Function Selection = 1** and **Input Enable = 1**.
For other specific GPIO initialization, please refer the SW programming guide of KBx930.

4.2.1 GPIO Function Description

The GPIO module is flexible for different applications. Each GPIO pin can be configured as alternative input or alternative output mode. The alternative function can be selected by register setting. A summary table is given as below for more detail.

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.
GPIO00	GA20		GPIO00	GPIOFS00.[0]
GPIO01	KBRST#		GPIO01	GPIOFS00.[1]
GPIO02*			GPIO02	GPIOFS00.[2]
GPIO03*			GPIO03	GPIOFS00.[3]
GPIO04			GPIO04	GPIOFS00.[4]
GPIO05		PCIRST#	GPIO05	GPIOFS00.[5]
GPIO06*			GPIO06	GPIOFS00.[6]
GPIO07	i_clk_8051		GPIO07	GPIOFS00.[7]
GPIO08	i_clk_peri		GPIO08	GPIOFS08.[0]
GPIO09*			GPIO09	GPIOFS08.[1]
GPIO0A	OWM	RLC_RX2 / OWM	GPIO0A	GPIOFS08.[2] OWMCFG[7]
GPIO0B	ESB_CLK		GPIO0B	GPIOFS08.[3]
GPIO0C	ESB_DAT	ESB_DAT	GPIO0C	GPIOFS08.[4]
GPIO0D	RLC_TX2		GPIO0D	GPIOFS08.[5]
GPIO0E	SCI#		GPIO0E	GPIOFS08.[6]
GPIO0F	PWM0		GPIO0F	GPIOFS08.[7]
GPIO10	PWM1		GPIO10	GPIOFS10.[0]
GPIO11	PWM2		GPIO11	GPIOFS10.[1]
GPIO12	FANPWM0		GPIO12	GPIOFS10.[2]
GPIO13	FANPWM1		GPIO13	GPIOFS10.[3]
GPIO14		FANFB0	GPIO14	GPIOFS10.[4]
GPIO15		FANFB1	GPIO15	GPIOFS10.[5]
GPIO16	E51TXD		GPIO16	GPIOFS10.[6]
GPIO17	E51CLK	E51RXD	GPIO17	GPIOFS10.[7]
GPIO18			GPIO18	GPIOFS18.[0]
GPIO19	PWM3		GPIO19	GPIOFS18.[1]
GPIO1A	NUMLED#		GPIO1A	GPIOFS18.[2]
GPIO1B*			GPIO1B	GPIOFS18.[3]
GPIO1C*			GPIO1C	GPIOFS18.[4]

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.
GPIO1D	CLKRUN#	CLKRUN#	GPIO1D	GPIOFS18.[5]
GPIO1E *			GPIO1E	GPIOFS18.[6]
GPIO1F *			GPIO1F	GPIOFS18.[7]
GPIO20	KSO00	TP_TEST	GPIO20	GPIOFS20.[0]
GPIO21	KSO01	TP_PLL	GPIO21	GPIOFS20.[1]
GPIO22	KSO02	TP_TMUX	GPIO22	GPIOFS20.[2]
GPIO23	KSO03	TP_ISP	GPIO23	GPIOFS20.[3]
GPIO24	KSO04		GPIO24	GPIOFS20.[4]
GPIO25	KSO05	PCICLK (LPC)	GPIO25	GPIOFS20.[5] GPIO_MISC2[7]
GPIO26	KSO06	PCIRST# (LPC)	GPIO26	GPIOFS20.[6] GPIO_MISC2[7]
GPIO27	KSO07 SERIRQ (LPC)	SERIRQ (LPC)	GPIO27	GPIOFS20.[7] GPIO_MISC2[7]
GPIO28	KSO08	LFRAME# (LPC)	GPIO28	GPIOFS28.[0] GPIO_MISC2[7]
GPIO29	KSO09		GPIO29	GPIOFS28.[1]
GPIO2A	KSO10		GPIO2A	GPIOFS28.[2]
GPIO2B	KSO11 LAD0 (LPC)	LAD0 (LPC)	GPIO2B	GPIOFS28.[3] GPIO_MISC2[7]
GPIO2C	KSO12 LAD1 (LPC)	LAD1 (LPC)	GPIO2C	GPIOFS28.[4] GPIO_MISC2[7]
GPIO2D	KSO13 LAD2 (LPC)	LAD2 (LPC)	GPIO2D	GPIOFS28.[5] GPIO_MISC2[7]
GPIO2E	KSO14 LAD3 (LPC)	LAD3 (LPC)	GPIO2E	GPIOFS28.[6] GPIO_MISC2[7]
GPIO2F	KSO15		GPIO2F	GPIOFS28.[7]
GPIO30		KSI0	GPIO30	GPIOFS30.[0]
GPIO31		KSI1	GPIO31	GPIOFS30.[1]
GPIO32		KSI2	GPIO32	GPIOFS30.[2]
GPIO33		KSI3	GPIO33	GPIOFS30.[3]
GPIO34		KSI4 / EDI_CS	GPIO34	GPIOFS30.[4]
GPIO35		KSI5 / EDI_CLK	GPIO35	GPIOFS30.[5]
GPIO36		KSI6 / EDI_DIN	GPIO36	GPIOFS30.[6]
GPIO37	EDI_DO	KSI7	GPIO37	GPIOFS30.[7]
GPI38		AD0	GPI38	GPIOFS38.[0]
GPI39		AD1	GPI39	GPIOFS38.[1]
GPI3A		AD2	GPI3A	GPIOFS38.[2]
GPI3B		AD3	GPI3B	GPIOFS38.[3]
GPO3C	DA0		GPO3C	GPIOFS38.[4] ★
GPO3D	DA1		GPO3D	GPIOFS38.[5] ★
GPO3E	DA2		GPO3E	GPIOFS38.[6] ★

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.
GPO3F	DA3		GPO3F	GPIOFS38.[7] ★
GPIO40		CIR_RX	GPIO40	GPIOFS40.[0]
GPIO41	CIR_RLC_TX / PECI	PECI	GPIO41	GPIOFS40.[1] GPIO_MISC2[0]
GPI42		AD4	GPI42	GPIOFS40.[2]
GPI43		AD5	GPI43	GPIOFS40.[3]
GPIO44	SCL0		GPIO44	GPIOFS40.[4]
GPIO45	SDA0		GPIO45	GPIOFS40.[5]
GPIO46	SCL1		GPIO46	GPIOFS40.[6]
GPIO47	SDA1		GPIO47	GPIOFS40.[7]
GPIO48	KSO16		GPIO48	GPIOFS48.[0]
GPIO49	KSO17		GPIO49	GPIOFS48.[1]
GPIO4A	PSCLK1 / SCL2		GPIO4A	GPIOFS48.[2] GPIO_MISC2[4]
GPIO4B	PSDAT1 / SDA2		GPIO4B	GPIOFS48.[3] GPIO_MISC2[4]
GPIO4C	PSCLK2 / SCL3		GPIO4C	GPIOFS48.[4] GPIO_MISC2[5]
GPIO4D	PSDAT2 / SDA3		GPIO4D	GPIOFS48.[5] GPIO_MISC2[5]
GPIO4E	PSCLK3		GPIO4E	GPIOFS48.[6]
GPIO4F	PSDAT3		GPIO4F	GPIOFS48.[7]
GPIO50			GPIO50	GPIOFS50.[0]
GPIO51 *			GPIO51	GPIOFS50.[1]
GPIO52	E51CS#		GPIO52	GPIOFS50.[2]
GPIO53	CAPSLED#	E51TMR1	GPIO53	GPIOFS50.[3]
GPIO54	WDT_LED#	E51TMR0	GPIO54	GPIOFS50.[4]
GPIO55	SCORLED#	E51INT0	GPIO55	GPIOFS50.[5]
GPIO56		E51INT1	GPIO56	GPIOFS50.[6]
GPIO57	XCLK32K		GPIO57	GPIOFS50.[7]
GPIO58	SPICLK		GPIO58	GPIOFS58.[0]
GPIO59		TEST_CLK/SPICLK	GPIO59	GPIOFS58.[1]
GPXIOA00	SDICS#			GPIO_MISC.[2]
GPXIOA01	SDICLK			GPIO_MISC.[2]
GPXIOA02	SDIMOSI			GPIO_MISC.[2]
GPXIOA03		FANFB2		FANTMCFG0[0]
GPXIOA04		FANFB3		FANTMCFG1[0]
GPXIOA05				
GPXIOA06	VOUT			GPX_MISC[0]
GPXIOA07				
GPXIOA08				
GPXIOA09				
GPXIOA10				
GPXIOA11				

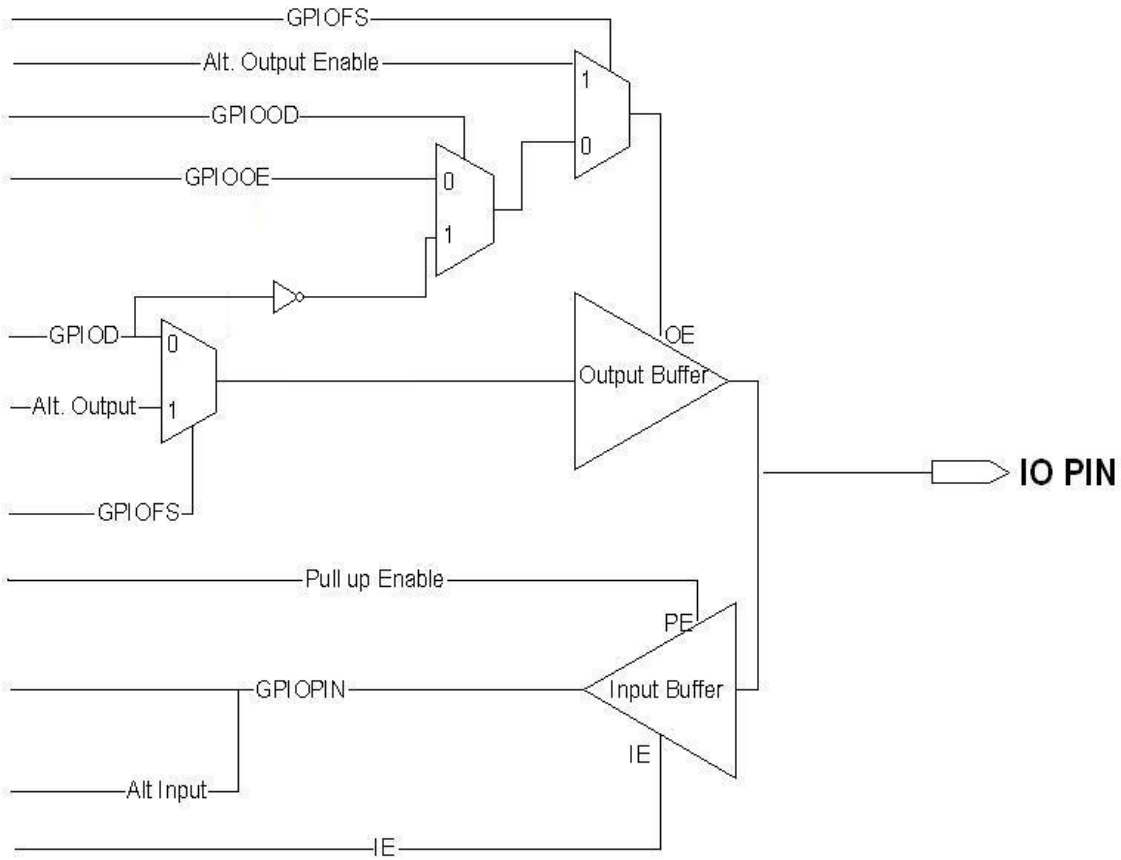
GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.
GPXIOD00		SDIMISO / VCIN0		VCCSR[0]
GPXIOD01				
GPXIOD02				
GPXIOD03		VCIN1		VCCSR[1]
GPXIOD04				
GPXIOD05				
GPXIOD06				
GPXIOD07				

＊ In KBx930, these GPIO pins no more exist. The corresponding register bits do not work.

★ If DAC function selected, please do not set this register bit.

4.2.2 GPIO Structures

In this section, the GPIO structure is illustrated as following diagram. The upper part is alternative output circuit and the lower part is alternative input circuit. In the figure, **GPIOFS** is used to enable alternative output. **GPIOOD** is for open-drain setting with output function. **GPIOOE** is the switch for data output. As shown in the figure, the alternative input embedded with pull-high and interrupt feature.



4.2.3 GPIO Attribution Table

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.	Input Enable	Output Enable	Pull Up (40KΩ)	Open Drain	Output Current
GPIO00	GA20		GPIO00	GPIOFS00.[0]	V	V	V	V	2-4mA
GPIO01	KBRST#		GPIO01	GPIOFS00.[1]	V	V	V	V	2-4mA
GPIO02*			GPIO02	GPIOFS00.[2]					
GPIO03*			GPIO03	GPIOFS00.[3]					
GPIO04			GPIO04	GPIOFS00.[4]	V	V	V	V	2-4mA
GPIO05		PCIRST#	GPIO05	GPIOFS00.[5]	V	V		V	8-16mA
GPIO06*			GPIO06	GPIOFS00.[6]					
GPIO07	i_clk_805)		GPIO07	GPIOFS00.[7]	V	V	V	V	2-4mA
GPIO08	i_clk_peri		GPIO08	GPIOFS08.[0]	V	V	V	V	2-4mA
GPIO09*			GPIO09	GPIOFS08.[1]					
GPIO0A	OWM	RLC_RX2 OWM	GPIO0A	GPIOFS08.[2] OWMCFG[7]	V	V	V	V	2-4mA
GPIO0B	ESB_CLK		GPIO0B	GPIOFS08.[3]	V	V	V	V	8-16mA
GPIO0C	ESB_DAT	ESB_DAT	GPIO0C	GPIOFS08.[4]	V	V	V	V	4~8mA
GPIO0D	RLC_TX2		GPIO0D	GPIOFS08.[5]	V	V	V	V	2-4mA
GPIO0E	SCI#		GPIO0E	GPIOFS08.[6]	V	V	V	V	2-4mA
GPIO0F	PWM0		GPIO0F	GPIOFS08.[7]	V	V	V	V	8-16mA
GPIO10	PWM1		GPIO10	GPIOFS10.[0]	V	V	V	V	2-4mA
GPIO11	PWM2		GPIO11	GPIOFS10.[1]	V	V	V	V	2-4mA
GPIO12	FANPWM0		GPIO12	GPIOFS10.[2]	V	V	V	V	2-4mA
GPIO13	FANPWM1		GPIO13	GPIOFS10.[3]	V	V	V	V	2-4mA
GPIO14		FANFB0	GPIO14	GPIOFS10.[4]	V	V	V	V	2-4mA
GPIO15		FANFB1	GPIO15	GPIOFS10.[5]	V	V	V	V	2-4mA
GPIO16	E51TXD		GPIO16	GPIOFS10.[6]	V	V	V	V	2-4mA
GPIO17	E51CLK	E51RXD	GPIO17	GPIOFS10.[7]	V	V	V	V	2-4mA
GPIO18			GPIO18	GPIOFS18.[0]	V	V	V	V	2-4mA
GPIO19	PWM3		GPIO19	GPIOFS18.[1]	V	V		V	8-16mA
GPIO1A	NUMLED#		GPIO1A	GPIOFS18.[2]	V	V		V	8-16mA
GPIO1B*			GPIO1B	GPIOFS18.[3]					
GPIO1C*			GPIO1C	GPIOFS18.[4]					
GPIO1D	CLKRUN#	CLKRUN#	GPIO1D	GPIOFS18.[5]	V	V		V	8-16mA
GPIO1E*			GPIO1E	GPIOFS18.[6]					
GPIO1F*			GPIO1F	GPIOFS18.[7]					
GPIO20	KSO00	TP_TEST	GPIO20	GPIOFS20.[0]	V	V	V	V	2-4mA
GPIO21	KSO01	TP_PLL	GPIO21	GPIOFS20.[1]	V	V	V	V	2-4mA
GPIO22	KSO02	TP_TMUX	GPIO22	GPIOFS20.[2]	V	V	V	V	2-4mA
GPIO23	KSO03	TP_ISP	GPIO23	GPIOFS20.[3]	V	V	V	V	2-4mA
GPIO24	KSO04		GPIO24	GPIOFS20.[4]	V	V	V	V	2-4mA
GPIO25	KSO05	PCICLK(LPC)	GPIO25	GPIOFS20.[5] GPIO_MISC2[7]	V	V	V	V	8-16mA
GPIO26	KSO06	PCIRST#(LPC)	GPIO26	GPIOFS20.[6] GPIO_MISC2[7]	V	V	V	V	2-4mA
GPIO27	KSO07 / SERIRQ(LPC)	SERIRQ(LPC)	GPIO27	GPIOFS20.[7] GPIO_MISC2[7]	V	V	V	V	2-4mA
GPIO28	KSO08	LFRAME#(LPC)	GPIO28	GPIOFS28.[0] GPIO_MISC2[7]	V	V	V	V	2-4mA
GPIO29	KSO09		GPIO29	GPIOFS28.[1]	V	V	V	V	8-16mA
GPIO2A	KSO10		GPIO2A	GPIOFS28.[2]	V	V	V	V	8-16mA

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.	Input Enable	Output Enable	Pull Up (40KΩ)	Open Drain	Output Current
GPIO2B	KSO11 / LAD0(LPC)	LAD0(LPC)	GPIO2B	GPIOFS28.[3] GPIO_MISC2[7]	V	V	V	V	2-4mA
GPIO2C	KSO12 / LAD1(LPC)	LAD1(LPC)	GPIO2C	GPIOFS28.[4] GPIO_MISC2[7]	V	V	V	V	2-4mA
GPIO2D	KSO13 / LAD2(LPC)	LAD2(LPC)	GPIO2D	GPIOFS28.[5] GPIO_MISC2[7]	V	V	V	V	2-4mA
GPIO2E	KSO14 / LAD3(LPC)	LAD3(LPC)	GPIO2E	GPIOFS28.[6] GPIO_MISC2[7]	V	V	V	V	2-4mA
GPIO2F	KSO15		GPIO2F	GPIOFS28.[7]	V	V	V	V	2-4mA
GPIO30		KSI0	GPIO30	GPIOFS30.[0]	V	V	V	V	2-4mA
GPIO31		KSI1	GPIO31	GPIOFS30.[1]	V	V	V	V	2-4mA
GPIO32		KSI2	GPIO32	GPIOFS30.[2]	V	V	V	V	2-4mA
GPIO33		KSI3	GPIO33	GPIOFS30.[3]	V	V	V	V	2-4mA
GPIO34		KSI4 / EDI_CS	GPIO34	GPIOFS30.[4]	V	V	V	V	2-4mA
GPIO35		KSI5 / EDI_CLK	GPIO35	GPIOFS30.[5]	V	V	V	V	2-4mA
GPIO36		KSI6 / EDI_DIN	GPIO36	GPIOFS30.[6]	V	V	V	V	2-4mA
GPIO37	EDI_DO	KSI7	GPIO37	GPIOFS30.[7]	V	V	V	V	2-4mA
GPI38		AD0		GPIOFS38.[0]	V				
GPI39		AD1		GPIOFS38.[1]	V				
GPI3A		AD2		GPIOFS38.[2]	V				
GPI3B		AD3		GPIOFS38.[3]	V				
GPO3C	DA0		GPO3C	GPIOFS38.[4]		V			2-4mA
GPO3D	DA1		GPO3D	GPIOFS38.[5]		V			2-4mA
GPO3E	DA2		GPO3E	GPIOFS38.[6]		V			2-4mA
GPO3F	DA3		GPO3F	GPIOFS38.[7]		V			2-4mA
GPIO40		CIR_RX	GPIO40	GPIOFS40.[0]	V	V		V	2-4mA
GPIO41	CIR_RLC_TX / PECI	PECI	GPIO41	GPIOFS40.[1] GPIO_MISC2[0]	V	V	V	V	2-4mA
GPI42		AD4		GPIOFS40.[2]	V				2-4mA
GPI43		AD5		GPIOFS40.[3]	V				2-4mA
GPIO44	SCL0		GPIO44	GPIOFS40.[4]	V	V		V	2-4mA
GPIO45	SDA0		GPIO45	GPIOFS40.[5]	V	V		V	2-4mA
GPIO46	SCL1		GPIO46	GPIOFS40.[6]	V	V		V	2-4mA
GPIO47	SDA1		GPIO47	GPIOFS40.[7]	V	V		V	2-4mA
GPIO48	KSO16 /		GPIO48	GPIOFS48.[0]	V	V	V	V	2-4mA
GPIO49	KSO17		GPIO49	GPIOFS48.[1]	V	V	V	V	2-4mA
GPIO4A	PSCLK1 / SCL2		GPIO4A	GPIOFS48.[2] GPIO_MISC2[4]	V	V		V	2-4mA
GPIO4B	PSDAT1 / SDA2		GPIO4B	GPIOFS48.[3] GPIO_MISC2[4]	V	V		V	2-4mA
GPIO4C	PSCLK2 / SCL3		GPIO4C	GPIOFS48.[4] GPIO_MISC2[5]	V	V		V	8-16mA
GPIO4D	PSDAT2 / SDA3		GPIO4D	GPIOFS48.[5] GPIO_MISC2[5]	V	V		V	8-16mA
GPIO4E	PSCLK3		GPIO4E	GPIOFS48.[6]	V	V		V	2-4mA
GPIO4F	PSDAT3		GPIO4F	GPIOFS48.[7]	V	V		V	2-4mA
GPIO50			GPIO50	GPIOFS50.[0]	V	V		V	2-4mA
GPIO51*			GPIO51	GPIOFS50.[1]					
GPIO52	E51CS#		GPIO52	GPIOFS50.[2]	V	V		V	8-16mA

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.	Input Enable	Output Enable	Pull Up (40KΩ)	Open Drain	Output Current
GPIO53	CAPSLED#	E51TMR1	GPIO53	GPIOFS50.[3]	V	V		V	8-16mA
GPIO54	WDT_LED#	E51TMR0	GPIO54	GPIOFS50.[4]	V	V		V	8-16mA
GPIO55	SCORLED#	E51INT0	GPIO55	GPIOFS50.[5]	V	V		V	8-16mA
GPIO56		E51INT1	GPIO56	GPIOFS50.[6]	V	V	V	V	2-4mA
GPIO57	XCLK32K		GPIO57	GPIOFS50.[7]	V	V	V	V	2-4mA
GPIO58	SPICLK		GPIO58	GPIOFS58.[0]	V	V	V	V	8-16mA
GPIO59		TEST_CLK/ SPICLK	GPIO59	GPIOFS58.[1]	V	V	V	V	2-4mA
GPXIOA00	SDICS#			GPIO_MISC.[2]	V	V	V		2-4mA
GPXIOA01	SDICLK			GPIO_MISC.[2]	V	V	V		2-4mA
GPXIOA02	SDIMOSI			GPIO_MISC.[2]	V	V	V		2-4mA
GPXIOA03		FANFB2		FANTMCFG0[0]	V	V	V		2-4mA
GPXIOA04		FANFB3		FANTMCFG1[0]	V	V	V		2-4mA
GPXIOA05	VCOUT			GPX_MISC[0]	V	V	V		2-4mA
GPXIOA06					V	V	V		2-4mA
GPXIOA07					V	V	V		2-4mA
GPXIOA08					V	V	V		8-16mA
GPXIOA09					V	V	V		8-16mA
GPXIOA10					V	V	V		8-16mA
GPXIOA11					V	V	V		8-16mA
GPXIOD00		SDIMISO / VCIN0		VCCSR[0]	V	V	V		2-4mA
GPXIOD01					V	V	V		2-4mA
GPXIOD02					V	V	V		2-4mA
GPXIOD03		/ VCIN1		VCCSR[1]	V	V	V		2-4mA
GPXIOD04					V	V	V		2-4mA
GPXIOD05					V	V	V		2-4mA
GPXIOD06					V	V	V		2-4mA
GPXIOD07					V	V	V		2-4mA

＊ Denotes that these pins do not exist in KBx930

4.2.3 GPIO Registers Descriptions

Function Selection Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x00	GPIOFS00	R/W	GPIO00~GPIO07 Function Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x01	GPIOFS08	R/W	GPIO08~GPIO0F Function Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x02	GPIOFS10	R/W	GPIO10~GPIO17 Function Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x03	GPIOFS18	R/W	GPIO18~GPIO1F Function Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x04	GPIOFS20	R/W	GPIO20~GPIO27 Function Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x05	GPIOFS28	R/W	GPIO28~GPIO2F Function Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x06	GPIOFS30	R/W	GPIO30~GPIO37 Function Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x07	GPIOFS38	R/W	GPIO3C~GPIO3F Function Selection bit[4]~bit[7] stand for GPIO3C~GPIO3F separately **0:** General purpose output function selected **1:** Alternative output function selected. **GPI38~GPI3B without alternative output function.*	0x00	0xFC
0x08	GPIOFS40	R/W	GPIO40~41, 44~47 Function Selection bit[0:1], bit[4:7] stand for GPIO40~41, 44~47 separately **0:** General purpose output function selected **1:** Alternative output function selected. **GPI42~GPI43 without alternative output function.*	0x00	0xFC
0x09	GPIOFS48	R/W	GPIO48~GPIO4F Function Selection bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x0A	GPIOFS50	R/W	GPIO50~GPIO57 Function Selection bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x02	0xFC
0x0B	GPIOFS58	R/W	GPIO58~GPIO59 Function Selection bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC

Output Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x10	GPIOOE00	R/W	GPIO00~GPIO07 Output Enable bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x11	GPIOOE08	R/W	GPIO08~GPIO0F Output Enable bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x12	GPIOOE10	R/W	GPIO10~GPIO17 Output Enable bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x13	GPIOOE18	R/W	GPIO18~GPIO1F Output Enable bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x14	GPIOOE20	R/W	GPIO20~GPIO27 Output Enable bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x15	GPIOOE28	R/W	GPIO28~GPIO2F Output Enable bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x16	GPIOOE30	R/W	GPIO30~GPIO37 Output Enable bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x17	GPIOOE38	R/W	GPIO3C~GPIO3F Output Enable bit[4]~bit[7] stand for GPIO3C~GPIO3F separately **0:** Output Disable **1:** Output Enable ** GPI38~GPI3A without output enable feature.*	0x00	0xFC
0x18	GPIOOE40	R/W	GPIO40~41 , 44~47 Output Enable bit[0:1], bit[4:7] stand for GPIO40~1, 44~47 separately **0:** Output Disable **1:** Output Enable **GPI42~GPI43 without output enable.*	0x00	0xFC
0x19	GPIOOE48	R/W	GPIO48~GPIO4F Output Enable bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1A	GPIOOE50	R/W	GPIO50~GPIO57 Output Enable bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Output Disable **1:** Output Enable	0x02	0xFC
0x1B	GPIOOE58	R/W	GPIO58~GPIO59 Output Enable bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1C	GPXAOE00	R/W	GPXIOA00~GPXIOA07 Output Enable bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC

Output Enable Register (Continued)					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x1D	GPXAOE08	R/W	GPXIOA08~GPXIOA11 Output Enable bit[0]~bit[3] stand for GPXIOA08~GPXIOA11 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1E	RSV	RSV	Reserved	RSV	0xFC
0x1F	GPXDOE00	R/W	GPXIOD00~GPXIOD07 Output Enable bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC

Output Data Port Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x20	GPIOD00	R/W	GPIO00~GPIO07 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO00~GPIO07 separately	0x00	0xFC
0x21	GPIOD08	R/W	GPIO08~GPIO0F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO08~GPIO0F separately	0x00	0xFC
0x22	GPIOD10	R/W	GPIO10~GPIO17 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO10~GPIO17 separately	0x00	0xFC
0x23	GPIOD18	R/W	GPIO18~GPIO1F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO18~GPIO1F separately	0x00	0xFC
0x24	GPIOD20	R/W	GPIO20~GPIO27 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO20~GPIO27 separately	0x00	0xFC
0x25	GPIOD28	R/W	GPIO28~GPIO2F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO28~GPIO2F separately	0x00	0xFC
0x26	GPIOD30	R/W	GPIO30~GPIO37 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO30~GPIO37 separately	0x00	0xFC
0x27	GPIOD38	R/W	GPIO3C~GPIO3F Output Data Port for output function. Bit[4]~bit[7] stand for GPIO3C~GPIO3F separately ** GPI38~GPI3B have no output data ports.*	0x00	0xFC
0x28	GPIOD40	R/W	GPIO40~41, 44~47 Output Data Port for output function. Bit[0:1],bit[4:7] stand for GPIO40~41, 44~47 separately ** GPI42~GPI43 have no output data ports.*	0x00	0xFC
0x29	GPIOD48	R/W	GPIO48~GPIO4F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO48~GPIO4F separately	0x00	0xFC
0x2A	GPIOD50	R/W	GPIO50~GPIO57 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO50~GPIO57 separately	0x00	0xFC
0x2B	GPIOD58	R/W	GPIO58~GPIO59 Output Data Port for output function. Bit[0]~bit[1] stand for GPIO58~GPIO59 separately	0x00	0xFC
0x2C	GPXAD00	R/W	GPXIOA00~GPXIOA07 Output Data Port for output function. Bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately	0x00	0xFC
0x2D	GPXAD08	R/W	GPXIOA08~GPXIOA11 Output Data Port for output function. Bit[0]~bit[3] stand for GPXIOA08~GPXIOA11 separately	0x00	0xFC
0x2E	RSV	RSV	Reserved	RSV	0xFC
0x2F	GPXDD00	R/W	GPXIOD00~GPXIOD07 Output Data Port for output function. Bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately	0x00	0xFC

Input Data Port Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x30	GPIOIN00	R	GPIO00~GPIO07 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO00~GPIO07 separately	0xFF	0xFC
0x31	GPIOIN08	R	GPIO08~GPIO0F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO08~GPIO0F separately	0xFF	0xFC
0x32	GPIOIN10	R	GPIO10~GPIO17 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO10~GPIO17 separately	0xFF	0xFC
0x33	GPIOIN18	R	GPIO18~GPIO1F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO18~GPIO1F separately	0xFF	0xFC
0x34	GPIOIN20	R	GPIO20~GPIO27 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO20~GPIO27 separately	0xFF	0xFC
0x35	GPIOIN28	R	GPIO28~GPIO2F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO28~GPIO2F separately	0xFF	0xFC
0x36	GPIOIN30	R	GPIO30~GPIO37 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO30~GPIO37 separately	0xFF	0xFC
0x37	GPIOIN38	R	GPIO38~GPIO3B Input Data Port for input function. Bit[0]~bit[3] stand for GPIO38~GPIO3B separately ** GPO3C~GPO3F have no input data ports.*	0x0F	0xFC
0x38	GPIOIN40	R	GPIO40~GPIO47 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO40~GPIO47 separately	0xFF	0xFC
0x39	GPIOIN48	R	GPIO48~GPIO4F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO48~GPIO4F separately	0xFF	0xFC
0x3A	GPIOIN50	R	GPIO50~GPIO57 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO50~GPIO57 separately	0xFF	0xFC
0x3B	GPIOIN58	R	GPIO58~GPIO59 Input Data Port for input function. Bit[0]~bit[1] stand for GPIO58~GPIO59 separately	0x01	0xFC
0x3C	GPXAIN00	R	GPXIOA00~GPXIOA07 Input Data Port for input function. Bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately	0xFF	0xFC
0x3D	GPXAIN08	R	GPXIOA08~GPXIOA11 Input Data Port for input function. Bit[0]~bit[3] stand for GPXIOA08~GPXIOA11 separately	0xFF	0xFC
0x3E	RSV	RSV	Reserved	RSV	0xFC
0x3F	GPXDIN00	R	GPXIOD00~GPXIOD07 Input Data Port for input function. Bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately	0xFF	0xFC

Pull-up Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x40	GPIOPU00	R/W	GPIO00~04, 06~07 Internal Pull-Up Resistor Enable for input function bit[0:4],bit[6:7] stand for GPIO00~04, 06~07 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable * GPIO05 (bit 5)do not exist internal pull-up resistor	0x00	0xFC
0x41	GPIOPU08	R/W	GPIO08~GPIO0F Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable *The ESB_CLK Pull-Up is changed to default off.*	0x00	0xFC
0x42	GPIOPU10	R/W	GPIO10~GPIO17 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x43	GPIOPU18	R/W	GPIO18, 1B~1C, 1E~1F Internal Pull-Up Resistor Enable for input function bit[0], bit[3:4], bit[6:7] stand for GPIO18, 1B~1C, 1E~1F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable. * GPIO19/1A/1D (bit 1/2/5)do not exist internal pull-up resistor	0x00	0xFC
0x44	GPIOPU20	R/W	GPIO20~GPIO27 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x0F	0xFC
0x45	GPIOPU28	R/W	GPIO28~GPIO2F Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x46	GPIOPU30	R/W	GPIO30~GPIO37 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0xFF	0xFC
0x47	RSV	RSV	Reserved		0xFC
0x48	GPIOPU40	R/W	GPIO41 Internal Pull-Up Resistor Enable for input function bit[1] stand for GPIO41 **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x49	GPIOPU48	R/W	GPIO48~GPIO49 Internal Pull-Up Resistor Enable for input function bit[0]~bit[1] stand for GPIO48~GPIO49 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable.	0x00	0xFC
0x4A	GPIOPU50	R/W	GPIO56/57 Internal Pull-Up Resistor Enable for input function bit[6]~bit[7] stand for GPIO56~57 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable.	0x00	0xFC
0x4B	GPIOPU58	R/W	GPIO58~GPIO59 Internal Pull-Up Resistor Enable for input function bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable.	0x00	0xFC

Pull-up Enable Register (Continued)					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x4C	GPXAPU00	R/W	GPXIOA00~ GPXIOA07 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPXIO00~ GPXIO07 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x4D	GPXAPU08	R/W	GPXIOA08~ GPXIOA11 Internal Pull-Up Resistor Enable for input function bit[0]~bit[3] stand for GPXIOA08~ GPXIOA11 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x4E	RSV	RSV	Reserved	RSV	0xFC
0x4F	GPXDPU00	R/W	GPXIOD00~ GPXIOD07 Internal Pull-Up Resistor Enable for input function bit[0]~bit[1] stand for GPXIOD00~ GPXIOD07 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC

Open Drain Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x50	GPIOOD00	R/W0C	GPIO00~GPIO07 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x51	GPIOOD08	R/W0C	GPIO08~GPIO0F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x52	GPIOOD10	R/W0C	GPIO10~GPIO17 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x53	GPIOOD18	R/W0C	GPIO18~GPIO1F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x54	GPIOOD20	R/W0C	GPIO20~GPIO27 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x55	GPIOOD28	R/W0C	GPIO28~GPIO2F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x56	GPIOOD30	R/W0C	GPIO30~GPIO37 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x57	RSV	RSV	RSV		0xFC
0x58	GPIOOD40	R/W0C	GPIO40~41, 44~47 Open Drain Enable for output function bit[0:1], bit[4:7] stand for GPIO40~41, 44~47 separately **0:** Open drain disable **1:** Open drain enable. * GPI42/43 do not exist open drain function	0x00	0xFC
0x59	GPIOOD48	R/W0C	GPIO48~GPIO4F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x5A	GPIOOD50	R/W0C	GPIO50~GPIO57 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x5B	GPIOOD58	R/W0C	GPIO58~GPIO59 Open Drain Enable for output function bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC

Input Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x60	GPIOIE00	R/W	GPIO00~GPIO07 Input Enable for input function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x20	0xFC
0x61	GPIOIE08	R/W	GPIO08~GPIOF Input Enable for input function bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x62	GPIOIE10	R/W	GPIO10~GPIO17 Input Enable for input function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x63	GPIOIE18	R/W	GPIO18~GPIO1F Input Enable for input function bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x64	GPIOIE20	R/W	GPIO20~GPIO27 Input Enable for input function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x0F	0xFC
0x65	GPIOIE28	R/W	GPIO28~GPIO2F Input Enable for input function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x66	GPIOIE30	R/W	GPIO30~GPIO37 Input Enable for input function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0xFF	0xFC
0x67	GPIOIE38	R/W	GPIO38~GPIO3B Input Enable for input function bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** GPIO input mode disable **1:** GPIO input mode enable. ** GPO3C~GPO3F have no input functions.*	0x00	0xFC
0x68	GPIOIE40	R/W	GPIO40~GPIO47 Input Enable for input function bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x69	GPIOIE48	R/W	GPIO48~GPIO4F Input Enable for input function bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6A	GPIOIE50	R/W	GPIO50~GPIO57 Input Enable for input function bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6B	GPIOEE58	R/W	GPIO58~GPIO59 Input Enable for input function bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x03	0xFC

Input Enable Register (Continued)					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x6C	GPXAIE00	R/W	GPXIOA00~GPXIOA07 Input Enable for input function bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6D	GPXAIE08	R/W	GPXIOA08~GPXIOA11 Input Enable for input function bit[0]~bit[3] stand for GPXIOA08~GPXIOA11 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6E	RSV	RSV	Reserved	RSV	0xFC
0x6F	GPXDIE00	R/W	GPXIOD00~GPXIOD07 Input Enable for input function bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC

GPIO_MISC Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x70	GPIO_MISC	7	R/W	ESB_DAT(GPIO0C) output current selection 0: 4mA 1: 8mA	0x60	0xFC
		6	R/W	SPICLK(GPIO58) output current selection 0: 8mA 1: 16mA		
		5	R/W	ESB_CLK(GPIO0B) output current selection 0: 8mA 1: 16mA		
		4	R/W	RSV		
		3	R/W	GPIO17 / GPIO18 are featured with signal bypass function. Signal input via GPIO17 can be directly passed through GPIO18. **0:** Pass through function disable **1:** Pass through function enable		
		2	R/W	Alternative functions select for GPXIOA00~GPXIOA02. **0:** GPXIOA00~GPXIOA02 remain default output function **1:** GPXIOA00~GPXIOA02 become SDICS#, SDICLK, and SDIMOSI functions.		
		1	RSV	Reserved		
		0	R/W	Beep glue logic switch. GPIO12 can be output a specific function as following formula. *GPIO12 = PWM2 ⊕ GPIO16(input) ⊕ GPIO17(input)* **0:** Beep glue logic function disable **1:** Beep glue logic function enable		

GPIO_MISC 2 Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x71	GPIO_MISC2	7	R/W	LPC bus redirection enable, will redirect LPC bus to relative KSO pins: 0: Disable 1: Enable PCICLK to GPIO25(KSO5) PCIRST# to GPIO26(KSO6) SERIRQ to GPIO27(KSO7) LFRAME# to GPIO28(KSO8) LAD3 to GPIO2B(KSO11) LAD2 to GPIO2C(KSO12) LAD1 to GPIO2D(KSO13) LAD0 to GPIO2E(KSO14)	0x00	0xFC
		6	R/W	Select GPIO25(KSO5) output current 4mA/16mA =0, Select Output Current 4mA for GPIO25(KSO5) =1, Select Output Current 16mA for GPIO25(KSO5)		
		5	R/W	Enable SMBus port 3 (SCL3/SDA3) 0:Disable 1:Enable		
		4	R/W	Enable SMBus port 2 (SCL2/SDA2) 0:Disable 1:Enable		
		3	RSV	Reserved		
		2	RSV	Reserved		
		1	RSV	Reserved		
		0	R/W	PECI function enable to GPIO41 0:Disable 1:Enable		

GPIO Test Mux Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x72	GPIO_TMR	7	R/W	Enable Test Mux Mode **0:** Disable **1:** Enable	0x00	0xFC
		6~4	RSV	Reserved		
		3~0	RO	Test Mux Mode Counter Show Current Test Mode		

GPX MISC Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x73	GPX_MISC	7~3	RSV	Reserved	0x00	0xFC
		2	R/W	GPXIOA07 output power fail flag enable **0:** Disable **1:** Enable		
		1	R/W	GPXIOA03 output power fail flag enable **0:** Disable **1:** Enable		
		0	R/W	GPXA06 output VC(Voltage comparator) Enable **0:** Disable **1:** Enable		

4.2.4 GPIO Programming Sample

In this section gives some programming sample to control GPIO module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of GPIO filed application.

Example	
PIN	**Function**
GPIO00 (GA20)	Output
GPIO01 (KBRST#)	Output
GPIO02 (GPIO) *	Input
GPIO03 (GPIO) *	Input
GPIO04 (GPIO)	Output
GPIO05 (PCIRST#)	Input
GPIO06 (GPIO) *	Input
GPIO07 (GPIO)	Output
Programming model	
1. Set function selection register. GPIOFS00 (0xFC00) = 0x03 2. Set related pins to be output enable. GPIOOE00 (0xFC10) = 0x93 3. Set related pins to be input enable. GPIOIE00 (0xFC60) = 0x6C * GPIO02/03/06 do not exist in KBx930 chip	

4.3 Keyboard and Mouse Control Interface (KBC)

4.3.1 KBC I/F Function Description

The KBC is compatible with i8042 and responsible for keyboard/mouse accessing via legacy 60h/64h ports. The port 60h is the data port and port 64h is the command port. The legacy IRQ1 for keyboard devices and IRQ12 for mouse devices can be generated. The KBC interface provides fast GA20 control for legacy application.

KBC data register can be accessed by host or KBC firmware. Writing this register will setup a **OBF** (**O**utput **B**uffer **F**ull) flag, which can be clear by firmware. While the host issues I/O write to 60h/64h port, an **IBF** (**I**nput **B**uffer **F**ull) flag will assert. The interrupts can be programmed to issue while the flag of IBF/OBF asserting.

The following table gives a summary about port 60h/64h accessing.

Port	Access	Type	Register	Flag	Comment
60h	I/O Write	Data	KBCDAT (0xFC85)	IBF	Write data to keyboard/mouse
64h	I/O Write	Command	KBCCMD (0xFC84)	IBF	Write command to keyboard/mouse
60h	I/O Read	Data	KBCDAT (0xFC85)	OBF	Read data from keyboard/mouse
64h	I/O Read	Status	KBCSTS (0xFC86)		Read status from keyboard/mouse

KBC data register, **KBCDAT**, keeps data from host or data written by KBC firmware.

Bit	7	6	5	4	3	2	1	0
Name	Keyboard/Mouse Data Register							

KBC command register, **KBCCMD**, is used to keep the command from host. This register is read only.

Bit	7	6	5	4	3	2	1	0
Name	Keyboard/Mouse Command Register							

KBC status register, **KBCSTS**, keeps the status as the following table. For more detail please refer to the section, **KBC Registers Description**.

Bit	7	6	5	4	3	2	1	0
Name	Parity Error	Time Out	Aux. Data Flag	Un-inhibited	Address (A2)	System Flag	IBF	OBF

4.3.2 KBC Registers Description

KBC Command Byte Register (KBC command 20h/60h)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x80	KBCCB	7	R/W	PS/2 hardware mode enable **0**: Disable **1**: Enable If the host issues command 20h via port 64h, and the KBC returns data via port 60h. This bit will always be read as **zero**.	0x40	0xFC
		6	R/W	Scan code set2 conversion enable (PS/2 scan code set2 converts to set 1) **0**: Disable **1**: Enable		
		5	R/W	Disable Auxiliary device **0**: Enable **1**: Disable		
		4	R/W	Disable Keyboard device **0**: Enable **1**: Disable		
		3	R/W	Inhibit Override **0**: Disable **1**: Enable		
		2	R/W	System Flag (warm boot flag) **0**: cold boot **1**: warm boot		
		1	R/W	IRQ12 Enable While KBCSTS[5]=1(Auxiliary Data Flag) and KBCSTS[0]=1 (OBF), then IRQ12 will issue. **0**: Disable **1**: Enable		
		0	R/W	IRQ1 Enable While KBCSTS[5]=0 (Auxiliary Data Flag) and KBCSTS[0]=1 (OBF), then IRQ1 will issue. **0**: Disable **1**: Enable		

KBC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x81	KBCCFG	7	R/W	Keyboard lock enable **0**: Disable **1**: Enable	0x00	0xFC
		6	R/W	Fast gate A20 control **0**: Disable gate A20 control **1**: Enable gate A20 control		
		5	R/W	KBC hardware command sets (90h~93h, D4h) enable. **0**: Disable **1**: Enable		
		4	R/W	KBC hardware command sets (60h, A7h~ABh, Adh~Aeh) enable. **0**: Disable **1**: Enable		
		3	R/W	Keyboard lock flag status **0**: keyboard not lock or not inhibit **1**: keyboard lock or inhibit		
		2	R/W	KBC hardware command sets (A4h, A6h) enable. **0**: Disable **1**: Enable		
		1	R/W	IBF (KBCSTS[1]) interrupt enable. (IBF from 0 to 1) **0**: Disable **1**: Enable		
		0	R/W	OBF (KBCSTS[0]) interrupt enable (OBF from 1 to 0) **0**: Disable **1**: Enable		

KBC Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x82	KBCIF	7-3	RSV	Reserved	0x00	0xFC
		2	R/W1C	Status of KBC command handled by firmware While receiving KBC commands which need firmware to handle, the hardware will set this bit. Then the firmware will deal with all the following command until this bit is clear by firmware.		
		1	R/W1C	IBF interrupt pending flag **0**: no IBF interrupt occurs **1**: IBF interrupt occurs		
		0	R/W1C	OBF interrupt pending flag **0**: no OBF interrupt occurs **1**: OBF interrupt occurs		

KBC Hardware Command Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x83	KBCHWEN	7	R/W	KBC hardware command set (FEh) enable **0**: Disable **1**: Enable	0x00	0xFC
		6	R/W	KBC hardware command set (E0h) enable **0**: Disable **1**: Enable		
		5	R/W	KBC hardware command set (D3h) enable **0**: Disable **1**: Enable		
		4	R/W	KBC hardware command set (D2h) enable **0**: Disable **1**: Enable		
		3	R/W	KBC hardware command set (D1h) enable **0**: Disable **1**: Enable		
		2	R/W	KBC hardware command set (D0h) enable **0**: Disable **1**: Enable		
		1	R/W	KBC hardware command set (C0h) enable **0**: Disable **1**: Enable		
		0	R/W	KBC hardware command set (20h) enable **0**: Disable **1**: Enable		

KBC Command Buffer						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x84	KBCCMD	7-0	RO	Command written to port 64h will be stored in this register	0x00	0xFC

KBC Data Input/Output Buffer						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x85	KBCDAT	7-0	R/W	Data written to this register to make OBF set (OBF=1). The host read this register via port 60h.	0x00	0xFC

KBC Host Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x86	KBCSTS	7	R/W	Parity error **0**: No parity error occurs in PS/2 protocol **1**: Parity error occurs in PS/2 protocol.	0x00	0xFC
		6	R/W	Timeout **0**: No timeout occurs in PS/2 protocol **1**: Timeout occurs in PS/2 protocol.		
		5	R/W	Auxiliary data flag		
		4	RO	Uninhibited **0**: keyboard inhibited **1**: keyboard not inhibited		
		3	RO	Address (A2) **0**: output buffer data from 60h **1**: output buffer data from 64h		
		2	RO	System flag		
		1	R/W1C	IBF		
		0	R/W1C	OBF		

(Reserved)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x87	RSV	7-0	RSV	Reserved	0x00	0xFC

(Reserved)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x88	RSV	7-0	RSV	Reserved	0x00	0xFC

(Reserved)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x89	RSV	7~0	RSV	Reserved	0x00	0xFC

KBC Write Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8A	KBCDATR	7-0	RO	Read back port of KBCDAT, [0xFC85]	0x00	0xFC

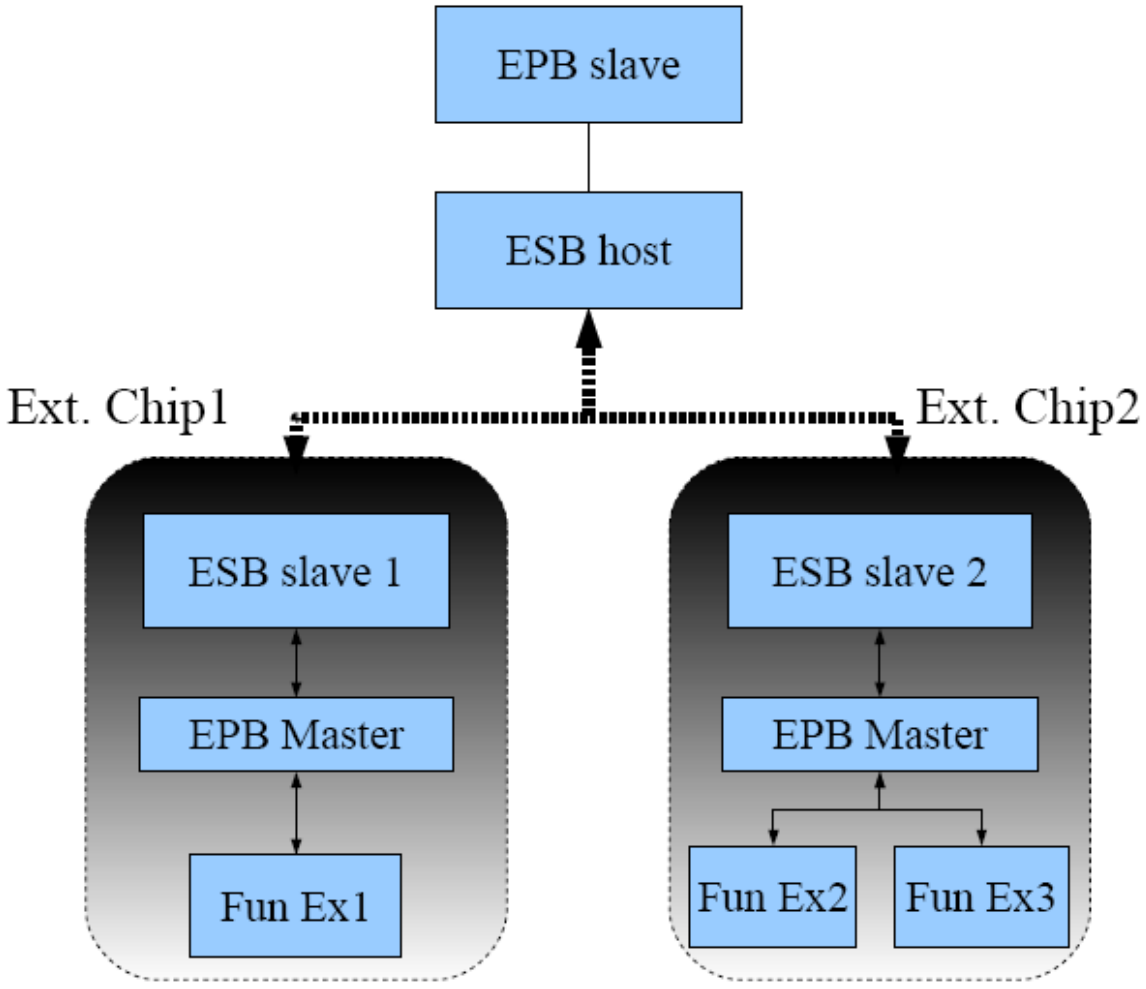
4.4 ENE Serial Bus Controller (ESB)

4.4.1 ESB Function Description

To extend the usage of the current design, an ENE serial bus interface is introduced. An external ESB device can be controlled by firmware transparently. As the following table, 4 memory address ranges are reserved for ESB devices.

	Memory Range
Range1	0xFCA0~0xFCAF
Range2	0xFCB0~0xFCBF
Range3	0xFCC0~0xFCCF
Range4	0xFD00~0xFDFF

In the ESB architecture, external ESB devices are supported. And each device can be configured with interrupt capability. A figure gives the topology of ENE Serial Bus as following.



The topology of ENE Serial Bus

4.4.2 ESB Registers Description

ESB Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x90	ESBCFG	7	R/W	Loop back test enable **0**: Disable **1**: Enable	0x00	0xFC
		6-5	R/W	ESB clock selection. **00**: (main clock) / 8 (2MHz) **01**: (main clock) / 4 (4MHz) **10**: (main clock) / 2 (8MHz) **11**: (main clock) / 1 (16MHz)		
		4	R/W	External device access mode. **0**: Access external device via 4 predefined memory ranges. (automatic mode) **1**: Access external devices via **ESBCA**, **ESBCD** and **ESBRD** registers. (byte mode)		
		3	R/W	ESB clock output enable **0**: Disable **1**: Enable		
		2	R/W	ESB interrupt enable **0**: Disable **1**: Enable		
		1	R/W	ESB host queries device interrupt status automatically. (when **ESBCFG**[3]=1) **0**: Disable **1**: Enable		
		0	R/W	ESB function enable **0**: Disable **1**: Enable		

ESB Command and Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x91	ESBCS	7	RSV	Reserved	0x00	0xFC
		6	R/W1C	Device resume signal flag **0**: no event **1**: event occurs.		
		5	R/W1C	ESB bus timeout status **0**: no timeout event **1**: bus timeout		
		4	R/W1C	Device data received status. **0**: no data received **1**: data received.		
		3	R	ESB host busy flag. **0**: not busy **1**: host busy		
		2	W	Start to send command, command byte in **ESBCD**, 0xFC94 Please write “0” will not work. **1**: send command		
		1-0	R/W	ESB access command type (while **ESBCFG**[3]=1) **00**: interrupt query **01**: read **10**: write **11**: Reserved		

ESB Interrupt Enable of External Device						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	ESBINTE	7	RSV	Reserved	0x00	0xFC
		6	R/W	Device resume signal interrupt enable **0**: Disable **1**: Enable		
		5	R/W	Bus timeout interrupt enable **0**: Disable **1**: Enable		
		4	R/W	Device data received interrupt enable **0**: Disable **1**: Enable		
		3	R/W	Interrupt enable (IRQ3) of external ESB device. **0**: Disable **1**: Enable		
		2	R/W	Interrupt enable (IRQ2) of external ESB device. **0**: Disable **1**: Enable		
		1	R/W	Interrupt enable (IRQ1) of external ESB device. **0**: Disable **1**: Enable		
		0	R/W	Interrupt enable (IRQ0) of external ESB device. **0**: Disable **1**: Enable		

ESB Command Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x93	ESBCA	7-0	R/W	External ESB device address to be accessed. (when **ESBCFG**[3]=1) The address is predefined according to different device.	0x00	0xFC

ESB Command Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	ESBCD	7-0	R/W	Write data port to external ESB device (when **ESBCFG**[3]=1)	0x00	0xFC

ESB Received Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	ESBRD	7-0	R/W	Read data port to external ESB device (when **ESBCFG**[3]=1) If loop back test enabled, ESBCFG[7]=1, the register will be writable, otherwise, read-only.	0x00	0xFC

ESB Enable for External Device						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	ESBED	7-5	RSV	Reserved	0x00	0xFC
		4	R/W	Low clock mode enable (clock source 32KHz) For performance and power saving consideration, while low clock mode enabled, please set the query function off. **0**: Disable **1**: Enable		
		3	R/W	Enable external ESB device decoding address 0xFEE0~0xFEFF **0**: Disable **1**: Enable		
		2	R/W	Enable external ESB device decoding address 0xFCC0~0xFCCF **0**: Disable **1**: Enable		
		1	R/W	Enable external ESB device decoding address 0xFCB0~0xFCBF **0**: Disable **1**: Enable		
		0	R/W	Enable external ESB device decoding address 0xFD00~0xFDFF. **0**: Disable **1**: Enable		

ESB Interrupt Event Pending Flag for External Chip						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x97	ESBINT	7	R/W1C	Interrupt event pending flag of IRQ7 (cascade mode only) **0**: no event **1**: event occurs	0x00	0xFC
		6	R/W1C	Interrupt event pending flag of IRQ6 (cascade mode only) **0**: no event **1**: event occurs		
		5	R/W1C	I Interrupt event pending flag of IRQ5 (cascade mode only) **0**: no event **1**: event occurs		
		4	R/W1C	Interrupt event pending flag of IRQ4 (cascade mode only) **0**: no event **1**: event occurs		
		3	R/W1C	Interrupt event pending flag of IRQ3 **0**: no event **1**: event occurs		
		2	R/W1C	Interrupt event pending flag of IRQ2 **0**: no event **1**: event occurs		
		1	R/W1C	Interrupt event pending flag of IRQ1 **0**: no event **1**: event occurs		
		0	R/W1C	Interrupt event pending flag of IRQ0 **0**: no event **1**: event occurs		

ESB Cascade Mode Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	ESBCAS	7	R/W	Interrupt enable of IRQ7 for external chip **0**: disable **1**: enable	0x00	0xFC
		6	R/W	Interrupt enable of IRQ6 for external chip **0**: disable **1**: enable		
		5	R/W	Interrupt enable of IRQ5 for external chip **0**: disable **1**: enable		
		4	R/W	Interrupt enable of IRQ4 for external chip **0**: disable **1**: enable		
		3-1	RSV	Reserved		
		0	R/W	Cascade mode enable **0**: disable **1**: enable		

ESB Programming Sample

In this section gives some programming sample to control ESB module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of ESB filed application.

Example
A device connecting to ESB master.
Programming model
GPIOFS08[4:3] (0xFC01[4:3])= 11b ; ESB function selection pin GPIOIE08[4] (0xFC61[4]) = 1b ; set related pin as an input ESBCFG (0xFC90) = 0x69 ; ESB clock=32MHz / EPB mode enable ESBED (0xFC96) = 0x02 ; enable ESB Now F/W can access register 0xFCC0~0xFCCF

4.5 Reserved

This page is leaved blank intentionally.

4.6 PECI

4.6.1 PECI Functional Description

The **Platform Environment Control Interface (PECI)** is a one-wire bus interface that provides a communication channel between Intel processor and chipset components to external monitoring devices. The PECI is a subset of SST**(Simple Serial Transport)** application. The PECI specification provides information for electrical requirements, platform topologies, power management handling, bus device enumeration, commands and addressing for Intel based system.

Please be noted that the PECI enable bit is in GPIO_MISC2, and should be set properly before PECI start to work.

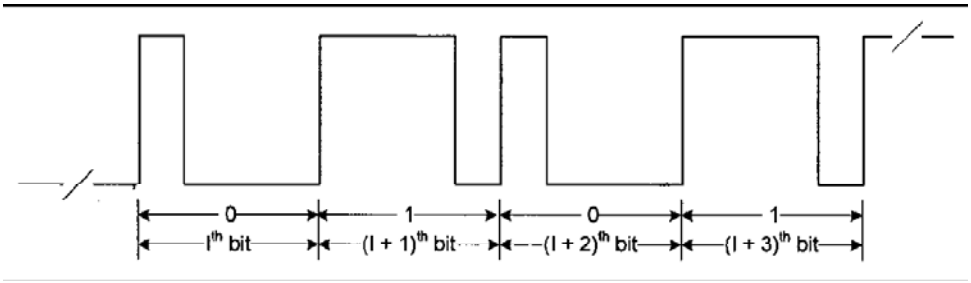
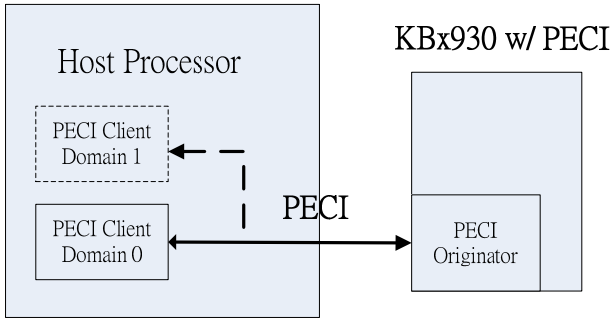


Figure 4.6.1 Example stream of 4 bits: "0101"

(Logic '0' encodes into 1000 pulse; Logic '1' encodes into 1110 pulse)



Conceptual Block Diagram

Not Intended to depict actual implementation

Figure 4.6.2 Conceptual Block Diagram for PECI application

4.6.2 PECI Register Description (Base address = FCD0h, 16 bytes)

PECI function configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD0	PECICFG	7~6	R/W	PECI operation frequency Selection, default support highest speed. **00**: 2M ~ 15.6k Hz **01**: 1M ~ 7.8k Hz **10**: 0.5M ~ 3.9k Hz **11**: 0.25M ~ 2k Hz	0x00	0xFC
		5	RSV	Reserved		
		4	R/W	Slow clock at idle state disable (for low power) **0**: enable **1**: disable		
		3	R/W	PECI Interrupt Enable (total enable)		
		2	R/W	Stealth cycle at quarter t_{BIT} time **0**: disable **1**: enable This bit is set, then quarter t_{BIT} time will be reduced 1T.		
		1	RSV	Reserved		
		0	R/W	PECI function enable PECI state machine will come back to idle state, when this bit is disabled.		

PECI function control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD1	PECICTL	7~3	RSV	Reserve	0x00	0xFC
		2	RSV	Reserve		
		1	R/W	Issue abort command This bit will be auto clear when abort behavior finish. The originator can't abort message when receives data state.		
		0	R/W	Issue package to client This bit will be auto clear when package transfer finish.		

PECI status observation						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD2	PECIST	7~6	RSV	Reserved	0x01	0xFC
		5	RO	TX active flag for transmitter state		
		4	RO	RX active flag for receiver state		
		3	RO	PECI bus line status for debugging		
		2	RO	Bus busy		
		1	RO	FIFO full flag for write/read state		
		0	RO	FIFO empty flag for write/read state		

PECI interrupt enable control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD3	PECIINTE	7	R/W	PECI data input de-bounce enable **0**: no de-bounce **1**: de-bounce enable	0x00	0xFC
		6	R/W	PECI output enable selection **0**: normal mode PECI output enable high, when issue package **1**: PECI output enable always high		
		5	R/W	PECI output data selection **0**: normal mode **1**: PECI output data always high for debugging		
		4	RSV	Reserved		
		3	R/W	Interrupt Enable of Client Abort		
		2	R/W	Interrupt Enable of FCS fault		
		1	R/W	Interrupt Enable of FIFO half		
		0	R/W	Interrupt Enable of FIFO error		

PECI interrupt status (event pending flag)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD4	PECIINT	7~4	RSV	Reserved	0x00	0xFC
		3	R/W1C	Interrupt Status of Client Abort The client reply to FCS is a one's complement. That means client will abort this message.		
		2	R/W1C	Interrupt Status of FCS fault The client reply to FCS is not correct. If FCS value is wrong then this bit will be set.		
		1	R/W1C	Interrupt Status of FIFO half If FIFO half, this bit will be set. That means FW must be write/read register PECIWD/PECIRD.		
		0	R/W1C	Interrupt Status of FIFO error If full flag is set and write data to PECIWD, this bit will be set; Otherwise, If empty flag is set and read data from PECIRD, then this bit will be set. If this bit is set, FIFO all pointers and data will be clear.		

PECI target address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD5	PECIADR	7~0	R/W	This is the address of the PECI device targeted to receive a message.	0x00	0xFC

PECI write length byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD6	PECIWLB	7~0	R/W	The Write Length byte in the PECI header is used to convey the number of bytes the originator will send to the target device. The length byte includes command and data byte.	0x00	0xFC

PECI read length byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD7	PECIRLB	7~0	R/W	The Read Length byte is used by the target to determine the number of data bytes it must supply to the originator before Returning the FCS over that data.	0x00	0xFC

PECI write data byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD8	PECIWD	7~0	R/W	PECI Write data. This includes both commands and data. All commands require at least one Command byte with the exception of Ping().	0x00	0xFC

PECI read data byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD9	PECIRD	7~0	RO	PECI Received (Read) data from client devices.	0x00	0xFC

PECI received FCS value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xDA	PECICFCS	7~0	RO	The FCS value received from client	0x00	0xFC

PECI generated FCS value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xDB	PECIOFCS	7~0	RO	The FCS value generated from originator	0x00	0xFC

PECI t_{bit} counter value observation						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xDC	PECIQTB	7~0	RO	The counter value of quarter tBIT time for debugging	0x01	0xFC

PECI FIFO write/read pointer observation						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xDD	PECIPOIN	7~4	RO	FIFO Read Pointer FIFO read pointer points to the location in the FIFO to read from next	0x00	0xFC
		3~0	RO	FIFO Write Pointer FIFO write pointer points to the location in the FIFO to write to next		

4.7 OWM

4.7.1 OWM Functional Description

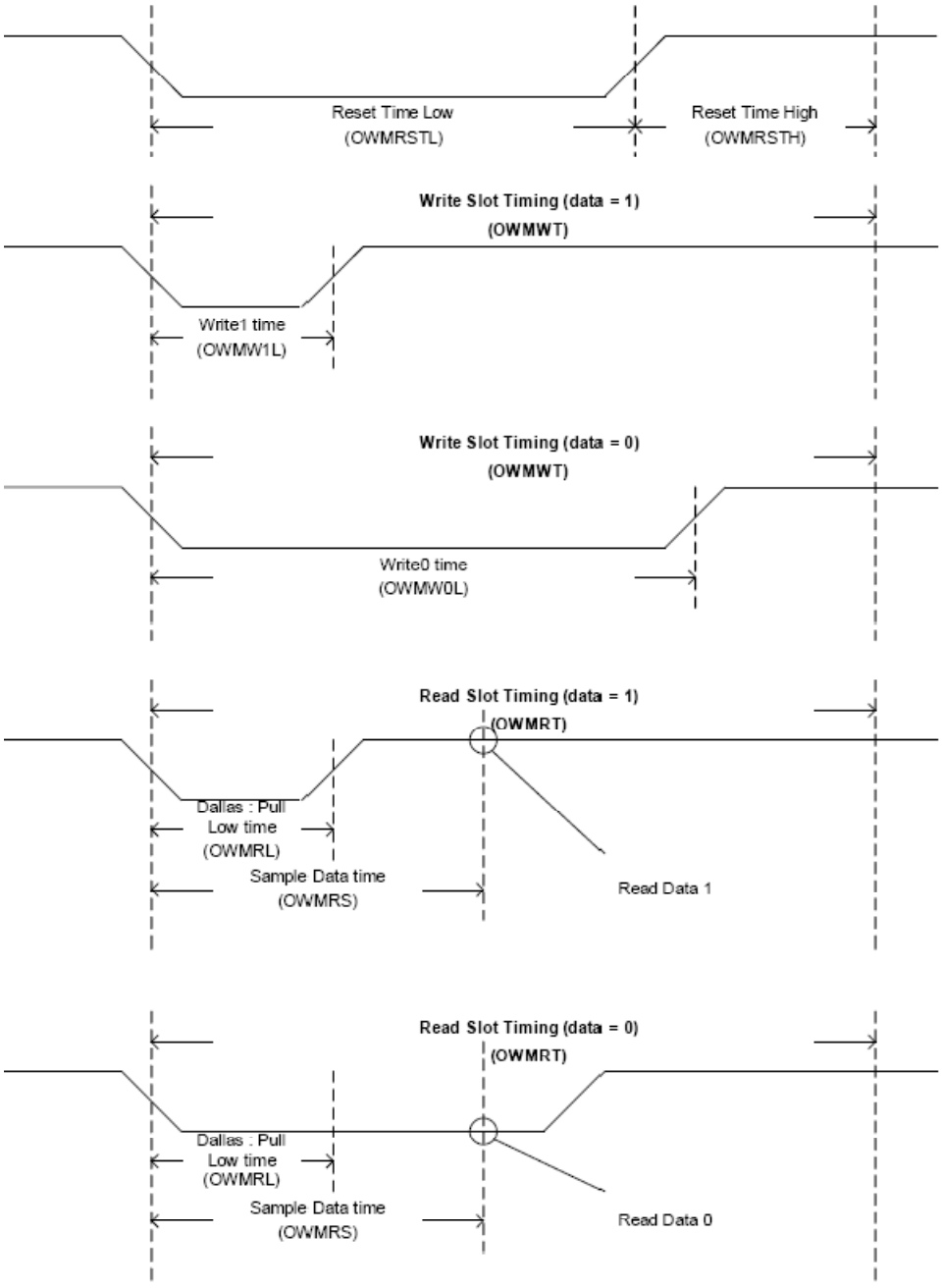
OWM is called One Wire Bus Master Interface (GPIO0A).

OWM supports Dallas One Wire Bus Master and TI HDQ protocol.

OWM supports Reset/Break, Read and Write command.

Separate 8-bit read and write buffers.

Configurable timing registers can be setting by F/W.



OWM bus master configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF0	OWMCFG	7	R/W	**EN** : One Wire Bus Master Interface Enable **0**: Disable One Wire Bus Master Interface **1**: Enable One Wire Bus Master Interface	0x00	0xFC
		6	R/W	TI/Dallas Mode Select **1**: TI mode **0**: Dallas mode		
		5~4	RSV	Reserved		
		3	R/W	**ETMOI**: Enable Timeout Interrupt. Interrupt occurs if timeout interrupt flag is set **0**: Disable **1**: Enable		
		2	R/W	**EWRI**: Enable Write Command Complete Interrupt. Interrupt occurs if write command complete flag is set **0**: Disable **1**: Enable		
		1	R/W	**ERDI**: Enable Read Command Complete Interrupt. Interrupt occurs if read command complete flag is set **0**: Disable **1**: Enable		
		0	R/W	**ERSTI**: Enable Reset/Break Completely Interrupt. Interrupt occurs if reset/break complete flag is set **0**: Disable **1**: Enable		

OWM bus master status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF1	OWMSR	7	RO	**BSY** : One Wire Host Busy Status **0**: Idle **1**: Busy	0x00	0xFC
		6~5	RO	Reserved		
		4	RO	**PDR**: Presence Detect Result. (for Dallas Only) The detect result status of the presence detect when reset/break complete interrupt occurs. **0**: Not Exist **1**: Exist		
		3	R/W1C	**TMO**: Timeout flag of read/write command for slave response. **0**: No timeout event **1**: Timeout event		
		2	R/W1C	**WRC**: Status flag of write command for operation completion **0**: Write command not complete **1**: Write command complete		
		1	R/W1C	**RDC** : Status flag of read command for operation completion **0**: Read command not complete **1**: Read command complete		
		0	R/W1C	**RSTC**: Status flag of reset/break for operation completion **0**: Reset/Break command not complete **1**: Reset/Break command complete (Set when the reset high time reached after reset low time)		

OWM bus master command						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF2	OWMCMD	7~2	RSV	Reserved	0x03	0xFC
		1~0	R/W	One Wire Interface Command **00**: Reset /Break **01**: Read **10**: Write **11**: No operation		

OWM bus master write data buffer (transmit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF3	OWMWB	7~0	R/W	The transmit data buffer send to a slave device	0x00	0xFC
OWM bus master read data buffer (receive)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF4	OWMRB	7~0	RO	The receive data buffer got from a slave device	0x00	0xFC

OWM reset/break low timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF5	OWMRSTL	7	RSV	Reserved	0x40	0xFC
		6~0	R/W	The Reset Time Low interval,, Clock time base = 8us		
OWM reset/break high timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF6	OWMRSTH	7	RSV	Reserved	0x40	0xFC
		6~0	R/W	The Reset Time High interval Clock time base = 8us		

OWM write slot timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF7	OWMWT	7~0	R/W	Write 1-bit Data time interval Clock time base = 2us	0x2D	0xFC
OWM write 1 low timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF8	OWMW1L	7~0	R/W	Write 1 time interval Clock time base = 1us	0x0A	0xFC
OWM write 0 low timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xF9	OWMW0L	7~0	R/w	Write 0 time interval Clock time base = 1us	0x50	0xFC

OWM read slot timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xFA	OWMRT	7	R/W	Host Read 1-bit Data time, clock time base = 2us .	0x2D	0xFC
OWM read low timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xFB	OWMRL	7~4	RSV	Reserved	0x03	0xFC
		3~0	R/W	For Dallas only, Host to pull low time Clock time base = 1us		

OWM read sample timing						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xFC	OWMRS	7~0	R/W	The time interval for Host to check read data 0 or 1, Clock time base = 1us.	0x14	0xFC

4.8 Pulse Width Modulation (PWM)

4.8.1 PWM Function Description

Pulse width modulation (PWM) is a powerful technique for controlling analog circuits with a processor's digital outputs. PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion.

The KBC supports 4 PWM channels. 2 channels (PWM0/PWM1) are for 8-bit resolution and 2 channels (PWM2/PWM3) are for 14-bit resolution. The PWM provides clock source selection which is defined in the register description.

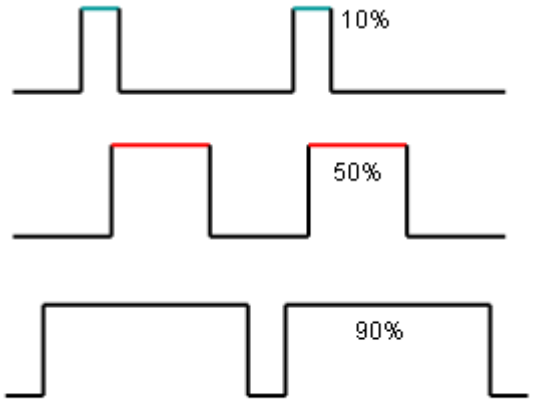


Figure. PWM Duty Cycle

. The duty cycle of PWM is illustrated as the above figure. The following table summarizes the relationship about the applications with the definition in the PWM registers description.

Definition	Formula	Comment
Duty Cycle	(PWM High Period Length+1)/(PWM Cycle Period Length+1) *100%	
Cycle Length	(PWM Cycle Length Register +1) * (PWM clock source)	For 8-bit
Cycle Length	(PWMCYC + 1) * 2 * (1 + Prescaler)/(Peripheral clock or fixed 1 MHz)	For 14-bit

For the limitation of current design, in some critical cases, the PWM output will be the one as the following table.

Condition	PWM Output
H>C	Always "1" (High)
H=0x00 and C=0x00	Always "1" (High)
H=0x00 and C=0xFF	A Short Pulse
H=0xFF and C=0x00	Always "1" (High)
Switch to GPIO mode and output low	Always "0" (Low)
H= High Period Length (PWMHIGH) , **C**= Cycle Period Length (PWMCYCL)	

4.8.2 PWM Registers Description

PWM Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x00	PWMCFG	7-6	R/W	PWM1 clock source selection **0**: 0.976 μs (1 μs) **1**: 62.5 μs (64 μs) **2**: 250 μs (256 μs) **3**: 3.99ms (4ms)	0x00	0xFE
		5	RSV	Reserved		
		4	R/W	PWM1 Enable **0**: Disable **1**: Enable		
		3-2	R/W	PWM0 clock source selection **0**: 0.976 μs (1 μs) **1**: 62.5 μs (64 μs) **2**: 250 μs (256 μs) **3**: 3.99ms (4ms)		
		1	RSV	Reserved		
		0	R/W	PWM0 Enable **0**: Disable **1**: Enable		

PWM0 High Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x01	PWMHIGH0	7-0	R/W	High Period Length of PWM0. This should be smaller than Cycle Length.	0x00	0xFE

PWM0 Cycle Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x02	PWMCYC0	7-0	R/W	Cycle Length of PWM0.	0x00	0xFE

PWM1 High Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x03	PWMHIGH1	7-0	R/W	High Period Length of PWM1. This should be smaller than Cycle Length.	0x00	0xFE

PWM1 Cycle Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x04	PWMCYC1	7-0	R/W	Cycle Length of PWM1	0x00	0xFE

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x05	RSV	7-0	RSV	RSV	0x00	0xFE

PWM2 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x06	PWMCFG2	7	R/W	PWM2 Enable **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	PWM2 pre-scaler clock selection **0**: peripheral clock **1**: 1MHz clock (fixed)		
		5-0	R/W	The 6-bit pre-scaler of PWM2 The pre-scalar value = register value + 1		

PWM3 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x07	PWMCFG3	7	R/W	PWM3 Enable **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	PWM3 pre-scaler clock selection **0**: peripheral clock **1**: 1MHz clock (fixed)		
		5-0	R/W	The 6-bit pre-scaler of PWM3 The pre-scaler value = register value + 1		

PWM2 High Period Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x08	PWMHIGH2H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x09	PWMHIGH2L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

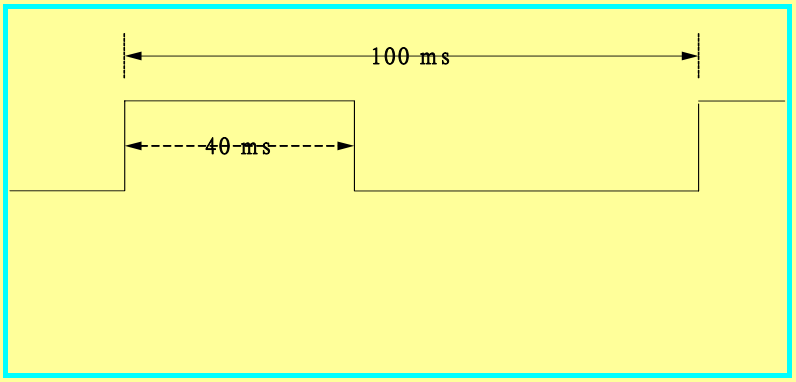
PWM2 Cycle Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0A	PWMCYC2H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x0B	PWMCYC2L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

PWM3 High Period Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0C	PWMHIGH3H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x0D	PWMHIGH3L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

PWM3 Cycle Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0E	PWMCYC3H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x0F	PWMCYC3L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

4.8.3 PWM Programming Sample

In this section gives some programming sample to control PWM module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of PWM filed application.

Example
Programming PWM0 with a period 100ms and high period 40ms, that is, duty cycle is 40%. 
Programming model
1. Set related GPIO function selection register. GPIOFS08[7] (0xFC01[7]) = 1b 2. Select clock source = 4ms , and enable PWM0 PWMCFG[3:0] (0xFE00[3:0]) = 1101b 3. Cycle = 4ms * (24+1) PWMCYCL0 (0xFE02) = 0x18 4. Duty cycle = 40/100 = 40% ; (X+1)/(24+1) = 40% -> X=9 PWMHIGH0 (0xFE01) = 0x09
For PWM2/3 as 800Hz pulse : The formula is as: (PWMCYC+1)*2*(1+Prescaler)*(1/11Mhz)= (1/800hz)…… (Set Prescaler=0) PWMCYC = 6874 = 0x1ADB

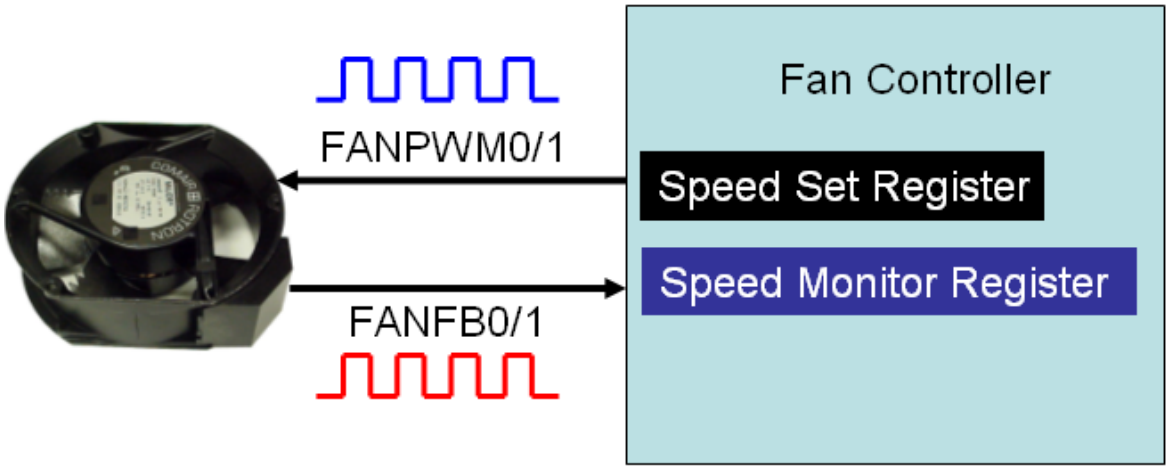
4.9 Fan Controller

4.9.1 Fan Function Description

The KBC provides 2 interfaces with speed monitor for fan control. Two clock selections for fan controller, one is based on main clock and the other is fixed 62.5 μs. The fan controller can be configured as a PWM function, as known as FANPWM.

4.9.1.1 Fan Tachometer Monitor

The fan tachometer is implemented by a 12-bit counter with four resolution(In Reg FANSTS0/FANSTS1) as follows, 62.5 μs, 31.25 μs, 15.625 μs, 7.8125 μs. The following figure gives an example for fan speed monitor and control with 62.5 μs. The KBC uses the pin FANPWM0/1 to drive external fan device, and the fan device feedback the speed via the pin FANFB0/1. The fan controller keeps the speed in the monitor register. The fan controller will compare the speed and check if the current speed is higher or slower than the expected one. If slower, then the controller will increase the frequency to drive FANPWM0/1 automatically, otherwise decrease the frequency. The expected speed can be programmable by F/W.



As following RPM table is given for programmers. In this table, the information between RPM and value for fan speed set is shown with 62.5 μs resolution. The target speed counter value is require when fan controller is operated under auto-fan mode.

RPM	Round/1min	Round/1sec	μs/Round	Value (Set Counter)
6000	6000	100	10000	160 (10000/62.5)
5000	5000	83.33	12000	192 (12000/62.5)
4000	4000	66.667	15000	240 (15000/62.5)
3000	3000	50	20000	320 (20000/62.5)
2000	2000	33.333	30000	480 (30000/62.5)
1000	1000	16.667	60000	960 (60000/62.5)
500	500	8.3	120000	1920(120000/62.5)

RPM (round/min) = 60,000,000 / (FANMON *62.5)

4.9.1.2 FANPWM Function

The fan controller can be used as a 12-bit PWM function. While PWM function applied, the fan controller will refer to the peripheral clock, and the PWM high period and cycle time can be determined as the following formula:

PWM Cycle Length = (PWM cycle register + 1) * peripheral clock resolution

PWM High Period = (PWM high period register + 1) * peripheral clock

Please note, to program the high pulse width of PWM (**FANPWMH0/FANPWML0** and **FANPWMH1/FANPWML1**, i.e., 0xFE26/0xFE27 and 0xFE36/0xFE37), *high-byte first and then low-byte in order.*

The fan controller could be operated in **Auto-fan** or **Fixed-fan mode**.

In Auto-fan mode, it's required to set the Fan Speed Set Counter Value based on the table in 4.9.1.1 (Be cautious that the resolution could be different by register value)

In Fixed-fan mode, it's required to set the PWM cycle length, PWM high period based on the formula in 4.9.1.2. By setting these registers, specific PWM frequency, duty cycle could be generated.

4.9.2 Fan Registers Description

Fan0 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x20	FANCFG0	7	R/W	FAN0 monitor clock selection. **0**: peripheral clock **1**: the monitor base clock will based on FANSTS0[6:5](0xFE21)	0x00	0xFE
		6	R/W	FAN0 speed monitor counter edge trigger selection. **0**: count pulse event on rising edge. **1**: count pulse event on rising and falling edge.		
		5	R/W	FANPWM0 cycle width enable **0**: Disable **1**: Enable		
		4	R/W	FANPWM0 enable. **0**: Disable **1**: Enable		
		3	R/W	FAN0 speed monitor interrupt enable **0**: Disable **1**: Enable		
		2	R/W	FAN0 speed monitor timeout error interrupt enable **0**: Disable **1**: Enable		
		1	R/W	Auto-fan mode control enable. FANCFG0[0] and FANCFG0[4] should be set at the same time to make it work. (The tachometer is required for feedback, PWM should also be enabled) **0**: Disable **1**: Enable		
		0	R/W	FAN0 tachometer monitor enable. **0**: Disable **1**: Enable		

Fan0 Control and Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x21	FANSTS0	7	R/W	FAN0 auto-load FANCPWM function enable **0**: Disable **1**: Enable	0x00	0xFE
		6-5	R/W	FANPWM clock resolution selection. FANCFG0[7](0xFE20) should be set for selection take effect **00**: 62.5us (default) **01**: 31.25us **10**: 15.625us **11**: 7.8125us		
		4	R/W	FAN0 digital noise filter enable. **0**: Disable **1**: Enable		
		3-2	RSV	Reserved		
		1	R/W1C	Flag of FAN0 speed monitor timeout error **0**: no timeout error **1**: timeout error event		
		0	R/W1C	Flag of FAN0 speed monitor update event. **0**: no update event. **1**: update event		

Fan0 Speed Monitor Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x22	FANMONH0	3-0	RO	High 4 bits of FAN0 speed monitor counter value	0x0F	0xFE
0x23	FANMONL0	7-0	RO	Low 8 bits of FAN0 speed monitor counter value	0xFF	0xFE

Fan0 Speed Set Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x24	FANSETH0	3-0	R/W	High 4 bits of target FAN0 speed counter value.	0x00	0xFE
0x25	FANSETL0	7-0	R/W	Low 8 bits of target FAN0 speed counter value.	0x00	0xFE
Notice: These two registers are used in auto-fan mode and are set as target fan speed counter value						

FANPWM0 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x26	FANPWMH0	3-0	R/W	High 4 bits of FANPWM0 high pulse width. (FANCFG0[1]=0 only)	0x00	0xFE
0x27	FANPWML0	7-0	R/W	Low 8 bits of FANPWM0 high pulse width. (FANCFG0[1]=0 only)	0x00	0xFE
Notice: These two registers are used in fixed-fan mode and are set as target FANPWM width to change effective fan speed ***PWM high period = (PWM high pulse register + 1) * peripheral clock***						

Current FANPWM0 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x28	FANCPWMH0	3-0	RO	High 4 bits of current FANPWM0 high pulse width.	0x00	0xFE
0x29	FANCPWML0	7-0	RO	Low 8 bits of current FANPWM0 high pulse width.	0x00	0xFE

FANPWM0 Cycle Length (12-bit)

Offset	Name	Bit	Type	Description	Default	Bank
0x2A	FANPWMCH0	3-0	R/W	High 4 bits of Cycle length of FANPWM0 (FANCFG0[5]=1)	0x00	0xFE
0x2B	FANPWMCHL0	7-0	R/W	Low 8 bits of Cycle length of FANPWM0 (FANCFG0[5]=1)	0x00	0xFE
Notice: These two registers are used in fixed-fan mode and are set as target FANPWM cycle ***Cycle length = (PWM cycle register + 1) * peripheral clock***						

FANPWM0 Auto-Load High Pulse Width Bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2C	FANUPWM0	7-4	RSV	Reserved	0x0F	0xFE
		3-0	R/W	If auto-load feature enabled (FANSTS0[7]=1), this register value will be auto-loaded into FANCPWMH0 registers and FANCPWML0 will be forced to be zero while monitor timeout occurs		

FAN tachometer monitor controller configuration for FANFB2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2D	FANTMCFG0	7-6	RSV	Reserved	0x00	0xFE
		5-4	R/W	FAN tachometer monitor speed sample range FANTMCFG0[1](0xFE2D) should be set for selection take effect **00**: 62.5us (default) **01**: 31.25us **10**: 15.625us **11**: 7.8125us		
		3	R/W1C	Flag bit for Fan tachometer monitor timeout error event. **0**: no timeout error **1**: timeout error event		
		2	R/W	FAN digital filter enable for Fan tachometer monitor **0**: Disable **1**: Enable		
		1	R/W	Test mode enable for Fan tachometer monitor **1**: the monitor base clock will be peripheral clock. **0**: the monitor base clock will be based on FANTMCFG0[5:4]		
		0	R/W	FAN tachometer monitor enable To enable addition FAN Tachometer Monitor FANFB2 **0**: Disable **1**: Enable		

FAN tachometer monitor speed monitor counter value for FANFB2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2E	FANTMMONH0	3-0	R/W	High 4 bits of FANFB2 speed monitor counter value	0x0F	0xFE
0x2F	FANTMMONL0	7-0	R/W	Low 8 bits of FANFB2 speed monitors counter value.	0xFF	0xFE

Fan1 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x30	FANCFG1	7	R/W	FAN1 monitor clock selection. **0**: peripheral clock **1**: the monitor base clock will based on FANSTS1[6:5](0xFE31)	0x00	0xFE
		6	R/W	FAN1 speed monitor counter edge trigger selection. **0**: count pulse event on rising edge. **1**: count pulse event on rising and falling edge.		
		5	R/W	FANPWM1 cycle width enable **0**: Disable **1**: Enable		
		4	R/W	FANPWM1 enable. **0**: Disable **1**: Enable		
		3	R/W	FAN1 speed monitor interrupt enable **0**: Disable **1**: Enable		
		2	R/W	FAN1 speed monitor timeout error interrupt enable **0**: Disable **1**: Enable		
		1	R/W	Automatic FANPWM control enable. FANCFG1[0] and FANCFG1[4] should be set at the same time to make it work. (The tachometer is required for feedback, PWM should also be enabled) **0**: Disable **1**: Enable		
		0	R/W	FAN1 tachometer monitor enable. **0**: Disable **1**: Enable		

Fan1 Control and Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x31	FANSTS1	7	R/W	FAN1 auto-load FANCPWM function enable **0**: Disable **1**: Enable	0x00	0xFE
		6-5	R/W	FANPWM clock resolution selection. FANCFG1[7](0xFE30) should be set for selection take effect **00**: 62.5us (default) **01**: 31.25us **10**: 15.625us **11**: 7.8125us		
		4	R/W	FAN1 digital noise filter enable. **0**: Disable **1**: Enable		
		3-2	R/W	Reserved		
		1	R/W	Flag of FAN1 speed monitor timeout error **0**: no timeout error **1**: timeout error event		
		0	R/W	Flag of FAN1 speed monitor update event. **0**: no update event. **1**: update event		

Fan1 Speed Monitor Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x32	FANMONH1	3-0	R/W	High 4 bits of FAN1 speed monitor counter value	0x0F	0xFE
0x33	FANMONL1	7-0	R/W	Low 8 bits of FAN1 speed monitor counter value	0xFF	0xFE

Fan1 Speed Set Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x34	FANSETH1	3-0	R/W	High 4 bits of target FAN1 speed counter value.	0x00	0xFE
0x35	FANSETL1	7-0	R/W	Low 8 bits of target FAN1 speed counter value.	0x00	0xFE
Notice: These two registers are used in auto-fan mode and are set as target fan speed counter value						

FANPWM1 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x36	FANPWMH1	3-0	R/W	High 4 bits of FANPWM1 high pulse width. (FANCFG1[1]=0 only)	0x00	0xFE
0x37	FANPWML1	7-0	R/W	Low 8 bits of FANPWM1 high pulse width. (FANCFG1[1]=0 only)	0x00	0xFE
Notice: These two registers are used in fixed-fan mode and are set as target FANPWM width to change effective fan speed ***PWM high period = (PWM high pulse register + 1) * peripheral clock***						

Current FANPWM1 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x38	FANCPWMH1	3-0	RO	High 4 bits of current FANPWM1 high pulse width.	0x00	0xFE
0x39	FANCPWML1	7-0	RO	Low 8 bits of current FANPWM1 high pulse width.	0x00	0xFE

FANPWM1 Cycle Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x3A	FANPWMCH1	3-0	R/W	High 4 bits of Cycle length of FANPWM1 (FANCFG1[5]=1)	0x00	0xFE
0x3B	FANPWMCHL1	7-0	R/W	Low 8 bits of Cycle length of FANPWM1 (FANCFG1[5]=1)	0x00	0xFE
Notice: These two registers are used in fixed-fan mode and are set as target FANPWM cycle ***Cycle length = (PWM cycle register + 1) * peripheral clock***						

FANPWM1 Update High Pulse Width Bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x3C	FANUPWM1	7-4	RSV	Reserved	0x0F	0xFE
		3-0	R/W	If auto-load feature enabled (FANSTS1[7]=1), this register value will be auto-loaded into FANCPWMH1 registers and FANCPWML1 will be forced to be zero while monitor timeout occurs		

FAN tachometer monitor controller configuration for FANFB3						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x3D	FANTMCFG1	7-6	RSV	Reserved	0x00	0xFE
		5-4	R/W	FAN tachometer monitor speed sample range FANTMCFG1[1](0xFE3D) should be set for selection take effect **00**: 62.5us (default) **01**: 31.25us **10**: 15.625us **11**: 7.8125us		
		3	R/W1C	Flag bit for Fan tachometer monitor timeout error event. **0**: no timeout error **1**: timeout error event		
		2	R/W	FAN digital filter enable for Fan tachometer monitor **0**: Disable **1**: Enable		
		1	R/W	Test mode enable for Fan tachometer monitor **1**: the monitor base clock will be peripheral clock. **0**: the monitor base clock will be based on FANTMCFG1[5:4]		
		0	R/W	FAN tachometer monitor enable To enable addition FAN Tachometer Monitor FANFB3 **0**: Disable **1**: Enable		

FAN tachometer monitor speed monitor counter value for FANFB3						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x3E	FANTMMONH1	3-0	R/W	High 4 bits of FANFB3 speed monitor counter value	0x0F	0xFE
0x3F	FANTMMONL1	7-0	R/W	Low 8 bits of FANFB3 speed monitors counter value.	0xFF	0xFE

4.9.3 Fan Programming Sample

In this section gives some programming sample to control FAN module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of FAN filed application.

Example
FAN0 @ 4000 rpm with automatic PWM control **FAN1 @ some rpm with fixed PWM control**
Programming model
For FAN0: 1. set related GPIO function select register to enable alternative output. GPIOFS10[2] (0xFC02[2]) = 1b 2. set related GPIO input enable. GPIOIE10[4] (0xFC62[4]) = 1b 3. set FAN0 configuration register FANCFG0 (0xFE20) = 0x93 4. set FAN0 speed monitor counter value FANMONH0 (0xFE24) = 0x00 FANMONL0 (0xFE25) = 0xF0 For FAN1: 1. set related GPIO function select register to enable alternative output. GPIOFS10[3] (0xFC02[3]) = 1b 2. set FAN1 configuration register FANCFG1 (0xFE30) = 0x90 3. set FAN1 speed monitor counter value FANPWMH1 (0xFE36) = 0x03 FANPWML2 (0xFE37) = 0xE8

4.10 General Purpose Timer (GPT)

4.10.1 GPT Function Description

The KBC provides 4 GPTs (General Purpose Timers), two 16-bit timers and two 8-bit timers. These 4 GPTs operate based on 32.768KHz and all timers have the interrupt capability. The GPT is simply a free run counter. While the timer meets the specific value in count register, for instance, 0xFE53 and 0xFE55, an interrupt issues (if interrupt enabled) and the counter reset to be zero.

- GPT0 and GPT1 are 8-bit timers.
- GPT2 and GPT3 are 16-bit timers.

4.10.2 GPT Registers Description

GPT Configuration

Offset	Name	Bit	Type	Description	Default	Bank
0x50	GPTCFG	7-5	RSV	Reserved	0x00	0xFE
		4	R/W	GPT test mode enable. In test mode, the GPT runs with main clock. **0**: Disable **1**: Enable		
		3	R/W	GPT3 counting and interrupt enable. **0**: Disable **1**: Enable		
		2	R/W	GPT2 counting and interrupt enable. **0**: Disable **1**: Enable		
		1	R/W	GPT1 counting and interrupt enable. **0**: Disable **1**: Enable		
		0	R/W	GPT0 counting and interrupt enable. **0**: Disable **1**: Enable		

GPT Pending Flag

Offset	Name	Bit	Type	Description	Default	Bank
0x51	GPTPF	7	WO	Writing “1” to this bit forces GPT3 restart.	0x00	0xFE
		6	WO	Writing “1” to this bit forces GPT2 restart.		
		5	WO	Writing “1” to this bit forces GPT1 restart.		
		4	WO	Writing “1” to this bit forces GPT0 restart.		
		3	R/W1C	Interrupt pending flag of GPT3.		
		2	R/W1C	Interrupt pending flag of GPT2.		
		1	R/W1C	Interrupt pending flag of GPT1.		
		0	R/W1C	Interrupt pending flag of GPT0.		

GPT0 Counter Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x53	GPT0	7-0	R/W	Once GPT0 counter meets this value, an interrupt issues. GPT0 restart to count from zero.	0x00	0xFE

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x54	RSV	7-0	RSV	Reserved	0x00	0xFE

GPT1 Counter Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x55	GPT1	7-0	R/W	Once GPT1 counter meets this value, an interrupt issues. GPT1 restart to count from zero.	0x00	0xFE

GPT2 Counter Value (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x56	GPT2H	7-0	R/W	High byte of GPT2 counter value Once GPT2 counter meets this 16-bit value, an interrupt issues. GPT2 restart to count from zero.	0x00	0xFE
0x57	GPT2L	7-0	R/W	Low byte of GPT2 counter value Once GPT2 counter meets this 16-bit value, an interrupt issues. GPT2 restart to count from zero.	0x00	0xFE

GPT3 Counter Value (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x58	GPT3H	7-0	R/W	High byte of GPT3 counter value. Once GPT3 counter meets this 16-bit value, an interrupt issues. GPT3 restart to count from zero.	0x00	0xFE
0x59	GPT3L	7-0	R/W	Low byte of GPT3 counter value. Once GPT2 counter meets this 16-bit value, an interrupt issues. GPT3 restart to count from zero.	0x00	0xFE

4.10.3 GPT Programming Sample

In this section gives some programming sample to control GPT module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of GPT filed application.

Example
Programming GPT0 to issue an interrupt every 5ms
Programming model
1. Set GPT configuration register, enable GPT0 interrupt. GPTCFG[0] (0xFE50[0]) = 1b 2. Fill the GPT counter value. GPT0 (0xFE53) = 0xA6 ; 5000/30 = 0xA6

4.11 SDI Host/Device Interface Controller

The SDI host/device controller can be programmed to a SPI Host or a SPI Device (0xFE74.7). The Default is the SPI Host. The SPI Host and Device use the same IO.

4.11.1 SDI Host/Device Interface Description

The Serial Peripheral Interface Bus or SPI (often pronounced “spy”) bus is a synchronous serial data link standard designed by Motorola that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame.

In KBCx930, the SDI host could support the SPI mode 0/1, and is configurable by SFICFG[1]

SDI slave could support the SPI mode 0.

4.11.2 SDI Host Interface Description

SDI host interface configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x70	SHICFG	7	RO	SDI host Idle flag. If this bit set, the SDI host is in an idle state. **0**: busy **1**: idle	0x00	0xFE
		6-5	RSV	Reserved		
		4	R/W	SDI host SDICS# Pin Control 0 : Set SDICS# High 1 : Set SDICS# Low		
		3-2	R/W	SDI host CLK divider. ***SPICLK frequency = peripheral clock / [(divider +1)*2]***		
		1	R/W	SDI Host SDIMOSI/SDIMISO Timing. **0**: SDIMOSI changes data at falling edge of SDICLK. (device latches at rising edge of SDICLK) SDIMISO latch data at rising edge of SDICLK. (device changes at falling edge of SDICLK). **1**: SDIMOSI changes data at rising edge of SDICLK. (device latches at falling edge of SDICLK) SDIMISO latch data at falling edge of SDICLK. (device changes at rising edge of SDICLK).		
		0	R/W	SDI host controller enable **0**: Disable **1**: Enable		

SDI host interface transmit data port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x71	SHITBUF	7-0	R/W	While SHICFG[7]=0 (SDI not busy), writing to this register forces data output to SDIMOSI in continuously serial 8 bits. MSB first.	0x00	0xFE
SDI host interface receive data port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x72	SHIRBUF	7-0	RO	SDI host reading port.	0x00	0xFE

4.11.2 SDI Device Interface Description

SDI device interface configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x78	SDICFG	7	RO	SDICS# status	0x44	0xFE
		6~4	RSV	Reserved		
		3	R/W	SDI command mode **0**: Disable. (Normal mode) **1**: Enable. (Command mode) (When enable this mode, SDICFG[2:1] would not take effect) (Configurable command : Read TX buffer in register SDICMD)		
		2	R/W	Enable SDI device TX. **0**: Disable **1**: Enable		
		1	R/W	Enable SDI device RX. **0**: Disable **1**: Enable		
		0	R/W	SDI device controller enable **0**: Disable **1**: Enable		

SDI device interface interrupt configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x79	SDIRS	7	RSV	Reserved	0x00	0xFE
		6	R/W1C	(Normal mode only) Transmit buffer empty pending flag		
		5	R/W1C	(Normal mode only) Receive buffer full pending flag		
		4	R/W1C	SDICS# rising edge pending flag		
		3	RSV	Reserved		
		2	R/W	(Normal mode only) Transmit buffer empty interrupt enable bit **0**: Disable **1**: Enable		
		1	R/W	(Normal mode only) Receive buffer full interrupt enable bit **0**: Disable **1**: Enable		
		0	R/W	SDICS# rising edge interrupt enable bit **0**: Disable **1**: Enable		

SDI device interface transmit status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x7A	SDITSTS	7	RSV	Reserved	0x00	0xFE
		6~4	RO	***Transmit buffer count*** **In normal mode:** The count is the number how many data in Tx Buffer aren't transmitted yet. **In command mode:** The count is the number of transmitted byte data in single transition.		
		3	RSV	Reserved		
		2	R/W1C	(Normal mode only) Transmit buffer underflow flag		
		1	RO	(Normal mode only) Transmit buffer full flag		
		0	R	(Normal mode only) Transmit buffer empty flag		
			W	***Write 1 to clear Tx buffer*** **Normal mode:** FIFO's write point and read point are both reset to point to position "0". **Command mode:** Only FIFO's write point is reset to point to position "0".		

SDI device interface receive status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x7B	SDIRSTS	7	RSV	Reserved	0x00	0xFE
		6~4	RO	***Receive Buffer count*** **Normal mode :** The count is the number how many data in Rx Buffer aren't read yet. **Command mode:** The count is the number of received byte data in single transition.		
		3	RSV	Reserved		
		2	R/W1C	(Normal mode only) Receive buffer overflow flag		
		1	RO	(Normal mode only) Receive buffer full flag		
		0	R	(Normal mode only) Receive buffer empty flag		
			W	***Write 1 to clear Rx buffer*** **Normal mode:** FIFO's write point and read point are both reset to point to position "0". **Command mode:** Only FIFO's write point is reset to point to position "0".		

SDI device interface transmit data port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x7C	SDITBUF	7~0	WO	***SDI Device Interface Transmitted Data Port*** (4 bytes buffers, External SPI Host must supply SPI clock) **Normal mode**: Please check full flag to finish the write operation. If the TX buffer is full, SDI will skip the newly data and preserve the previous data. If the TX buffer is empty, SDI will always Transmit data = 0x00.	0x00	0xFE

SDI device interface receive data port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x7D	SDIRBUF	7	RO	***SDI Device Interface Received Data Port*** (4 bytes buffers, Read the data from the external SPI Host) **Normal mode:** Please check empty flag to finish the reading operation. If the RX buffer is full, SDI will skip the newly data and preserve the previous data. If the RX buffer is empty, SDI will always read data = 0x00. **Command mode:** In a transaction, SDI will only receive 4 bytes data. If over 4 bytes data, SDI will skip the newly data and preserve the previous data. We can read the RX buffer according the Rx buffer's read point.	0x00	0xFE

Command : Read TX buffer						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x7E	SDICMD	7~0	R/W	***Configurable command. : Read TX buffer*** This function should be used along with SDI command mode (SDICFG[3], 0xFE78.3 = 1)	0x5A	0xFE

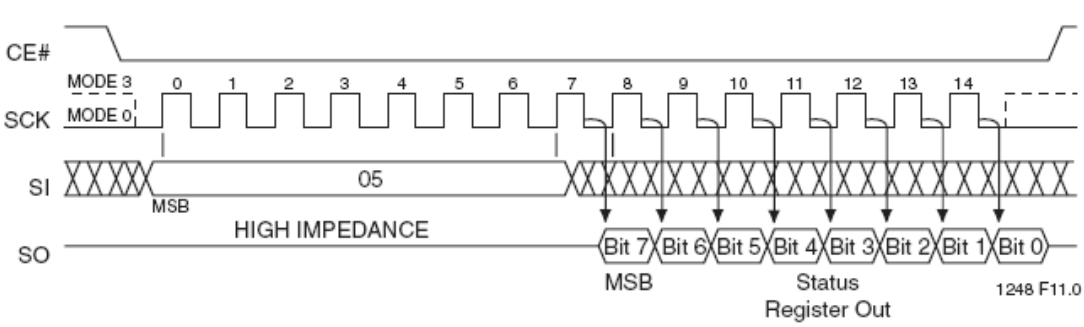
SDI TX/RX buffer write point and read point						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x7F	SDIPT	7~6	RO	**Normal mode:** When writing SDI Tx buffer, write point will increase 1 until SDI Tx buffer is full. **Command mode:** When writing SDI Tx buffer, write point will increase 1.	0x00	0xFE
		5~0	RSV	Reserved		

4.11.3 SDI Programming Sample

In this section gives some programming sample to control SDI module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of SDI filed application.

Example

A SPI device is attached. Here is a READ STATUS command.



Programming model

```
GPXAFS00[2:0]   (0xFC0C[2:0]) = 111b;    //Select SDI function pins
GPXDIE00[0] (0xFC6C[0])   = 1b;          //Enable SDI data input
SDICSR (0xFE70) = 0x01;                  //Set SDICS# low, SPI clock = Peripheral //clock/2
SDIBO   (0xFE71) = 0x05;                 //Transfer CMD(0x05) to device
Wait SDICSR[7]   (0xFE70[7]) = 1b;       //Wait bus idle
SDIBO (0xFE71) = 0x00;       //Transfer dummy byte to device and //device sends status byte to SDI
Wait SDICSR[7] (0xFE70[7])   = 1b;       //Wait bus idle
SDICSR (0xFE70) = 0x00;                  //Set SDICS# high
Read SDIBI (0xFE72);                     //Get device status
```

4.12 Watchdog Timer (WDT)

4.12.1 WDT Function Description

A Watchdog Timer (WDT) is a hardware timing device that triggers a system reset while the system encounters any unrecoverable situation. The WDT utilizes 32.768KHz for operation. The WDT triggers the system reset in three ways.

- Reset the 8051 microprocessor only.
- Reset the whole logic, except GPIO modules.
- Reset the whole logic, including GPIO modules.

Here gives the highlight of WDT features:

- 20 bit Watchdog
- Watchdog password protection.
- Interrupt support.
- WDT LED blinking support.
- New 24 bit timer (TMR) support.

4.12.2 WDT Registers Description

WDT Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x80	WDTCFG	7	R/W	WDT clock source selection **0**: DPLL 32.768KHz source **1**: Internal OSC or External Crystal 32.768KHz source	0x00	0xFE
		6~4	RSV	Reserved		
		3	RSV	Reserved		
		2	R/W	WDT test mode enable. **0**: normal mode **1**: test mode, clock driven by internal 32MHz. (WDTCFG[7] ignore)		
		1	R/W	WDT interrupt enable. **0**: Disable **1**: Enable		
		0	R/W	WDT reset enable. Once WDT resets, two WDT pending flags are clear. **0**: Disable **1**: Enable		

WDT Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x81	WDTPF	7-5	RSV	Reserved	0x00	0xFE
		1	R/W1C	WDT interrupt flag Once the timer counts to half of WDT (0xFE82), an interrupt occurs. If the timer counts to WDT(0xFE82), a WDT reset occurs. **0**: no event **1**: event occurs		
		0	R/W1C	WDT reset flag Once the timer counts to WDT (0xFE82), a WDT reset occurs and this flag is set. **0**: no event **1**: event occurs		

WDT High 8-bit Counter Value (for WDT reset system of 10 bits counter)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x82	WDT	7-0	R/W	The high 8-bits of WDT counter value. The WDT timer unit is 32ms. Please note, fill this value at least greater than or equal **3** (>=3) for hardware limitation.	0x00	0xFE

WDT Blinking LED Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x83	LEDCFG	7~6	R/W	The low 2-bits of WDT counter value. The WDT timer unit is 32ms. Please note, fill this value at least greater than or equal **3** (>=3) for hardware limitation.	0x00	0xFE
		5-3	RSV	Reserved		
		2-0	R/W	LED Blinking configuration. **0**: LED output keeps high **1**: LED output keeps low 500ms for every 1 sec. **2**: LED output keeps low 500ms for every 2 sec **3**: LED output keeps low 500ms for every 4 sec **4**: LED output keeps low 500ms for every 8 sec		

WDT TMR (24-bit Timer) Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x84	TMR_CFG	7	R/W	TMR enable **0**: Disbale/reset TMR **1**:Enable TMR	0x00	0xFE
		6~3	RSV	Reserved		
		2	RO	TMR interrupt pending flag overflow. While TMR interrupt flag (TMR_CFG[1]) is set and an interrupt event occurs again. This bit will be set and can be clear via writing TMR_CFG[7] with “0”. **0**: no event **1**: event occurs		
		1	R/W1C	TMR interrupt flag. When TMR counter[23:16] is equal to TMR_MATCH register. This bit will be set. **0**: no event **1**: event occurs		
		0	R/W	TMR counter start control. **0**: stop counting **1**: start counting		

WDT TMR (24-bit Timer) Counter Match Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x85	TMR_MATCH	7-0	R/W	The 8bit match value register. If the clock source is from 32.768KHz OSC, the time base is approximated as 2.048s. When timer counter[23:16] is reached this value, timer emits interrupt and TMR_CFG[1] is set to 1 .	0x00	0xFE

WDT TMR (24-bit Timer) Counter Value 1						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x86	TMR_V1	7-0	RO	Value for TMR counter[23:16]	0x00	0xFE

WDT TMR (24-bit Timer) Counter Value 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x87	TMR_V2	7-0	RO	Value for TMR counter[15:8]	0x00	0xFE

4.12.3 WDT Programming Sample

In this section gives some programming sample to control WDT module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of WDT filed application.

Example
Set WDT=512ms to reset system, and an interrupt occurs while WDT=256ms (half of WDT)
Programming model
WDT (0xFE82) = 0x10 ; set WDT=512ms WDTCFG (0xFE80) = 0x03 ; enable interrupt and WDT reset

4.13 Low Pin Count Interface (LPC)

4.13.1 LPC Function Description

The Low Pin Count (LPC) is an interface for modern ISA-free system. The KBC connects to the system via LPC interface. The following LPC cycle types are supported.

Type	Address	Data
LPC I/O Read	16-bit	8-bit
LPC I/O Write	16-bit	8-bit
LPC Memory Read	32-bit	8-bit
LPC Memory Write	32-bit	8-bit
FWH Read	28-bit	8-bit
FWH Write	28-bit	8-bit

4.13.2 LPC I/O Decode Range

Item	Port	Comment
Keyboard Controller	60h/64h	
Embedded Controller	62h/66h (default)	Programmable
Legacy I/O	68h/6Ch, 2Eh/2Fh	
EC Index-I/O	FF29h~FF2Bh/FF2Dh~FF2Fh(default)	2 Sets, Programmable.
Debug Port	80h	Only write cycle support interrupt

4.13.3 LPC Memory Decode Range

Memory Address (hex)	Size	Setting (LPCSCFG[3],LPCFWH[7:6])
000C_0000 ~ 000F_FFFF＊ FFFC_0000 ~ FFFF_FFFF	256K (default)	0b,00b
000C_0000 ~ 000F_FFFF＊ FFF8_0000 ~ FFFF_FFFF	512K	0b,01b
000C_0000 ~ 000F_FFFF＊ FFF0_0000 ~ FFFF_FFFF	1M	0b,10b
000C_0000 ~ 000F_FFFF＊ FFE0_0000 ~ FFFF_FFFF	2M	0b,11b
000C_0000 ~ 000F_FFFF＊ FFC0_0000 ~ FFFF_FFFF	4M	1b,00b
000E_0000 – 000F_FFFF FFFE_0000 – FFFF_FFFF	128K	1b,11b
＊ LPC module decodes low memory address only in 256K range.		

4.13.4 FWH Memory Decode Range

Memory Address (hex)	Size	Setting (LPCSCFG[3],LPCFWH[7:6])
00C_0000 ~ 00F_FFFF＊ FFC_0000 ~ FFF_FFFF	256K (default)	0b,00b
00C_0000 ~ 00F_FFFF＊ FF8_0000 ~ FFF_FFFF	512K	0b,01b
00C_0000 ~ 00F_FFFF＊ FF0_0000 ~ FFF_FFFF	1M	0b,10b
00C_0000 ~ 00F_FFFF＊ FE0_0000 ~ FFF_FFFF	2M	0b,11b
00C_0000 ~ 00F_FFFF＊ FC0_0000 ~ FFF_FFFF	4M	1b,00b
00E_0000 – 00F_FFFF FFE_0000 – FFF_FFFF	128K	1b,11b
＊ LPC module decodes low memory address only in 256K range.		

4.13.5 Index-I/O Port

The KBC provides a method to communicate with the host via legacy I/O port. The host can access the XRAM space inside the KBC. The I/O port is called Index-I/O. Two Index-I/Os are supported and programmable. The registers, **LPCIBAH** and **LPCIBAL** (0xFE92 and 0xFE93), are used to specify the desired I/O port base. To enable the 2nd Index-I/O, the **LPCSCFG**[5], (0xFE90[5]) should be set. The index-I/O base address will be **8** bytes align if the **LPCSCFG**[5] set, otherwise **4** bytes alignment . For example, while the base address is 0xFF2C and LPCSCFG[5] set, the 1st index-I/O address will be 0xFF29 (io_base +1).

The following table collects the port definition for the *host*. The base address of Index-I/O is assumed to be **io_base**.

1st Index-I/O		2nd Index-I/O (LPCSCFG[5]=1)	
XRAM address (high)	io_base+1	XRAM address (high)	io_base+5
XRAM address (low)	io_base+2	XRAM address (low)	io_base+6
XRAM data (high)	io_base+3	XRAM data (high)	io_base+7

Here is an example how to use an Index-I/O.

EC F/W	Host software
1. EC F/W setups the base address, for instance, 0x380. That is, LPCIBAH=0x03 and LPCIBAL=0x80. 2. If the 2nd Index-I/O is needed, turn on the enable bit. That is, LPCSCFG[5]=1 (0xFE90[5]=1).	1. Host setups the desired XRAM address: Port 0x381 = high byte of XRAM address Port 0x382 = low byte of XRAM address 2. And then the host can access the content/data via Port 0x383. 3. If the 2nd Index-I/O required. Port 0x385 = high byte of XRAM address Port 0x386 = low byte of XRAM address Port 0x387 = content/data of XRAM address

4.13.6 Extended I/O Port (Debug Port, Port80)

Developers may use legacy I/O port, 0x80 for debug. The KBC provides a debug interface for this application, called extended I/O port (debug port). The port address can be programmable in the KBC. The host software can use this interface not only for debug but also for special communication with the EC F/W. This interface provides interrupt capability as well. That is, while host accesses this I/O port, an interrupt to 8051 occurs. There is one thing should be reminded. The interrupt feature is only for **I/O-write** to this port, not for I/O-read. Please note, the interrupt capability is controlled in the register **ECCFG**[2] (0xFF04[2]).

4.13.7 LPC Registers Description

LPC SIRQ Configuration for Quiet Mode						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x90	LPCSCFG	7-6	R/W	LPC Register Bank Switch Registers, 0xFE91~0xFE9F, are mapping to 2 banks. **00**: Bank 0 **01**: Bank 1 **10**: Reserved **11**: Reserved	0x20	0xFE
		5	R/W	Enable 2nd index-I/O mode		
		4	R/W	Switch of CIR/User-defined IRQ Switch between CIR and User defined SIRQ, and the SIRQ channel is defined in LPCTCFG[3:0] **0**: User defined SIRQ **1**: CIR SIRQ (Any one from CIRPF [3:0],FEC2h)		
		3	R/W	Memory size 4MB enable (LPC/FWH). If this bit enable, please make sure LPCFWH[7:6]=00**b** **0**: Disable **1**: Enable		
		2	R/W	LPC I/O 2Eh/2Fh decode enable. If enabled, 0xFE9A/0xFE9B are configured to take in charge of LPC I/O 2Eh/2Fh. **0**: Disable **1**: Enable		
		1	Ro	LPC SIRQ mode **0**: Continuous mode **1**: Quiet mode		
		0	WO	Force LPC SIRQ cycle start. Writing "**1**" to this bit forces SIRQ signal low for a pulse.		

4.13.7.1 LPC Registers Bank0 Descriptions (LPCSCFG[7:6]=2'b00, 15 bytes)

LPC SIRQ Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x91	LPCSIRQ	7	R/W	Ignore A22 of FWH cycle. **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	SCI SIRQ enable **0**: Disable **1**: Enable		
		5	R/W	IRQ12 SIRQ enable **0**: Disable **1**: Enable		
		4	R/W	IRQ1 SIRQ enable **0**: Disable **1**: Enable		
		3-0	R/W	SCI SIRQ channel. **0x00**: no SIRQ **0x01**: IRQ1 **0x02**: SMI# **0x03**: IRQ3 **0x04**: IRQ4 …. **0x0F**: IRQ15		

LPC Index-I/O Base Address (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	LPCIBAH	7-0	R/W	High byte of LPC index-I/O address	0xFF	0xFE
0x93	LPCIBAL	7-0	R/W	Low byte of LPC index-I/O address (8-byte alignment required)	0x28	0xFE

LPC Firmware Hub Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	LPCFWH	7-6	R/W	Memory size selection (LPC/FWH) **00**: 256KB **01**: 512KB **10**: 1MB **11**: 2MB / (Select 128KB when LPCSCFG[3]=1)	0x00	0xFE
		5	R/W	FWH memory cycle enable **0**: Disable **1**: Enable		
		4	R/W	FWH IDSEL check enable **0**: Disable **1**: Enable		
		3-0	R/W	FWH ID		

LPC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	LPCCFG	7	R/W	LPC memory write protection (including FWH) **0**: Disable **1**: Enable	0x80	0xFE
		6	R/W	Index-I/O port enable **0**: Disable **1**: Enable		
		5	R/W	KBC 60h/64h I/O port enable **0**: Disable **1**: Enable		
		4	R/W	Debug port (port 80) enable **0**: Disable **1**: Enable		
		3	R/W	EC I/O port enable (default port 62h/66h) **0**: Disable **1**: Enable		
		2	R/W	LPC memory cycle enable (not including FWH) **0**: Disable **1**: Enable		
		1	R/W	SIRQ always in continuous mode enable **0**: Disable **1**: Enable		
		0	R/W	LPC CLKRUN# enable **0**: Disable **1**: Enable		

LPC Extended (Debug) I/O Base Address (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	LPCXBAH	7-0	R/W	High byte of Extended I/O (debug port)	0x00	0xFE
0x97	LPCXBAL	7-0	R/W	Low byte of Extended I/O (debug port)	0x80	0xFE

LPC EC I/O Base Address (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	LPCEBAH	7-0	R/W	High byte of EC I/O	0x00	0xFE
0x99	LPCEBAL	7-0	R/W	Low byte of EC I/O	0x62	0xFE

LPC I/O 0x2E/0x2F Configuration and Status (LPCSCFG[2]=1)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9A	LPC2ECFG	7-4	RSV	Reserved	0x00	0xFE
		3	RO	The previous access type of 2Eh/2Fh **0**: Read **1**: Write		
		2	R/W1C	Interrupt flag of accessing 2Fh I/O. **0**: no event **1**: event occurs		
		1	R/W	2Fh I/O interrupt enable If this bit set, while host accesses 2Fh I/O, an interrupt will issue. **0**: Disable **1**: Enable		
		0	R/W	Decode 2Eh/2Fh I/O enable. **0**: Disable **1**: Enable		

LPC USER SIRQ Configuration (LPCSCFG[2]=0)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9B	LPCTCFG	7-6	RSV	Reserved	0x00	0xFE
		5	R/W	User defined SIRQ Setting. **0**: Low **1**: High		
		4	R/W	User defined SIRQ channel enable **0**: Disable **1**: Enable		
		3~0	R/W	User defined SIRQ channel number **0x00**: no SIRQ **0x01**: IRQ1 **0x02**: SMI# **0x03**: IRQ3 **0x04**: IRQ4 …. **0x0F**: IRQ15		

LPC I/O 2E Read Port Register (LPCSCFG[2]=1)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9B	LPCTCFG	7-0	RO	Host writes data to I/O port 0x2E, EC F/W could read data from this register.	0x00	0xFE

LPC Read/Write Data of I/O 0x2F (LPCSCFG[2]=1)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9C	LPC2FDAT	7-0	R	Host writes data to I/O port 0x2F, EC F/W could read data from this register.	0x00	0xFE
		7-0	W	If host issue any read access to I/O port 0x2F, the host will get the data which kept in this register		

LPC I/O 0x68/0x6C Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9D	LPC68CFG	7	R/W	LPC decode I/O port 68h/6Ch enable **0**: Disable **1**: Enable	0x00	0xFE
		6-2	RSV	Reserved		
		1	R/W	IBF interrupt enable Interrupt issues while IBF rising (LPC write I/O 68h/6Ch) **0**: Disable **1**: Enable		
		0	R/W	OBF interrupt enable Interrupt issues while OBF falling (LPC read I/O 68h) **0**: Disable **1**: Enable		

LPC I/O 0x68/0x6C Configuration and Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9E	LPC68CSR	7	R/W1C	I/O 68h/6Ch busy flag. EC F/W can write "**1**" to clear this flag. A write cycle to port 6Ch with data 0xFF also clear this flag **0**: not busy **1**: busy	0x00	0xFE
		6	RO	Indicator of write port. **0**: write 68h occurs **1**: write 6Ch occurs.		
		5-4	RSV	Reserved		
		3	R/W1C	IBF interrupt flag Interrupt flag while IBF rising (LPC write I/O 68h/6Ch) **0**: no event **1**: event occurs		
		2	R/W1C	OBF interrupt flag Interrupt flag while OBF falling (LPC read I/O 68h) **0**: no event **1**: event occurs		
		1	R/W1C	IBF of port 68h/6Ch		
		0	R/W1C	OBF of port 68h/6Ch		

LPC I/O 0x68/0x6C Data Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9F	LPC68DAT	7-0	R	Host writes data to I/O port 0x68/0x6C, EC F/W could read data from this register.	0x00	0xFE
		7-0	W	If host issue any read access to I/O port 0x68/0x6C, the host will get the data which kept in this register		

4.13.7.2 LPC Registers Bank1 Descriptions (LPCSCFG[7:6]=2'b01, 15 bytes)

LPC MEM/FWH Configuration register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x91	LPCFPCFG	7~2	RSV	Reserved	0x00	0xFE
		1	R/W	MEM / FWH access XRAM **0**: Disable **1**: Enable **LPCBIXAR** (0xFE9F bank 1) to set the base address of LPC to map to XRAM.		
		0	R/W	Protection enable **0**: Disable **1**: Enable		

LPC MEM/FWH protection segment						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	LPCFPSEG	7-6	RSV	Reserved	0x00	0xFE
		5-0	R/W	Protection Segment setting **LPCSCFG[3]** (0xFE90) , LPCFWH[7:6] (0xFE94 bank 0) to set the decoding memory size (4M/2M/1M/512K/256K) 1 Segment Size = 64KByte 6 bits valid for 4MB, Max 64 Segment 5 bits valid for 2MB, Max 32 Segment 4 bits valid for 1MB, Max 16 Segment 3 bits valid for 512K, Max 8 Segment 2 bits valid for 256K, Max 4 Segment (0~3)		

LPC MEM/FWH block number						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x93	LPCFPBKN	7-6	RSV	Reserved	0x00	0xFE
		5-0	R/W	Protection Block Number **00 0000**: indicates 1 block(1Kbyte), **11 1111**: indicates 64 blocks		

LPC misc register set 0						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	LPCMISC0	7	R/W	Schmitt Trigger for PCI clock input switch **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W1C	SIRQ start frame detection flag Set 1 by hardware, and clear by firmware		
		5	RO	Latched status of SERIRQ (pin3)		
		4	RO	Latched status of LFRAME# (pin 4)		
		3~0	RO	Latched status of LAD[3:0] ports		

LPC control and status register for clock detection function						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	LPC_CDCSR	7~5	RSV	Reserved	0x00	0xFE
		4	R/W1C	Clock detection pending flag When clock stopping detected, this bit will be high and clock monitoring will be stopped. Clearing the pending flag will cause detection start again. **0**:Clock alive detected, and clock source is still in monitoring **1**:Clock stop detected		
		3	RSV	Reserved		
		2	R/W	CLKRUN# Pull Down Enable The signal of CLKRUN# will be pulled down by satisfying following conditions: 1. The port of CLKRUN# is at Pull Up state. 2. PCI Clock stopping detected. 3. There are latched SIRQ request to be emitted. **0**:Disable **1**:Enable		
		1	R/W	Clock source select for detection **0**: Select PCI clock signal for detection. **1**: Select clock path of 32K OSC to detect.		
		0	R/W	Clock detection enable **0**:Disable **1**:Enable		

LPC raw counter value output for clock detection function						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	LPC_CDCV	7-0	RO	Referenced output of counter value for debugging purpose	0x00	0xFE

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x97	RSV	7-0	RSV	Reserved	0x00	0xFE

LPC transaction debug output register 0						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	LPCTDR0	7	RO	Transaction data valid indication	0x00	0xFE
		6	RSV	Reserved		
		5~4	RO	Transaction Toggle bits It will be accumulated after a valid transaction done		
		3~0	RSV	Reserved		

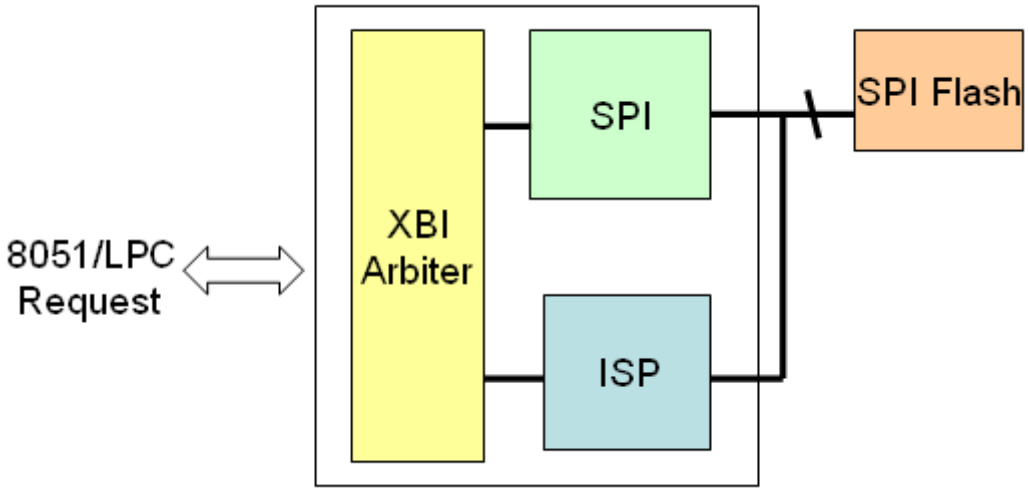
Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x99~9E	Reserved	7-0	RSV	Reserved	0x00	0xFE

LPC Bank Index for XRAM Access Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9F	LPCBIXAR	7-0	RSV	The base address for address map to access 4K XRAM	0x00	0xFE

4.14 X-Bus Interface (XBI)

4.14.1 XBI Function Description

The KBC implements a XBI module to handle the related request from 8051/LPC to flash device. The following figure is operation illustration.



The XBI module also takes the responsibility for the In-System-Programming (ISP) mechanism to update system BIOS. The detail steps to update system BIOS via ISP mode, please refer to the section of ISP. Here gives the feature of XBI module.

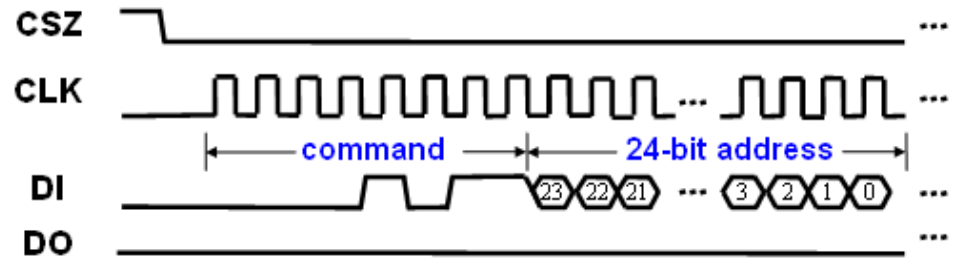
- Two 8051 code segments, one for 16K and the other for 48K.
- XBI arbiter to handle the transaction of 8051 and LPC request.
- XBI pre-fetch code mechanism support for better performance.
- Flash write-protection support.
- ISP flash update support.
- SHC (SPI host controller)
    1. 4 byte buffer for TX/RX
    2. Support mode 0 (clock rising latch data, clock falling drive data)
    3. SPI clock speed is 66 / 33 MHz (Depending on 0xFF0D.6)
    4. Programmable Tx/Rx length

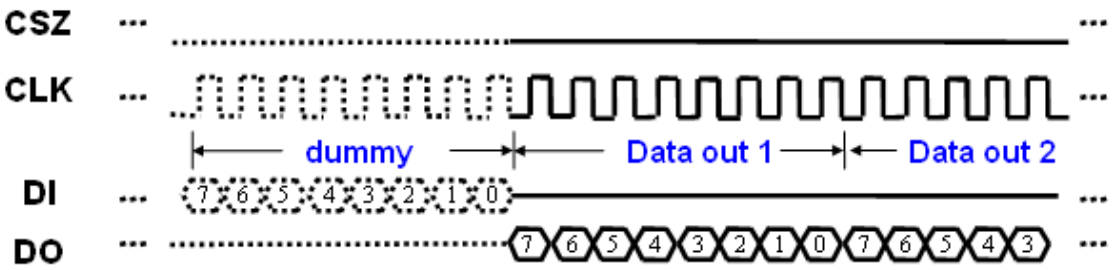
4.14.2 XBI SPI Enhancement

The 8051 microprocessor executes machine codes from SPI flash and the performance is determined by the read operation. To enhance the performance of SPI flash fetching, special protocols are introduced. They are **Dual/Quad** protocols. The following sections give a brief introduction.

4.14.2.1 Original Read Protocol

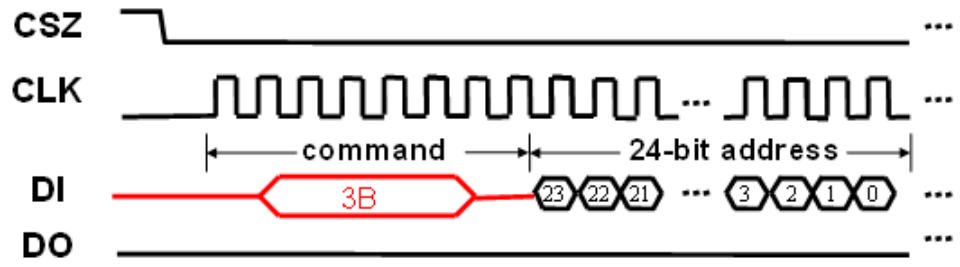
An original SPI read protocol to read flash is as the following figure. A chip select is asserted to the specific flash device and the SPI flash controller drives clock out. A 24-bit address phase follows 8-bit command phase. After a dummy phase, the SPI flash device returns data. Please note, the KBC currently supports 4MB (22bit) size SPI flash, therefore, address bit, A23~A22 are all zero.

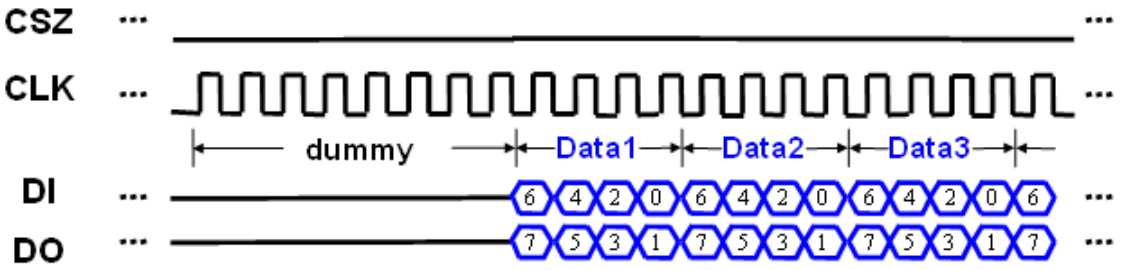




4.14.2.2 Dual Input Protocol

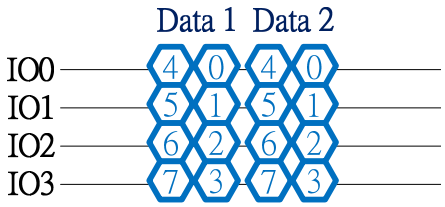
An improvement of data phase to increase performance is also introduced. The method is called **Dual Input** mode. In this mode, data output pin works as another input. The bit stream is shown as the following figure. Please note, the high nibble of this protocol is fixed to be 0x3. This is not a standard protocol and SPI flash devices should implement this feature to make it work.





4.14.2.3 Quad Protocol

Another improvement of data phase to increase performance is introduced, **Quad** mode. Extra pins are served as another IO ports (GPXIO06 / GPXIO07). The data bit stream illustration is as the following figure. The details could be found in register: SPICFG. This is not a standard protocol and SPI flash devices should implement this feature to make it work.



4.14.3 XBI Registers Description

8051 Address Segment 0 Mapping Configuration (0x0000~0x3FFF)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA0	XBISEG0	7	R/W	8051 code segment SEG0 remapping enable. **0**: Disable **1**: Enable	0x00	0xFE
		6	RSV	Reserved		
		5-0	R/W	SEG0 XBI Address SEG0 XBI Address = XBISEG0[5:0]*16K + 8051 Address[13:0]		

8051 Address Segment 1 Mapping Configuration (0x4000~0xFFFF)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA1	XBISEG1	7	R/W	8051 code segment SEG1 remapping enable. **0**: Disable **1**: Enable	0x00	0xFE
		6-4	RSV	Reserved		
		3-0	R/W	SEG1 XBI Address SEG1 XBI Address = XBISEG1[3:0]*64K + 8051 Address[15:0]		

SPI host controller configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA2	SHCCFG	7	R/W	Enable SHC Function **0**: Disable **1**: Enable When enable SHC , The address 0xFEA8 ~ 0xFEAC, 0xFEAE is changed to another bank for register control.	0x00	0xFE
		6~4	R/W	Tx length If Tx length is more than 4 , SHC translate data “0x00”		
		3	RSV	Reserved		
		2~0	R/W	Rx length If Rx length is more than 4 , SHC only receive 4 bytes data.		

LPC Read Buffer Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA3	XBI_LPBCFG	7	R/W	LPC buffer read enable **0**: disable (default) **1**: enable	0x0F	0xFE
		6	R/W	LPC buffer auto pre-fetch next 16-byte **0**: disable (default) **1**: enable		
		5-4	RSV	Reserved		
		3~0	R/W	Code-Memory Region Selection enable Code-Memory by XBICS[3] (0xFEA6), and 8051 can fetch code from XRAM region for following setting **0**: 0xEC00~0xECFF **1**: 0xEC00~0xEDFF **2**: 0xEC00~0xEEFF **3**: 0xEC00~0xEFFF ... **F**: 0xEC00~0xFBFF		

XBI XIO Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA4	RSV	7-0	RSV	Reserved	0x00	0xFE

XBI Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA5	XBICFG	7	RSV	Reserved	0x07	0xFE
		6	R/W	8051 instruction fetch (sustaining access) **0**: Disable **1**: Enable		
		5~0	RSV	Reserved		

XBI E51CS# Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA6	XBICS	7-6	RSV	Reserved	0x00	0xFE
		5	R/W	XBI arbitration priority. **0**: Disable **1**: Enable		
		4	R/W	Reset code segment enable. Once the 8051 reset, the code segment SEG0 or SEG1 can be reset if the corresponding code segment enabled. (XBISEG0[7]/XBISEG1[7]) **0**: Disable **1**: Enable		
		3	R/W	Enable XRAM as code-memory Enable 8051 to fetch code from XRAM; the code momory region is selected by XBILPBCFG[3:0] (0xFEA3). Please note, users should move codes from flash to XRAM, jump to XRAM and then enable this bit. **0**: Disable **1**: Enable		
		2	R/W	Reset XBI arbiter while in idle/stop mode. **0**: Disable **1**: Enable		
		1	R/W	EHB fast accessing enable. Enable this bit gets better performance in EHB. **0**: Disable **1**: Enable		
		0	RSV	Reserved		

XBI Write Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA7	XBIWE	7-0	R/W	XBI write command. **00h**: exit SRAM test mode **C5h**: enter SRAM test mode	0x00	0xFE

4.14.3.1 XBI Registers Bank 0 Description (SHCCFG[7]='0')

XBI SPI Flash Address (22-bit) = [SPIA2(6bit) : SPIA1(8bit) : SPIA0(8bit)]						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA8	SPIA0	7-0	R/W	SPI Address lower 8-bits (A7:A0)	0x00	0xFE
0xA9	SPIA1	7-0	R/W	SPI Address middle 8-bits (A15:A8)	0x00	0xFE
0xAA	SPIA2	5-0	R/W	SPI Address upper 6-bits (A21:A16)	0x00	0xFE

XBI SPI Flash Output/Input Data Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAB	SPIDAT	7-0	R/W	Input (read) / Output (write) data port of SPI flash interface.	0x00	0xFE

XBI SPI Flash Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAC	SPICMD	7-0	R/W	Commands support for SPI flash. Writing this register will force the SPI protocol start. Please note, the address phases must be prior to command phase. **SPI command support:** 01h: Write Status Register 02h: Byte Program 03h: Read 04h: Write Disable 05h: Read Status Register 06h: Write Enable 0Bh: High Speed Read 20h: Sector Erase (SST) 3Bh: Fast Read Dual Output (Windbond, AMIC) 6Bh Fast Read Dual IO(Winbond, ESMT) BBh Fast Read Quad Output(Winbond, ESMT,MXIC,AMIC) EBh Fast Read Quad IO(Winbond, ESMT) 50h: Enable Write Status Register (SST) 52h: Block Erase (SST) 60h: Chip Erase (SST) B9h: Power Down C7h: Chip Erase (PMC, NexFlash) D7h: Sector Erase (PMC) D8h: Block Erase (PMC, NexFlash)	0x00	0xFE

XBI SPI Flash Configuration/Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAD	SPICFG	7	R/W	Fast read dual mode enable (IO Mode or Read Output Mode is setting by SPICFG[5]) Please note, if this bit set, the SPICFG[2] will be ignored. **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	Fast Read Quad mode enable (IO Mode or Read Output Mode is setting by SPICFG[5]) Pin mapping as following: MOSI(P120): DI(IO0) MISO(P119): DO(IO1) GPXIOD6(P117): WP#(IO2) GPXIOD7(P118): HOLD#(IO3) Note that This bit has higher priority with bit 7. **0**: Disable **1**: Enable		
		5	R/W	Dual/Quad IO Mode or Output Mode Select. 0: Read Output Mode 1: IO Mode		
		4	R/W	SPICS# force low **0**: SPICS# high **1**: SPICS# low		
		3	R/W	Write enable of **SPICMD** register,0xFEAC. **0**: Disable **1**: Enable		
		2	R/W	Dummy byte of read command. **0**: Disable **1**: Enable		
		1	RO	SPI controller accessing in busy status. **0**: not busy **1**: busy		
		0	R/W	SPI flash busy status check enable Automatic SPI status check after a SPICMD issued, until SPI busy flag, clear, i.e., (SPICFG[1]=0). **0**: Disable **1**: Enable		

XBI SPI Flash Output Data for Read compare						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAE	SPIDATR	7-0	RO	Output data to SPI flash interface.	0x00	0xFE

XBI SPI Flash Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAF	RSV	7-0	RSV	Reserved	0x00	0xFE

4.14.3.2 XBI Registers Bank 1 Description (SHCCFG[7]='1')

Transmit Buffer Data Byte Write Port 2 / Receive Buffer Data Byte Read Port 3						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA8	SHC_TX2	7~0	W	Transmit Buffer Data Byte Write Port 2	0x00	0xFE
0xA8	SHC_RX3	7~0	R	Receive Buffer Data Byte Read Port 3	0x00	0xFE

Transmit Buffer Data Byte Write Port 1 / Receive Buffer Data Byte Read Port 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA9	SHC_TX1	7~0	W	Transmit Buffer Data Byte Write Port 1	0x00	0xFE
0xA9	SHC_RX2	7~0	R	Receive Buffer Data Byte Read Port 2	0x00	0xFE

Transmit Buffer Data Byte Write Port 0 / Receive Buffer Data Byte Read Port 1						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAA	SHC_TX0	7~0	W	Transmit Buffer Data Byte Write Port 0	0x00	0xFE
0xAA	SHC_RX1	7~0	R	Receive Buffer Data Byte Read Port 1	0x00	0xFE

Transmit Buffer Data Byte Write Port 3 / Receive Buffer Data Byte Read Port 0						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAB	SHC_TX3	7~0	W	Transmit Buffer Data Byte Write Port 3	0x00	0xFE
0xAB	SHC_RX0	7~0	R	Receive Buffer Data Byte Read Port 0	0x00	0xFE

SPI host control command register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAC	SHC_CMD	7~0	R/W	When writing this register, SHC will start to transmit / receive data according Tx/Rx length SHCCFG(0xFEA2) .	0x00	0xFE

SPI host Tx data read path						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAE	SHC_DEB	7~1	RSV	Reserved	0x00	0xFE
		0	R/W	Tx buffer read path enable bit **0**: Disable **1**: Enable When we enable this bit, we can read Tx buffer data.		

4.14.3.3 XBI/SPI extension registers description

XBI error pending flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB0	XBI_ERR_PF	7~2	RSV	Reserved	0x00	0xFE
		1	W1C	**Write Protection Hit Pending Flag** The pending flag would be asserted if any write protection access hit when enable Write Protection.		
		0	W1C	**Byte Mode Error Access Pending Flag** The pending flag would be asserted if any unexpected access flash when enable SPICS manual mode (0xFEAD.5 =1),		

SPI idle configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB1	SPI_IDLE_CFG	7~2	RSV	Reserved	0x00	0xFE
		1	R/W	Pull-up SPICS when SPI controller in IDLE state (Only when SPI_IDLE_CFG[0], 0xFEB1 is set to 1)		
		0	R/W	High-Z SPICS, SPICK, SPIDO when SPI controller in IDLE state		

XBI Write Protection Configuration Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB2	XBIWPCR	7	R/W	XBI Write Protect Enable When enable , the following commands are inactive: **60h Chip Erase (SST)** **C7h Chip Erase (PCM, NexFlash)** The following commands with protection region are inactive: **02h Byte Program** **20h Sector Erase (SST)** **50h Enable Write Status Register (SST)** **52h Block Erase (SST)** **D8h Block Erase (PMC, NexFlash)** **D7h Sector Erase (PCM)** **0**: Disable **1**: Enable	0x00	0xFE
		6~3	RSV	Reserved		
		2~0	R/W	XBI write protection size (Unit: 16KB) **000**: 16 KB **001**: 32 KB **010**: 48 KB ... **111**: 128KB		

XBI Write Protection Configuration Register 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB3	XBIWPCR2	7~3	RSV	Reserved	0x00	0xFE
		2~0	R/W	XBI write protection base address (Unit: 16KB) **000**: 00 0000h **001**: 00 4000h **010**: 00 8000h ... **111**: 01 C000h		

4.15 Consumer IR Controller (CIR)

4.15.1 CIR Function Description

The KBC embeds with a native hardware Consumer IR controller, which connects to system via LPC interface. Popular protocols are supported, such as RC-5/RC-6/NEC/RLC. The CIR controller handles the protocol of RC-5/RC-6/NEC/RLC for receiving, and only RLC for transmit. IRQ and I/O port are implemented. An extended function is implemented to support learning application. The basic features are list as the following table. The CIR functionality of KBx930 series is compatible to KBx926 series.

	926D	930
RX carrier demodulation	V	V
TX carrier modulation	V	V
RX protocol support	RC5/RC6/NEC/RLC	RC5/RC6/NEC/RLC
TX protocol support	RLC	RLC
RX carrier frequency measurement	V	V

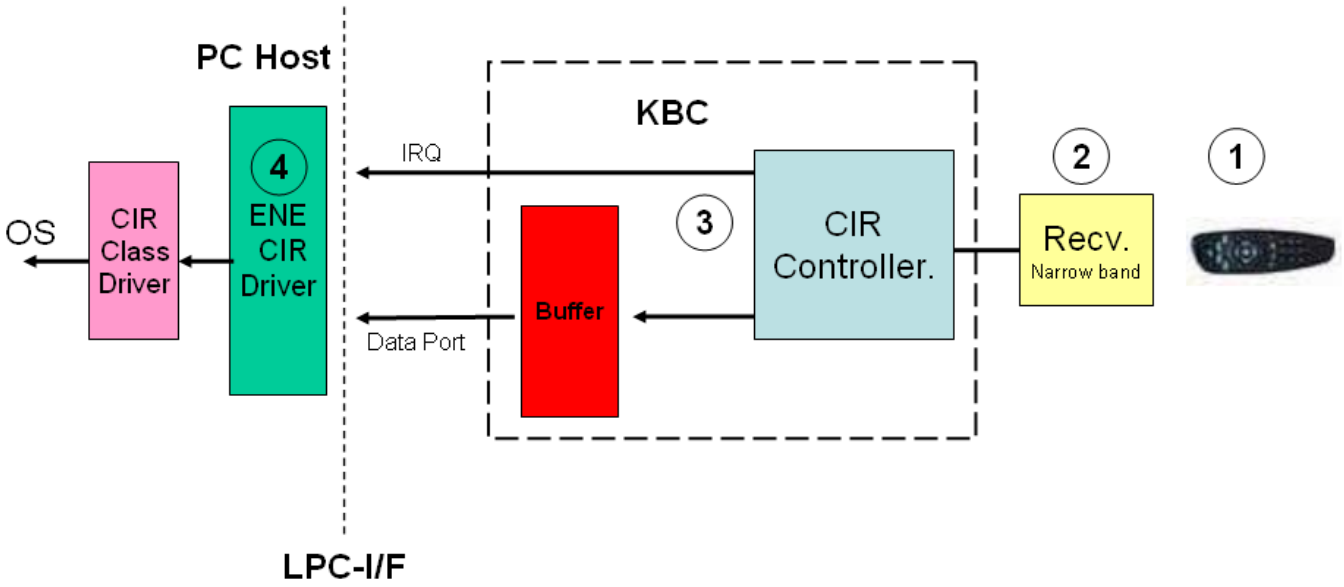
A SIRQ channel can be assigned for CIR application. The related programming registers are summarized as following table.

Register	Description
LPCSCFG[4] (0xFE90[4])	SIRQ selection for LPCTCFG[3:0] (0xFE9B[3:0]) **0**: User defined IRQ **1**: CIR IRQ enable
LPCTCFG[3:0] (0xFE9B[3:0])	SIRQ channel number. **0x00**: IRQ0 **0x01**: IRQ1 … **0x0F**: IRQ15

Here is the features highlight.

- Native hardware protocol decoder, such as RC5/RC6/NEC and RLC.
- I/O and IRQ resource for CIR controller.
- Support **2** sets of RX/TX in one chip, and RX/TX works simultaneously.
- RX carrier demodulation/ TX carrier modulation support.
- Wide range of carrier frequency support, **15K~1MHz**. (The carrier frequency is 30K~60KHz in normal application)
- More flexible in carrier sample frequency, **1μs**.**~128μs** (The sample frequencies are 25, 50 and 100μs for normal application).
- Remote controller learning support.

The following figure shows an example how a CIR controller works with narrow band receiver.



Here gives the guidance for programming CIR.

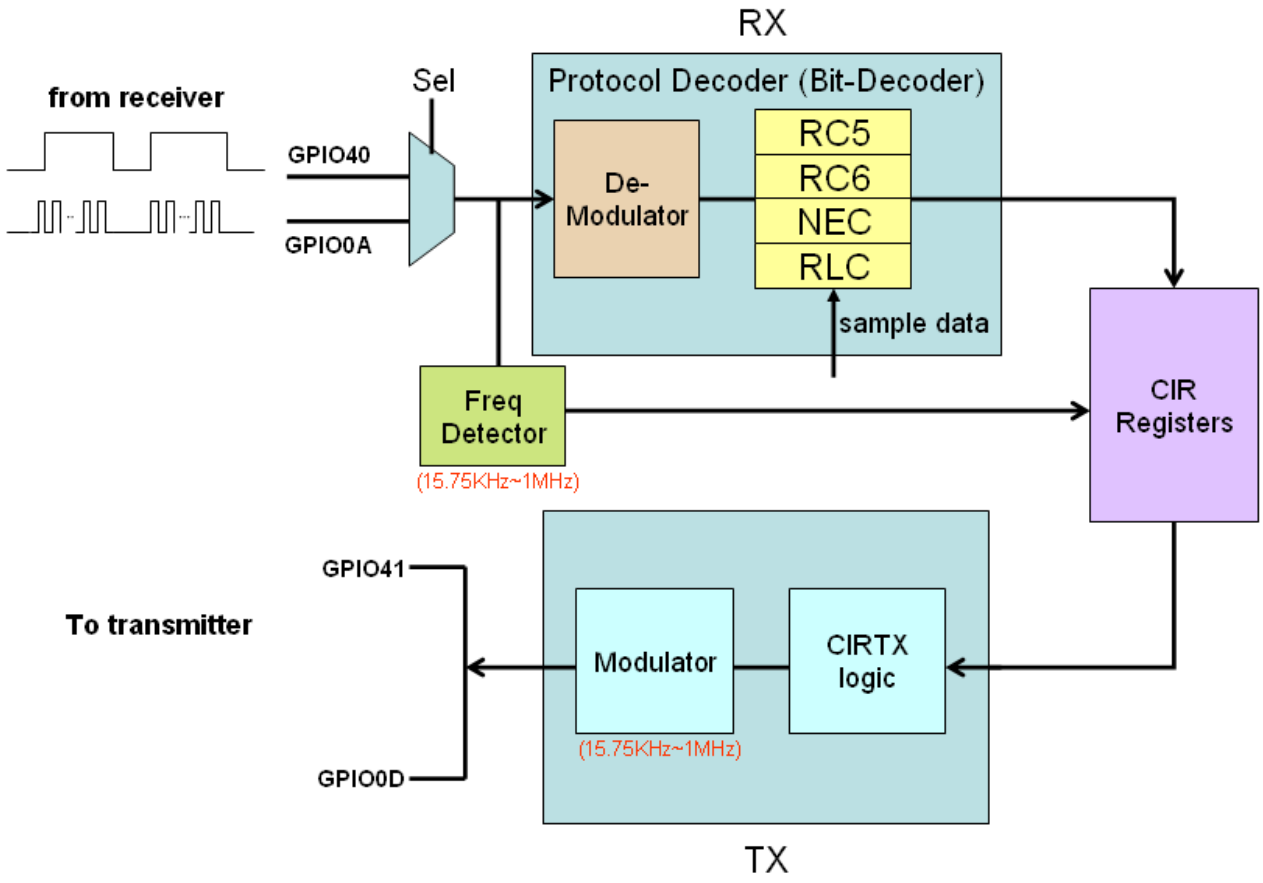
For Receive	For Transmit
1. Select protocol via setting CIRCFG2 (0xFEC1) 2. According to the selected protocol, setup CIRHIGH/CIRBIT/CIRSTART/CIRSTART2, i.e., 0xFEC3~0xFEC6 3. Enable protocol and other configuration setting via CIRCFG (0xFEC0) 4. EC F/W waits for data-in by pooling or interrupt.	1. Select RLC protocol and enable via setting CIRCFG (0xFEC0) 2. Writing to CIRRLC_OUT0, 0xFEC9, will start to transmit. 3. If CIRRLC_OUT0 (0xFEC9) and CIRRLC_OUT1 (0xFECA) are written at the same time, it start to transmit CIRRLC_OUT0 and then CIRRLC_OUT1. 4. If only CIRRLC_OUT0 (0xFEC9) is written, the hardware will transmit CIRRLC_OUT0 first and then CIRRLC_OUT1. 5. Each byte transmit completion, an interrupt will occur.

4.15.2 CIR Block Diagram

The CIR controller supports two RX ports (GPIO40/GPIO0A) and two TX ports (GPIO41/GPIO0D). A register bit, **CIRCFG2**[5] (0xFEC1[5]), is used to determine RX source. For example, if CIRCFG2[5]=0, GPIO40 is the RX source, otherwise GPIO0A. The TX port is selected according to the GPIO function selection register. The following table gives an example of RX/TX combination.

	GPIOFS08[5]=0b, GPIOFS[1]=1b	GPIOFS08[5]=1b, GPIOFS[1]=0b
CIRCFG2[5]=0b	(RX,TX)=(GPIO40, GPIO41)	(RX,TX)=(GPIO40, GPIO0D)
CIRCFG2[5]=1b	(RX,TX)=(GPIO0A, GPIO41)	(RX,TX)=(GPIO0A, GPIO0D)

The CIR controller could detect the carrier frequency and demodulate the carrier. This provides a *learning* feature for CIR application. The frequency detection range is from 15.75KHz to 1MHz. After demodulation, the CIR controller handles remote signals with hardware decoder which supports ***RC5/RC6/NEC/RLC*** protocols. If transmit function needed, the CIR controller could modulate the carrier and send it out via GPIO41/GPIO0D. The output carrier frequency range is the same as input (15.75KHz~1MHz). *The RX and TX can work simultaneously in the current design*. The following diagram gives more detail about CIR controller.



4.15.3 CIR Remote Protocol

In this section, brief introduction of protocols supported in the CIR is given. Four protocols are supported, Philips RC5/RC6, NEC and Run-Length-Code. Only features and protocol definition listed. For more detail please refer to the related specifications.

4.15.3.1 Philips RC5 Protocol

Here highlights the features of Philips RC5 protocol.

- Manufacturer Philips.
- Carrier frequency 36KHz.
- Bi-phase coding.
- 5 bits address / 6 bits command lengths

RC5 Protocol													
Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14
S1	S2	T	Address					Command					

S1/S2: start bits, always “1”
T: toggle bit, This bit is inverted every time a key is released and pressed again.
Address: IR device address, MSB first.
Command: IR command, MSB first.

4.15.3.2 Philips RC6 Protocol

Here highlights the features of Philips RC6 protocol.

- Manufacturer Philips.
- Carrier frequency 36KHz.
- Bi-phase coding.
- 5 bits address
- Variable command lengths based on the operation mode.

RC6 Protocol																					
LS	SB	MB2	MB1	MB0	T	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0	C 7	C 6	C 5	C 4	C 3	C 2	C 1	C 0	
Header						Control								Information								SF
Header Phase (ENE CIR)						*Data Phase (ENE CIR)*																

LS: Leader symbol
SB: Start bit, always “1”
MB2-MB0: Mode bits, operation mode selection.
T: Trailer bit, this bit can be served as a toggle bit.
A7-A0: Address
C7-C0: Command
SF: Signal free time, 2.666ms.

4.15.3.3 NEC Protocol

Here highlights the features of NEC protocol.

- Manufacturer NEC.
- Carrier frequency 38KHz.
- Pulse distance modulation.
- 8 bit address / 8 bit command length
- Address/Command transmitted twice.
- Total transmit time is constant.

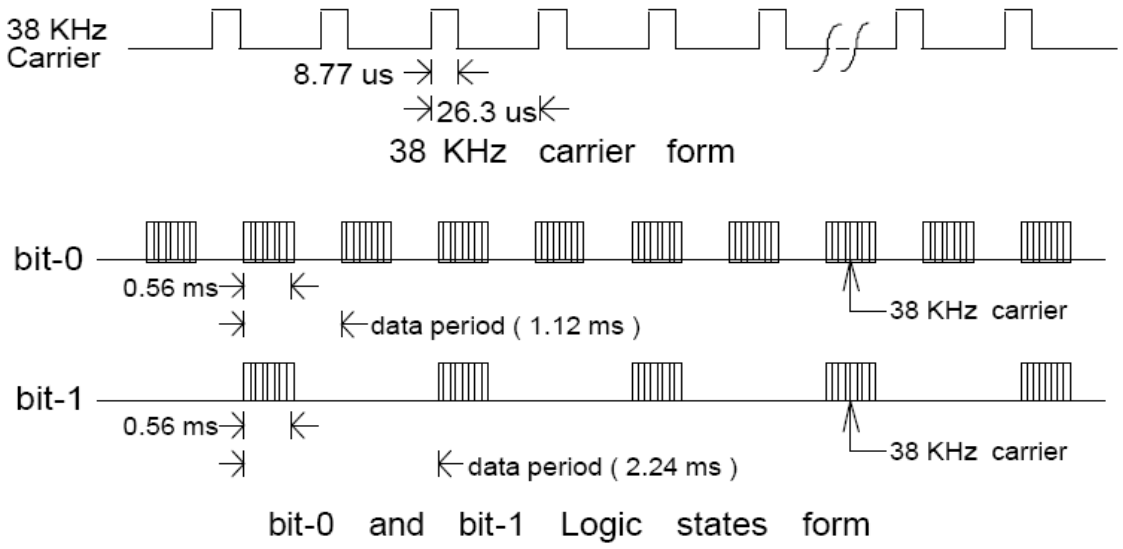
NCE Protocol					
AGC burst	space	Address	～Address	Command	～Command
9ms	4.5ms	8bit	8bit	8bit	8bit

AGC burst: set gain of IR remote controller, 9ms long
Space: follow by AGC burst, 4.5ms.
Address: 8-bit address, LSB first.
~Address: inverted 8-bit address, LSB first.
Command: 8-bit command, LSB first.
~Command: inverted 8-bit command, LSB first

4.15.4 CIR Automatic Carrier Frequency Detection and Modulation

To support learning feature, wide-band transmitter and receiver will be used in a system. The KBC introduces a new mechanism to provide carrier frequency information of wide-band receiver to the host.

The CIR controller needs to be programmed with two parameters for the detection. Register **CIRCAR_PULS** is used to determine these two parameters. **CIRCAR_PULS**[7:4] keeps the discard number of carrier pulse and **CIRCAR_PULS**[3:0] keeps the average number to detect. The **CIRCAR_PULS**[7:4] tells the controller to discard the specific number of carrier pulse from the beginning. The controller then gets the average number of carriers pulse as sample data and analyzes. The detection of carrier period is kept in **CIRCAR_PRD**[6:0], and the valid flag is kept in **CIRCAR_PRD**[7]. Please note, the detection range is from *15.75KHz~1MHz*. (The general application is from 30K~60KHz).



Here gives an example as the above waveform. Bit stream with 38KHz carrier is shown as bit-0. Each bit is 0.56ms in length and 38KHz carrier period is 26.3 μs, that is, there will be about 21 carrier pulses in a bit. If **CIRCAR_PULS**[7:4]=5 and **CIRCAR_PULS**[3:0]=10, once the detection enabled, the CIR controller will get 6th carrier pulse as the first one and analyze the sequential 10 pluses. The detection result can be obtained via register **CIRCAR_PRD**.

The related registers for automatic carrier frequency detection are listed as following.

Register	Address	Description
CIRCFG2[5:4]	0xFEC1[5:4]	Bit5=1, select wide-band as bit-decoder input. Bit4=1, enable wide-band frequency detection
CIRCAR_PULS	0xFECB	CIRCAR_PULS[7:4] = discard number of carrier pulse CIRCAR_PULS[3:0] = average number of carrier pulse
CIRCAR_PRD	0xFECC	Detection of wide-band carrier period
CIRCAR_HPRD	0xFECD	Detection of wide-band carrier period, pulse width high.

The KBC provides the modulation ability for RLC transmit. The carrier frequency of modulation can be programmable. Before the carrier modulation, the programmer should notice the modulation polarity. That is, if the data bus (TX) is kept low in idle state, only data in high state will be modulated and the bit, **CIRMOD_PRD**[7], should be "**1**".

The related registers for RLC modulation is summarized as below.

Register	Address	Description
CIRCFG[7]	0xFEC0	RLC output modulation enable.
CIRMOD_PRD	0xFECE	CIRMOD_PRD[7] = modulation polarity selection CIRMOD_PRD[6:0] = modulation carrier period
CIRMOD_HPRD	0xFECF	CIRMOD_HPRD[6:0] = modulation carrier period, pulse width high.

4.15.5 CIR Registers Description

CIR Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC0	CIRCFG	7	R/W	Output carrier modulator for RLC (TX) **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	Output polarity reversed for RLC. (TX) **0**: Disable **1**: Enable		
		5	R/W	Interrupt while transmit completes with RLC protocol. (TX) **0**: Disable **1**: Enable		
		4	R/W	Output enable for RLC protocol. (TX) Once the data filled into CIRRLC_OUT1 (0xFECA), the controller starts the transmit with RLC protocol **0**: Disable **1**: Enable		
		3	R/W	Input carrier demodulator. (RX) **0**: Disable **1**: Enable		
		2	R/W	Input polarity reversed. (RX) **0**: Disable **1**: Enable		
		1	R/W	Interrupt enable. (RX) Two conditions issue interrupt. 1. After decode a byte in RX 2. Once receive the “Repeat” in NEC protocol **0**: Disable **1**: Enable		
		0	R/W	Protocol decode enable. (RX) The protocol type is determined by CIRCFG2[3:0] (0xFEC1) **0**: Disable **1**: Enable,		

CIR Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC1	CIRCFG2	7	R/W	Fast sample (data phase, not leader phase) enable for input signal. If this bit set, the sample period changes. For RC5/RC6, period changes from 30 μs to 16 μs For NEC, period changes from 64 μs to 30 μs **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	Fast sample (leader phase) enable for input signal. If this bit set, the sample period changes. For RC6, period changes from 64 μs to 30 μs **0**: Disable **1**: Enable		
		5	R/W	Input selection for protocol decoder (bit-decoder) **0**: from GPIO40 **1**: from GPIO0A		
		4	R/W	Frequency detection enable. **0**: Disable **1**: Enable		
		3-0	R/W	CIR Protocol selection. (valid while CIRCFG[0]=1) **000**: RLC **001**: RC5 **010**: RC6 **011**: NEC **others**: reserved.		

CIR Pending Flag and Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC2	CIRPF	7	RO	Hardware RX idle state. **0**: not idle state **1**: idle state	0x00	0xFE
		6	RO	Hardware TX (RLC) idle state. **0**: not idle state **1**: idle state		
		5-4	RSV	Reserved		
		3	R/W1C	Pending flag of RLC transmit complete **0**: no event **1**: event occurs		
		2	R/W1C	Pending flag of RLC receive counter overflow **0**: no event **1**: event occurs		
		1	R/W1C	Pending flag of NEC repeat protocol **0**: no event **1**: event occurs		
		0	R/W1C	Pending flag of data-in This bit is set while data received and stored in **CIRDAT_IN**. **0**: no event **1**: event occurs		

Value for High Pulse Width						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC3	CIRHIGH	5-0	R/W	This register determines the high pulse width of a "logic bit". High pulse width = Decoder sample period * **CIRHIGH**	0x00	0xFE

Value for Bit Width(RC5/RC6) / Logic Bit-One (NEC)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC4	CIRBIT	6-0	R/W	This register determines the bit width of a "logic bit". (RC5/RC6) Bit width = Decoder sample period * **CIRBIT** This register determines the "logic bit-one". (NEC) Logic bit-one = Decoder sample period * **CIRBIT**	0x00	0xFE

Value for Leader Pulse Width (RC6/NEC) for Normal Packet						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC5	CIRSTART	6-0	R/W	This register determines the leader pulse width for normal packet (RC6/ENC) Leader pulse width = Decoder sample period * **CIRSTART**	0x00	0xFE

Value for Tailer Bit Width (RC6) / Leader Width of Repeat Packet (NEC)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC6	CIRSTART2	6-0	R/W	This register determines the bit width of trailer (RC6) trailer bit width = Decoder sample period * **CIRSTART2** This register determines the leader width of repeat packet (NEC) Leader width(repeat) = Decoder sample period * **CIRSTART2**	0x00	0xFE

CIR Decode Data Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC7	CIRDAT_IN	7-0	RO	Received data to decode.	0x00	0xFE

CIR Counter Value for RLC Sample Period						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC8	CIRRLC_CFG	7	R/W	Counter overflow control bit. **0**: if overflow, the counter will stop. **1**: if overflow, an interrupt issues and the counter keeps counting.	0x00	0xFE
		6-0	R/W	CIR RLC sample period, The unit is **1** μs. Please note CIRRLC_CFG[6:0] can not be zero.		

CIR RLC Output 1st Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC9	CIRRLC_OUT0	7-0	R/W	Output (TX) 1st byte for RLC protocol.	0x00	0xFE

CIR RLC Output 2nd Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCA	CIRRLC_OUT1	7-0	R/W	Output (TX) 2nd byte for RLC protocol.	0x00	0xFE

CIR Carrier Discard/Average Pulse Number Setting for Automatic Carrier Detection.						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCB	CIRCAR_PULS	7-4	R/W	Discard carrier pulse number F/W should specify the number of pulse to discard	0x44	0xFE
		3-0	R/W	Average carrier pulse number F/W should specify the average number to calculate the carrier period.		

CIR Detected Carrier Period						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCC	CIRCAR_PRD	7	RO	Detected carrier period valid. **0**: carrier detection not completed. **1**: carrier detection completed.	0x00	0xFE
		6-0	RO	Detected carrier period. Detected carrier period = CIRCAR_PRD[6:0] x 500ns		

CIR Detected Pulse Width High of Carrier						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCD	CIRCAR_HPRD	7	RSV	Reserved	0x00	0xFE
		6-0	R/W	Detected pulse width high of carrier Pulse width high = CIRCAR_HPRD[6:0] x 500ns		

CIR Modulation Carrier Period (RLC only)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCE	CIRMOD_PRD	7	R/W	Carrier modulation selection. **0**: If TX idle state is high ,only low signal in TX will be modulated **1**: If TX idle state is low, only high signal in TX will be modulated	0x00	0xFE
		6-0	R/W	Modulation carrier period. This register determines the modulation carrier period. The unit is 500ns. The value can be chosen from 0x02 to 0x7F, i.e., the period is from 15.87KHz~1MHz. The period = CIRMOD_PRD[6:0] x 500 ns.		

CIR Pulse Width High of Modulation Carrier (RLC only)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCF	CIRMOD_HPRD	7	R/W	Reserved	0x00	0xFE
		6-0	R/W	Pulse width high of modulation carrier. This register determines the pulse width high of modulation carrier. The unit is 500ns. The value can be chosen from 0x01 to 0x7E. *Please note, the pulse width high can not be larger than the carrier period.* The pulse width high = CIRMOD_HPRD[6:0] x 500 ns.		

4.15.3 CIR Programming Sample

In this section gives some programming sample to control CIR module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of CIR filed application.

Example
A RC6 receiver which filters out carrier is connected to CIR RX pin.
Programming model

GPIOIE40[0] (0xFC68[0]) = 1; //Enable CIR Rx input
CIRCFG (0xFEC0) = 0x07; //Enable Rx interrupt and protocol
CIRCFG2 (0xFEC1) = 0x02; //Select RC-6 protocol
CIRHIGH (0xFEC3) = 0x0B; //High width = 32*11 = 352 us
CIRBIT (0xFEC4) = 0x22; //Bit width = 32*34 = 1088 us
CIRSTART (0xFEC5) = 0x3B; //Leader width = 64*59 = 3776 us
CIRSTART2 (0xFEC6) = 0x4A; //Trailer width = 32*74 = 2368 us
When CIRPF[0] (0xFEC2[0]) = 1, Read CIRDAT_IN (0xFEC7) to get data.

4.16 PS/2 Interface (PS/2)

This page is leaved blank intentionally.

4.17 Embedded Controller (EC)

4.17.1 EC Function Description

The ACPI specification defined for the embedded controller (EC) interface requires either three separate host interfaces (KBC, SCI, SMI) or two interfaces (KBC, and shared SCI/SMI). The ENE KBC supports KBC and SCI interface, and SMI interface can be shared with SCI or use a dedicated GPIO. The embedded controller also provides some features which are collected as following:

- Handles EC standard commands from host, firmware mode support.
- Handles EC extended commands from host, only firmware mode support.
- SCI generation capability.
- Extended I/O write interface, i.e., debug port (port 80) support.
- KBC/EC clock configuration.
- A/D and D/A control.
- Power management control.
- Miscellaneous control.

The host queries (read) EC status and issues (write) EC command via port **66h**. The EC data port is **62h**. The status of EC is defined as the below table:

Status Bit	Name	Description
7	RSV	Reserved
6	RSV	Reserved
5	SCI	SCI event flag. Please note, this bit will not be set if standard EC commands (80h~84h) issued by host. **0:** No SCI event occurs **1:** SCI event occurs
4	Burst Enable	The burst enable flag **0:** Disable **1:** Enable
3	Command/Data Flag	**0:** Previous access port is data port. (EC_DAT) **1:** Previous access port is command/status port. (EC_CMD/EC_STS)
2	RSV	Reserved
1	IBF	**I**nput **B**uffer **F**ull flag of EC
0	OBF	**O**utput **B**uffer **F**ull flag of EC

The EC commands are defined as following, for more detail please refer to ACPI, *Advanced Configuration Power Interface Specification. 2.0*

Value	Command	Description
80h	EC Read	Read EC space registers
81h	EC Write	Write EC space registers
82h	EC Burst Enable	Enable EC operation in burst mode
83h	EC Burst Disable	Disable EC operation in burst mode
84h	EC Query	Query SCI events
Others	Firmware Command	Extended commands and handled with F/W mode.

4.17.2 EC Command Program Sequence

The following table summarizes the standard EC commands programming flow. Port **66h** is the EC command and status port and port **62h** is the EC data port.

Command Byte	Command	Program Sequence
80h	EC Read	1. Host writes command byte 80h (EC_Read) to port 66h. 2. EC will issue SCI to host while IBF=0 3. Host writes address to port 62h. 4. EC will issue SCI to host while OBF=1 5. Host reads data via port 62h.
81h	EC Write	1. Host writes command byte 81h (EC_Write) to port 66h. 2. EC will issue SCI to host while IBF=0 3. Host writes address to port 62h. 4. EC will issue SCI to host while IBF=0 5. Host writes data to port 62h. 6. EC will issue SCI to host while IBF=0
82h	Burst Enable	1. Host writes command byte 82h (Burst_Enable) to port 66h. 2. EC will issue SCI to host while OBF=1. 3. Host reads via port62h. If 90h obtained, it's Burst Ack.
83h	Burst Disable	1. Host writes command byte 83h (Burst_Disable) to port 66h. 2. EC will issue SCI to host while IBF=0
84h	Query EC	1. Host writes command byte 84h (Query_EC) to port 66h. 2. EC will issue SCI to host while OBF=1. 3. Host reads data via port 62h. The data obtained is SCI_ID number.

4.17.3 EC SCI Generation

The EC can generate SCI with independent enable control and status flag. Plenty of hardware SCI events are predefined, and a firmware SCI event gives more flexible use for different applications. There is a F/W SCI command port located at **SCID** (0xFF0B). As the F/W writes any **non-zero** value to this port, and corresponding enable bit (SCIE0[6]) is set. A hardware SCI signal will issue to host in sequence. Then the host uses standard EC_Query (84h) command to get the **SCI ID** which is written by F/W before. The below table summarizes the information about SCI events, SCI IDs and the priorities.

SCI ID	Event	Switch	Applications	Priority
00h	Nothing	N/A		0(Highest)
01h-07h	RSV	N/A	Reserved	1
08h	WDT	SCIE0[0]	Watchdog	2
09h	LPC_IO2F / OWM	SCIE0[1]	LPC I/O 0x2F R/W accessing interrupt / OWM	3
0Ah	PS2	SCIE0[2]	PS/2 event	4
0Bh	KBC	SCIE0[3]	IBF rising (LPC write I/O 60h/64h) OBF falling (LPC read I/O 60h)	5
0Ch	IKB	SCIE0[4]	IKB	6
0Dh	LPC_IO686C	SCIE0[5]	IBF rising (LPC write I/O 68h/6Ch) OBF falling (LPC read I/O 68h)	7
0Eh	LPC_IO6266	SCIE0[6]	IBF rising (LPC write I/O 62h/66h) OBF falling (LPC read I/O 62h)	8
FW_SCIID	*FW_SCI*	*SCIE0[7]*	*EC F/W SCI event*	9
10h	FAN0	SCIE1[0]	FAN0 monitor event (update/overflow)	10
11h	FAN1	SCIE1[1]	FAN1 monitor event (update/overflow)	11
12h	SMBus	SCIE1[2]	SMBus events	12
13h	CIR	SCIE1[3]	CIR events	13
14h	GPT0	SCIE1[4]	GPT0 event	14
15h	GPT1	SCIE1[5]	GPT1 event	15
16h	GPT2	SCIE1[6]	GPT2 event	16
17h	GPT3 / SDI	SCIE1[7]	GPT3 event /SDI	17
18h	EXTWIO / PECI	SCIE3[0]	Write extended I/O (LPC I/O port 80)	18
19h	GPIO00~GPIO0F	SCIE3[1]	GPIO00~GPIO0F	19
1Ah	GPIO10~GPIO1F	SCIE3[2]	GPIO10~GPIO1F	20
1Bh	GPIO20~GPIO2F	SCIE3[3]	GPIO20~GPIO2F	21
1Ch	GPIO30~GPIO3F	SCIE3[4]	GPIO30~GPIO3F	22
1Dh	GPIO40~GPIO4F	SCIE3[5]	GPIO40~GPIO4F	23
1Eh	GPIO50~GPIO5F	SCIE3[6]	GPIO50~GPIO59 / GPXIOD00~GPXIOD07	24
1Fh	ADC	SCIE3[7]	ADC update	25(Lowest)

The SCI pulse width is programmable for different applications. Two unit basis, 16μs and 64 μs can be chosen. To change the SCI pulse width, register **PXCFG**[2] (0xFF14) is to select the timing base unit and **SCICFG**[3:0] (0xFF03) is to decide another coefficient. The SCI pulse is decided by the following equation. Please refer to registers description for details.

*SCI Pulse Width = SCICFG[3:0] * Unit (16μs or 64 μs)*

4.17.4 EC/KBC Clock Configuration

The EC provides programmable interface to adjust the microprocessor and peripheral frequency. By default, the microprocessor runs at 8MHz and peripherals are at 4MHz. The microprocessor can operate at 32MHz as the highest frequency, and the peripheral runs up to 7.2 MHz. The programming interface is located at register **CLKCFG/CLKCFG2** (0xFF0D/0xFF1E) and **PLLCFG/PLLCFG2** (0xFF0F/0xFF1F). The figure 4-1 (in section **Clock Domain**) illustrates the clock scheme applied in the KBC.

4.17.5 A/D Converter Control

The control interface of A/D is in the EC space. Details SPEC of the A/D converters could be found in the electronics characteristic chapter.

The following table summarizes the related registers of these 6 A/D converters.

Name	Address	Description
ADDAEN[3:0]	0xFF15	ADC port enable bits of ADC3~ADC0 Bit3: ADC3 Bit2: ADC2 Bit1: ADC1 Bit0: ADC0 If ADC selected, please do not set related IE register.
ADCTRL[6:5]	0xFF18	ADC port enable bits of ADC5~ADC4 Bit6: ADC5 Bit5: ADC4
ADCTRL[4:2]	0xFF18	ADC channels selection to be converted and put in ADCDAT and ECIF[7:6]
ADCDAT	0xFF19	This stands for higher bit9~bit2 of 10bit A/D result.
ECIF[7:6]	0xFF1A	This stands for lower bit1~bit0 of 10bit A/D result.

The following gives the programming sample to control ADC.

Example
Using ADC0 to get input analog signal
Programming model
1. Clear IE of the related pin GPIOIE38[0] (0xFC67[0]) = 0b 2. Enable ADC function ADDAEN[0] (0xFF15[0]) = 1b 3. Enable ADC control ADCTRL (0xFF18) = 0x01 Waiting ADC interrupt. 4. Read ADCDAT (0xFF19) and ECIF (0xFF1A)

4.17.6 D/A Converter Control

The control interface of D/A is in the EC space. Details SPEC of the D/A converters could be found in the electronics characteristic chapter.

The following table summarizes the related registers of these 4 D/A converters.

Name	Address	Description
ADDAEN[7:4]	0xFF15	DAC port Enable bits of DAC3~DAC0 Bit7: DAC3 Bit6: DAC2 Bit5: DAC1 Bit4: DAC0 If DAC selected, please do not set related GPIO function selection register.
DAC0	0xFF10	DAC0 Output Value
DAC1	0xFF11	DAC1 Output Value
DAC2	0xFF12	DAC2 Output Value
DAC3	0xFF13	DAC3 Output Value

The following gives the programming sample to control a DAC.

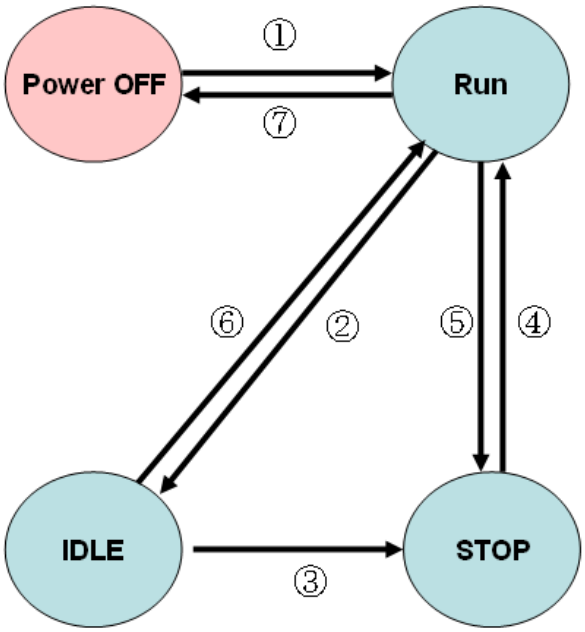
Example
Using DAC2
Programming model
1. Clear the alternative function selection of the related pin GPIOFS38[6] (0xFC07[6]) = 0b ; 2. Enable DAC function ADDAEN[6] (0xFF15[6]) = 1b 3. Fill the value to be converted. DAC2 (0xFF12) = specific value to convert

4.17.7 Power Management Control

Two power modes are defined, one is **STOP** mode and the other is **IDLE** mode. The register **PMUCFG** (0xFF0C) is used to configure the power management. The following table gives more detail about the definition for these two power modes.

Mode	Description
STOP	All clock sources stop, except external PCI clock and 32.768KHz.
IDLE	Only clock of 8051 microprocessor stops.
RUN	System operations in normal mode.
OFF	All power supply removed, including AC and battery

The diagram below shows the relationship between each power mode.



① power supply on

② enter idle mode from run mode
set PMUCFG[6]=1

③ enter stop mode from idle mode
set PMUCFG[7]=1 (by Host)

④ back to run mode from stop mode
by wakeup source

⑤ enter stop mode from run mode
set PMUCFG[7]=1

⑥ back to run mode from idle mode
by wakeup source

⑦ all power removed

4.17.8 EC Registers Description

EC Hardware Revision ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x00	ECHV	7-0	RO	EC Hardware version	0xA0	0xFF

EC Firmware Revision ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x01	ECFV	7-0	R/W	EC firmware version This register will be a data port, **ADC_test_data**[7:0] in ADC test mode (ADCTRL[1]=1).	0x00	0xFF

EC High Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x02	ECHA	7-6	R/W	These two bits will be a data port, **ADC_test_data**[9:8] in ADC test mode (ADCTRL[1]=1).	0x0F	0xFF
		5	R/W	Write protection of PXCFG[1], PXCFG[4]. **0**: writable. **1**: write protection.		
		4	R/W	Index-I/O mode access control. **0**: access range 0xF400~0xFFFF **1**: access range 0xF400~0xF403 and 0xFC00~0xFFFF		
		3-0	RSV	Reserved		

EC SCI Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x03	SCICFG	7	R/W	Standard EC commands generate SCI. **0**: Disable **1**: Enable	0x90	0xFF
		6	R/W	**SCIID** port enable. (F/W SCI write port enable) **0**: Disable **1**: Enable		
		5	R/W	SCI polarity **0**: Low active (default) **1**: High active		
		4	R/W	SCIE0/SCIE1/SCIE2 (0xFF05~0xFF07) enable. **0**: Disable **1**: Enable		
		3-0	R/W	SCI pulse width. (max. 1ms) *SCI pulse width = SCICFG[3:0] * (time unit)* where time unit is determined by PXCFG[2], **64µs or 16µs** If SCICFG[3:0]=0, SCI pulse width = width of system clock.		

EC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x04	ECCFG	7	R/W	EPB fast access enable. To enhance EPB performance. **0**: Disable **1**: Enable	0x00	0xFF
		6	R/W	Test mode selection **0**: Normal mode **1**: Test mode.		
		5~3	RSV	Reserved		
		2	R/W	Extended I/O (debug I/O, port 80) interrupt enable. Only available while write cycle to port 80 from the host. **0**: Disable **1**: Enable		
		1	R/W	Reserved		
		0	R/W	OBF interrupt enable. EC data port interrupt enable. CPU reads data from EC data port. **0**: Disable **1**: Enable		

EC SCI Interrupt Enable (SCIE0,SCIE1,SCIE3)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x05	SCIE0	7-0	R/W	SCI Event0 enable **0**: Disable **1**: Enable	0x00	0xFF
0x06	SCIE1	7-0	R/W	SCI Event1 enable **0**: Disable **1**: Enable	0x00	0xFF
0x07	SCIE3	7-0	R/W	SCI Event3 enable **0**: Disable **1**: Enable	0x00	0xFF

EC SCI Flag (SCIF0,SCIF1,SCIF3)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x08	ECIF0	7-0	R/W1C	SCI Event0 flag **0**: no event **1**: event occurs	0x00	0xFF
0x09	ECIF1	7-0	R/W1C	SCI Event1 flag **0**: no event **1**: event occurs	0x00	0xFF
0x0A	ECIF3	7-0	R/W1C	SCI Event3 flag **0**: no event **1**: event occurs	0x00	0xFF

EC SCI ID Write Port (to Generate SCI Event)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0B	SCID	7-0	R/W	Firmware SCI write port	0x00	0xFF

EC PMU Control/Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0C	PMUCFG	7	WO	Write "**1**" to enter STOP mode.	0x2F	0xFF
		6	WO	Write "**1**" to enter Idle mode.		
		5	R/W	LPC cycle wakeup system from STOP mode. **0**: Disable **1**: Enable		
		4	R/W	Reset 8051 while in STOP mode. **0**: Disable **1**: Enable		
		3	R/W	SCI wakeup system **0**: Disable **1**: Enable		
		2	R/W	WDT wakeup system from STOP mode. **0**: Disable **1**: Enable		
		1	R/W	GPWU wakeup system from STOP mode. **0**: Disable **1**: Enable		
		0	R/W	Interrupt wakeup system from Idle mode. **0**: Disable **1**: Enable		

EC Clock Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0D	CLKCFG	7	R/W	Flash clock from external clock (GPIO59). **0**: Disable **1**: Enable	0x00	0xFF
		6	R/W	Flash clock control. **0**: Half speed. (DPLL_CLK/2) **1**: Full speed (DPLL_CLK) please note, while CLKCFG[6]=0 and CLKCFG[3:2]=0 (power-on default), the SPI flash clock is always 16MHz.		
		5	R/W	Enable DPLL to generate 32.768 MHz **0**: Disable **1**: Enable		
		4	R/W	DPLL enters low power state while in STOP mode. **0**: Disable **1**: Enable		
		3-2	R/W	8051/Peripheral clock selection. **11b**: 32 MHz / 16 MHz **10b**: 22 MHz / 11 MHz **01b**: 16 MHz / 8 MHz **00b**: 8 MHz / 4 MHz (default)		
		1	R/W	Peripheral slow down to 1MHz automatically. If no host access, the peripheral clock will slow down to 1MHz automatically. **0**: Disable **1**: Enable		
		0	R/W	Clock slow down to 2MHz / 1MHz (8051 / Peripheral) in Idle mode. If this bit set, the clock of flash will be stopped in idle mode. **0**: Disable **1**: Enable		

EC Extended I/O (Debug Port) Write Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0E	EXTIOW	7-0	R/W	If the host write data to extended I/O (debug port, port80), an interrupt occurs, and then the firmware read it back via this register.	0x00	0xFF

EC PLL Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0F	PLLCFG	7-0	R/W	DPLL initial value. (low 8-bit) After reset, the DPLL will output frequency about 32MHz with default value 0xD0. DPLL initial value is 10-bit, the higher two bits are located at 0xFF1F, **PLLCFG2**[7:6].	0xD0	0xFF

EC DAC0 Output Value (ECMISC[1:0]=00b) / Extended Command (ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x10	DAC0	7-0	R/W	The digital data to be converted in DAC0.	0x00	0xFF
0x10	EXTCMD	7-0	R/W	8051 extended command port. Once the command is filled, two events may occur. - if non-zero command written, 8051 interrupt issues. - If zero command written, SCI event issues. Please note, EXTARG0/EXTARG1/EXTARG2 must be ready before filling this register.	0x00	0xFF

EC DAC1 Output Value (ECMISC[1:0]=00b)/ Extended Command Argument 0(ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x11	DAC1	7-0	R/W	The digital data to be converted in DAC1.	0x00	0xFF
0x11	EXTARG0	7-0	R/W	Extended command argument0	0x00	0xFF

EC DAC2 Output Value (ECMISC[1:0]=00b)/ Extended Command Argument 1(ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x12	DAC2	7-0	R/W	The digital data to be converted in DAC2.	0x00	0xFF
0x12	EXTARG1	7-0	R/W	Extended command argument1	0x00	0xFF

EC DAC3 Output Value (ECMISC[1:0]=00b)/ Extended Command Argument 2(ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x13	DAC3	7-0	R/W	The digital data to be converted in DAC3.	0x00	0xFF
0x13	EXTARG2	7-0	R/W	Extended command argument2	0x00	0xFF

EC 8051 On-Chip Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x14	PXCFG	7-5	RSV	Reserved	0x00	0xFF
		4	R/W	Setting for WDT timeout reset of GPIO This field is only valid when PXCFG[1]='0' To write this bit, set the field on ECHA[5]='0' **0**: GPIO module when WDT timeout reset will not be reset. **1**: GPIO module when WDT timeout reset will be reset.		
		3	RSV	Reserved		
		2	R/W	SCI pulse width time unit. **0**: 64μs **1**: 16μs		
		1	R/W	WDT timeout reset selection **0**: reset whole KBC, except GPIO module. **1**: reset 8051 only To write this bit, set the field on ECHA[5]='0'		
		0	R/W	8051 program counter control **0**: program counter starts to execute. **1**: 8051 reset and PC=0 PC will keep 0 (reset vector) until this bit is written to "0"		

EC ADC/DAC Function Switch						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x15	ADDAEN	7-4	R/W	DAC3~DAC0 Function Enable Bit7~Bit4 represents DAC3~DAC0 respectively **0:** DAC Disable **1:** DAC Enable If DAC enable, please **do not** set related GPIO function selection register.	0x00	0xFF
		3-0	R/W	ADC3~ADC0 Function Enable Bit3~Bit0 represents ADC3~ADC0 respectively **0:** ADC Disable **1:** ADC Enable. If ADC enable, please **do not** set related GPIO bit with input enable (IE).		

EC PLL Frequency Register (High Byte)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x16	PLLFRH	7-0	R/W	DPLL frequency = 32.768KHz(external) * PLLFR PLLFR[11:0] =(**PLLFRH**[7:0] : PLLFRL[7:4]) To generate 32.768MHz, PLLFR = 1000 (decimal) = 0x3E8 i.e., PLLFRH=0x3E	0x3E	0xFF

EC PLL Frequency Register (Low Byte)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x17	PLLFRL	7-4	R/W	DPLL frequency = 32.768KHz * PLLFR PLLFR[11:0] =(PLLFRH[7:0] : **PLLFRL**[7:4]) To generate 32.768MHz, PLLFR = 1000 (decimal) = 0x3E8 i.e., PLLFRL[7:4]=0x8	0x83	0xFF
		3	R/W	DPLL lock value presented in **CHIPID** (0xFF1E~0xFF1F). **0**: Disable **1**: Enable.		
		2	R/W	DPLL test mode enable **0**: Disable **1**: Enable.		
		1-0	RSV	Reserved		

EC ADC Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x18	ADCTRL	7	RSV	Reserved	0x00	0xFF
		6-5	R/W	ADC5, ADC4 enable. Bit6 and Bit5 represent for ADC5 and ADC4 respectively. **0**: Disable **1**: Enable.		
		4-2	R/W	Convert ADC channel selection. **000**: ADC0 **001**: ADC1 **010**: ADC2 **011**: ADC3 **100**: ADC4 **101**: ADC5		
		1	R/W	ADC test mode enable. **0**: Disable **1**: Enable.		
		0	R/W	ADC convert start and force interrupt after converting. **0**: ADC stops converting, interrupt disable **1**: ADC starts converting, interrupt enable		

EC ADC Data Output Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x19	ADCDAT	7-0	RO	Converted data by ADC. ADC output[9:2]=**ADCDAT**[7:0]	0x00	0xFF

EC Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1A	ECIF	7-6	RO	Converted data by ADC. ADC output[1:0]=**ECIF**[7:6]	0x00	0xFF
		5-3	RSV	Reserved		
		2	R/W1C	EC firmware mode flag. If EC command handled by F/W, this flag will be set		
		1	R/W1C	EC IBF interrupt pending flag **0**: no event **1**: event occurs		
		0	R/W1C	EC OBF interrupt pending flag **0**: no event **1**: event occurs		

EC Data Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1B	ECDAT	7-0	R/W	EC data port. If **ECDAT** written, **ECSTS**[0] (OBF) becomes “1”.	0x00	0xFF

EC Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1C	ECCMD	7-0	RO	This register keeps EC command issued by the host.	0x00	0xFF

EC Control and Status Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1D	ECSTS	7	R/W	Reserved	0x00	0xFF
		6	R/W	Reserved		
		5	RO	SCI pending flag **0**: no event **1**: event occurs		
		4	R/W	Burst enable status. **0**: EC burst mode disable **1**: EC burst mode enable.		
		3	R/W	EC I/O write port indicator **0**: host writes for data (writes I/O port 62h) **1**: host writes for command (writes I/O port 66h)		
		2	R/W	Register 0xFF1E and 0xFF1F function select. **0**: CHIPID display selected **1**: CLKCFG2/PLLCFG2 function selected		
		1	R/W1C	IBF (Input Buffer Full) **0**: buffer not full **1**: buffer full		
		0	R/W1C	OBF (Output Buffer Full) **0**: buffer not full **1**: buffer full		

EC Clock Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1E	CHIPID_H	7-0	R/W	**CHIPID** high byte. (ECSTS[2]=0)	0x39	0xFF
0x1E	CLKCFG2	7-0	R/W	Divider of **(DPLL Freq)/2** to generate 1μs (ECSTS[2]=1) Eg: DPLL outputs 64MHz (by default), to generate 1μs, the divider should be 32. That is the CLKCFG2 will be 0x1F.	0x1F	0xFF

EC PLL Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1F	CHIPID_L	7-0	R/W	**CHIPID** low byte. (ECSTS[2]=0)	0x30	0xFF
0x1F	PLLCFG2	7-6	R/W	High 2 bits of DPLL initial value. (ECSTS[2]=1) DPLL initial value is 10-bit, the low 8 bits are located at 0xFF0F, **PLLCFG**[7:0].	0x21	0xFF
		5	R/W	DPLL reference selection. **0**: Reference PCI clock **1**: Reference 32.768KHz source. (default)		
		4	R/W	DPLL source clock divider. **0**: Disable. (default) **1**: Enable. If PLLCFG2[5]=1, then this bit should be “0”. If PLLCFG2[5]=0, this bit should be “1”.		
		3-0	R/W	DPLL low speed state setting in Idle mode. The default value is **0001b**, the DPLL will provide 2MHz (8051)/ 1MHz (Peripheral) clock.		

EC MISC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x20	ECMISC	7	RO	8051 state. **0**: Idle state **1**: Normal state	0x80	0xFF
		6	R/W1C	GPXIOA07 output 8051 IDLE State Flag **0**: 8051 in Normal mode **1**: 8051 in IDLE(STOP) mode This bit is set only by HW when 8051 enters IDLE state This bit is write 1 clear by firmware only		
		5~4	R/W	Reserved		
		3	R/W	GPXIOA07 output 8051 IDLE State Enable When setting this bit and GPX_MISC (FF73h) bit 0, GPXA07 will output high only if ECMISC (FF20h) bit 6 is set to 1. And GPXIOA07 will return to previous state if ECMISC (FF20h) bit 6 is clear. The output priority of power fail status is higher than 8051 idle state in GPXIOA07		
		2	R/W	8051 extended command (**ExtCMD**, 0xFF10) interrupt enable. **0**: Disable **1**: Enable		
		1	R/W	Register function select of 0xFF10~0xFF13 for LPC index-I/O **0**: DAC **1**: LPC index-I/O Extended command related registers		
		0	R/W	Register function select of 0xFF10~0xFF13 for 8051. **0**: DAC **1**: 8051 Extended command related registers		

EC Extended I/O (Debug I/O) Data Port by Host						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x21	EXTIOR	7-0	R/W	The host reads extended I/O port and gets data from this register. *No interrupt occurs*.	0x00	0xFF

Embedded Debug Interface Feature Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x22	EDIF	7	R/W	EDI feature enable **0**: disable **1**: enable	0x00	0xFF
		6-0	RSV	Reserved		

Embedded Debug Interface Active Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x23	EDIAS	7	R/W	EDI active status **0**: not active **1**: active	0x00	0xFF
		6-0	RSV	Reserved		

Embedded Debug Version ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x24	EDIID	7-0	RO	EDI version	0x02	0xFF

RSV						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x25	RSV	7~0	RSV	Reserved	0x00	0xFF

RSV						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x26	RSV	7~0	RSV	Reserved	0x00	0xFF

Voltage comparator control and status register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x27	VCCSR	7	RO	GPXIOD03 VC (Voltage comparer) status	0x00	0xFF
		6	RO	GPXIOD00 VC (Voltage comparer) status		
		5~4	R/W	Voltage comparator De-bounce setting **00**: No De-bounce **01**: continually trigger twice **10**: continually trigger 4 times **11**: continually trigger 8 times		
		3~2	RSV	Reserved		
		1	R/W	GPXIOD03 voltage comparator enable To enable the voltage comparator input from GPXIOD03 pad, detecting the voltage is over 1.2V. **Note: GPXIOD00/GPXIOD03 is as VC input, GPXIOA06 is as VC output, and GPX_MISC[0] is enable bit**		
		0	R/W	GPXIOD00 voltage comparator enable To enable the voltage comparator input from GPXIOD00 pad, detecting the voltage is over 1.2V. **Note: GPXIOD00/GPXIOD03 is as VC input, GPXIOA06 is as VC output, and GPX_MISC[0] is enable bit**		

Power fail control and status register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x28	PFCSR	7	R/W1C	Power Fail status flag This bit is set by hardware if voltage of power is under 2.7V, and write 1 clear by firmware or system reset occur	0x00	0xFF
		6	RSV	Reserved		
		5~4	R/W	Power fail De-bounce setting **00**: No De-bounce **01**: continually trigger twice **10**: continually trigger 4 times **11**: continually trigger 8 times		
		3~1	RSV	Reserved		
		0	R/W	Power fail status enable GPXIOA03 or GPXIOA07 will output low to indicate the system power is under 2.7V. The output pin select is controlled by GPX_MISC[2:1]. **0**: Disable **1**: Enable **Note: GPXIOA03/GPXIOA07 will return to previous state if PFCSR[7] is written 1 clear.** **The output priority of power fail status is higher than 8051 idle state in GPXIOA07.**		

Internal oscillator control register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x29	IOSCCR	7	R/W	Oscillator current setting enable	0x00	0xFF
		6~5	RSV	Reserved		
		4~0	R/W	Oscillator current setting		

Crystal 32k control register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2A	CRY32CR	7~4	RSV	Reserved	0x01	0xFF
		3	R/W	PS2 / GPT / CIR / FAN / FANMON / PWM clock selection **1**: clock source from external 32KHz crystal **0**: clock source from DPLL divider output 32KHz		
		2	R/W	32Khz clock crystal selection **0**: Select 32KHz from original external crystal **1**: Select 32KHz from internal oscillator (embedded) **Note: Before change this bit, the relative clock should be stable.**		
		1	R/W	Internal oscillator enable		
		0	R/W	External crystal enable		

4.18 General Purpose Wake-up Controller (GPWU)

4.18.1 GPWU Function Description

The GPIO module provides flexible methods to wakeup the KBC or to generate interrupt. Once the input function is determined, plenty of features for wakeup can be setup. Here is the table to summarize all the features.

Wakeup Enable **0:** Disable **1:** Enable	**Polarity** **0:** ↓ / L **1:** ↑ / H	**Edge/Level** **0:** Edge **1:** Level	**Toggle** **0:** Disable **1:** Enable	**Description**
0	X	X	X	No wakeup events occur
1	X	X	1	Signal toggle trigger
1	0	0	0	Falling edge trigger
1	0	1	0	Low level trigger
1	1	0	0	Rising edge trigger
1	1	1	0	High level trigger

4.18.2 GPWU Registers Description

GPIO Wakeup Event Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x30	GPWUEN00	7-0	R/W	GPIO00~GPIO07 Wakeup Event Switch bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x31	GPWUEN08	7-0	R/W	GPIO08~GPIO0F Wakeup Event Switch bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x32	GPWUEN10	7-0	R/W	GPIO10~GPIO17 Wakeup Event Switch bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x33	GPWUEN18	7-0	R/W	GPIO18~GPIO1F Wakeup Event Switch bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x34	GPWUEN20	7-0	R/W	GPIO20~GPIO27 Wakeup Event Switch bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x35	GPWUEN28	7-0	R/W	GPIO28~GPIO2F Wakeup Event Switch bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x36	GPWUEN30	7-0	R/W	GPIO30~GPIO37 Wakeup Event Switch bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x37	GPWUEN38	7-0	R/W	GPIO38~GPIO3B Wakeup Event Switch bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** Wakeup event disable **1:** Wakeup event enable ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x38	GPWUEN40	7-0	R/W	GPIO40~GPIO47 Wakeup Event Switch bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x39	GPWUEN48	7-0	R/W	GPIO48~GPIO4F Wakeup Event Switch bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x3A	GPWUEN50	7-0	R/W	GPIO50~GPIO57 Wakeup Event Switch bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x3B	GPWUEN58	7-0	R/W	GPIO58~GPIO59 Wakeup Event Switch bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x3C	GXWUEN00	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Event Switch bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF

GPIO Wakeup Event Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x40	GPWUPF00	7-0	R/W1C	GPIO00~GPIO07 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x41	GPWUPF08	7-0	R/W1C	GPIO08~GPIO0F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x42	GPWUPF10	7-0	R/W1C	GPIO10~GPIO17 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x43	GPWUPF18	7-0	R/W1C	GPIO18~GPIO1F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x44	GPWUPF20	7-0	R/W1C	GPIO20~GPIO27 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x45	GPWUPF28	7-0	R/W1C	GPIO28~GPIO2F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x46	GPWUPF30	7-0	R/W1C	GPIO30~GPIO37 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x47	GPWUPF38	7-0	R/W1C	GPIO38~GPIO3B Wakeup Event Pending Flag bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** No wakeup event **1:** Wakeup event pending ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x48	GPWUPF40	7-0	R/W1C	GPIO40~GPIO47 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x49	GPWUPF48	7-0	R/W1C	GPIO48~GPIO4F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x4A	GPWUPf50	7-0	R/W1C	GPIO50~GPIO57 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x4B	GPWUPF58	7-0	R/W1C	GPIO58~GPIO59 Wakeup Event Pending Flag bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x4C	GXWUPF00	7-0	R/W1C	GPXIOD00~GPXIOD07 Wakeup Event Pending Flag bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF

GPIO Wakeup Polarity Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x50	GPWUPS00	7-0	R/W	GPIO00~GPIO07 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x51	GPWUPS08	7-0	R/W	GPIO08~GPIO0F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x52	GPWUPS10	7-0	R/W	GPIO10~GPIO17 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x53	GPWUPS18	7-0	R/W	GPIO18~GPIO1F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x54	GPWUPS20	7-0	R/W	GPIO20~GPIO27 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x55	GPWUPS28	7-0	R/W	GPIO28~GPIO2F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x56	GPWUPS30	7-0	R/W	GPIO30~GPIO37 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x57	GPWUPS38	7-0	R/W	GPIO38~GPIO3B Wakeup Polarity Selection bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger) ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x58	GPWUPS40	7-0	R/W	GPIO40~GPIO47 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x59	GPWUPS48	7-0	R/W	GPIO48~GPIO4F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x5A	GPWUPS50	7-0	R/W	GPIO50~GPIO57 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x5B	GPWUPS58	7-0	R/W	GPIO58~GPIO59 Wakeup Polarity Selection bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x5C	GXWUPS00	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Polarity Selection bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF

GPIO Wakeup Level/Edge Trigger Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x60	GPWUEL00	7-0	R/W	GPIO00~GPIO07 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x61	GPWUEL08	7-0	R/W	GPIO08~GPIO0F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x62	GPWUEL10	7-0	R/W	GPIO10~GPIO17 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x63	GPWUEL18	7-0	R/W	GPIO18~GPIO1F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x64	GPWUEL20	7-0	R/W	GPIO20~GPIO27 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x65	GPWUEL28	7-0	R/W	GPIO28~GPIO2F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x66	GPWUEL30	7-0	R/W	GPIO30~GPIO37 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x67	GPWUEL38	7-0	R/W	GPIO38~GPIO3B Wakeup Level/Edge Selection bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** Edge trigger **1:** Level trigger ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x68	GPWUEL40	7-0	R/W	GPIO40~GPIO47 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x69	GPWUEL48	7-0	R/W	GPIO48~GPIO4F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x6A	GPWUEL50	7-0	R/W	GPIO50~GPIO57 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x6B	GPWUEL58	7-0	R/W	GPIO58~GPIO59 Wakeup Level/Edge Selection bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x6C	GXWUEL00	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Level/Edge Selection bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF

GPIO Wakeup Input Change (Toggle) Trigger Selection Note:This setting will ignore the corresponding bit of GPWUELxx.						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x70	GPWUCHG00	7-0	R/W	GPIO00~GPIO07 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x71	GPWUCHG08	7-0	R/W	GPIO08~GPIO0F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x72	GPWUCHG10	7-0	R/W	GPIO10~GPIO17 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x73	GPWUCHG18	7-0	R/W	GPIO18~GPIO1F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x74	GPWUCHG20	7-0	R/W	GPIO20~GPIO27 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x75	GPWUCHG28	7-0	R/W	GPIO28~GPIO2F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x76	GPWUCHG30	7-0	R/W	GPIO30~GPIO37 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x77	GPWUCHG38	7-0	R/W	GPIO38~GPIO3B Wakeup Input Change (Toggle) Trigger bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** Toggle trigger disable **1:** Toggle trigger enable ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x78	GPWUCHG40	7-0	R/W	GPIO40~GPIO47 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x79	GPWUCHG48	7-0	R/W	GPIO48~GPIO4F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x7A	GPWUCHG50	7-0	R/W	GPIO50~GPIO57 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x7B	GPWUCHG58	7-0	R/W	GPIO58~GPIO59 Wakeup Input Change (Toggle) Trigger bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x7C	GXWUCHG00	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Input Change (Toggle) Trigger bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF

4.18.3 GPWU Programming Sample

In this section gives some programming sample to control GPWU module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of GPWU filed application.

Example	
PIN	**Function**
GPIO02	Low level trigger
GPIO03	Rising edge trigger
GPIO05	Falling edge trigger
GPIO06	Edge change trigger
Programming model	
1. Set related wakeup enable register. GPWUEN00 (0xFF30) = 0x6C 2. Set related wakeup polarity register GPWUPS00 (0xFF50) = 0x08 3. Set related wakeup edge/level trigger register GPWUEL00 (0xFC60) = 0x04 4. Set related wakeup input change register GPWUCHG00 (0xFF70) = 0x40	

4.19 System Management Bus Controller (SMBus)

4.19.1 SMBus Function Description

The SMBus is a two wire interface design based on I^2C bus. The SMBus controller in the KBC supports SMBus 2.0 and supports both master and slave mode with 4 interfaces. The SMBus controller supports 12 command protocols as following table. For more detail about each command protocol, please refer to the *System Management Bus Specification 2.0.*

Command Byte	Command	Command Byte	Command
02h	Quick Write	08h	Write Word
03h	Quick Read	09h	Read Word
04h	Send Byte	0Ah	Write Block
05h	Receive Byte	0Bh	Read Block
06h	Write Byte	0Ch	Word Process
07h	Read Byte	0Dh	Block Process

The SMBus introduces new mechanism to communicate with I^2C devices, called **Byte mode**. If the SMBus operates in this mode, only 3 protocols are supported, ***05h*** *(Receive Byte),* ***0Ah*** *(Write Block) and* ***0Bh*** *(Read Block)*. Here gives the brief programming guide of how to use Byte mode as following table.

05h, Receive Byte	0Ah, Write Block	0Bh, Read Block
1. Set the address in SMBADR (0xFF9A). 2. Set the ACK or NACK bit in SMBPF (0xFF96[6]). 3. Set the protocol in SMBPRTCL (0xFF98). 4. Once one byte data received, the interrupt pending flag will be set (0xFF96[5]). And the F/W could obtain the data via pooling or interrupt method. 5. If more than one byte received, the F/W must set the ACK or NACK response in advance. (the same as step 2), then continue to the step 4 until all bytes complete.	1. Set the address in SMBADR (0xFF9A). 2. Set the data array in SMBDAT (0xFF9C). 3. Set the count number in SMBCNT (0xFFBC). 4. Set the protocol in SMBPRTCL (0xFF98).	1. Set the address in SMBADR (0xFF9A). 2. Set the count number in SMBCNT (0xFFBC). 3. Set the protocol in SMBPRTCL (0xFF98).

The SMBus controller works as a host (master). The controller can be programmed to enable slave mode. In slave mode, the controller will response to its slave address which is programmable. A slave device could communicate with the SMBus host controller via **SMBus Alert** or **Host Notify** protocols. The **SMBus Alert** protocol can be implemented via optional SMBAlert# signal or periodical ARA (Alert Response Address) command. As to **Host Notify** protocol, The controller provides registers for F/W to achieve different applications. The following gives the brief summary between Host Notify protocol and SMBus register interface.

1bit	7bit	1bit	1bit	7bit	1bit	8bit	1bit	8bit	1bit	1bit
S	SMB Host Addr.	Wr	A	Device Addr.	A	Data Low Byte	A	Data High Byte	A	P

SMB Host Addr : stored in **SMBAADR**, 0xFFBD.
Device Addr : stored in **SMBAADR**, 0xFFBD.
Data Low Byte: stored in **SMBADAT0**, 0xFFBE.
Data High Byte: stored in **SMBADAT1**, 0xFFBF.
S: Start bit P: Stop bit

Slave (SMBus device) to Master

Master (SMBus host) to Slave

4.19.2 SMBus Register Description

SMBus Selection bank						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x80	SMBBAK	7-0	RSV	Reserved	0x00	0xFF
		1	R/W	SMBus selection and registers bank selection , to select SMBus controller 0: Controller 0, for SCL0/SDA0 and SCL1/SDA1 1: Controller 1, for SCL2/SDA2 and SCL3/SDA3		

Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x81~ 0x91	Reserved	7-0	RSV	Reserved	0x00	0xFF

SMBus CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	SMBTCRC	7-0	RO	CRC value transmit to SMBus.	0x00	0xFF

SMBus Pin Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x93	SMBPIN	7	R/W	SMBus data line forced to low. Write "**0**" to force **SDA0** or **SDA1** low. (Based on SMBPIN[1:0]) If SMBBAK is selected as controller 1, Write "0" to force SDA2 or SDA3 low.(Based on SMBPIN[1:0])	0x00	0xFF
		6	R/W	SMBus clock line forced to low. Write "**0**" to force **SCL0** or **SCL1** low. (Based on SMBPIN[1:0]) If SMBBAK is selected as controller 1, Write "0" to force SCL2 or SCL3 low.(Based on SMBPIN[1:0])		
		5	RO	Status of SDA0 or SDA1 or SDA0 wired SDA1..		
		4	RO	Status of SCL0 or SCL1 or SCL0 wired SCL1.		
		3	R/W	Byte mode function enable 3 protocols support, **Write Block/Read Block/Receive Byte.** Protocols are defined via register SMBPRTCL[6:0] **0**: Disable **1**: Enable		
		2	R/W	SCL/SDA input debounce enable. **0**: Disable **1**: Enable		
		1	R/W	SCL1/SDA1 pin connected to SMBus controller. **0**: Disable **1**: Enable If SMBBAK is selected as controller 1, Select SCL3/SDA3 pin connected to SMBus controller.		
		0	R/W	SCL0/SDA0 pin connected to SMBus controller. **0**: Disable **1**: Enable If SMBBAK is selected as controller 1, Select SCL2/SDA2 pin connected to SMBus controller.		

SMBus Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	SMBCFG	7	R/W	SMBus master disable **0**: Enable master function. **1**: Disable master function	0x06	0xFF
		6	R/W	SMBus host alarm protocol disable (0xFFBD~0xFFBF disable) **0**: Enable slave function. **1**: Disable slave function		
		5	RSV	Reserved		
		4-0	R/W	SMBus clock period If **SMBCFG[4:0]>0 and SMBPIN[2]=1,** the period is SMBus clock period = (SMBCFG[4:0]+1) * 4 μs If **SMBCFG[4:0]>0 and SMBPIN[2]=0**, the period is SMBus clock period = SMBCFG[4:0] * 4 μs Please **do not** set these bits to "0".		

SMBus Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	SMBEN	7	RO	SMBus host controller status **0**: not busy **1**: busy	0x00	0xFF
		6-4	RSV	Reserved		
		3	R/W	SMBus slave protocol selection. **0**: word read/write **1**: byte read/write		
		2	R/W	SMBus slave mode enable. **0**: Disable **1**: Enable		
		1	R/W	SMBus alert (host notify protocol) interrupt **0**: Disable **1**: Enable		
		0	R/W	SMBus protocol completion interrupt **0**: Disable **1**: Enable		

SMBus Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	SMBPF	7	RSV	Reserved	0x00	0xFF
		6	R/W	ACK bit of Receive Byte (Byte Mode) protocol **0**: ACK, the Receive Byte protocol keeps going **1**: NACK, once the F/W ready to obtain the last Receive Byte, F/W set this bit in advance. After this last byte transferred, the controller issues NACK to device and the protocol stop.		
		5	R/W1C	Read data interrupt flag of Receive Byte (Byte Mode) protocol **0**: no event **1**: event occurs		
		4	RO	Read protocol interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		3	R/W1C	Interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		2-0	RSV	Reserved		

SMBus Received CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x97	SMBRCRC	7-0	RO	The CRC value received from SMBus slave device.	0x00	0xFF

SMBus Protocol						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	SMBPRTCL	7	R/W	SMBus transaction with PEC (Packet Error Check) **0**: Disable **1**: Enable.	0x00	0xFF
		6-0	R/W	Command protocol. **02h**: Quick Write **03h**: Quick Read **04h**: Send Byte **05h**: Receive Byte / Receive Byte (Byte Mode) **06h**: Write Byte **07h**: Read Byte **08h**: Write Word **09h**: Read Word **0Ah**: Write Block **/** Write Block (Byte Mode) **0Bh**: Read Block / Read Block (Byte Mode) **0Ch**: Word Process **0Dh**: Block Process others: Reserved		

SMBus Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x99	SMBSTS	7	R/W	SMBus command done flag **0**: no event (Write 0 to clear) **1**: event occurs	0x00	0xFF
		6	R/W	SMBus alarm (host notify protocol) interrupt flag **0**: no event (Write 0 to clear) **1**: event occurs		
		5	R/W	SMBus block data array protocol control. F/W could control the protocol progress via this bit. **0**: Block Data Array protocol keeps going. **1**: Block Data Array protocol stops		
		4-0	R/W	Error code. **00h**: no error **07h**: unknown address failure. **10h**: device address no ACK **12h**: command no ACK **13h**: device data no ACK **17h**: device access deny **18h**: SMBus timeout **19h:** unsupported protocol **1Ah**: SMBus busy **1Fh**: PEC (Packet Error Check) error others: Reserved		

SMBus Address Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9A	SMBADR	7-0	R/W	SMBus address (7-bits long), bit0 ignored.	0x00	0xFF
0x9A	SMBADR (SMBPIN[3]=1)	7-1	R/W	SMBus address (7-bits long).	0x00	0xFF
		0	R/W	Data direction bit **0**: Write **1**: Read		

SMBus Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9B	SMBCMD	7-0	R/W	SMBus command port	0x00	0xFF

SMBus Data Array (8 Bytes)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9C	SMBDAT0	7-0	R/W	Data port for ***Send/Receive/Read Byte/Write Byte*** protocol	0x00	0xFF
0x9D	SMBDAT1	7-0	R/W	Data port for ***Read Word/Write Word*** protocol, 2nd byte data	0x00	0xFF
0x9E	SMBDAT2	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0x9F	SMBDAT3	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA0	SMBDAT4	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA1	SMBDAT5	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA2	SMBDAT6	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA3	SMBDAT7	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF

SMBus Slave Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA4	SMBRSA	7-0	R/W	SMBus slave address (7-bits long), bit0 ignores. Only valid when SMBEN[2]=1	0x00	0xFF

SMBus Block Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBC	SMBCNT	7-5	RSV	Reserved	0x00	0xFF
		4~0	R/W	SMBus block count. "0x00", for 32-byte length in a block transfer.		

SMBus Alarm (Host Notify Protocol) Address / SMBus Slave Received Command Code						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBD	SMBAADR	7-0	R/W	This register is alarm address or SMBus Slave Command Code for Response Slave Address.	0x00	0xFF

SMBus Alarm Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBE	SMBDAT0	7-0	R/W	Alarm data (low byte)	0x00	0xFF
0xBF	SMBDAT1	7-0	R/W	Alarm data (high byte)	0x00	0xFF

SMBus Programming Sample

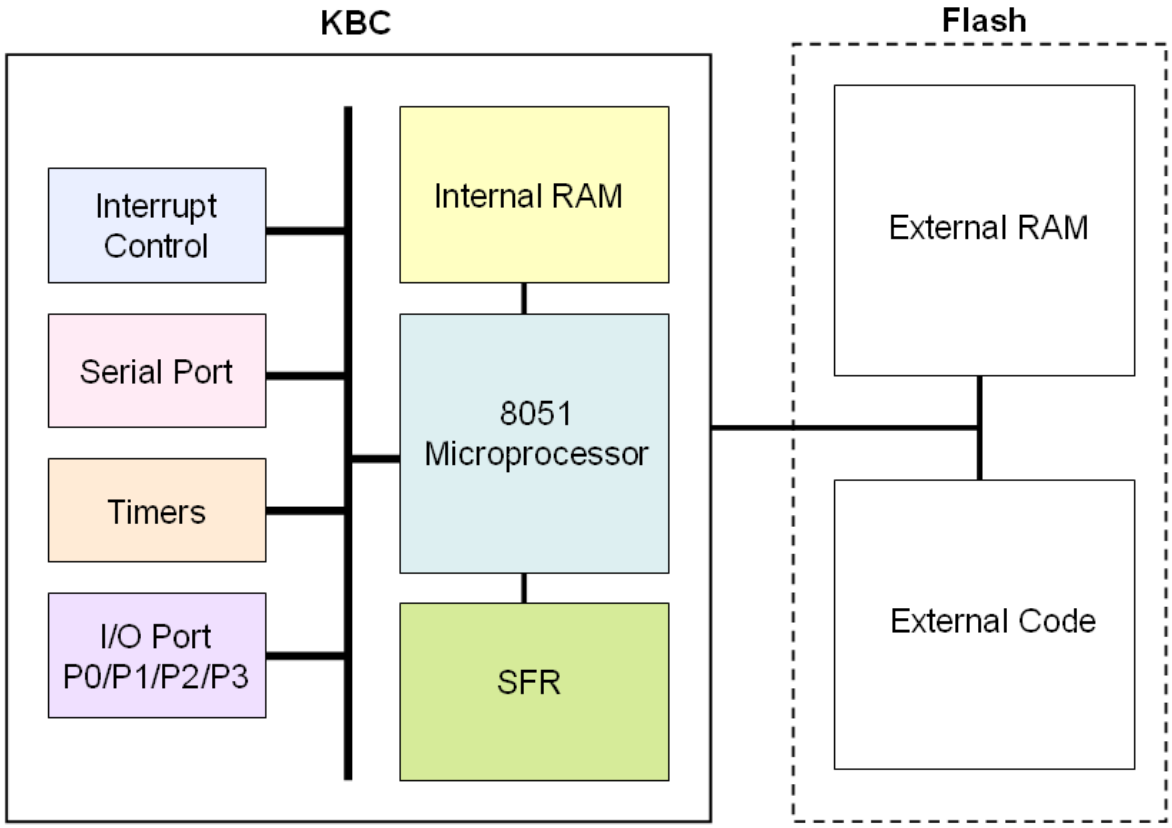
In this section gives some programming sample to control SMBus module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of SMBus filed application.

Example
Reading status of a battery (address 0x0A)
Programming model
SMBADR (0xFF9A) = 0x0A ; battery address SMBCMD (0xFF9B) = 0x12 SMBPTCL (0xFF98) = 0x07 Wait SMBSTS (0xFF99[7]) = 1b ; command complete Check if SMBSTS[4:0] = 0000b ; no error Read SMBDAT (0xFF9C) ; the current status

4.20 8051 Microprocessor

4.20.1 8051 Microprocessor Function Description

The Microprocessor inside KBC is an industrial compatible i8051. The 8051 is featured with 128bytes Special Function Register (SFR), Serial port, 2 16-bit Timers and 3 I/O ports with interrupt capability. The 8051 operates based on external crystal and runs at 8MHz by default. The following figure gives an illustration of the 8051 architecture. Except the standard 128bytes SFR, 8051 in KBx930 series is designed with overall 256 bytes internal memory



4.20.2 8051 Microprocessor Instruction

The instruction of 8051 microprocessor is fully compatible with industrial i8051. The instruction sets are as following table. The **OpCode** is in *Hexadecimal* and (b) means *Binary*. **B** stands for *byte number of instruction*. **C** stands for *number of cycle needed*.

Arithmetic				
Mnemonic	**OP code**	**Byte**	**Cycle**	**Description**
ADD A, #data	24	2	2	Add immediate data to Accumulator
ADD A, direct	25	2	2	Add direct byte to Accumulator
ADD A, @ R_N	26~27	1	2	Add indirect RAM to Accumulator (@R0~R1, OP 0x26~0x27)
ADD A, R_N	28~2F	1	2	Add register to Accumulator (R0~R7, OP 0x28~0x2F)
ADDC A, #data	34	2	2	Add immediate data to Accumulator with Carry
ADDC A, direct	35	2	2	Add direct byte to Accumulator with Carry
ADDC A, @ R_N	36~37	1	2	Add indirect RAM to Accumulator with Carry (@R0~R1, OP 0x26~0x27)
ADDC A, R_N	38~3F	1	2	Add register to Accumulator with Carry (R0~R7, OP 0x38~0x3F)
SUBB A, #data	94	2	2	Subtract immediate data from ACC with Borrow
SUBB A, direct	95	2	2	Subtract direct byte from ACC with Borrow
SUBB A, @ R_N	96~97	1	2	Subtract indirect RAM from ACC with Borrow (R0~R1, OP 0x96~0x97)
SUBB A, R_N	98~9F	1	2	Subtract register from Accumulator with Borrow (R0~R7, OP 0x98~0x9F)
INC A	04	1	2	Increment Accumulator
INC direct	05	2	2	Increment direct byte
INC @ R_N	06~07	1	2	Increment indirect RAM (R0~R1, OP 0x06~0x07)
INC R_N	08~0F	1	2	Increment Register (R0~R7, OP 0x08~0x0F)
DEC A	14	1	2	Decrement Accumulator
DEC direct	15	2	2	Decrement direct byte
DEC @ R_N	16~17	1	2	Decrement indirect RAM (R0~R1, OP 0x16~0x17)
DEC R_N	18~1F	1	2	Decrement Register (R0~R7, OP 0x18~0x1F)
INC DPTR	A3	1	2	Increment Data Pointer
MUL AB	A4	1	2	Multiply A & B
DIV AB	84	1	2	Divide A by B
DA A	D4	1	2	Decimal Adjust Accumulator

Logic & Byte Operation				
Mnemonic	**OP code**	**Byte**	**Cycle**	**Description**
ANL direct, A	52	2	2	AND Accumulator to direct byte
ANL direct, #data	53	3	2	AND immediate data to direct byte
ANL A, #data	54	2	2	AND immediate data to Accumulator
ANL A, direct	55	2	2	AND direct byte to Accumulator
ANL A, @ R_N	56~57	1	2	AND indirect RAM to Accumulator (R0~R1, OP 0x56~0x57)
ANL A, R_N	58~58	1	2	AND Register to Accumulator (R0~R7, OP 0x58~0x5F)
ORL direct, A	42	2	2	OR Accumulator to direct byte
ORL direct, #data	43	3	2	OR immediate data to direct byte
ORL A, #data	44	2	2	OR immediate data to Accumulator
ORL A, direct	45	2	2	OR direct byte to Accumulator
ORL A, @ R_N	46~47	1	2	OR indirect RAM to Accumulator (R0~R1, OP 0x46~0x47)
ORL A, R_N	48~4F	1	2	OR Register to Accumulator (R0~R7, OP 0x48~0x4F)
XRL direct, A	62	2	2	XOR Accumulator to direct byte
XRL direct, #data	63	3	2	XOR immediate data to direct byte
XRL A, #data	64	2	2	XOR immediate data to Accumulator
XRL A, direct	65	2	2	XOR direct byte to Accumulator
XRL A, @ R_N	66~67	1	2	XOR indirect RAM to Accumulator (R0~R1, OP 0x66~0x67)
XRL A, R_N	68~6F	1	2	XOR Register to Accumulator (R0~R7, OP 0x68~0x6F)
CLR A	E4	1	2	Clear Accumulator
CPL A	F4	1	2	Complement Accumulator
RL A	2 3	1	2	Left rotate Accumulator
RLC A	3 3	1	2	Left rotate Accumulator through Carry
RR A	0 3	1	2	Right rotate Accumulator
RRC A	1 3	1	2	Right rotate Accumulator through Carry
SWAP A	C 4	1	2	Swap Accumulator Nibbles

Data Movement				
Mnemonic	**OP code**	**Byte**	**Cycle**	**Description**
MOV A, R_N	E8~EF	1	2	Move Register to Accumulator (R0~R7, OP 0xE8~0xEF)
MOV A, direct	E5	2	2	Move direct byte to Accumulator
MOV A, @ R_N	E6~E7	1	2	Move indirect RAM to Accumulator (R0~R1, OP 0xE6~0xE7)
MOV A, #data	74	2	2	Move immediate data to Accumulator
MOV R_N, A	F8~FF	1	2	Move Accumulator to Register (R0~R7, OP 0xF8~0xFF)
MOV R_N, direct	A8~AF	2	2	Move direct byte to Register (R0~R7, OP 0xA8~0xAF)
MOV R_N, #data	78~7F	2	2	Move immediate data to Register (R0~R7, OP 0x78~0x7F)
MOV direct, A	F5	2	2	Move Accumulator to direct byte
MOV direct, @ R_N	86~87	2	2	Move indirect RAM to direct byte (R0~R1, OP 0x86~0x87)
MOV direct, R_N	88~8F	2	2	Move Register to direct byte (R0~R7, OP 0x88~0x8F)
MOV direct, #data	75	3	2	Move immediate data to direct byte
MOV direct, direct	85	3	2	Move direct byte to direct byte
MOV @ R_N, direct	A6~A7	2	2	Move direct byte to indirect RAM (R0~R1, OP 0xA6~0xA7)
MOV @ R_N, A	F6~F7	1	2	Move Accumulator to indirect RAM (R0~R1, OP 0xF6~0xF7)
MOV @ R_N, #data	76~77	2	2	Move immediate to indirect RAM (R0~R1, OP 0x76~0x77)
MOV DPTR,#data16	90	3	2	Load Data Pointer with a 16bit constant
MOVC A,@ A+PC	83	1	>33	Move Code byte relative to PC to Accumulator
MOVC A,@ A+DPTR	93	1	>33	Move Code byte relative to DPTR to Accumulator
MOVX A, @ DPTR	E0	1	>=5	Move External RAM to Accumulator
MOVX A, @ R_N	E2~E3	1	>=5	Move External RAM to Accumulator (R0~R1, OP 0xE2~0xE3)
MOVX @ DPTR, A	F0	1	>=4	Move Accumulator to External RAM
MOVX @ R_N, A	F2~F3	1	>=4	Move Accumulator to External RAM (R0~R1, OP 0xF2~0xF3)
POP direct	D0	2	2	POP direct byte from Stack
PUSH direct	C0	2	2	Push direct byte to Stack
XCH A, direct	C 5	2	2	Exchange direct byte with Accumulator
XCH A, @ R_N	C6~C7	1	2	Exchange indirect RAM with Accumulator (R0~R1, OP 0xC6~0xC7)
XCH A, R_N	C8~CF	1	2	Exchange Register with Accumulator (R0~R7, OP 0xC8~0xCF)
XCHD A, @ R_N	D6~D7	1	2	Exchange low order nibble of indirect RAM with Accumulator (R0~R1, OP 0xD6~0xD7)

Bit Operation				
Mnemonic	**OP code**	**Byte**	**Cycle**	**Description**
SETB bit	D2	2	2	Set direct bit
SETB C	D3	1	2	Set Carry
CLR bit	C2	2	2	Clear direct bit
CLR C	C3	1	2	Clear Carry
CPL bit	B2	2	2	Complement direct bit
CPL C	B3	1	2	Complement Carry
ANL C, bit	82	2	2	AND direct bit to Carry
ANL C, /bit	B0	2	2	AND complement of direct bit to Carry
ORL C, bit	72	2	2	OR direct bit to Carry
ORL C, /bit	A0	2	2	OR complement of direct bit to Carry
MOV C, bit	92	2	2	Move direct bit to Carry
MOV bit, C	A2	2	2	Move Carry to direct bit
JC relative	4 0	2	2	Jump if Carry is set
JNC relative	5 0	2	2	Jump if Carry is NOT set
JB bit, relative	2 0	3	2	Jump if direct bit is set
JBC bit, relative	1 0	3	2	Jump if direct bit is set & clear bit
JNB bit, relative	3 0	3	2	Jump if direct bit is NOT set

Program Branching				
Mnemonic	**OP code**	**Byte**	**Cycle**	**Description**
ACALL address11	bbb1 0001	2	3	Absolute sub-routine call
AJMP address11	bbb0 0001	2	2	Absolute jump
LCALL address16	12	3	3	Long sub-routine call
LJMP address16	02	3	2	Long jump
SJMP relative	80	2	2	Short jump (relative address)
JMP @ A+DPTR	73	1	2	Jump indirect relative to the DPTR
JNZ relative	70	2	2	Jump if Accumulator is NOT zero
JZ relative	60	2	2	Jump if Accumulator is zero
CJNE A, #data, relative	B4	3	2	Compare immediate to Accumulator and Jump if NOT equal
CJNE A, direct, relative	B5	3	2	Compare direct byte to Accumulator and Jump if NOT equal
CJNE @ R_N, #data, relative	B6~B7	3	2	Compare immediate to indirect and Jump if NOT equal (R0~R1, OP 0xB6~0xB7)
CJNE R_N, #data, relative	B8~BF	3	2	Compare immediate to Register and Jump if NOT equal (R0~R7, OP 0xB8~0xBF)
DJNZ direct, relative	D5	3	2	Decrement direct byte and Jump if NOT zero
DJNZ R_N, relative	D8~DF	2	2	Decrement register and Jump if NOT zero (R0~R7, OP 0xD8~0xDF)
RET	22	1	3	Return from sub-routine
RETI	32	1	3	Return form interrupt

Special Instruction				
Mnemonic	**OP code**	**Byte**	**Cycle**	**Description**
NOP	00	1	2	No Operation

4.20.3 8051 Interrupt Controller

In order to support more application, the 8051 in KBC extends interrupt channel to 24 for internal peripherals, that is, I/O port P0, P1 and P3 are with interrupt capability. The *interrupt priority for each channel is fixed* and no nested interrupt is supported. Here is the table to summarize the implementation of the interrupt controller.

Int. Source	Vector Address	Applications	Priority
IE0	0x0003	8051 external interrupt 0	0(Highest)
TF0	0x000B	8051 Timer 0	1
IE1	0x0013	8051 external interrupt 1	2
TF1	0x001B	8051 Timer 1	3
RI & TI	0x0023	8051 Serial port TX/RX interrupt	4
P0I[0]	0x0043	Watchdog	5
P0I[1]	0x004B	LPC I/O 0x2F R/W accessing interrupt / OWM	6
P0I[2]	0x0053	PS/2 event	7
P0I[3]	0x005B	KBC	8
P0I[4]	0x0063	IKB	9
P0I[5]	0x006B	68h/6Ch ports	10
P0I[6]	0x0073	EC	11
P0I[7]	0x007B	ESB events	12
P1I[0]	0x0083	FAN0 monitor event (update/overflow)	13
P1I[1]	0x008B	FAN1 monitor event (update/overflow)	14
P1I[2]	0x0093	SMBus events	15
P1I[3]	0x009B	CIR events	16
P1I[4]	0x00A3	GPT0 event	17
P1I[5]	0x00AB	GPT1 event	18
P1I[6]	0x00B3	GPT2 event	19
P1I[7]	0x00BB	GPT3 event / SDI	20
P3I[0]	0x00C3	Write extended I/O (LPC I/O port 80) / PECI	21
P3I[1]	0x00CB	GPIO00~GPIO0F	22
P3I[2]	0x00D3	GPIO10~GPIO1F	23
P3I[3]	0x00DB	GPIO20~GPIO2F	24
P3I[4]	0x00E3	GPIO30~GPIO3F	25
P3I[5]	0x00EB	GPIO40~GPIO4F	26
P3I[6]	0x00F3	GPIO50~GPIO59 / GPXIOD00~GPXIOD07	27
P3I[7]	0x00FB	ADC update	28(Lowest)

4.20.4 Interrupt Enable/Flag Table

Application	Interrupt Enable			Pending Flag		
	address	bit	behavior	address	bit	type
8051 external interrupt0 (GPIO1A)	A8h (IE)	0	2	88h (TCON)	1	2
8051 Timer0	A8h (IE)	1	2	88h (TCON)	5	2
8051 external interrupt0 (GPIO1B)	A8h (IE)	2	2	88h (TCON)	3	2
8051 Timer1	A8h (IE)	3	2	88h (TCON)	7	2
8051 Serial Port	A8h (IE)	4	2	98h (SCON)	1~0	1
WDT	FE80h (WDTCFG)	1	1	FE81h (WDTPF)	1	1
				FE81h (WDTPF)	0	1
RTC	FE84h (TMR_CFG)	7,0	1	FE84h (TMR_CFG)	1	1
LPC I/O R/W 0x2F	FF20h (ECMISC)	2	1	-	-	-
	FF9Ah (LPC2ECFG)	1	1	FE9Ah (LPC2ECFG)	2	1
PS/2	FEE0h (PS2CFG)	3~0	2	FEE1h (PS2PF)	3~0	2
KBC	FC81h (KBCCFG)	1,0	1	FC82h (KBCIF)	1,0	2
IKB	FCA3h (IKBIE)	5~0	1	FCA4h (IKBPF)	5~0	2
LPC 68h/6Ch IBF_Rising OBF_Falling	FE9Dh (LPC68CFG)	1,0	1	FE9Eh (LPC68CSR)	1,0	1
				FE9Eh (LPC68CSR)	3,2	2
EC host interrupt	FF04h (ECCFG,IBF)	1	4	FF1Ah (ECIF,IBF)	1	1
	FF04h (ECCFG,OBF)	0	2	FF1Ah (ECIF,OBF)	0	2
behavior	1. IE bit = 1, interrupt asserts when trigger event occurs 2. IE bit = 1, interrupt asserts when trigger event occurs but if PF not clear, interrupt will continue asserting 3. IE = 1, interrupt asserts when trigger event occurs or IE bit is from low to high(0 -> 1) when Pending Flag(PF) is = 1 4. No matter IE bit = 1 or 0, interrupt asserts when trigger event occurs					
type	1. When trigger event occurs, PF will be set to 1. PF cleared to 0 by W1C/W0C 2. IE bit = 1, when event occurs, PF will be set to 1. PF is cleared to 0 by W1C/W0C					

4.20.4 Interrupt Enable/Flag Table (Continued)

Application	Interrupt Enable			Pending Flag		
	address	bit	behavior	address	bit	type
ESB	FC90h (ESBCFG)	2	3	-	-	-
	FC92h (ESBINTE)	6~4	3	FC91h (ESBCS)	6~4	1
	FC92h (ESBINTE)	3~0	3	FC97h (ESBINT)	7~0	1
	FC98h (ESBCAS)	7~4	3	FC97h (ESBINT)		1
FAN	FE20h (FANCFG0)	3,2	3	FE21h (FANSTS0)	1,0	1
	FE30h (FANCFG1)	3,2	3	FE31h (FANSTS1)	1,0	1
SMBus	FF95h (SMBEB)	0	1	FF99h (SMBSTS)	7,5	1
				FF96h (SMBPF)	5	1
	FF95h (SMBEB)	1	1	FF99h (SMBSTS,alarm)	6	1
	FF95h (SMBEB)	2	1	FF96h (SMBPF,Slave)	3	1
CIR TX	FEC0h (CIRCFG, TX)	5	1	FEC2h (CIRPF, TX)	3	1
CIR RX	FEC0h (CIRCFG, RX)	1	1	FEC2h (CIRPF)	2~0	1
GPT0~GPT3	FE50h (GPTCFG)	3~0	1	FE51h (GPTPF)	3~0	2
Write Extended I/O	FE95h (LPCCFG)	4	1	-	-	-
GPWU	FF3xh (GPWUENxx)	7~0	3	FF4xh (GPWUPFxx)	7~0	1
ADC	FF18h (ADCTRL)	0	1	-	-	-
behavior	Interrupt Behavior => (Interrupt Occurs) (1) IE bit = 1, interrupt asserts when trigger event occurs (2) IE bit = 1, interrupt asserts when trigger event occurs but if PF not clear, interrupt will continue asserting (3) IE = 1, interrupt asserts when trigger event occurs or IE bit is from low to high(0 -> 1) when Pending Flag(PF) is = 1 (4) No matter IE bit = 1 or 0, interrupt asserts when trigger event occurs					
type	Pending Flag(PF) => 6. When trigger event occurs, PF will be set to 1. PF cleared to 0 by WC1/WC0 (2) IE bit = 1, when event occurs, PF will be set to 1. PF is cleared to 0 by WC1/WC0					

4.20.5 8051 Special Function Register (SFR)

The Special Function Registers are located in the internal RAM of 8051 microprocessor. The internal address is from 0x80 to 0xFF, sized with 128 bytes. All the SFRs are compatible with the standard ones. Some SFRs are redesigned with new features for flexible application. The following table gives a brief summary.

P3IE, P1IE, P0IE are read/write registers used as Interrupt Enable (IE) to their corresponding interrupt inputs. These three registers are original 8051 port registers with 8-bits. For the embedded 8051 inside KB910, the 3 ports are used for interrupt input (always rise pulses) extensions. The overall interrupt events are 24.

P3IF, P1IF, P0IF are Interrupt Flag(IF) corresponding to the 24 interrupt inputs. The IFs are set by external interrupt event (always a rising pulse, one clock width), and are cleared by software (execute IRET instruction for active interrupt). The original alternate 8051 port 3 functions are not related with P3IE and P3IF.

For more detail, please refer to the section of register description.

80	P0IE	SP	DPL	DPH			PCON2	PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1IE								97
98	SCON	SBUF	SCON2	SCON3	SCON4				9F
A0	P2								A7
A8	IE								AF
B0	P3IE								B7
B8	IP								BF
C0									C7
C8									CF
D0	PSW								D7
D8	P0IF								DF
E0	ACC								E7
E8	P1IF								EF
F0	B								F7
F8	P3IF								FF
	★								

1. The blue parts are changed from standard features and the green ones are the new design for special features. And all the others are the standard features of conventional 8051.

2. The registers listed in the column with ★mark are all *bit addressable*.

4.20.6 8051 Microprocessor Register Description

The SFR registers are located at internal RAM 0x80 ~ 0xFF.

P0 Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x80	P0IE	7-0	R/W	P0 interrupt enable. Bit0~7 for P0[0]~P0[7] respectively. **0**: Disable **1**: Enable	0x00

Stack Pointer					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x81	SP	7-0	R/W	8051 stack pointer register	0x07

Data Pointer Low Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x82	DPL	7-0	R/W	Low byte of DPTR	0x00

Data Pointer High Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x83	DPH	7-0	R/W	High byte of DPTR	0x00

Processor Control Register 2					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x86	PCON2	7	R/W	Reserved but this bit should be “0”.	0x20
		6	R/W	Timer0/Timer1 test mode enable. **0**: Disable **1**: Enable	
		5	R/W	Reserved	
		4	R/W	KBC modules write control. Once this bit set, 8051 could issue write access to external modules. **0**: Disable **1**: Enable	
		3	R/WC0	Same interrupt source pending flag. If the 8051 is handling some interrupt event, at the same time, the same source asserting the interrupt again, this flag will be set. If this flag set, the 8051 will re-enter ISR again once executing IRET. Writing “**0**” to clear this flag.	
		2-1	RSV	Reserved	
		0	R/W	Not fetching instruction while in idle loop. **0**: Disable **1**: Enable	

Processor Control Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x87	PCON	7-6	RSV	Reserved	0x00
		5	R/W	Interrupt vector offset address1 **0**: Interrupt vector address offset adding 0x0 **1**: Interrupt vector address offset adding 0x8000	
		4	R/W	Interrupt vector offset address2 **0**: Interrupt vector address offset adding 0x0 **1**: Interrupt vector address offset adding 0x4000 Please note, if PCON[5]=1 and PCON[4]=1 then the result of interrupt vector address will be added 0xC000.	
		3	R/W	General purpose flag 1 **0**: no event **1**: event occurs	
		2	R/W	General purpose flag 2 **0**: no event **1**: event occurs	
		1	WO	Stop mode enable. All clock stop except the external 32.768K OSC and PCICLK. **1**: Enable (write “0” no work)	
		0	WO	Idle mode enable. The clock of 8051 stops. **1**: Enable (write “0” no work)	

Timer/Counter Control Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x88	TCON	7	R/W1C	**TF1**, Timer1 overflow flag **0**: no event **1**: event occurs	0x00
		6	R/W	**TR1**, Timer1 start control. **0**: stop to count **1**: start to count	
		5	R/W1C	**TF0**, Timer0 overflow flag **0**: no event **1**: event occurs	
		4	R/W	**TR0**, Timer0 start control. **0**: stop to count **1**: start to count	
		3	R/W1C	**IE1**, External interrupt 1 flag **0**: no event **1**: event occurs	
		2	R/W	**IT1**, External interrupt 1 trigger selection **0**: low level trigger **1**: falling edge trigger	
		1	R/W1C	**IE0**, External interrupt 0 flag **0**: no event **1**: event occurs	
		0	R/W	**IT0**, External interrupt 0 trigger selection **0**: low level trigger **1**: falling edge trigger	

Timer Mode Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x89	TMOD	7	R/W	**GATE1**, this bit is the gate control of TR1 and INT1 **0**: Disable **1**: Enable	0x00
		6	R/W	**CT1**, Timer1 timer/counter selection **0**: Timer **1**: Counter	
		5-4	R/W	**TM1**, Timer1 mode selection **0**: 13-bit timer **1**: 16-bit timer **2**: 8-bit auto reload timer **3**: Timer 1 stops.	
		3	R/W	**GATE0**, this bit is the gate control of TR0 and INT0 **0**: Disable **1**: Enable	
		2	R/W	**CT0**, Timer0 timer/counter selection **0**: Timer **1**: Counter	
		1-0	R/W	**TM0**, Timer0 mode selection **0**: 13-bit timer **1**: 16-bit timer **2**: 8-bit auto reload timer **3**: TL0 and TH0 are two 8-bit timers.	

Timer 0 Low Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8A	TL0	7-0	R/W	Low byte of timer 0	0x00

Timer 1 Low Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8B	TL1	7-0	R/W	Low byte of timer 1.	0x00

Timer 0 High Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8C	TH0	7-0	R/W	High byte of timer 0	0x00

Timer 1 High Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8D	TH1	7-0	R/W	High byte of timer 1	0x00

Port1 Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x90	P1IE	7-0	R/W	Port 1 interrupt enable. Bit0~7 for P1[0]~P1[7] respectively **0**: Disable **1**: Enable	0x00

Serial Port Control Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x98	SCON	7-6	R/W	**SM1,SM0**, serial port mode **00**: 8-bit shift register, E51RX will be shift clock of E51CLK. **01**: 8-bit serial port (variable) **10**: 9-bit serial port (variable) **11**: 9-bit serial port (variable)	0x50
		5	RSV	Reserved	
		4	R/W	**REN**, serial port receive function enable. **0**: Disable **1**: Enable	
		3	R/W	**TB8**, The 9th bit of transmit data in mode2 and mode3.	
		2	R/W	**RB8**, The 9th bit of receive data	
		1	R/W0C	**TI**, TX interrupt flag **0**: no event **1**: event occurs	
		0	R/W0C	**RI**, RX interrupt flag **0**: no event **1**: event occurs	

Serial Port Data Buffer Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x99	SBUF	7-0	R/W	Serial port data buffer	0x00

Serial Port Control Register 2					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x9A	SCON2	7-0	R/W	High byte of 16-bit counter for baud rate	0x00

Serial Port Control Register 3					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x9B	SCON3	7-0	R/W	Low byte of 16-bit counter for baud rate	0x00

Serial Port Control Register 4					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x9C	SCON4	7-2	RSV	Reserved	0x00
		1~0	R/W	Serial Port mode 0 baud- rate setting (E51 clock set in CLKCFG, 0xFF0D) **00:** E51 clock divide 2 **01:** E51 clock divide 4 **10:** E51 clock divide 8 **11**: E51 clock divide 16	

Port 2 Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xA0	P2	7-0	R/W	Port 2 register	0x00

Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xA8	IE	7	R/W	**EA**, all interrupts enable. **0**: Disable **1**: Enable	0x00
		6	R/W	**EP**, Change P0IF, P1IF, P3IF Interrupt event trigger flag to Interrupt event pending flag **0**: Disable **1**: Enable	
		6-5	RSV	Reserved	
		4	R/W	**ES**, serial port interrupt enable **0**: Disable **1**: Enable	
		3	R/W	**ET1**, timer1 overflow interrupt enable **0**: Disable **1**: Enable	
		2	R/W	**EX1**, external interrupt 1 enable. **0**: Disable **1**: Enable	
		1	R/W	**ET0**, timer0 overflow interrupt enable **0**: Disable **1**: Enable	
		0	R/W	**EX0**, external interrupt 0 enable. **0**: Disable **1**: Enable	

Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xB0	P3IE	7-0	R/W	Port 3 interrupt enable. Bit0~7 for P3[0]~P3[7] respectively **0**: Disable **1**: Enable	0x00

Interrupt Priority Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xB8	IP	7-5	RSV	Reserved	0x00
		4	R/W	Serial port interrupt priority **0**: Low **1**: High	
		3	R/W	Timer1 interrupt priority **0**: Low **1**: High	
		2	R/W	External interrupt 1 priority **0**: Low **1**: High	
		1	R/W	Timer 0 interrupt priority **0**: Low **1**: High	
		0	R/W	External interrupt 0 priority **0**: Low **1**: High	

Processor Status Word Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xD0	PSW	7	R/W	**CY**, carry flag	0x00
		6	R/W	**AC**, auxiliary carry flag.	
		5	R/W	**F0**, for user general purpose.	
		4	R/W	**RS1**, register bank selector 1.	
		3	R/W	**RS0**, register bank selector 0.	
		2	R/W	**OV**, overflow flag	
		1	R/W	**F1**, flag 1 for user general purpose	
		0	R/W	**P**, parity flag	

Port0 Interrupt Flag Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xD8	P0IF	7-0	R/W	Port 0 interrupt flag.	0x00

Accumulator, ACC					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xE0	ACC	7-0	R/W	Accumulator	0x00

Port1 Interrupt Flag Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xE8	P1IF	7-0	R/W	Port 1 interrupt flag.	0x00

B Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xF0	B	7-0	R/W	**B** register, for MUL and DIV instructions.	0x00

Port3 Interrupt Flag Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xF8	P3IF	7-0	R/W	Port 3 interrupt flag.	0x00

5. Electrical Characteristics

5.1 Absolute Maximum Rating

Symbol	Parameter	Condition	Rating	Unit
V_{CC}	Power Source Voltage	All voltages are referred to GND.	-0.3 ~ 3.6	V
V_i	Input Voltage		-0.3 ~ 3.6	V
V_o	Output Voltage		-0.3 ~ 3.6	V
T_{STG}	Storage Temperature		-65 ~ 150	℃
	ESD	Human Body Mode (HBM)	TBD	V
		Machine Mode (MM)	TBD	

5.2 DC Electrical Characteristics

BQCZ16HIV

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.23		V	
Input High Threshold	V_{t+}		1.90		V	
Hysteresis	V_{TH}		0.67		V	
Output Low Voltage	V_{OL}			0.4	V	16mA Sink
Output High Voltage	V_{OH}	2.8			V	16mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		TBD		Ω	V_I=0V
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

BQC04HIV

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.23		V	
Input High Threshold	V_{t+}		1.90		V	
Hysteresis	V_{TH}		0.67		V	
Output Low Voltage	V_{OL}			0.4	V	4mA Sink
Output High Voltage	V_{OH}	2.8			V	4mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		40K		Ω	V_I=0V
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

BQCW16HIV

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.23		V	
Input High Threshold	V_{t+}		1.90		V	
Hysteresis	V_{TH}		0.67		V	
Output Low Voltage	V_{OL}			0.4	V	16mA Sink
Output High Voltage	V_{OH}	2.8			V	16mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		40K		Ω	V_I=0V
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

BCC16HI

(No Schmitt Trigger, No Pull-Up resistance function)

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold *	V_{t-}		--		V	
Input High Threshold *	V_{t+}		--		V	
Hysteresis *	V_{TH}		--		V	
Output Low Voltage	V_{OL}			0.4	V	16mA Sink
Output High Voltage	V_{OH}	2.8			V	16mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance *	R_{PU}		--		Ω	
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

BQC04HI

(No Pull-Up resistance function)

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.23		V	
Input High Threshold	V_{t+}		1.90		V	
Hysteresis	V_{TH}		0.67		V	
Output Low Voltage	V_{OL}			0.4	V	4mA Sink
Output High Voltage	V_{OH}	2.8			V	4mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance*	R_{PU}		--		Ω	
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

IQTHI (ADC cell)

(Input only, No Pull-Up resistance function)

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.26		V	
Input High Threshold	V_{t+}		1.77		V	
Hysteresis	V_{TH}		0.51		V	
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		--		Ω	
Input Capacitance	C_{PU}		5.5		pF	

OCT04H (DAC cell)

(Output only)

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Output Low Voltage	V_{OL}			0.4	V	4mA Sink
Output High Voltage	V_{OH}	2.8			V	4mA Source
Output Capacitance	C_{OUT}		5.5		pF	

BQC08HIV

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.23		V	
Input High Threshold	V_{t+}		1.90		V	
Hysteresis	V_{TH}		0.67		V	
Output Low Voltage	V_{OL}			0.4	V	8mA Sink
Output High Voltage	V_{OH}	2.8			V	8mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		40K		Ω	
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

BQC04HIVPECI

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.23		V	
Input High Threshold	V_{t+}		1.90		V	
Hysteresis	V_{TH}		0.67		V	
Input Low Threshold	V_{t-}		0.37		V	PECI Enable
Input High Threshold	V_{t+}		0.68		V	PECI Enable
Hysteresis	V_{TH}		0.31		V	PECI Enable
Output Low Voltage	V_{OL}			0.4	V	4mA Sink
Output High Voltage	V_{OH}	2.8			V	4mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		40K		Ω	
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

5.3 A/D & D/A Characteristics

ADC characteristics

Parameter	Limits			Unit
	Min	Typ	Max	
Resolution		10		Bit
Integral Non-linearity Error (INL)		TBD		LSB
Differential Non-linearity Error (DNL)		TBD		LSB
Offset Error		TBD		LSB
Gain Error		TBD		LSB
A/D Input Voltage Range	$0.1V_{cca}$		$0.9V_{cca}$	V
A/D Input Leakage Current		<0.5		uA
A/D Input Resistance	10			MΩ
A/D Input Capacitance			2	pF
A/D Clock Frequency		1		MHz
Voltage Conversion Time		256		uS

DAC characteristics

Parameter	Limits			Unit
	Min	Typ	Max	
Resolution		8		Bit
Integral Non-linearity Error (INL)			±2	LSB
Differential Non-linearity Error (DNL)			±1	LSB
Offset Error			±1	LSB
Gain Error			±2	LSB
D/A Output Voltage Range	0		V_{cca}	V
D/A Output Setting Time			1.12	uS
D/A Output Resistance		3.5		kΩ
D/A Output Capacitance		1		pF

5.4 Recommend Operation Condition

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
Vcc	Power Source Voltage	3.0	3.3	3.6	V
GND	Ground Voltage	-0.3	0	0.3	V
V_{CCA}	Analog Reference Voltage (for A/D and D/A)	3.0	3.3	3.6	V
AGND	Analog Ground Voltage	-0.3	0	0.3	V
T_{op}	Operating Temperature	0	25	70	℃

5.5 Operating Current

Symbol	Parameter	Limits	Unit
		Typ	
Icc	Typical current consumption in operating state under Windows environment. All clock domains are running, and no keyboard/mouse activities.	20	mA

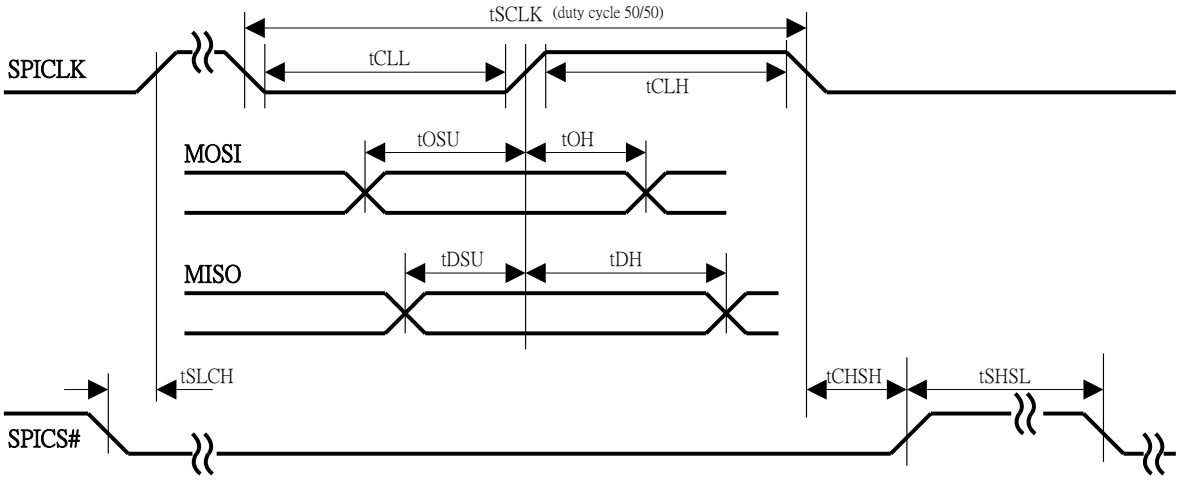
5.6 Package Thermal Information

Thermal resistance (degrees C/W). $Theta_{JA}$ values for KBx930.

	$Theta_{JA}$ @ 0 m/s	**$Theta_{JA}$ @ 1 m/s**	**$Theta_{JA}$ @ 2 m/s**
128-Pin LQFP	59.1	53.3	51.4

5.7 AC Electrical Characteristics

5.7.1 SPI Flash Timing



Parameter	Symbol	Spec.			Unit	Condition
		Min.	Typ.	Max.		
SPICLK period	**tSCLK**	15.2[*1]	--	--	ns	
SPICLK High Period	**tCLH**	4	--	--	ns	**tSCLK** = 15.2ns, C_L=12pF
SPICLK Low Period	**tCLL**	4	--	--	ns	**tSCLK** = 15.2ns, C_L=12pF
MOSI Setup Time	**tOSU**	--	tSCLK/2 – 5[*2]	--	ns	C_L=12pF
MOSI Hold Time	**tOH**	--	tSCLK/2 + 5[*2]	--	ns	C_L=12pF
SPICS# Active Setup Time	**tSLCH**	--	tSCLK	--	ns	
SPICS# Not Active Hold Time	**tCHSH**	--	tSCLK/2	--	ns	
SPICS# Deselect Time	**tSHSL**	110	--	--	ns	
MISO Setup Time	**tDSU**	0	--	--	ns	
MISO Hold Time	**tDH**	tSCLK/2 – 4	--	--	ns	

1. Tolerance +/- 3% (need to count in the DPLL tolerance)
2. For characteristic only.

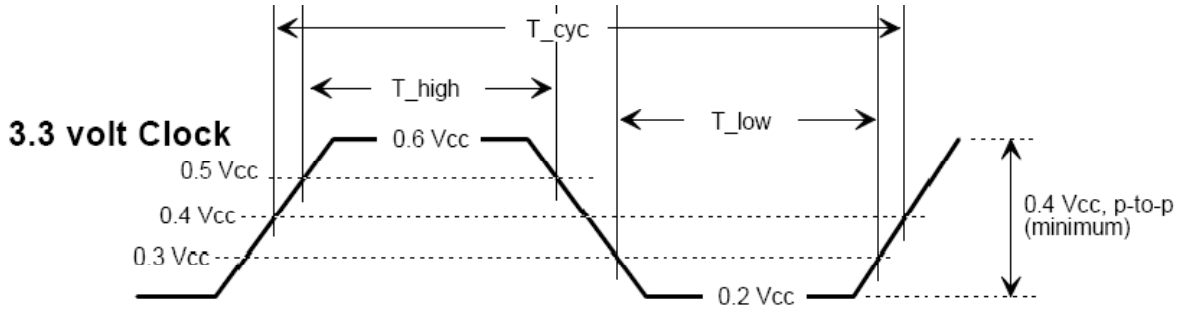
5.7.2 LPC interface Timing

Note: All AC characteristics of the LPC interface meet the PCI Local Bus SPEC for 3.3V DC signaling.

Clock & Reset：

Symbol	Parameter	Min	Max	Units	Notes
T_{cyc}	**CLK** Cycle Time	30	33	ns	1,4
T_{high}	**CLK** High Time	11		ns	
TI_{ow}	**CLK** Low Time	11		ns	
	CLK Slew Rate	1	4	V/ns	2
	Reset Slew Rate	50		mV/ns	3

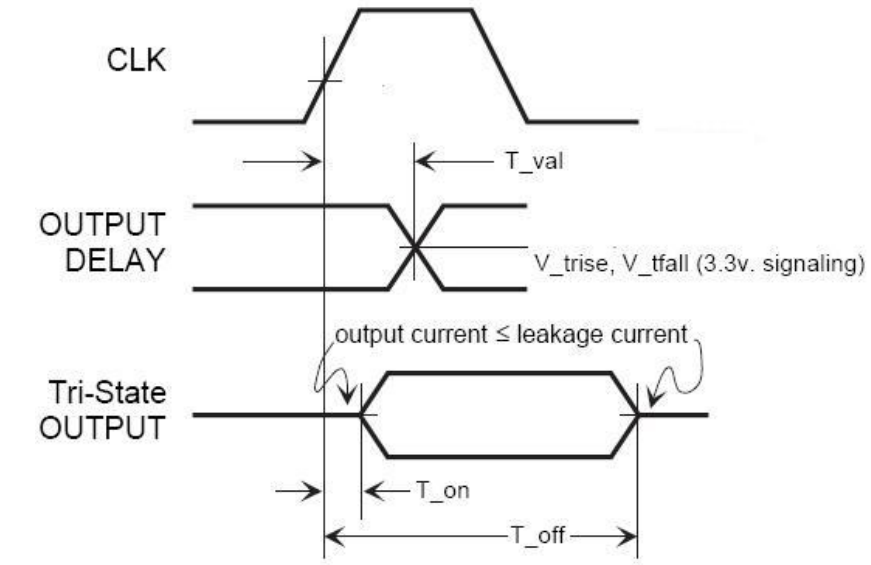
1. In general, all PCI components must work within clock frequency constrain. The clock frequency may be changed at any time during the operation of the system so long as the clock edges remain "clean" (monotonic) and the minimum cycle and high and low times are not violated. The clock may only be stopped in a low state.
2. Rise and fall times are specified in terms of the edge rate measured in V/ns. This slew rate must be met across the minimum peak-to-peak portion of the clock waveform as shown below.
3. The minimum RST# slew rate applies only to the rising (de-assertion) edge of the reset signal and ensures that system noise cannot render an otherwise monotonic signal to appear to bounce in the switching range.
4. Device operational parameters at frequencies under 16 MHz may be guaranteed by design rather than by testing.

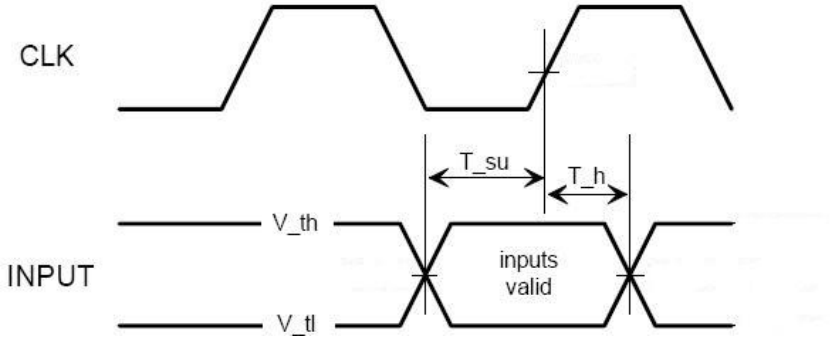


Timing Parameters

Symbol	Parameter	Min	Max	Units	Notes
T_{val}	**CLK** to Signal Valid Delay	2	11	ns	
T_{on}	Float to Active Delay	2		ns	1
T_{off}	Active to Float Delay		28	ns	1
T_{su}	Input Setup Time to **CLK**	7		ns	2,3
T_h	Input Hold Time from **CLK**	0		ns	3

1. For purposes of Active/Float timing measurements, the Hi-Z or "off" state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification.
2. Setup time applies only when the device is not driving the pin. Devices cannot drive and receive signals at the same time.
3. Refer the timing measurement conditions as below

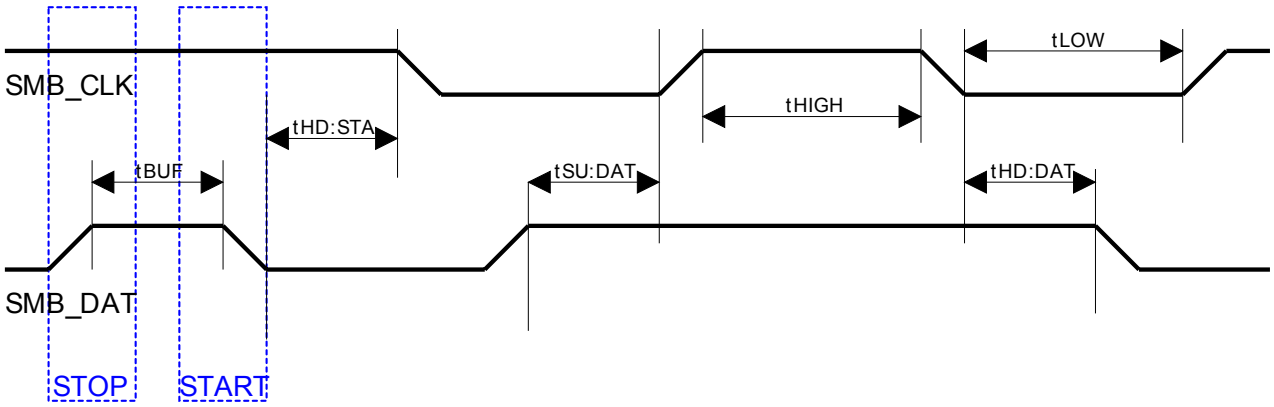


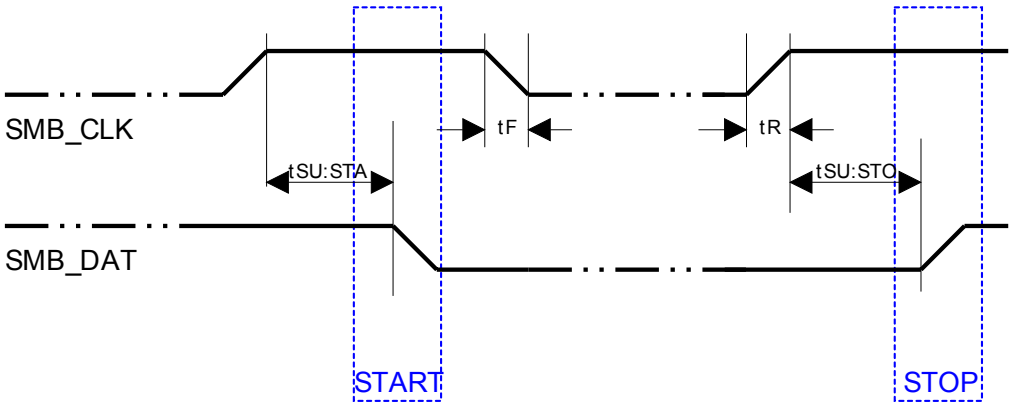


5.7.3 PS/2 interface Timing

This page is leaved blank intentionally.

5.7.4 SMBus interface Timing





Timing Parameters

Symbol	Parameter	Min	Typ.	Max	Units	Notes
T_{buf}	Bus free time between Stop and Start Condition	4.7			µs	
$T_{hd:sta}$	Hold time after (repeated) star condition. After this period, the first clock is generated.	4.0			µs	
$T_{su:sta}$	Repeated start condition setup time	4.7			µs	
$T_{su:sto}$	Stop confition setup time	4.0			µs	
$T_{hd:dat}$	Data hold time	300			ns	
$T_{su:dat}$	Data setup time	250			ns	
$T_{timeout}$	Detect clock low timeout	25		35	ms	
T_{low}	Clock low period	4.7			µs	2
T_{high}	Clock high period	4.0		50	µs	2
T_f	Data fall time			300	ns	
T_r	Data rise time			1000	ns	

1. For characteristic only
2. SMBUS frequencry dependant

5.7.5 PECI interface Timing

This page is leaved blank intentionally.

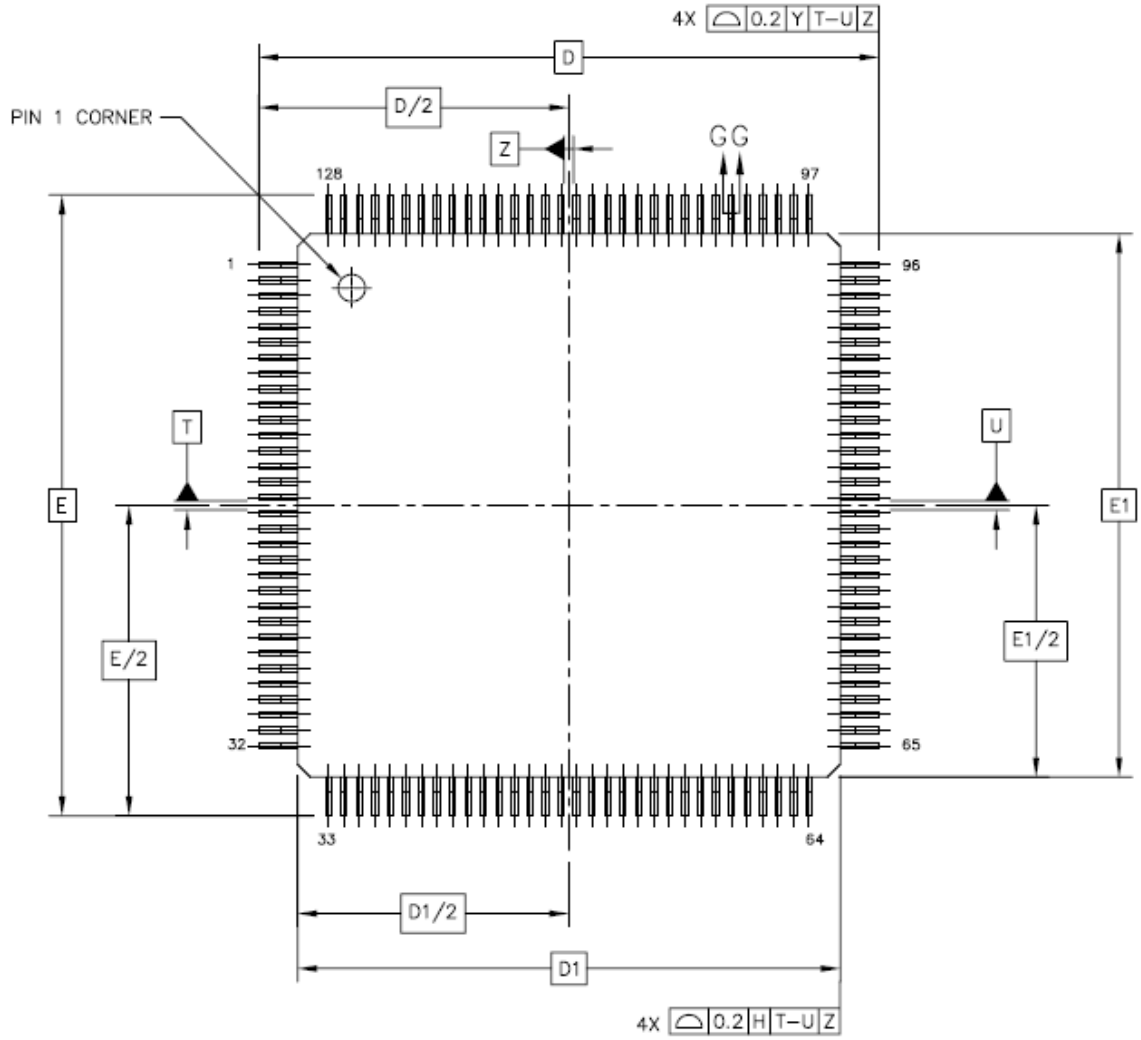
5.7.6 OWM interface Timing

This page is leaved blank intentionally.

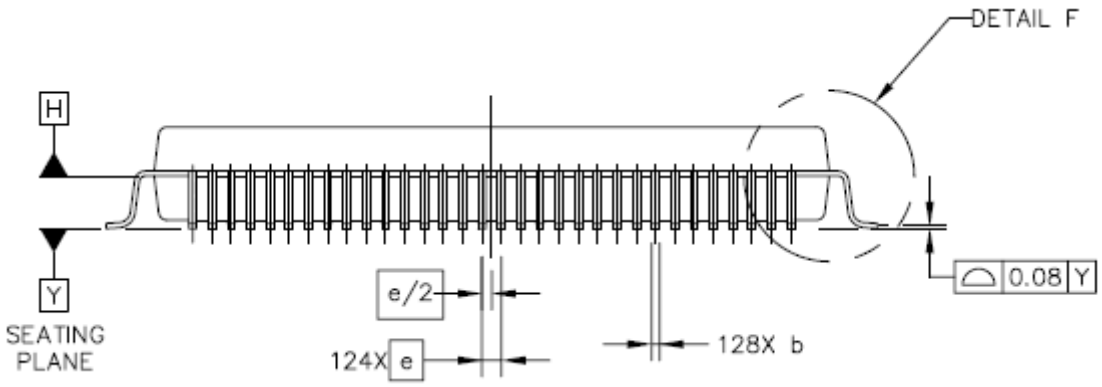
6. Package Information

6.1 LQFP 128-Pin Outline Diagram

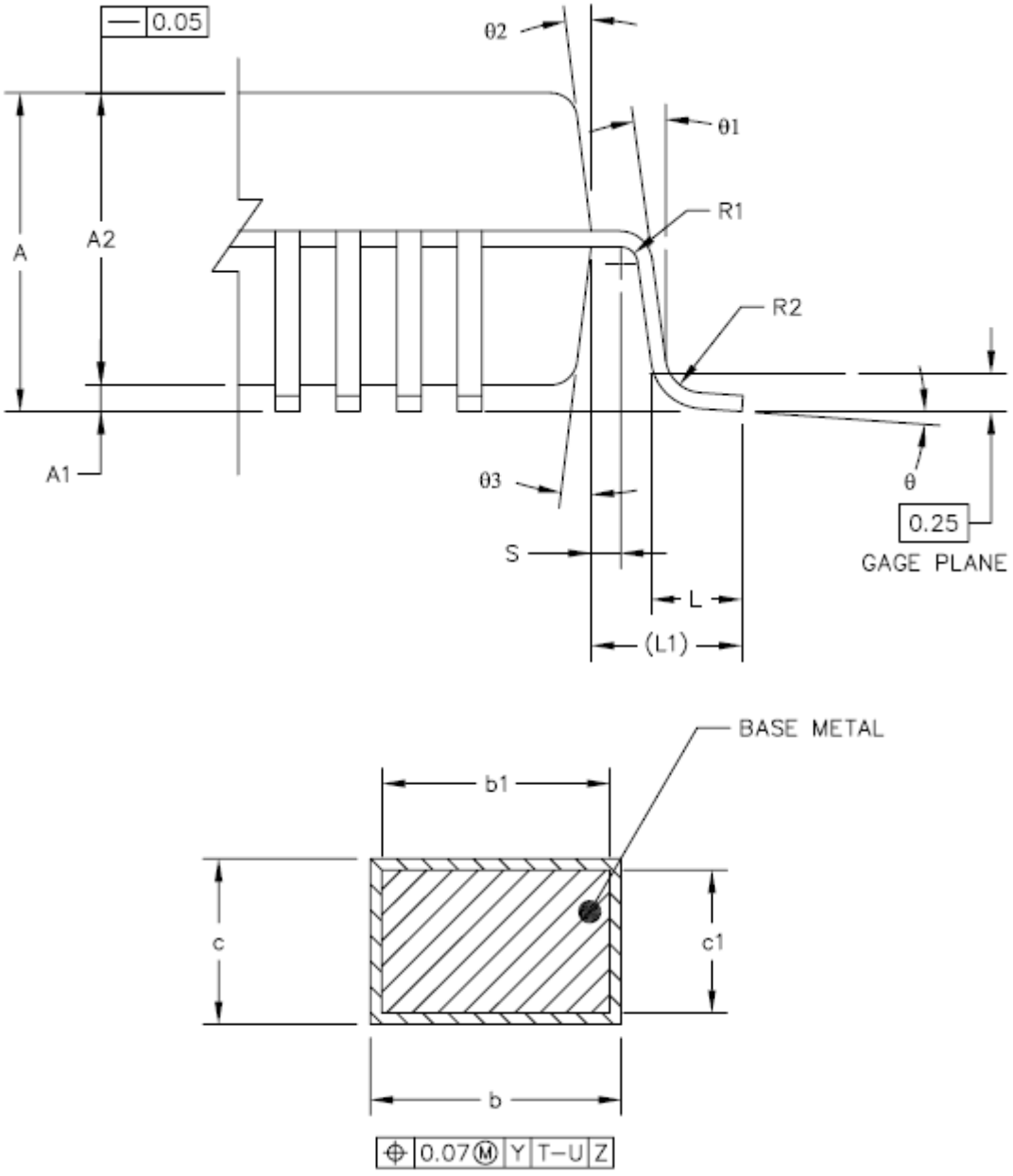
6.1.1 Top View



6.1.2 Side View



6.1.3 Lead View



6.1.4 LQFP Outline Dimensions

DIM	Min.	Typ.	Max.	DIM	Min.	Typ.	Max.
A	———		1.6	E1		14 BSC	
A1	0.05		0.15	L	0.45	0.6	0.75
A2	1.35	1.4	1.45	L1		1 REF	
b	0.13	0.16	0.23	R1	0.08		———
b1	0.13		0.19	R2	0.08		0.2
c	0.09		0.2	S	0.2		———
c1	0.09		0.16	θ	0°	3.5°	7°
D		16 BSC		θ1	0°		———
D1		14 BSC		θ2	11°	12°	13°
e		0.4 BSC		θ3	11°	12°	13°
E		16 BSC					
Unit	mm						
Package	14x14x1.4						
Pitch POD	0.4						

6.2 LFBGA 128-Pin Outline Diagram

6.2.1 Top View

This page is leaved blank intentionally.

6.2.2 Side View

This page is leaved blank intentionally.

6.2.3 Bottom View

This page is leaved blank intentionally.

6.2.4 LFBGA Outline Dimensions

This page is leaved blank intentionally.

6.3 Part Number Description

Part Number	Package Size	Lead Free Process
KB3930QF A1	14mm * 14mm 128 pins LQFP	Lead Free