∴ KB3910 Cx Datasheet

Rev. 0.1

1. Revision History....................5

2. Features....................6

2.1 FEATURE SUMMARY....................6

2.2 KB3910 CX /KB3910 BX FEATURE COMPARISON....................8

3. Pin Assignment....................9

3.1 176 PIN LQFP DIAGRAM TOP VIEW....................9

3.2 PIN ASSIGNMENT....................10

3.2.1 Pin Assignment Side A....................11

3.2.2 Pin Assignment Side B....................12

3.2.3 Pin Assignment Side C....................13

3.2.4 Pin Assignment Side D....................14

3.3 PIN DESCRIPTIONS....................15

3.3.1 Low Pin Count Interface Pin Descriptions....................15

3.3.2 X-BUS Interface Pin Descriptions....................15

3.3.3 PS2 Interface Pin Descriptions....................16

3.3.4 Internal Keyboard Encoder Pin Descriptions....................16

3.3.5 SMBus Pin Descriptions....................17

3.3.6 FAN Pin Descriptions....................17

3.3.7 Pulse Width Modulation Pin Descriptions....................17

3.3.8 General Purpose Wake Up Pin Descriptions....................17

3.3.9 General Purpose IO Pin Descriptions....................18

3.3.10 General Purpose Timer Pin Descriptions....................26

3.3.11 Analog To Digital Pin Descriptions....................27

3.3.12 Digital To Analog Pin Descriptions....................27

3.3.13 Expanded I/O Pin Descriptions....................27

3.3.14 8051 External Interface Pin Description....................28

3.3.15 Clock Pin Descriptions....................28

3.3.16 Miscellaneous Pin Descriptions....................28

3.3.17 Power Pin Descriptions....................28

3.4 Hardware Strap....................30

4. Module Descriptions....................31

4.1 SYSTEM ARCHITECTURE....................31

4.2 CHIP ARCHITECTURE....................33

4.2.1 Power Planes....................33

4.2.2 Clock Domains 34

4.2.3 Reset Domains 35

4.2.4 KBC/IKB/EC Hardware and 8051 Internal Communication 36

4.2.5 Internal Memory Map........ 37

4.4 XRAM MODULE........ 38

4.5 GPIO MODULE........ 39

4.5.1 Functional Description 39

4.5.2 GPIO Register Descriptions (Base Address = FC00h, Space 128 bytes).... 41

4.6 KBC MODULE........ 46

4.6.1 Functional Description 46

4.6.2 KBC Register Descriptions (Base Address = FC00h, Space 32 bytes)........ 49

4.7 IKB MODULE........ 54

4.7.1 IKB Funtional Descriptions 54

4.7.2 IKB Register Descriptions (Base address = FC00h, Space = 352 bytes) 58

4.8 PWM MODULE........ 60

4.8.1 PWM Funtional Description 60

4.8.2 PWM Register Descriptions (Base address = FE00h, Space 32 bytes)....... 61

4.9 FAN MODULE........ 62

4.9.1 FAN Funtional Descriptions 62

4.9.2 FAN Register Descriptions (Base Address = FE00h, Space = 48 bytes)..... 63

4.10 GPT MODULE 64

4.10.1 GPT Funtional Description 64

4.10.2 GPT Register Descriptions (Base Address = FE00h, Space = 32 bytes)... 65

4.11 WDT MODULE 67

4.11.1 WDT Funtional Description 67

4.11.2 WDT Register Descriptions (Base Address = FE00h, Space = 16 bytes)... 68

4.12 LPC MODULE 70

4.12.1 LPC/FWH Funtional Description 70

4.12.2 LPC Register Descriptions (Base Address = FE00h, Space = 16 bytes) ... 71

4.13 XBI MODULE........ 73

4.13.1 XBI Funtional Description........ 73

4.13.2 XBI Register Descriptions (Base Address = FE00h, Space = 48 bytes) 74

4.14 XIO MODULE 77

4.14.1 XIO Funtional Description 77

4.14.2 XIO Register Descriptions (Base Address = FE00h, Space = 16 bytes).... 77

4.15 PS2 MODULE........ 78

4.15.1 PS2 Funtional Description 78

4.15.2 PS2 Register Descriptions (Base Address = FE00h, Space = 32 bytes).... 82

4.16 EC MODULE..86
4.16.1 EC Funtional Description 86
4.16.2 EC Register Descriptions (Base Address = FF00h, Space = 32 bytes) 91
4.17 GPWU MODULE..95
4.17.1 GPWU Funtional Description.. 95
4.17.2 GPWU Register Descriptions (Base Addr= FF00h, Space = 96 bytes)...... 97
4.18 SMBUS MODULE..99
4.18.1 SMBus Funtional Description 99
4.18.2 SMBus Register Descriptions 102
4.19 POWER MANAGEMENT..104
4.20 8051 MODULE..105
4.20.1 8051 Funtional Description..105
4.20.2 8051 SFR Descriptions (Direct Addressing 80h~FFh) 107
5 Electronic Characteristics..110
5.1 ABSOLUTE MAXIMUM RATINGS..110
5.2 RECOMMENDED OPERATING CONDITIONS 110
5.3 OPERATING CURRENT..110
5.4 A/D CHARACTERISTICS 110
5.5 D/A CHARACTERISTICS 110
6. Package Information..112
6.1 176 LQFP (24X24) PACKAGE DIMENSION 112
6.2 176 LQFP (20X20) PACKAGE DIMENSION 114
6.3 PART NUMBER DESCRIPTIONS..116

1. Revision History

Rev.	Preliminary/Changes	Date
Draft 0.1	Initial draft datasheet for engineer usage only.	Apr. 29th, 2005

ENE reserves the right to amend this document without notice at any time. ENE assumes no responsibility for any errors appeared in the document, and ENE disclaims any express or implied warranty, relating to sale and/or use of ENE products including liability or warranties relating to fitness for a particular purpose, or infringement of any patent, copyright or other intellectual property right.

ENE Head Ofice:
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:
9F, No. 88, Bauchiau Rd.,
Shindian City
Taipei, Taiwan, R.O.C.
TEL: 886-2-89111525
FAX: 886-2-89111523

2. Features

2.1 Feature Summary

Low Pin Count Host Interface (LPC)

- ✓ LPC Specification 1.0 compliant
- ✓ LPC or Firmware Hub (FWH) Bus protocols
- ✓ Serial IRQ interface supporting IRQ1, IRQ12, and SCI interrupt
- ✓ CLKRUN# supporting LPC power management

X-Bus Bus Interface (XBI)

- ✓ Interfaces to ISA-Type memory and I/O devices
- ✓ External memory devices with addressable range up to 2MB
- ✓ External I/O devices with addressable range up to 64KB
- ✓ 8 external I/O chip select pins
- ✓ Independently programmable memory and I/O read/write cycle timing
- ✓ Map 8051 64K memory space to 4 different 16KB pages within the external chip

8051 Micro Controller

- ✓ Instruction sets compliant with standard 8051
- ✓ Operate on 2, 4, 8, 16 MHz clocks
- ✓ Fast instruction fetching from XBI
- ✓ 3~5 cycles per instruction.
- ✓ Built-in 2 KB + 128 bytes SRAM
- ✓ 24 extended interrupt sources supporting built-in peripherals
- ✓ Von Neumann Architecture (code and data space are the same)

Keyboard and PS/2 Controller

- ✓ Full hardware implemented standard keyboard and PS2 command processing
- ✓ Fast response time, low power consumption, low 8051 performance required
- ✓ Password quick-lock for system security
- ✓ 3 external PS/2 devices and one internal keyboard encoder (IKB)
- ✓ Hot-swap function allows dynamic swapping of PS/2 devices
- ✓ Hot-plug and Hot-removal for dynamic PS/2 devices plug-in and removed
- ✓ Support intelli-mouse, active PS/2 multiplexing, and PS/2 deviceswake-up

Internal Keyboard Encoder (IKB)

- ✓ 18 X 8 scan matrix
- ✓ Downloadable internal keyboard scan code
- ✓ Support W2K internet/multimedia keys
- ✓ Support up to 76 hot-key events processed by 8051
- ✓ Programmable scan timing

- ✓ Ghost key phenomenon cancellation

Embedded Controller (EC)

- ✓ ACPI Specification 2.0 compliant
- ✓ Support custom extended commands forwarded to 8051
- ✓ Programmable EC I/O port addressing (default 62h/66h)
- ✓ Support Index Addressing Mode allowing Host to access EC space directly
- ✓ SCI Event triggered by Embedded peripheral devices
- ✓ Two identical and independent SMBus Host Controller compliant with ACPI 2.0

SMBus Host Controller (SMBUS)

- ✓ Two built-in full Hardware SMBus Host Controllers
- ✓ SMBus Specification 2.0 compliant
- ✓ SMBus interrupts output to SCI and 8051
- ✓ Wake-up from low-power modes.

Digital To Analog Converter (DAC)

- ✓ 8 built-in DACs with 8-bit resolution
- ✓ The DAC pins can be alternatively configured as GPOs

Analog To Digital Converter (ADC)

- ✓ 8 built-in ADCs with 8-bit resolution
- ✓ The ADC pins can be alternatively configured as GPIs

Pulse Width Modulator (PWM)

- ✓ 8 built-in PWMs
- ✓ Selectable clock sources: 1MHz/16KHz/4KHz/256Hz
- ✓ Configurable cycle time and duty cycle
- ✓ Programmable output polarity

Watchdog Timer (WDT)

- ✓ 32.768KHz input clock with 5-bit prescaler
- ✓ 16-bit watchdog timer

General Purpose Timer (GPT)

- ✓ 4 built-in general purpose timer modules
- ✓ Selectable clock sources: 32.768KHz/System Clock (4MHz) / External Events
- ✓ 16-Bit programmable timer

General Purpose Wake-Up (GPWU)

- ✓ 8 GPWUs with debounce input
- ✓ Individual enable bit and event pending flag bit
- ✓ Configurable active-high or active -low
- ✓ Configurable edge or level Trigger
- ✓ All GPI pins can optionally generate interrupts or wake-up events

General Purpose Input/Output (GPIO)

- ✓ 48 GPIOs
- ✓ GPOs with configurable output enable and output data
- ✓ GPIs with configurable pull-up, high/low active, edge/level trigger
- ✓ KSO, DAC and PWM Pins can be optionally configured as GPOs
- ✓ KSI and ADC pins can be optionally configured as GPIs
- ✓ 16 GPOs equipped with selectable 4mA/16mA output buffer for LED driving

Fan Controller (FAN)

- ✓ 3 fan controllers with automatic fan speed control from 600rpm to 6000rpm
- ✓ Automatic fan and PWM counting clock selection

Power Management

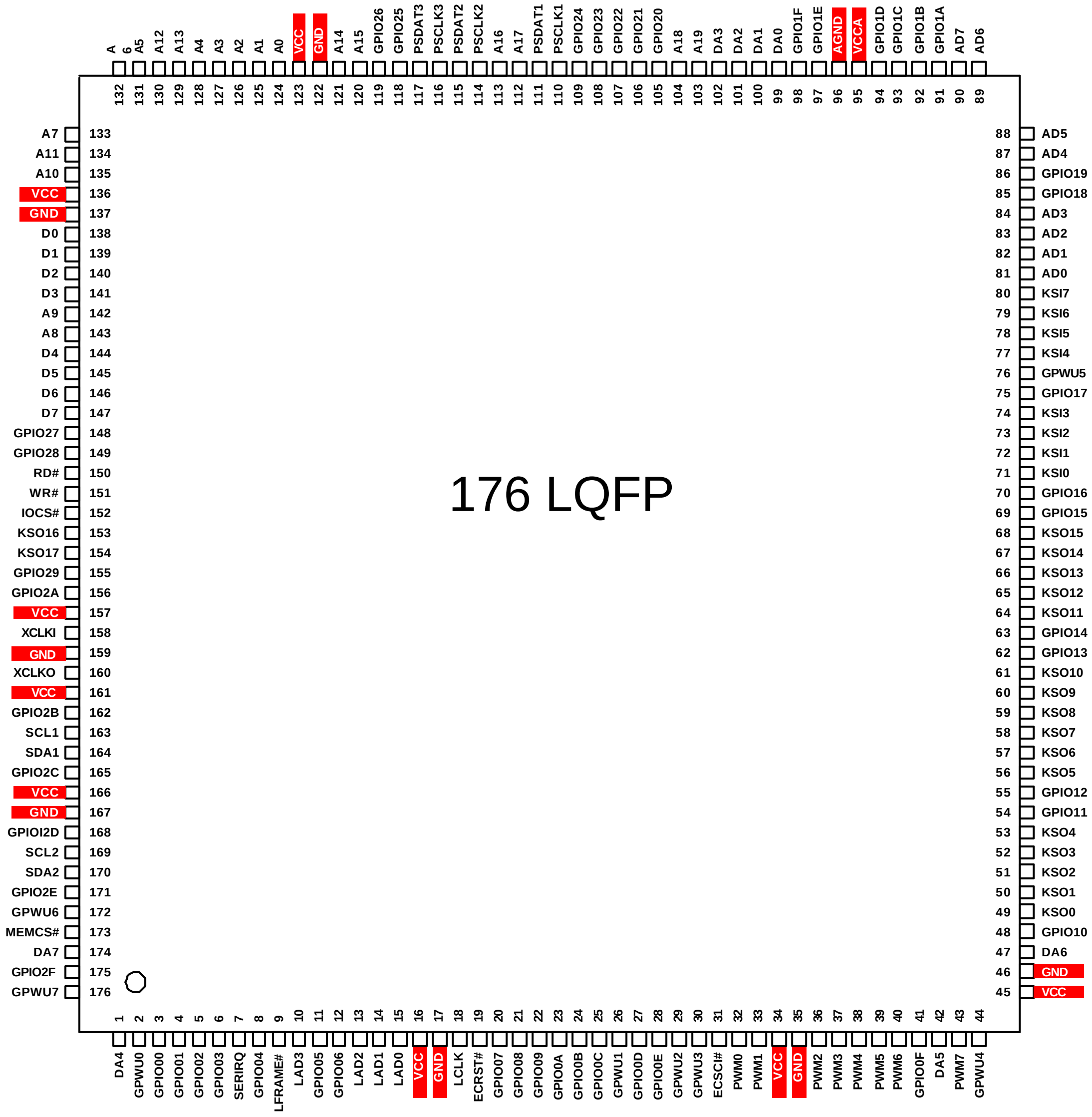
- ✓ Sleep State: 8051 Program Counter (PC) stopped
- ✓ Deep Sleep State: Stop all internal clocks; all internal modules are idle

2.2 KB3910 Cx /KB3910 Bx Feature Comparison

	KB3910 Cx	KB3910Bx
X-Bus Memory Range	2M bytes	2M bytes
RTC	Without	Embedded
ADC	8	8
DAC	8	8
PWM	8	8
Ecternal PS/2 devices	3	3
GPIOs	48	48
KB matrix scan	18x8	18x8
FAN	3	3
GPWU	8	8
Smbus	2	2
IO Chip Select (XIOCS#)	8	8

3. Pin Assignment

3.1 176 Pin LQFP Diagram Top View



3.2 Pin Assignment

IO Buffer Description

The internal pull-up scheme for all I/O buffers alike is pull-up to 3.3V via an 80K ohm resistor. Care should be taken if these pins are applied with external pull-up to 5V. Under such condition there is a small amount leakage current flowing through the 5V into the internal 3.3V power well and consumes power.

For easy understanding, different colors are used in the following tables to represent various pin groups and IO buffers.

IO Buffer Table:

Port IO Name	IO	I	O	OE	IE	HI	AE	H	PE	Output Current
BQC04HU	v	V	V	v	v				80K	2~4mA
BQC16HU	v	V	V	v	v	V			80K	2~4/8~16mA
IQU	v		V		v				80K	2~4mA
OC04	v	V		v						2~4mA
BCC16H	v	V	V	v	v			V		8~16mA
IQA	v		V		v		V			2~4mA
OC04A	v	V		v			V			2~4mA

KB3910 IO Descriptions:

IO Name	Descriptions	Applications
BQC04HU	Output / Pull high / Input Enable, schmitter trigger, 2~4mA Output/Sink Current	GPIO
BQC16HU	Output / Pull high / Input Enable, schmitter trigger, 2~4mA(default)/8~16mA select	GPIO for LED output
IQU	Pull high / Input Enable, schmitter trigger	Keyboard Scanin
OC04	Output Enable	Keyboard Scanout
BCC16H	8~16mA Output/Sink Current, Output / Input Enable, 5 Voltage Tolerance	LPC Interface
IQA	mixed mode IO, ADC Enable, with GPI, 2~4mA Sink Current, Input Enable	ADC, GPIN
OC04A	mixed mode IO, DAC Enable, with GPO, 2~4mA Output Current ,Output Enable	DAC, GPOUT

IO Buffer Naming Convention:

I--IO Buffer Input
O—IO Buffer Output
OE—IO Buffer Output Enable
IE—IO Buffer Input Enable
PE—IO Buffer Pull High Enable
HI—IO Buffer Output High Current
AE—IO Buffer Analog mode Enable(AE > OE)
Q—Schimitter Trigger
H—5V Tolerant

Legend: color code for pin groups:

DAC0~7
ADC0~7
GPWU0~7
GPIO00~2F
Power
Ground

3.2.1 Pin Assignment Side A

176 pin	**Pin**	**Alt. Output**	**Alt. Input**	**GPIO**	**Default**	**State During ECRST# Low / Hi**	**IO Cell**
1	DA4	GPODA4		GPODA4	DA4	HiZ / HiZ	OC04A
2	GPWU0			GPWU0	GPWU0	HiZ / HiZ	BQC04HU
3	GPIO00		E51IT0	GPIO00	GPIO00	HiZ / HiZ	
4	GPIO01		E51IT1	GPIO01	GPIO01	HiZ / HiZ	
5	GPIO02	GA20		GPIO02	GA20	OL / OL	
6	GPIO03	KBRST#		GPIO03	KBRST#	OH / OH	
7	SERIRQ				SERIRQ	HiZ / HiZ	BCC16H
8	GPIO04			GPIO04	GPIO04	HiZ / HiZ	BQC04HU
9	LFRAME#		FWH4		LFRAME#	HiZ / HiZ	BCC16H
10	LAD3	FWH3	FWH3		LAD3	HiZ / HiZ	
11	GPIO05	FAN3PWM	TEST_TP	GPIO05	GPIO05	IE / HiZ	BQC04HU
12	GPIO06		FANFB3 DPLL_TP	GPIO06	GPIO06	IE / HiZ	
13	LAD2	FWH2	FWH2		LAD2	HiZ / HiZ	BCC16H
14	LAD1	FWH1	FWH1		LAD1	HiZ / HiZ	
15	LAD0	FWH0	FWH0		LAD0	HiZ / HiZ	BCC16H
16	VCC						VCC
17	GND						GND
18	LCLK				LCLK		BCC16H
19	ECRST#				ECRST#		BQC04HU
20	GPIO07			GPIO07	GPIO07	HiZ / HiZ	
21	GPIO08			GPIO08	GPIO08	HiZ / HiZ	
22	GPIO09			GPIO09	GPIO09	HiZ / HiZ	BQC04HU
23	GPIO0A	NumLock#		GPIO0A	GPIO0A	HiZ / HiZ	BQC16HU
24	GPIO0B			GPIO0B	GPIO0B	HiZ / HiZ	BCC16H
25	GPIO0C	CLKRUN#	CLKRUN#	GPIO0C	CLKRUN#	HiZ / HiZ	
26	GPWU1			GPWU1	GPWU1	HiZ / HiZ	BQC04HU
27	GPIO0D			GPIO0D	GPIO0D	HiZ / HiZ	
28	GPIO0E			GPIO0E	GPIO0E	HiZ / HiZ	
29	GPWU2			GPWU2	GPWU2	HiZ / HiZ	BQC04HU
30	GPWU3			GPWU3	GPWU2	HiZ / HiZ	BQC04HU
31	ECSCI#				ECSCI#	HiZ / HiZ	BQC04HU
32	PWM0			GPOW0	PWM0	HiZ / HiZ	
33	PWM1			GPOW1	PWM1	HiZ / HiZ	
34	VCC						VCC
35	GND						GND
36	PWM2	FAN1PWM		GPOW2	PWM2	HiZ / HiZ	BQC04HU
37	PWM3			GPOW3	PWM3	HiZ / HiZ	
38	PWM4			GPOW4	PWM4	HiZ / HiZ	
39	PWM5			GPOW5	PWM5	HiZ / HiZ	
40	PWM6			GPOW6	PWM6	HiZ / HiZ	
41	GPIO0F	ScrollLock#		GPIO0F	GPIO0F	HiZ / HiZ	BQC16HU
42	DA5	GPODA5		GPODA5	GPODA5	HiZ / HiZ	OC04A
43	PWM7	FAN2PWM		GPOW7	PWM7	HiZ / HiZ	BQC04HU
44	GPWU4			GPWU4	GPWU4	HiZ / HiZ	BQC04HU

Legend:
HiZ=Input/Output Disabled. IE= Input Enable, OL=Output Low, OH=Output High
PU=Input/Output Disabled with Internal Pull-up or pull_Down determined by hardware strap option

3.2.2 Pin Assignment Side B

176 pin	Pin	Alt. Output	Alt. Input	GPIO	Default	State During ECRST# Low/Hi	IO Cell
45	VCC						VCC
46	GND						GND
47	DA6	GPODA6		GPODA6	DA6	HiZ / HiZ	OC04A
48	GPIO10			GPIO10	GPIO10	HiZ / HiZ	BQC04HU
49	KSO0	GPOK0		GPOK0	KSO0	HiZ / HiZ	OC04
50	KSO1	GPOK1		GPOK1	KSO1	HiZ / HiZ	
51	KSO2	GPOK2		GPOK2	KSO2	HiZ / HiZ	
52	KSO3	GPOK3		GPOK3	KSO3	HiZ / HiZ	OC04
53	KSO4	GPOK4		GPOK4	KSO4	HiZ / HiZ	
54	GPIO11	CapLock#		GPIO11	GPIO11	HiZ / HiZ	BQC16HU
55	GPIO12	FnLock#		GPIO12	GPIO12	HiZ / HiZ	
56	KSO5	GPOK5		GPOK5	KSO5	HiZ / HiZ	OC04
57	KSO6	GPOK6		GPOK6	KSO6	HiZ / HiZ	
58	KSO7	GPOK7		GPOK7	KSO7	HiZ / HiZ	
59	KSO8	GPOK8		GPOK8	KSO8	HiZ / HiZ	OC04
60	KSO9	GPOK9		GPOK9	KSO9	HiZ / HiZ	
61	KSO10	GPOK10		GPOK10	KSO10	HiZ / HiZ	
62	GPIO13			GPIO13	GPIO13	HiZ / HiZ	BQC04HU
63	GPIO14			GPIO14	GPIO14	HiZ / HiZ	
64	KSO11	GPOK11		GPOK11	KSO11	HiZ / HiZ	OC04
65	KSO12	GPOK12		GPOK12	KSO12	HiZ / HiZ	
66	KSO13	GPOK13		GPOK13	KSO13	HiZ / HiZ	OC04
67	KSO14	GPOK14		GPOK14	KSO14	HiZ / HiZ	
68	KSO15	GPOK15		GPOK15	KSO15	HiZ / HiZ	
69	GPIO15			GPIO15	GPIO15	HiZ / HiZ	BQC04HU
70	GPIO16			GPIO16	GPIO15	HiZ / HiZ	
71	KSI0	GPIK0	GPIK0	GPIK0	KSI0	IE /IE	IQU
72	KSI1		GPIK1	GPIK1	KSI1	IE /IE	IQU
73	KSI2		GPIK2	GPIK2	KSI2	IE /IE	
74	KSI3		GPIK3	GPIK3	KSI3	IE /IE	
75	GPIO17			GPIO17	GPIO17	HiZ / HiZ	BQC04HU
76	GPWU5			GPWU5	GPWU5	HiZ / HiZ	
77	KSI4		GPIK4	GPIK4	KSI4	IE /IE	IQU
78	KSI5		GPIK5	GPIK5	KSI5	IE /IE	
79	KSI6		GPIK6	GPIK6	KSI6	IE /IE	IQU
80	KSI7		GPIK7	GPIK7	KSI7	IE /IE	
81	AD0		GPIAD0	GPIAD0	AD0	HiZ / HiZ	IQA
82	AD1		GPIAD1	GPIAD1	AD1	HiZ / HiZ	
83	AD2		GPIAD2	GPIAD2	AD2	HiZ / HiZ	
84	AD3		GPIAD3	GPIAD3	AD3	HiZ / HiZ	
85	GPIO18	XIO8CS#		GPIO18	GPIO18	PU / PU	BQC16HU
86	GPIO19	XIO9CS#		GPIO19	GPIO19	PU / PU	
87	AD4		GPIAD4	GPIAD4	AD4	HiZ / HiZ	IQA
88	AD5		GPIAD5	GPIAD5	AD5	HiZ / HiZ	

Legend:
HiZ=Input/Output Disabled. IE= Input Enable, OL=Output Low, OH=Output High
PU=Input/Output Disabled with Internal Pull-up or pull_Down determined by hardware strap option

3.2.3 Pin Assignment Side C

176 pin	Pin	Alt. Output	Alt. Input	GPIO	Default	State During ECRST# Low/Hi	IO Cell
89	AD6		GPIAD6	GPIAD6	AD6	HiZ / HiZ	IQA
90	AD7		GPIAD7	GPIAD7	AD7	HiZ / HiZ	
91	GPIO1A	XIOACS#		GPIO1A	GPIO1A	PU / PU	BQC16HU
92	GPIO1B	XIOBCS#		GPIO1B	GPIO1B	PU / PU	
93	GPIO1C	XIOCCS#		GPIO1C	GPIO1C	PU / PU	
94	GPIO1D	XIODCS#		GPIO1D	GPIO1D	PU / PU	
95	VCCA						VCCA
96	AGND						AGND
97	GPIO1E	XIOECS#		GPIO1E	GPIO1E	PU / PU	BQC16HU
98	GPIO1F	XIOFCS#		GPIO1F	GPIO1F	PU / PU	
99	DA0	GPODA0		GPODA0	DA0	HiZ / HiZ	OC04A
100	DA1	GPODA1		GPODA1	DA1	HiZ / HiZ	
101	DA2	GPODA2		GPODA2	DA2	HiZ / HiZ	
102	DA3	GPODA3		GPODA3	DA3	HiZ / HiZ	
103	A19				A19	OL / OL	
104	A18				A18	OL / OL	
105	GPIO20	E51CS#		GPIO20	GPIO20	IE / HiZ	
106	GPIO21		E51RXD	GPIO21	GPIO21	IE / HiZ	
107	GPIO22	E51TXD		GPIO22	GPIO22	IE / HiZ	
108	GPIO23	A20		GPIO23	A20	HiZ / HiZ	
109	GPIO24			GPIO24	GPIO24	HiZ / HiZ	
110	PSCLK1				PSCLK1	IE / IE	
111	PSDAT1				PSDAT1	IE / IE	
112	A17				A17	OL / OL	
113	A16				A16	OL / OL	
114	PSCLK2				PSCLK2	IE / IE	
115	PSDAT2				PSDAT2	IE / IE	
116	PSCLK3				PSCLK3	IE / IE	
117	PSDAT3				PSDAT3	IE / IE	
118	GPIO25			GPIO25	GPIO25	HiZ / HiZ	
119	GPIO26			GPIO26	GPIO26	HiZ / HiZ	
120	A15				A15	OL / OL	
121	A14				A14	OL / OL	
122	GND						GND
123	VCC						VCC
124	A0				A0	OL / OL	
125	A1		XIOP_TP		A1	IE / OL	
126	A2				A2	OL /OL	
127	A3				A3	OL / OL	
128	A4		DMRP_TP		A4	IE / OL	
129	A13				A13	OL / OL	
130	A12				A12	OL / OL	
131	A5		EMWB_TP		A5	IE / OL	
132	A6				A6	OL / OL	

Legend:
HiZ=Input/Output Disabled. IE= Input Enable, OL=Output Low, OH=Output High
PU=Input/Output Disabled with Internal Pull-up or pull_Down determined by hardware strap option

3.2.4 Pin Assignment Side D

176 pin	Pin	Alt. Output	Alt. Input	GPIO	Default	State During ECRST# Low/Hi	IO Cell
133	A7				A7	OL / OL	
134	A11				A11	OL / OL	
135	A10				A10	OL / OL	
136	VCC						VCC
137	GND						GND
138	D0				D0	HiZ / HiZ	BQC04HU
139	D1				D1	HiZ / HiZ	
140	D2				D2	HiZ / HiZ	
141	D3				D3	HiZ / HiZ	
142	A9				A9	OL / OL	
143	A8				A8	OL / OL	
144	D4				D4	HiZ / HiZ	
145	D5				D5	HiZ / HiZ	
146	D6				D6	HiZ / HiZ	
147	D7				D7	HiZ / HiZ	
148	GPIO27			GPIO27	GPIO27	HiZ / HiZ	
149	GPIO28			GPIO28	GPIO28	HiZ / HiZ	
150	RD#				RD#	OH / OH	
151	WR#				WR#	OH /OH	
152	IOCS#				IOCS#	HiZ / HiZ	
153	KSO16	GPOK16		GPOK16	KSO16	HiZ / HiZ	OC04
154	KSO17	GPOK17		GPOK17	KSO17	HiZ / HiZ	OC04
155	GPIO29			GPIO29	GPIO29	HiZ / HiZ	BQC04HU
156	GPIO2A			GPIO2A	GPIO2A	HiZ / HiZ	
157	VCC						VCC
158	XCLKI				XCLK1		
159	GND						GND
160	XCLKO				XCLKO		
161	VCCB						VCC
162	GPIO2B			GPIO2B	GPIO2B	HiZ / HiZ	
163	SCL1				SCL1	IE / IE	
164	SDA1				SDA1	IE / IE	
165	GPIO2C		LRST#	GPIO2C	GPIO2C	IE / IE	
166	VCC						VCC
167	GND						GND
168	GPIO2D			GPIO2D	GPIO2D	HiZ / HiZ	
169	SCL2				SCL2	IE / IE	
170	SDA2				SDA2	IE / IE	
171	GPIO2E	TOUT1	FANFB1	GPIO2E	GPIO2E	HiZ / HiZ	
172	GPWU6		TIN1	GPWU6	GPWU6	HiZ / HiZ	
173	MEMCS#				MEMCS#	HiZ / HiZ	
174	DA7	GPODA7		GPODA7	DA7	HiZ / HiZ	OC04A
175	GPIO2F	TOUT2		GPIO2F	GPIO2F	HiZ / HiZ	
176	GPWU7		TIN2 FANFB2	GPWU7	GPWU7	HiZ / HiZ	

Legend:
HiZ=Input/Output Disabled. IE= Input Enable, OL=Output Low, OH=Output High
PU=Input/Output Disabled with Internal Pull-up or pull_Down determined by har dware strap option

3.3 Pin Descriptions

3.3.1 Low Pin Count Interface Pin Descriptions

PN176	Pin name	Direction	Description
15	**LAD0**	I/O	LPC LAD0: Multiplexed Address, Data and Command. FWH0: I/O communication
14	**LAD1**	I/O	LPC LAD1: Multiplexed Address, Data and Command. FWH1: I/O communication
13	**LAD2**	I/O	LPC LAD2: Multiplexed Address, Data and Command. FWH2: I/O communication
10	**LAD3**	I/O	LPC LAD3: Multiplexed Address, Data and Command. FWH3: I/O communication
9	**LFRAME#**	I	LPC: The Frame# signal FWH4: Input communication
165	**LRST#**	I	Reset signal used to reset the LPC interface. This pin MUST be connected to the PCIRST# signal of the system.
18	**LCLK**	I	Clock for LPC interface. This pin MUST be connected to the PCI clock signal of the system.
7	**SERIRQ**	I/O	Serialized IRQ.
25	**CLKRUN#**	OD	This pin is an optional function for PCI CLKRUN#. This signal is used to control the start/stop of PCI clock. The CLKRUN# may be not used in KB3910 application.

3.3.2 X-BUS Interface Pin Descriptions

PN176	Pin name	Direction	Description
150	**RD#**	O	Read pulse to external devices
151	**WR#**	O	Write pulse to external devices
173	**MEMCS#**	O	Memory cycle chip select pulse
152	**IOCS#**	O	I/O Cycle chip select pulse
138	**D0**	I/O	Data bit 0
139	**D1**	I/O	Data bit 1
140	**D2**	I/O	Data bit 2
141	**D3**	I/O	Data bit 3
144	**D4**	I/O	Data bit 4
145	**D5**	I/O	Data bit 5
146	**D6**	I/O	Data bit 6
147	**D7**	I/O	Data bit 7
124	**A0**	O	X-bus Address 0
125	**A1**	O	X-bus Address 1
126	**A2**	O	X-bus Address 2
127	**A3**	O	X-bus Address 3
128	**A4**	O	X-bus Address 4
131	**A5**	O	X-bus Address 5
132	**A6**	O	X-bus Address 6
133	**A7**	O	X-bus Address 7
143	**A8**	O	X-bus Address 8
142	**A9**	O	X-bus Address 9
135	**A10**	O	X-bus Address 10
134	**A11**	O	X-bus Address 11
130	**A12**	O	X-bus Address 12
129	**A13**	O	X-bus Address 13

121	**A14**	O	X-bus Address 14
120	**A15**	O	X-bus Address 15
113	**A16**	O	X-bus Address 16
112	**A17**	O	X-bus Address 17
104	**A18**	O	X-bus Address 18
103	**A19**	O	X-bus Address 19
108	**A20**	O	X-bus Address 20
105	**E51CS#**	O	The E51CS# can be used to select secondary memory spaces for 8051 development environment support. Register BI_MS2 (FEA7h) controls the mapping of the four 16-K segments SEG3-0 memory spaces individually into secondary memories. When a particular memory space is being configured to map into secondary memory, E51CS# will be asserted instead of MEMCS#. The assertion of E51CS# and MEMCS# is mutual exclusive.

3.3.3 PS2 Interface Pin Descriptions

PN176	Pin name	Direction	Description
110	**PSCLK1**	I/O	PS2 port_0 clock
111	**PSDAT1**	I/O	PS2 Port_0 Data
114	**PSCLK2**	I/O	PS2 port_1 clock
115	**PSDAT2**	I/O	PS2 port_1 data
116	**PSCLK3**	I/O	PS2 port_2 clock
117	**PSDAT3**	I/O	PS2 port_2 data

3.3.4 Internal Keyboard Encoder Pin Descriptions

PN176	Pin name	Direction	Description
49	**KSO0**	O	Keyboard Scan-Out 0. The pre-charge time and driving time for KSO0~17 is programmable in register STCTL (FCAAh). The KSO0~17 can be alternatively configured as general purpose outputs (GPO)
50	**KSO1**	O	Keyboard Scan-Out 1
51	**KSO2**	O	Keyboard Scan-Out 2
52	**KSO3**	O	Keyboard Scan-Out 3
53	**KSO4**	O	Keyboard Scan-Out 4
56	**KSO5**	O	Keyboard Scan-Out 5
57	**KSO6**	O	Keyboard Scan-Out 6
58	**KSO7**	O	Keyboard Scan-Out 7
59	**KSO8**	O	Keyboard Scan-Out 8
60	**KSO9**	O	Keyboard Scan-Out 9
61	**KSO10**	O	Keyboard Scan-Out 10
64	**KSO11**	O	Keyboard Scan-Out 11
65	**KSO12**	O	Keyboard Scan-Out 12
66	**KSO13**	O	Keyboard Scan-Out 13
67	**KSO14**	O	Keyboard Scan-Out 14
68	**KSO15**	O	Keyboard Scan-Out 15
153	**KSO16**	O	Keyboard Scan-Out 16
154	**KSO17**	O	Keyboard Scan-Out 17
71	**KSI0**	I	Keyboard Scan-In 0. The KSI0~7 pins can be alternatively configured as general purpose inputs (GPI)
72	**KSI1**	I	Keyboard Scan-In 1
73	**KSI2**	I	Keyboard Scan-In 2
74	**KSI3**	I	Keyboard Scan-In 3

77	**KSI4**	I	Keyboard Scan-In 4
78	**KSI5**	I	Keyboard Scan-In 5
79	**KSI6**	I	Keyboard Scan-In 6
80	**KSI7**	I	Keyboard Scan-In 7

3.3.5 SMBus Pin Descriptions

PN176	Pin name	Direction	Description
163	**SCL1**	I/O	SMBus 1 clock
164	**SDA1**	I/O	SMBus 1 Data
169	**SCL2**	I/O	SMBus 2 clock
170	**SDA2**	I/O	SMBus 2 Data

3.3.6 FAN Pin Descriptions

PN176	Pin name	Direction	Description
171	**FANFB1**	I	FAN_1 Tachometer Input
176	**FANFB2**	I	FAN_2 Tachometer Input
12	**FANFB3**	I	FAN_3 Tachometer Input
43	**FAN2PWM**	O	FAN_2 PWM output to control the Fan Speed
36	**FAN1PWM**	O	FAN_1 PWM output to control the Fan Speed
11	**FAN3PWM**	O	FAN_3 PWM output to control the Fan Speed

3.3.7 Pulse Width Modulation Pin Descriptions

PN176	Pin name	Direction	Description
32	**PWM0**	O	PWM_0 output. The PWM0~7 pins can be alternatively configured as general purpose outputs
33	**PWM1**	O	PWM_1 output
36	**PWM2**	O	PWM_2 output
37	**PWM3**	O	PWM_3 output
38	**PWM4**	O	PWM_4 output
39	**PWM5**	O	PWM_5 output
40	**PWM6**	O	PWM_6 output
43	**PWM7**	O	PWM_7 output

3.3.8 General Purpose Wake Up Pin Descriptions

PN176	Pin name	Direction	Description
2	**GPWU0**	I	General Purpose Wake-up pin 0. The GPWU0~7 are used to generate wake-up events to bring the KB3910 back to normal operation state while in Sleep mode or Deep sleep mode. GPWU0~7 are equipped with input de-bounce logic for eliminating external glitches and spikes introduced by push buttons or Ring-in signals...etc. GPWU0~7 events can be configured to generate a 8051 interrupt or a SCI event to inform the system host. The polarity and edge-/level- trigger can be configured for each pin. GPWU0~7 pins can be alternatively used as general purpose inputs (GPI) pins.
26	**GPWU1**	I	General Purpose Wake-up pin 1
29	**GPWU2**	I	General Purpose Wake-up pin 2
30	**GPWU3**	I	General Purpose Wake-up pin 3
44	**GPWU4**	I	General Purpose Wake-up pin 4
76	**GPWU5**	I	General Purpose Wake-up pin 5
172	**GPWU6**	I	General Purpose Wake-up pin 6
176	**GPWU7**	I	General Purpose Wake-up pin 7

3.3.9 General Purpose IO Pin Descriptions

PN176	Name	In/Out	Description						
3	GPIO00	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: E51IT0						
			Power-on Default: GPIO00						
4	GPIO01	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: E51IT1						
			Power-on Default: GPIO01						
5	GPIO02	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: GA20						
			Power-on Default: GA20						
6	GPIO03	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: KBRST#						
			Power-on Default: KBRST#						
8	GPIO04	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO04						
11	GPIO05	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: FAN3PWM						
			Power-on Default: GPIO05						
12	GPIO06	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: FANFB3, DPLL_TP						
			Power-on Default: GPIO06						
20	GPIO07	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO07						
21	GPIO08	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO08						
22	GPIO09	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO09						
23	GPIO0A	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: NumLock#						
			Power-on Default: GPIO0A						
24	GPIO0B	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			16mA	v	v	v	v	v	v

			Alternative Function: LPCPD#						
			Power-on Default: GPIO0B						
25	**GPIO0C**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			16mA	v	v	v	v	v	v
			Alternative Function: CLKRUN#						
			Power-on Default: CLKRUN#						
27	**GPIO0D**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO0D						
28	**GPIO0E**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO0E						
41	**GPIO0F**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4/16mA	v	v	v	v	v	v
			Alternative Function: ScrollLock#						
			Power-on Default: GPIO0F						
48	**GPIO10**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO10						
54	**GPIO011**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4/16mA	v	v	v	v	v	v
			Alternative Function: CapLock#						
			Power-on Default: GPIO11						
55	**GPIO12**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4/16mA	v	v	v	v	v	v
			Alternative Function: FnLock#						
			Power-on Default: GPIO12						
62	**GPIO13**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO13						
63	**GPIO14**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO14						
69	**GPIO15**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO15						
70	**GPIO16**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v
			Alternative Function: none						
			Power-on Default: GPIO16						
75	**GPIO17**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	v	v	v	v	v	v

			Alternative Function: none						
			Power-on Default: GPIO17						
85	GPIO18	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIO8CS#						
			Power-on Default: GPIO18						
86	GPIO19	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIO9CS#						
			Power-on Default: GPIO19						
91	GPIO1A	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIOACS#						
			Power-on Default: GPIO1A						
92	GPIO1B	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIOBCS#						
			Power-on Default: GPIO1B						
93	GPIO1C	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIOCCS#						
			Power-on Default: GPIO1C						
94	GPIO1D	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIODCS#						
			Power-on Default: GPIO1D						
97	GPIO1E	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIOECS#						
			Power-on Default: GPIO1E						
98	GPIO1F	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4/16mA	v	v	v	v	v	v
			Alternative Function: XIOFCS#						
			Power-on Default: GPIO1F						
105	GPIO20	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: E51CS#						
			Power-on Default: GPIO20						
106	GPIO21	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: E51RXD						
			Power-on Default: GPIO21						
107	GPIO22	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v
			Alternative Function: E51TXD						
			Power-on Default: GPIO22						
108	GPIO23	I/O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA	v	v	v	v	v	v

			Alternative Function: A20						
			Power-on Default: A20						
109	**GPIO24**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO24						
118	**GPIO25**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO25						
119	**GPIO26**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO26						
148	**GPIO27**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO27						
149	**GPIO28**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO28						
155	**GPIO29**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO29						
156	**GPIO2A**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO2A						
162	**GPIO2B**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO2B						
165	**GPIO2C**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: LRST#						
			Power-on Default: GPIO2C						
168	**GPIO2D**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: none						
			Power-on Default: GPIO2D						
171	**GPIO2E**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
			Alternative Function: FANFB1, TOUT1						
			Power-on Default: GPIO2E						
175	**GPIO2F**	I/O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ

			Alternative Function: TOUT2						
			Power-on Default: GPIO2F						
49	GPOK0	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO0						
			Power-on Default: KSO0						
50	GPOK1	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO1						
			Power-on Default: KSO1						
51	GPOK2	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO2						
			Power-on Default: KSO2						
52	GPOK3	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO3						
			Power-on Default: KSO3						
53	GPOK4	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO4						
			Power-on Default: KSO4						
56	GPOK5	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO5						
			Power-on Default: KSO5						
57	GPOK6	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO6						
			Power-on Default: KSO6						
58	GPOK7	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO7						
			Power-on Default: KSO7						
59	GPOK8	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO8						
			Power-on Default: KSO8						
60	GPOK9	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function : KSO9						
			Power-on Default: KSO9						
61	GPOK10	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO10						
			Power-on Default: KSO10						
64	GPOK11	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				

			Alternative Function: KSO11						
			Power-on Default: KSO11						
65	GPOK12	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO12						
			Power-on Default: KSO12						
66	GPOK13	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO13						
			Power-on Default: KSO13						
67	GPOK14	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO14						
			Power-on Default: KSO14						
68	GPOK15	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO15						
			Power-on Default: KSO15						
153	GPOK16	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO16						
			Power-on Default: KSO16						
154	GPOK17	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		ˇ				
			Alternative Function: KSO17						
			Power-on Default: KSO17						
71	GPIK0	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				ˇ			ˇ		ˇ
			Alternative Function: KSI0						
			Power-on Default: KSI0						
72	GPIK1	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				ˇ			ˇ		ˇ
			Alternative Function: KSI1						
			Power-on Default: KSI1						
73	GPIK2	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				ˇ			ˇ		ˇ
			Alternative Function: KSI2						
			Power-on Default: KSI2						
74	GPIK3	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				ˇ			ˇ		ˇ
			Alternative Function: KSI3						
			Power-on Default: KSI3						
77	GPIK4	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				ˇ			ˇ		ˇ
			Alternative Function: KSI4						
			Power-on Default: KSI4						
78	GPIK5	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				ˇ			ˇ		ˇ

<table>
<tr><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td colspan="7">Alternative Function: KSI5</td></tr>
<tr><td colspan="7">Power-on Default: KSI5</td></tr>
<tr><td rowspan="4">79</td><td rowspan="4">GPIK6</td><td rowspan="4">I</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td></td><td>v</td><td></td><td></td><td>v</td><td></td><td>v</td></tr>
<tr><td colspan="7">Alternative Function: KSI6</td></tr>
<tr><td colspan="7">Power-on Default: KSI6</td></tr>
<tr><td rowspan="4">80</td><td rowspan="4">GPIK7</td><td rowspan="4">I</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td></td><td>v</td><td></td><td></td><td>v</td><td></td><td>v</td></tr>
<tr><td colspan="7">Alternative Function: KSI7</td></tr>
<tr><td colspan="7">Power-on Default: KSI7</td></tr>
<tr><td rowspan="4">99</td><td rowspan="4">GPODA0</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA0</td></tr>
<tr><td colspan="7">Power-on Default: DA0</td></tr>
<tr><td rowspan="4">100</td><td rowspan="4">GPODA1</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA1</td></tr>
<tr><td colspan="7">Power-on Default: DA1</td></tr>
<tr><td rowspan="4">101</td><td rowspan="4">GPODA2</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA2</td></tr>
<tr><td colspan="7">Power-on Default: DA2</td></tr>
<tr><td rowspan="4">102</td><td rowspan="4">GPODA3</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA3</td></tr>
<tr><td colspan="7">Power-on Default: DA3</td></tr>
<tr><td rowspan="4">1</td><td rowspan="4">GPODA4</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA4</td></tr>
<tr><td colspan="7">Power-on Default: DA4</td></tr>
<tr><td rowspan="4">42</td><td rowspan="4">GPODA5</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA5</td></tr>
<tr><td colspan="7">Power-on Default: DA5</td></tr>
<tr><td rowspan="4">47</td><td rowspan="4">GPODA6</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA6</td></tr>
<tr><td colspan="7">Power-on Default: DA6</td></tr>
<tr><td rowspan="4">174</td><td rowspan="4">GPODA7</td><td rowspan="4">O</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td>4mA</td><td></td><td>v</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: DA7</td></tr>
<tr><td colspan="7">Power-on Default: DA7</td></tr>
<tr><td rowspan="4">81</td><td rowspan="4">GPIAD0</td><td rowspan="4">I</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td></td><td>v</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Alternative Function: AD0</td></tr>
<tr><td colspan="7">Power-on Default: AD0</td></tr>
<tr><td rowspan="2">82</td><td rowspan="2">GPIAD1</td><td rowspan="2">I</td><td>Driving</td><td>Input</td><td>Output</td><td>5V Tol.</td><td>Pullup</td><td>OpenDrain</td><td>Sch. Trg</td></tr>
<tr><td></td><td>v</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

			Alternative Function: AD1						
			Power-on Default: AD1						
83	GPIAD2	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				v					
			Alternative Function: AD2						
			Power-on Default: AD2						
84	GPIAD3	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				v					
			Alternative Function: AD3						
			Power-on Default: AD3						
87	GPIAD4	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				v					
			Alternative Function: AD4						
			Power-on Default: AD4						
88	GPIAD5	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				v					
			Alternative Function: AD5						
			Power-on Default: AD5						
89	GPIAD6	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				v					
			Alternative Function: AD6						
			Power-on Default: AD6						
90	GPIAD7	I	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
				v					
			Alternative Function: AD7						
			Power-on Default: AD7						
32	GPOW0	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		v	v	v	v	
			Alternative Function: PWM0						
			Power-on Default: PWM0						
33	GPOW1	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		v	v	v	v	
			Alternative Function: PWM1						
			Power-on Default: PWM1						
36	GPOW2	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		v	v	v	v	
			Alternative Function: PWM2, FAN1PWM						
			Power-on Default: PWM2						
37	GPOW3	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		v	v	v	v	
			Alternative Function: PWM3						
			Power-on Default: PWM3						
38	GPOW4	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		v	v	v	v	
			Alternative Function: PWM4						
			Power-on Default: PWM4						
39	GPOW5	O	Driving	Input	Output	5V Tol.	Pullup	OpenDrain	Sch. Trg
			4mA		v	v	v	v	

			Alternative Function: PWM5						
			Power-on Default: PWM5						
40	**GPOW6**	O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA		v	v	v	v	
			Alternative Function: PWM6						
			Power-on Default: PWM6						
43	**GPOW7**	O	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
			4mA		v	v	v	v	
			Alternative Function: PWM7, FAN2PWM						
			Power-on Default: PWM7						
2	**GPWU0**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: none						
			Power-on Default: GPWU0						
26	**GPWU1**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: none						
			Power-on Default: GPWU1						
29	**GPWU2**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: none						
			Power-on Default: GPWU2						
30	**GPWU3**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: none						
			Power-on Default: GPWU3						
44	**GPWU4**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: none						
			Power-on Default: GPWU4						
76	**GPWU5**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: none						
			Power-on Default: GPWU5						
172	**GPWU6**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: TIN1						
			Power-on Default: GPWU6						
176	**GPWU7**	I	**Driving**	**Input**	**Output**	**5V Tol.**	**Pullup**	**OpenDrain**	**Sch. Trg**
				v		v	v		v
			Alternative Function: TIN2, FANFB2						
			Power-on Default: GPWU7						

3.3.10 General Purpose Timer Pin Descriptions

PN176	Pin name	Direction	Description
171	**TOUT1**	O	General Purpose Timer_1 Output. This pin toggles when the timer counts up to the pre-load value, and the resulting pulse is 50% duty cycle.
175	**TOUT2**	O	General Purpose Timer_2 Output..
172	**TIN1**	I	General Purpose Timer_1 Clock Input. Input to this pin can be used as

			the clock for the timer.
176	**TIN2**	I	General Purpose Timer_2 Clock Input. Input to this pin can be used as the clock for the timer.

3.3.11 Analog To Digital Pin Descriptions

PN176	Pin name	Direction	Description
81	**AD0**	I	Analog to Digital Conversion Input 0. The internal analog to digital conversion circuit with 8-bit resolution performs analog to digital conversion from ***AD0~AD7*** inputs in a round robin way. It polls ***AD0~AD7*** periodically with 4ms cycle time. ***AD0~7*** pins can alternatively configured to be general purpose inputs.
82	**AD1**	I	Analog to Digital Conversion Input 1.
83	**AD2**	I	Analog to Digital Conversion Input 2.
84	**AD3**	I	Analog to Digital Conversion Input 3.
87	**AD4**	I	Analog to Digital Conversion Input 4.
88	**AD5**	I	Analog to Digital Conversion Input 5.
89	**AD6**	I	Analog to Digital Conversion Input 6.
90	**AD7**	I	Analog to Digital Conversion Input 7.

3.3.12 Digital To Analog Pin Descriptions

PN176	Pin name	Direction	Description
99	**DA0**	O	Digital to Analog Conversion Output 0 with 8-bit resolution. DA0~7 pins can alternatively configured to be general purpose outputs.
100	**DA1**	O	Digital to Analog Conversion Output 1 with 8-bit resolution.
101	**DA2**	O	Digital to Analog Conversion Output 2 with 8-bit resolution.
102	**DA3**	O	Digital to Analog Conversion Output 3 with 8-bit resolution.
1	**DA4**	O	Digital to Analog Conversion Output 4 with 8-bit resolution.
42	**DA5**	O	Digital to Analog Conversion Output 5 with 8-bit resolution.
47	**DA6**	O	Digital to Analog Conversion Output 6 with 8-bit resolution.
174	**DA7**	O	Digital to Analog Conversion Output 7 with 8-bit resolution.

3.3.13 Expanded I/O Pin Descriptions

PN176	Pin name	Direction	Description
85	**XIO8CS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_8 of one of registers XIO_DA [F:0], depending on which of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.
86	**XIO9CS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_9 of one of registers XIO_DA[F:0], depending on whic h of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.
91	**XIOACS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_10 of one of registers XIO_DA[F:0], depending on which of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.
92	**XIOBCS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_11 of one of registers XIO_DA[F:0], depending on which of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.
93	**XIOCCS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_12 of one of registers XIO_DA[F:0], depending on which of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.
94	**XIODCS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_13 of one of registers XIO_DA[F:0], depending on which of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.

97	**XIOECS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_14 of one of registers XIO_DA[F:0], depending on which of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.
98	**XIOFCS#**	O	Expanded I/O Chip Select Pin. This pin reflects the state of bit_15 of one of registers XIO_DA[F:0], depending on which of the Expanded I/O ports XIO[F:0] is being addressed. XIO0 port corresponds to XIO_DA0 and so on.

3.3.14 8051 External Interface Pin Description

PN176	Pin name	Direction	Description
105	**E51CS#**	O	The ***E51CS#*** can be used to select secondary memory spaces for 8051 development environment support. Register BI_MS2 (FEA7h) controls the mapping of the four 16-K segments SEG3-0 memory spaces individually into secondary memories. When a particular memory space is being configured to map into secondary memory, ***E51CS#*** will be asserted instead of ***MEMCS#***. The assertion of ***E51CS#*** and ***MEMCS#*** is mutual exclusive.
107	**E51TXD**	O	The Transmit signal of the embedded serial controller of 8051
106	**E51RXD**	I	The Receive signal of the embedded serial controller of 8051
	E51IE0	I	The embedded 8051 Interrupt input port IE0
	E51IE1	I	The embedded 8051 Interrupt input port IE1
3	**E51IT0**	I	The embedded 8051 Timer_0 output
4	**E51IT1**	I	The embedded 8051 timer_1 output

3.3.15 Clock Pin Descriptions

PN176	Pin name	Direction	Description
158	**XCLKI**	I	32.768KHz Input.
160	**XCLKO**	O	32.768KHz Output

3.3.16 Miscellaneous Pin Descriptions

PN176	Pin name	Direction	Description
31	**ECSCI#**	O	The ***ECSCI#*** signal is output to system chipset to indicate an outstanding SCI event
5	**GA20**	O	The ***GA20*** signal is output to system chipset.
11	**TEST_TP**	I	Test Mode. Do Not Use. This pin MUST be pulled down externally for normal application. Refer to chapter 3 Hardware Strap for details.
125	**XIOP_TP**	I	This pin should be either pulled down or pulled up externally for normal application. Refer to chapter 3 Hardware Strap for details.
128	**DMRP_TP**	I	Test Mode. Do Not Use. This pin should be pulled up externally for normal application. Refer to chapter 3 Hardware Strap for details
131	**EMWB_TP**	I	Test Mode. Do Not Use. This pin should be pulled up externally for normal application. Refer to chapter 3 Hardware Strap for details.
6	**KBRST#**	O	The ***KBRST#*** is output to system chipset to generate a system reset.
19	**ECRST#**	I	ECRST# is used as the global reset signal for resetting all the chip's internal modules. The input usually comes from a power-on reset circuit.
41	**ScrollLock#**	O	Used to drive the LED indicating ***ScrollLock#***
23	**NumLock#**	O	Used to drive the LED indicating ***NumLock#***
54	**CapLock#**	O	Used to drive the LED indicating ***CaplLock#***
55	**FnLock#**	O	Used to drive the LED indicating ***FnLock#***

3.3.17 Power Pin Descriptions

PN176	Pin name	Direction	Description

16,34,45, 123,136, 157,161, 166	**VCC**	-	Power Supply to all internal modules except Analog portions of **ADC** and **DAC**
17,35,46, 122,137, 159,167	**GND**	-	Digital Ground
95	**VCCA**	-	Analog power to analog portions of **ADC** and **DAC**
96	**AGND**	-	Analog Ground (paired with ***VCCA***)

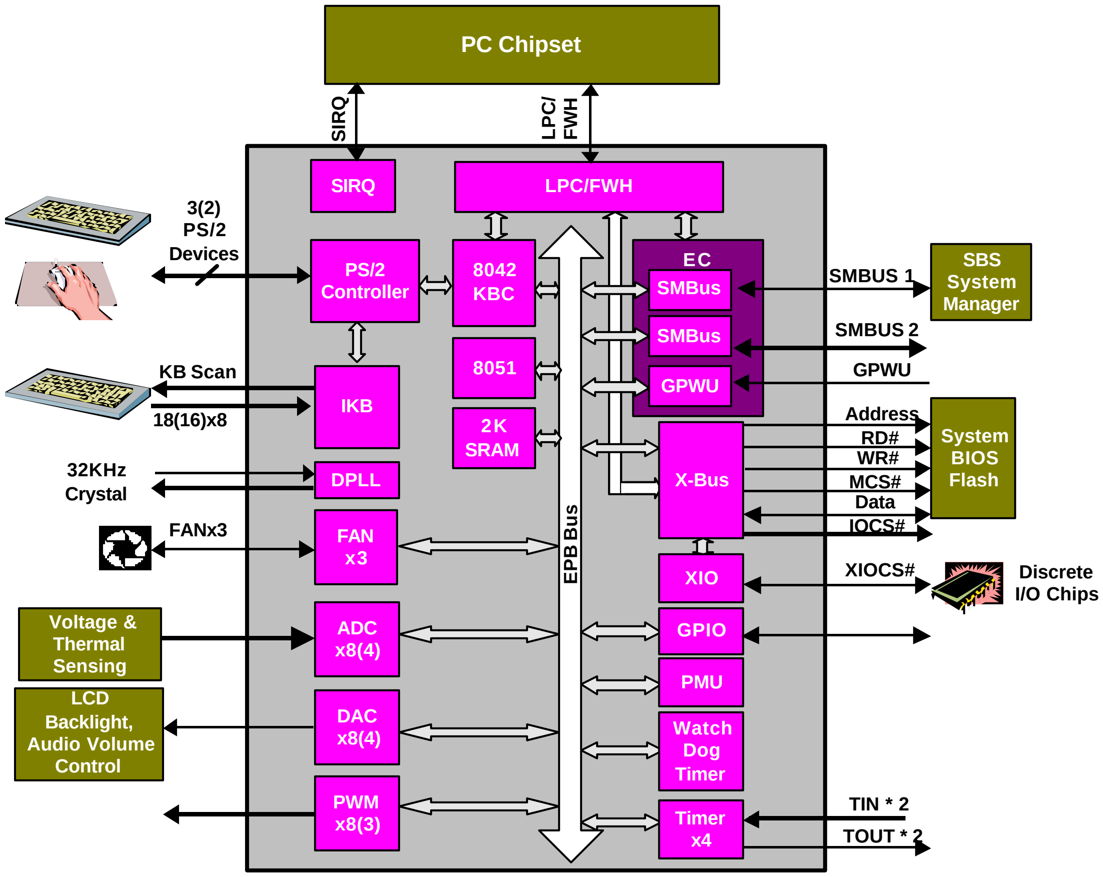
3.4 Hardware Strap

Hardware strap pins will latch the external signal levels at the rising edge of ***ECRST#***. Either a High or Low value will be stored internally to serve as control signals as described below.

PIN176	Pin name	HW Strap Value
125	**A1**	**XIOP_TP**: XIO priority selection. This signal is used to enable or disable the internal pull-up resistors on ***XIOCS [F:0]*** signals. The purpose is to provide a stable signal level on these pins when system is power-on, so that pre-determined signal levels either high or low can be expected on these pins. For example, when one of ***XIOCS [F:0]*** is connected to an external device with active-high chip select input pin, then it is recommended to disable the internal pull-up resistor and add pull-down resistor on board so that the external device will not accidentally be written upon powering up. Likewise, if one of ***XIOCS [F:0]*** is connected to external devices with active-low chip select input pins, then it is recommended to enable the internal pull-up resistors High: Enable the internal pull-up resistors on ***XIOCS [F:0]*** pins Low: Disable the internal pull-up resistors on ***XIOCS [F:0]*** pins
128	**A4**	**DMRP_TP**: Disable Memory Remapping Process The **DMRP_TP** is an internal signal used to disable the remapping of LPC memory addresses onto different memory spaces on X-Bus. The remapping process involves replacing address bits A [20:17] on LPC bus to values stored in **LF_RMA** register (FE96h). If not disabled, LPC address will be forwarded onto X-bus with default mapping scheme. The DMRP_TP signal stores the latched value and reflects this value (being inverted) on bit_5 of **LF_CSM** register (FE95h) High: Disable DMRP **(Recommended)** Low: Enable DMRP
131	**A5**	**EMWB_TP**: Enable Memory while Boot The EMWB_TP is an internal signal used to enable system host accessing BIOS memory located on KB3910's X-Bus. The EMWB_TP signal stores the latched value and reflects this value on bit [3:0] of **LF_EN** register (FE90h), and on bit_4 of **LF_FWHID** register (FE94h) High: Enable EMWB **(Recommended for application using shared BIOS)** Low: Disable EMWB
11	**GPIO05**	**TEST_TP**: Clock Test Mode High: Test Mode. Low: 32KHz clock in normal running (Recommend)
12	**GPIO06**	**DPLL_TP**: DPLL Test Mode High: Test Mode (***KSOUT0~15*** become DPLL internal data outputs, ***KSO16*** becomes internal power-on reset output) Low: Normal operation **(Recommended)**

4. Module Descriptions

4.1 System Architecture



Conventional keyboard controllers for notebook are typically based on 8/16-bit MCUs and multiple general purpose I/Os configurable by firmware for various applications like keyboard scan Input/Output, PS/2 Controller, SMBus controller…etc. While this kind of " MCU-based with GPIOs" chip solution provides greater flexibility for notebook designers to implement custom features in their systems, the underlying firmware effort often becomes much more heavier. As the complexity of today' s notebook design increases, so does the firmware development effort, code size, power consumption and time to market.

A detailed analysis of the firmware code for a typical notebook application reveals that the majority of code are used to handle repeated actions and routine functions which can otherwise be taken over or accelerated by hardware logic. Functions/Modules fall within this category which are suited to be implemented by hardware logic includes: 8042 keyboard controller, PS/2 controller, keyboard scancode encoder, SMbus controller, fan tachometer and fan speed controller. The accompanying advantages of hardware implementation over firmware are:

1. Shorten the development cycle of a notebook model.
2. Less CPU MHz required for firmware processing; i.e., less power consumption
3. Less firmware code size; i.e., less flash memory required.
4. Less firmware code maintenance effort
5. Easy migration to other models

With these design philosophy in mind, KB3910 is engineered to fit the requirement of today' s notebook design with a perfect balance between the efficiency brought by hardware implementations and the flexibility brought by firmware implementations. KB3910 is an 8051-based MCU with abundant peripheral controllers. It includes the following hardware implemented peripheral controllers:

- 8042 keyboard controller; handles standard keyboard commands in hard wires.
- PS/2 Controller; supports 3 external PS/2 devices in hard wires.
- Internal keyboard encoder; handles keyboard scan-in/out, scancode generation
- SMbus host controller; full SMbus feature implementation
- X-bus interface; used to interface to ISA-like flash memory and I/O peripherals
- Fan tachometer and speed controller
- EC standard commands defined in ACPI2.0

In addition, to allow even greater flexibility, every hardware implemented functions can be extended under the control of firmware. Keyboard scancode are downloadable and keyboard hot-keys are supported by the embedded 8051. These features make fully customized keyboard features possible.

The embedded processor (EC) is a 8051-based MCU. It is instruction sets compatible with the standard 8051 microprocessor while its internal architecture has been modified to be Von Newman rather than Harvard. The 8051 code memory can be mapped to the flash memory location. The code size required for KB3910 is application dependent.

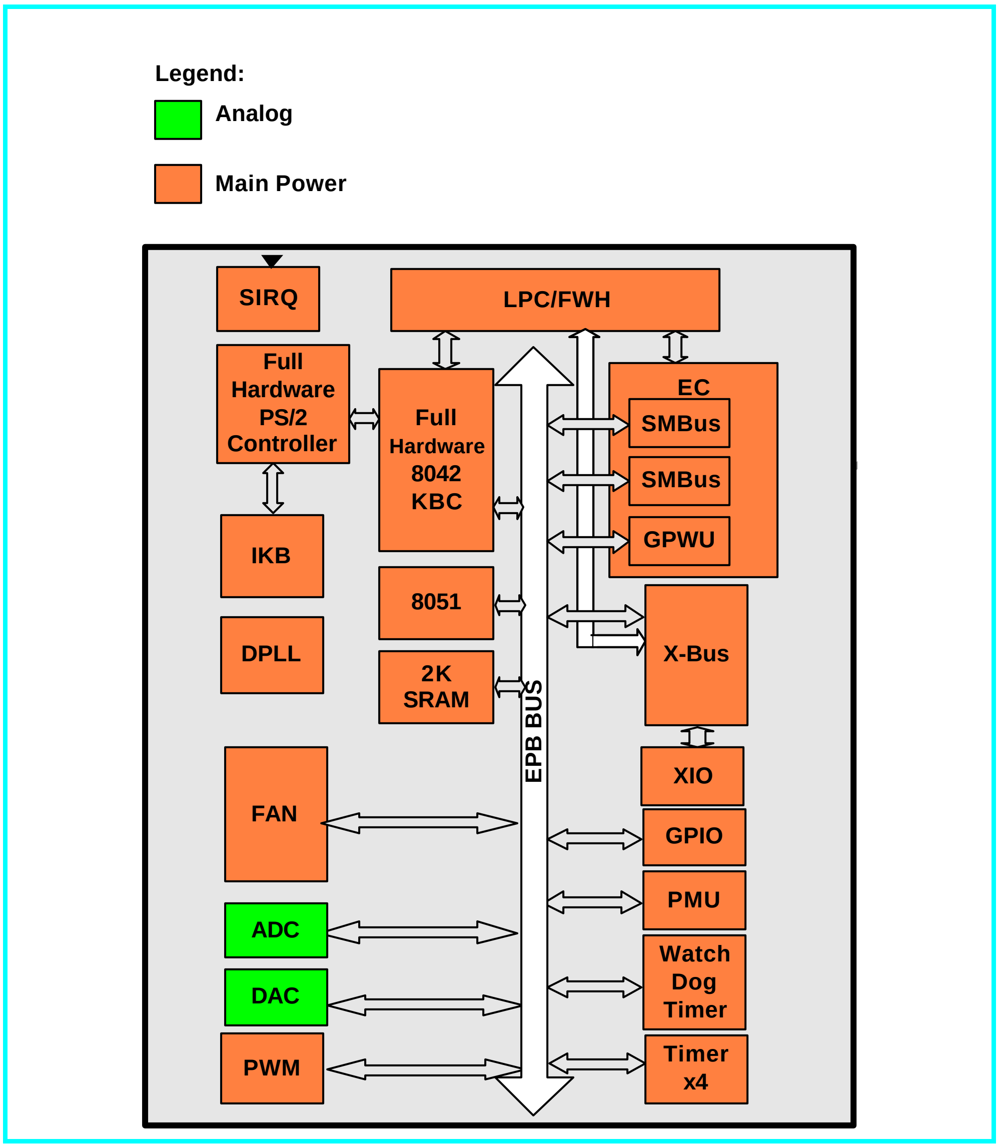
KB3910 requires only a 32.768KHz external crystal, from where it can generate 4/8/16/32MHz clock supplying most internal logic cells. Since most of the routine functions are carried out by hardware logic, the CPU/MHz required to process the remaining firmware implemented features can be reduced to minimum. Compared with other 16-bit MCU based keyboard controllers who may require up to 16M/20MHz clock to drive the whole chip, KB3910 can effectively reduce the power consumption.

Other peripherals of KB3910 include 4~8 pulse-width modulation (PWM) outputs, 5~8 Analog-to-digital (ADC) pins, 5~8 digital-to-analog (ADC) pins, 8 general purpose wake-up (GPWU) inputs, 48 GPIOs, two general purpose timer modules (GPT) and watchdog timer (WDT). KB3910 itself support two levels of power saving mode—sleep mode and deep sleep mode, in order to achieve maximum power saving.

The development tool of KB3910 is also 8051 compatible. ENE in-house uses the Keil-C tool for firmware development. Most of the codes are written in C language so that they are easily transferable to other projects. There are other in-house developed tools to streamline the debugging and code-download are also available upon request.

4.2 Chip Architecture

4.2.1 Power Planes



There are 3 internal power planes.

No.	Power Plane	Applied Modules	Power	GND
1	Main Logic	All internal logic modules except analog modules (ADC / DAC).	VCC	GND
2	Analog	ADC / DAC analog portions	VCCA	AGND

4.2.2. Clock Domains

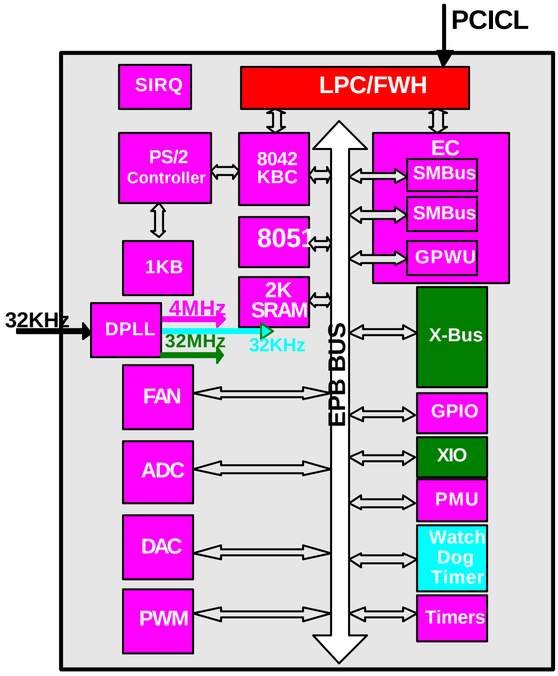
Legend:

🟩 **32MHz Clock Domain**

🟦 **32KHz clock Domain/ Internal DPLL 32KHz clock Domain**

🟪 **4M/8M/16M/32MHz Main Clock Domain**

🟥 **33MHz PCI Clock Domain**



There are 4 clock domains.

No.	Clock Domain	Applied Modules	Clock Source
1	32KHz	**WDT** modules	**XCLKI / XCLKO &** Internal **DPLL**
2	4/8/16MHz	All other modules. Programmable.	Internal **DPLL**
3	32MHz	XBI / XIO modules	32KHz multiplied 1000
4	33MHz	PCI Interface Modules	**PCICLK**

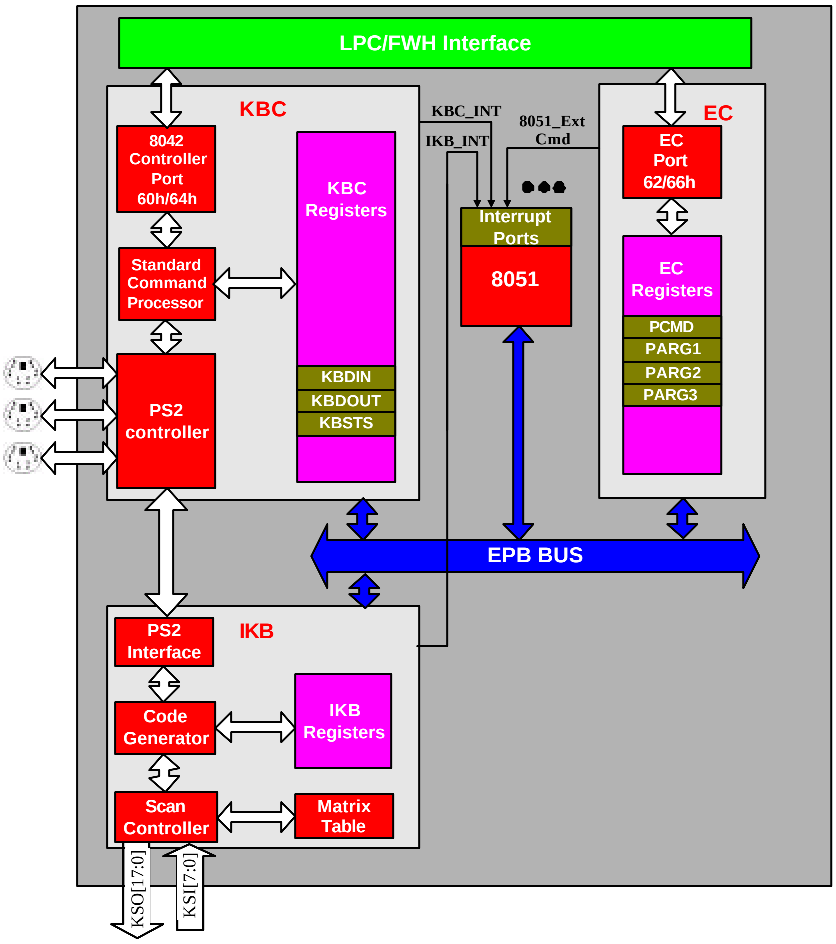
There are two 32KHz clock sources for WDT, one is from external crystal (XCLI/XCLO), another is from internal DPLL circuit, the power-on default clock source of WDT is from internal DPLL circuit. (In KB3910 Bx chip, there is only one 32KHz clock source from external crystal.) Please refer bit_2 of **WDTCTR (0xFE85h)**.

4.2.3 Reset Domains

KB3910 contains 4 reset domains.

No.	Reset Domain	Applied Modules	Reset Source
1	**LPC** Reset	**LPC** control and register logic.	1. ***LRST#*** pin input, or 2. ***ECRST#*** pin input
2	**EC** Reset	All modules (except LPC) logic reset including GPIO setting	1. ***ECRST#*** pin input, or 2. Internal WDT timeout (can be disable)
3	8051 Reset	8051 core logic includes 2 8051 16-bit timers, interrupt controller and serial port logic.	1. ***ECRST#*** pin input, or 2. Internal **WDT** timeout, or 3. Async. wakeup event in deep sleep mode, or 4. EC register 14h (**ECPXCFG**) bit 0 = 1

4.2.4 KBC/IKB/EC Hardware and 8051 Internal Communication

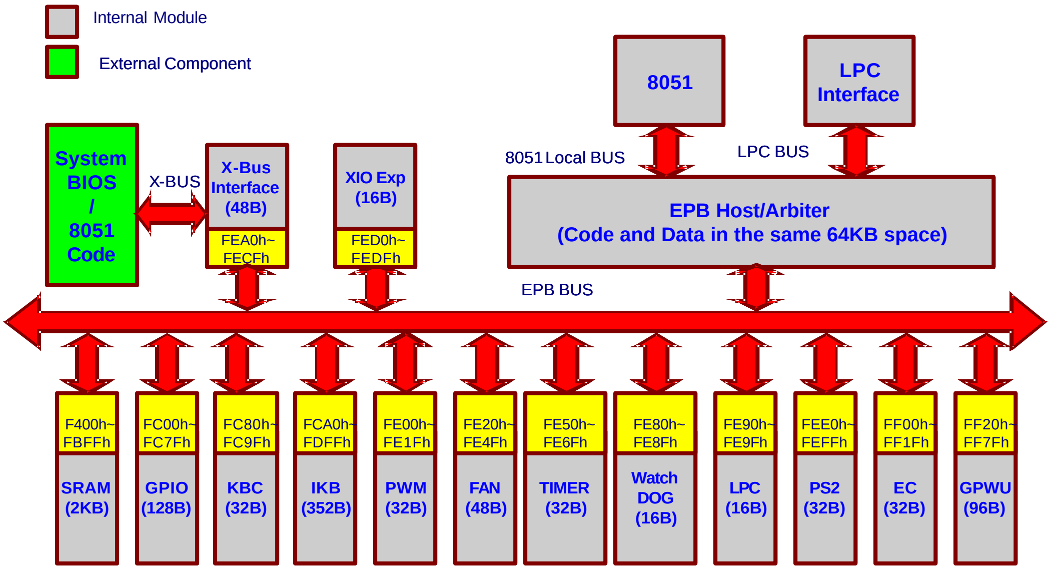


Standard KBC or EC commands will be automatically processed by hardware with result stored into **KBCOUT** and status updated in **KBCSTS**. Extended KBC commands will be forwarded to 8051 (by generating interrupt) for processing via registers **KBCIN**, **KBCOUT** and **KBCSTS**.

For extended EC commands, the system host and the 8051 can communicate via four registers, **ECPCMD, ECPARG1, ECPARG2** and **ECPARG3** with a pre-defined protocol including commands and arguments. Refer to the EC module section for an example of some of the pre-defined extended commands. The Extended EC commands are most useful for system host to access peripheral devices or KB3910 resources while eliminating the potential possibility that they are being accessed by the 8051 firmware simultaneously.

Hot-Keys are supported by firmware. When there is a hot-key being pressed (matrix values within B4h~FFh), 8051 will receive an interrupt so that firmware can process this event.

4.2.5 Internal Memory Map



On-chip peripheral devices are accessible either by the 8051 or by the Host. Access by 8051 is done by the **MOVX, MOVC** instructions (since these peripheral devices are located within the 8051's 64KB addressable range). Access by Host is accomplished in an indirect manner through extended EC commands: F0h and F1h.The EC Space (FF00h~FFFFh) is a flat byte-addressable I/O space as defined in ACPI 2.0. System Host can access this space via EC commands or Index IO. The on-chip peripheral address map is listed below.

No.	ABBR	Device Full Name	Address Range	Size (Byte)	
1	**Flash**	Program space mapped to system BIOS	0000h~F3FFh	61K	
2	**XRAM**	Embedded SRAM	F400h~FBFFh	2K	
3	**GPIO**	General Purpose IO (include ADC, DAC)	FC00h~FC7Fh	128	1K
4	**KBC**	Keyboard Controller	FC80h~FC9Fh	32	
5	**IKB**	Internal KB	FCA0h~FDFFh	352	
6	**PWM**	Pulse Width Modulation	FE00h~FE1Fh	32	
7	**FAN**	FAN Controller	FE20h~FE4Fh	48	
8	**GPT**	General Purpose 16-bit timer	FE50h~FE6Fh	32	
		Reserved.	FE70h~FE7Fh	16	
9	**WDT**	Watchdog Timer	FE80h~FE8Fh	16	
10	**LPC**	Low Pin Count	FE90h~FE9Fh	16	
11	**XBI**	X-BUS Interface	FEA0h~FECFh	48	
12	**XIO**	IO Expender	FED0h~FEDFh	16	
13	**PS2**	PS2	FEE0h~FEFFh	32	
14	**EC**	Embedded Controller **(in EC Space)**	FF00h~FF1Fh	32	
15	**GPWU**	General Purpose Wake-up **(in EC Space)**	FF20h~FE7Fh	96	
16	**SMBus**	System Management BUS **(in EC Space)**	FF80h~FFFFh	128	

Note: The original 8051 is Harvard architecture (code and data are in separated memory spaces), while the embedded 8051 in KB3910 is Von Neumann architecture, namely, it's code and data are in the same memory space. The code memory space of the embedded 8051 can be remapped to 4 different 16KB segments within a maximum addressable 2M-byte memory. Details of the remapping of 8051 memory space are described in the XBI section. The Keil C development tool also supports "banking codes" which allows code size larger than 64KB.

4.4 XRAM Module

KB3910 contains 2K bytes SRAM. The 8051 **MOVX** and **MOVC** instructions with address range F400h~FBFFh will hit the XRAM space. The XRAM will not respond to instruction-fetch cyles, thereby it cannot be used as the code mmory for 8051.

The SRAM in XRAM module will not be reset by the ***ECRST#,*** therefore, firmware must initialize the SRAM space before take advantage of this memeory space.

4.5 GPIO Module

4.5.1 Functional Description

Output Mode

When a GPIO pin is configured in output mode, there are three register bits associated with that pin to control its output behavior. They are

1. Output enable bit (with name appeared as **GPxxOExx**)
2. Output data bit (with appeared name as **GPxxD**)
3. Open drain output control bit (with name appeared as **GPxxOD**).

All GPIOs are capable of driving 4mA in output mode. Depending on the I/O buffer type, some GPIO pins are capable of driving 16mA for applications like driving a LED.

Input Mode

To configure a GPIO port to input mode, the associated Output Enable bit should be cleared to 0. The input pin status can be read through the corresponding GPIO Pin Input register. The Pull-up Enable registers are used to control the internal pull-up for input pins. The internal pull-up scheme for all I/O buffers alike is pull-up to 3.3V via an 80K ohm resistor. Care should be taken if these pins are applied with external pull-up to 5V. Under such condition there is a small amount leakage current flowing through the 5V into the internal 3.3V power well and consumes power. If a GPIO port is configured to generate a GPWU event, it should be set to input mode.

Some GPIOs can be configured as alternative (Alt.) functions. Refer to the table below.

GPIO name	Alt. Output Mode	Alt. Input Mode	Default	Alt. Select Reg.
GPIO02	GA20		GA20	GPIOFS00.2 = 1
GPIO03	KBRST#		KBRST#	GPIOFS00.3 = 1
GPIO05	FAN3PWM	TEST_TP	GPIO05	GPIOFS00.5 = 1
GPIO06		FANFB3, DPLL_TP	GPIO06	GPIOFS00.5 = 1
GPIO0A	NumLock#		GPIO0A	GPIOFS08.2=1
GPIO0C	CLKRUN#		GPIO0C	GPIOFS08.4 = 1
GPIO0F	ScrollLock#		GPIO0F	GPIOFS08.7=1
GPIO11	CapLock#		GPIO11	GPIOFS10.1=1
GPIO12	FnLock#		GPIO12	GPIOFS10.2=1
GPIO18~1F	XIO8CS#~XIOFCS#		GPIO18~1F	GPIOFS18 = FFh
GPIO20	E51CS#		GPIO20	GPIOFS20.0 = 1
GPIO21		E51RXD	GPIO21	GPIOFS20.1 = 1
GPIO22	E51TXD		GPIO22	GPIOFS20.2 = 1
GPIO23	A20 (X-BUS)		A20	GPIOFS20.3 = 1
GPIO2C		LRST#	GPIO2C	GPOWFS.7=1
GPIO2E	TOUT1	FANFB1	GPIO2E	GPIOFS28.6 = 1
GPIO2F	TOUT2		GPIO2F	GPIOFS28.7 = 1
PWM0~7	GPOW0~7		GPOW0~7	GPOWFS = FFh
PWM2	FAN1PWM		PWM2	GPIOKS16.4=1
PWM7	FAN2PWM		PWM7	GPIOKS16.5=1

KSO0~17	GPOK0~17		KSO0~17	GPOKFS00 = FFh
				GPOKFS08 = FFh
				GPOKFS16 = 03h

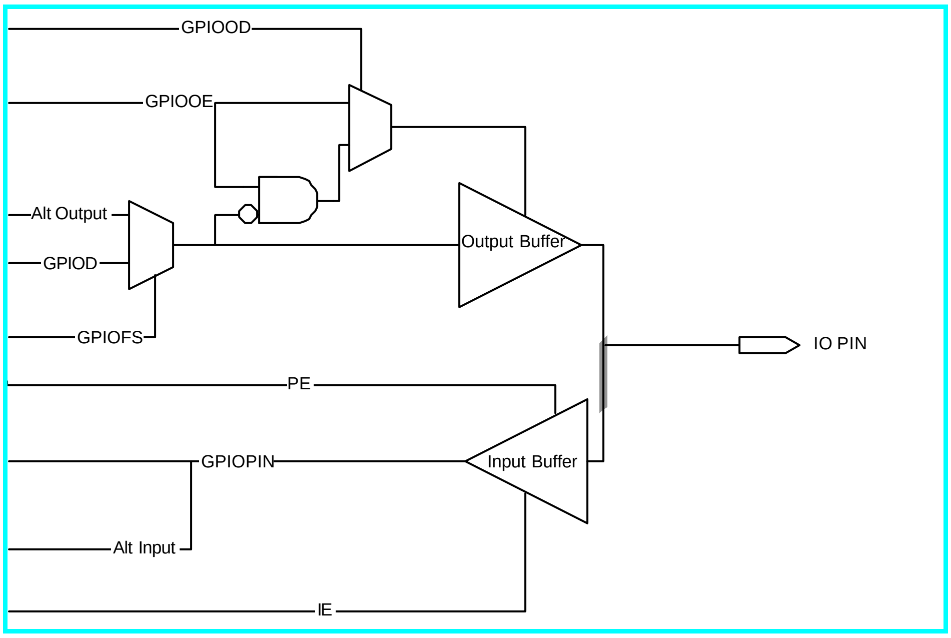
ADC Interface

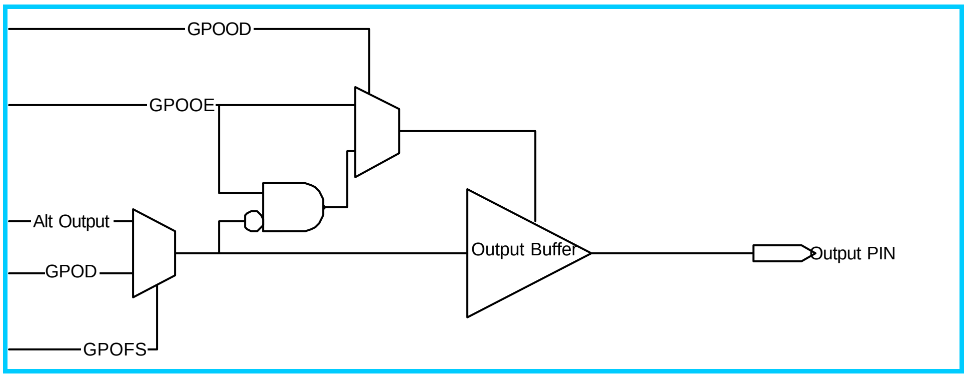
KB3910 includes 5~8 ADC channels. Before an ADC channel is active, the associated **ADCEN** should be set. If **ADCCFG** bit 7 is enabled, the ADC update interrupt will be generated periodically as defined in **ADCCFG** bits[5:0]. The latest converted digital values can be read in offset 70h~77h. No internal pull-up resistor associated with ADC digital inputs.

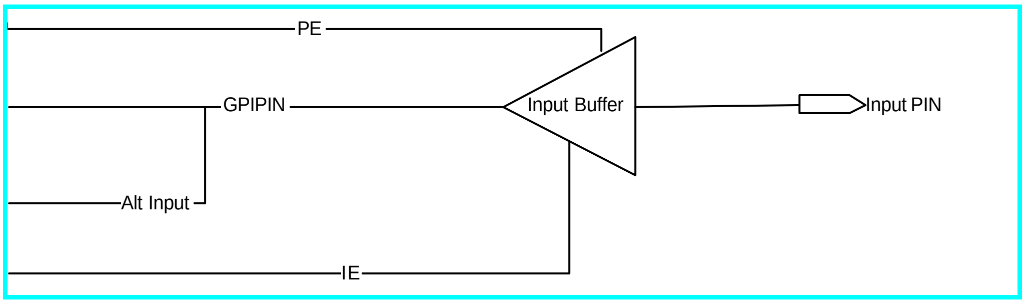
DAC Interface

KB3910 includes 5~8 DAC channels. Before a DAC channel is active, the corresponding **DACEN** bit should be set to 1. The output DAC values are stored in offset 78h~7Fh.

GPIO and IO Buffer Structure







4.5.2. GPIO Register Descriptions (Base Address = FC00h, Space 128 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
00h	GPIOFS00	GPIO00~07 Function Selection			0Ch	FCh
		7	R/W	0: ***GPIO07***. 1: intr_vector[5] (interrupt controller debugging output)		
		6	R/W	0: ***GPIO06/*** **FANFB3/ DPLL_TP** 1: intr_vector[4] (interrupt controller debugging output)		
		5	R/W	0: ***GPIO05***. 1: ***FAN3PWM***		
		4	R/W	0: ***GPIO04***. 1: intr_next_coming (interrupt controller debugging output)		
		3	R/W	0: ***GPIO03*** 1: ***KBRST#*** (default)		
		2	R/W	0: ***GPIO02*** 1: ***GA20*** (default)		
		1	R/W	0: ***GPIO01***. (***E51IT1*** is GPIO01 input) 1: intr_flag (interrupt controller debugging output)		
		0	R/W	0: ***GPIO00***. (***E51IT0*** is GPIO00 input) 1: intr_req (interrupt controller debugging output)		
01h	GPIOFS08	GPIO08~0F Function Selection			00h	FCh
		7	R/W	0: ***GPIO0F*** 1: ***ScrollLock#***		
		6	R/W	0: ***GPIO0E***		
		5	R/W	0: ***GPIO0D*** 1: intr_vector[3] (interrupt controller debugging output)		
		4	R/W	0: ***GPIO0C*** 1: ***CLKRUN#*** (default)		
		3	R/W	0: ***GPIO0B*** 1: kbc_host_obf (KBC debugging output)		
		2	R/W	0: ***GPIO0A*** 1: ***NumLock#***		
		1	R/W	0: ***GPIO09***. 1: intr_vector[7] (interrupt controller debugging output)		
		0	R/W	0: ***GPIO08***. 1: intr_vector[6] (interrupt controller debugging output)		
02h	GPIOFS10	GPIO10~17 Function Selection			00h	FCh
		7	R/W	0: ***GPIO17*** 1: kbc_ps2_rdy (KBC debugging output)		
		6	R/W	0: ***GPIO16*** 1: kbc_cmd_ps2Req (KBC debugging output)		
		5	R/W	0: ***GPIO15*** 1: ikb_ps2_clk (KBC debugging output)		
		4	R/W	0: ***GPIO14*** 1: ikb_ps2_data (KBC debugging output)		
		3	R/W	0: ***GPIO13*** 1: kbc_host_hwReq (KBC debugging output)		
		2	R/W	0: ***GPIO12*** 1: ***FN LOCK*** LED		
		1	R/W	0: GPIO11. 1: ***CAP LOCK*** LED		

		0	R/W	0: ***GPIO10***. 1: kbc_host_ibf (KBC debugging output)		
03h	GPIOFS18	**GPIO18~1F Function Selection**			00h	FCh
		7~0	R/W	Bit_0 corresponds to ***GPIO18*** and so on 0: ***GPIO18~1F*** (default is trapped by ***A1***, ***XIOP_TP***) 1: ***XIOCS8~F***		
04h	GPIOFS20	**GPIO20~27 Function Selection**			0Ch	FCh
		7	R/W	0: ***GPIO27*** 1: ec_ibf (EC debugging output)		
		6	R/W	0: ***GPIO26*** 1: ec_obf (EC debugging output)		
		5	R/W	0: ***GPIO25*** 1: ***IRQ12*** (EC debugging output)		
		4	R/W	0: ***GPIO24*** 1: ***IRQ1*** (EC debugging output)		
		3	R/W	0: ***GPIO23*** 1: ***A20*** (default)		
		2	R/W	0: ***GPIO22*** 1: ***E51TXD***		
		1	R/W	0: ***GPIO21*** (can also be ***E51RXD*** input) When used as E51RXD, be sure the associated input enable bit is enabled. 1: kbc_core_intr (KBC debugging output)		
		0	R/W	0: ***GPIO20*** 1: ***E51CS#***		
05h	GPIOFS28	**GPIO28~2F Function Selection**			00h	FCh
		7	R/W	0: ***GPIO2F*** 1: GPT ***TOUT2*** output		
		6	R/W	0: ***GPIO2E*** (***FANFB1*** input) 1: GPT ***TOUT1*** output		
		5	R/W	0: ***GPIO2D*** 1: lpc_kbcWrZ (EC debugging output)		
		4	R/W	0: ***GPIO2C/LRST#*** 1: lpc_kbcRdZ (EC debugging output)		
		3	R/W	0: ***GPIO2B*** 1: lpc_extec_csZ (EC debugging output)		
		2	R/W	0: ***GPIO2A*** 1: lpc_kbcCsZ (EC debugging output)		
		1	R/W	0: ***GPIO29*** 1: lpc_ecCsZ (EC debugging output)		
		0	R/W	0: ***GPIO28*** 1: clk_4mhz (EC debugging output)		
06h	GPOWFS	**GPOW Function Selection**			00h	FCh
		7~0	R/W	0: ***PWM0~7*** 1: ***GPOW0~7***.		
07h	GPODAFS	**GPODA Function Selection**			00h	FCh
		7~0	R/W	0: ***DA0~7*** 1: ***GPODA0~7***.		
08h	GPOKFS00	**GPOK0~7 Function Selection**			00h	FCh
		7~0	R/W	0: ***KSOUT0~7*** 1: ***GPOK0~7***.		
09h	GPOKFS08	**GPOK8~15 Function Selection**			00h	FCh
		7~0	R/W	0: ***KSOUT8~15*** 1: ***GPOK8~15***.		
0A~0Dh			NA	Reserved	00h	FCh

0Eh	**GPOKFS16**	**GPOK16~17 Function Selection**			00h	FCh
		6	R/W	0: ***GPIO2D*** is normal mode. 1: ***GPIO2D*** is GPWU0 echo output.		
		5	R/W	0: not output ***FAN2PWM*** 1: PWM7 is ***FAN2PWM***		
		4	R/W	0: not output ***FAN1PWM*** 1: PWM2 is ***FAN1PWM***		
		3	R/W	0: ***GPODA0~7*** digital are normal outputs 1: ***GPODA0*** digital are debug outputs ***GPODA0***: clk4_idle_cs ***GPODA1***: clk32_idle_cs ***GPODA2***: clk_32mhz		
		2	R/W	0: ***KSOUT0~17*** are normal outputs 1: ***KSOUT0~17*** are debug outputs ***KSOUT0*** = xepbm_psel_flash ***KSOUT1*** = xepbm_psel_xio ***KSOUT2*** = xLpc2XbiReq ***KSOUT3*** = xArbGrant2Lpc ***KSOUT4*** = xArb2CoreReq ***KSOUT5*** = flash_srdy ***KSOUT6*** = xArb2LpcRdy ***KSOUT7*** = xArb2PSRdy ***KSOUT8*** = xbi_inst_cs[1] ***KSOUT9*** = xbi_inst_cs[2] ***KSOUT10*** = xbi_inst_cs[3] ***KSOUT11***= xbi_inst_cs[4] ***KSOUT12*** = xbi_inst_cs[5] ***KSOUT13*** = xbi_inst_cs[6] ***KSOU14*** = xbi_inst_ok ***KSOUT15*** = epbm_ec_cs ***KSOUT16*** = epbm_ep_cs ***KSOUT17*** = e51_cpu_inst_fetch_cs		
		1~0	R/W	0: ***KSOUT16~17*** 1: ***GPOK16~17***.		
0Fh				Reserved		
10h 11h 12h 13h 14h 15h	**GPIOOE00** **GPIOOE08** **GPIOOE10** **GPIOOE18** **GPIOOE20** **GPIOOE28**	**GPIO Output Enable Register**			0Ch 00h 00h 00h 0Ch 00h	FCh
		7 – 0	R/W	Output Enable for ***GPIO00~2F***.		
16h	**GPOWOE**	**GPOW0~7 Output Enable Register**			00h	FCh
		7 – 0	R/W	Output Enable for ***GPOW0~7***, if function GPOW is selected.		
17h	**GPODAOE**	**GPODA0~7 Output Enable Register**			00h	FCh
		7 – 0	R/W	Output Enable for ***GPODA0~7***, if function GPODA is selected.		
18h 19h	**GPOKOE00** **GPOKOE08**	**GPOK0~17 Output Enable Register**			00h	FCh
		7 – 0	R/W	Output Enable for ***GPOK0~7***, if GPOK is selected. Output Enable for ***GPOK8~15***, if GPOK is selected.		
1A~1Dh				Reserved		
1Eh	**GPOKOE16**	**GPOK16~17 Output Enable Register**			00h	FCh
		7 – 0	R/W	Output Enable for ***GPOK16~17***, if GPOK is selected.		
1Fh			NA	Reserved	00h	FCh

20h 21h 22h 23h 24h 25h	**GPIOD00 GPIOD08 GPIOD10 GPIOD18 GPIOD20 GPIOD28**	**GPIO00~2F Output Data Register**			00h	FCh
		7 – 0	R/W	Output Data for ***GPIO00~2F***, if function GPIOxx output mode is enabled.		
26h	**GPOWD**	**GPOW0~7 Output Data Register**			00h	FCh
		7 – 0	R/W	Output Data for ***GPOW0~7***, if function GPOW is selected.		
27h	**GPODAD**	**GPODA0~7 Output Data Register**			00h	FCh
		7 – 0	R/W	Output Data for ***GPODA0~7***, if function GPODA is selected.		
28h 29h	**GPOKD00 GPOKD08**	**GPOK0~17 Output Data Register**			00h	FCh
		7 – 0	R/W	Output Data for ***GPOK0~7***, if GPOK is selected. Output Data for ***GPOK8~15***, if GPOK is selected.		
2A~2Dh				Reserved		
2Eh	**GPOKD16**	**GPOK16~17 Output Data Register**			00h	FCh
		7 – 0	R/W	Output Data for ***GPOK16~17***, if GPOK is selected.		
2Fh			NA	Reserved	00h	FCh
30h 31h 32h 33h 34h 35h	**GPIOPIN00 GPIOPIN08 GPIOPIN10 GPIOPIN18 GPIOPIN20 GPIOPIN28**	**GPIO Pin Input Register**			00h	FCh
		7 – 0	RO	***GPIO00~2F*** pin input port value.		
36~38h			NA	Reserved	00h	FCh
39h	**GPIKPIN**	**GPIK0~7 Pin Input Register**			00h	FCh
		7 – 0	RO	***GPIK0~7*** pin input port value.		
3A~3Dh				Reserved		
3Eh	**GPIADPIN**	**GPIAD0~7 Pin Input Register**			00h	FCh
		7 – 0	RO	***GPIAD0~7*** pin input port value.		
3Fh	**GPWUPIN**	**GPWU0~7 Pin Input Register**			00h	FCh
		7 – 0	RO	***GPWU0~7*** pin input port value.		
40h 41h 42h 43h 44h 45h	**GPIOPU00 GPIOPU08 GPIOPU10 GPIOPU18 GPIOPU20 GPIOPU28**	**GPIO Pull-Up Register**			00h	FCh
		7 – 0	R/W	***GPIO00~2F*** Pull-Up Enable. GPIOPU18 default value is dependent by XIO_TP.		
46~48h			NA	Reserved	00h	FCh
49h	**GPIKPU**	**GPIK0~7 Pull-Up Register**			00h	FCh
		7 – 0	R/W	***GPIK0~7*** Pull-Up Enable.		
4A~4Dh				Reserved		
4Eh	**GPIADPU**	**GPIAD0~7 Pull-Up Register**			00h	FCh
		7 – 0	R/W	***GPIAD0~7*** Pull-Up Enable.		
4Fh	**GPWUPU**	**GPWU0~7 Pull-Up Register**			00h	FCh
		7 – 0	R/W	***GPWU0~7*** Pull-Up Enable.		
50h 51h 52h 53h 54h 55h	**GPIOOD00 GPIOOD08 GPIOOD10 GPIOOD18 GPIOOD20 GPIOOD28**	**GPIO Open Drain Enable Register**			00h 00h 00h 00h 00h 00h	FCh
		7 – 0	R/W	***GPIO00~2F*** Open Drain Enable. ***GPIO0C*** is CLKRUN# default open drain enabled. 1: the output pin is Open_Drain 0: the output pin is Push-Pull		
56h	**GPOWOD**	**GPOWOD0~7 Open Drain Enable Register**			00h	FCh
		7 – 0	R/W	***GPOW0~7*** Open Drain Enable. 1: the output pin is Open_Drain 0: the output pin is Push-Pull		
57h	**GPODAOD**	**GPODA0~7 Open Drain Enable Register**			00h	FCh

		7 – 0	R/W	***GPODA0~7*** Open Drain Enable. 1: the output pin is Open_Drain 0: the output pin is Push-Pull		
58h			NA	Reserved	00h	FCh
59h	**GPIKIE**	**GPIK0~7 Input Enable Register**			FFh	FCh
		7 – 0	R/W	***GPIK0~7*** Input Enable.		
5A~5Dh				Reserved		
5Eh	**GPIADIE**	**GPIADIE0~7 Input Enable Register**			00h	FCh
		7 – 0	R/W	***GPIAD0~7*** Input Enable.		
5Fh	**GPWUIE**	**GPWU0~7 Input Enable Register**			00h	FCh
		7 – 0	R/W	***GPWU0~7*** Input Enable.		
60h 61h 62h 63h 64h 65h	**GPIOIE00** **GPIOIE08** **GPIOIE10** **GPIOIE18** **GPIOIE20** **GPIOIE28**	**GPIO Input Enable Register**			60h 00h 00h 00h 07h 10h	FCh
		7 – 0	R/W	Input Enable for ***GPIO00~2F***. ***GPIO05*** (Test clock hardware trap) ***GPIO2C*** (***LRST#***) ***GPIO20*** ***GPIO21*** (***RXD***) are default input enabled.		
66h	**GPIOCSX**	**Others GPIO Driving Current Selection Register**			00h	FCh
				0: 4mA output current 1: 16mA output current.		
		7~4	R/W	Reserved		
		3	R/W	***GPIO12*** Current Selection		
		2	R/W	***GPIO11*** Current Selection		
		1	R/W	***GPIO0A*** Current Selection		
		0	R/W	***GPIO0F*** Current Selection		
67h	**GPIOCS18**	**GPIO18~1F Driving Current Selection Register**			00h	FCh
		7 – 0	R/W	0: 4mA output current 1: 16mA output current.		
68h				Reserved		
69h	**ADCEN**	**ADC Enable**			00h	FCh
		7 – 0	R/W	ADC channel 0~7 enable control. 1: Pin definition is ADC 0: Pin definition is GPI		
6A~6Dh				Reserved		
6Eh	**ADCCFG**	**ADC Configuration**			00h	FCh
		7	R/W	ADC updated 8051 interrupt enable.		
		6	R/W	ADC digital loop back test enable.		
		5-0	R/W	ADC Update period = (update period + 1) x4 ms		
6Fh	**DACEN**	**DAC Enable**			00h	FCh
		7 – 0	R/W	DAC channel 0~7 enable control. 1: Pin definition is DAC 0: Pin defintion is GPO The related digital output enable should be disable if DAC is enabled.		
70~77h	**ADC0~7**	**ADC Read Value for Channel 0~7**			00h	FCh
		7 – 0	R/W	ADC channel digital value		
78~7Fh	**DAC0~7**	**DAC Output Value for Channel 0~7**			00h	FCh
		7 – 0	R/W	DAC channel output value		

4.6 KBC Module

4.6.1 Functional Description

KB3910 inherits the hardware logic of its predecessor-- KB3886, and integrates a hardwired standard 8042 controller. Sandard KB commands are directly processed by hardware, while extended commands can be processed by 8051.

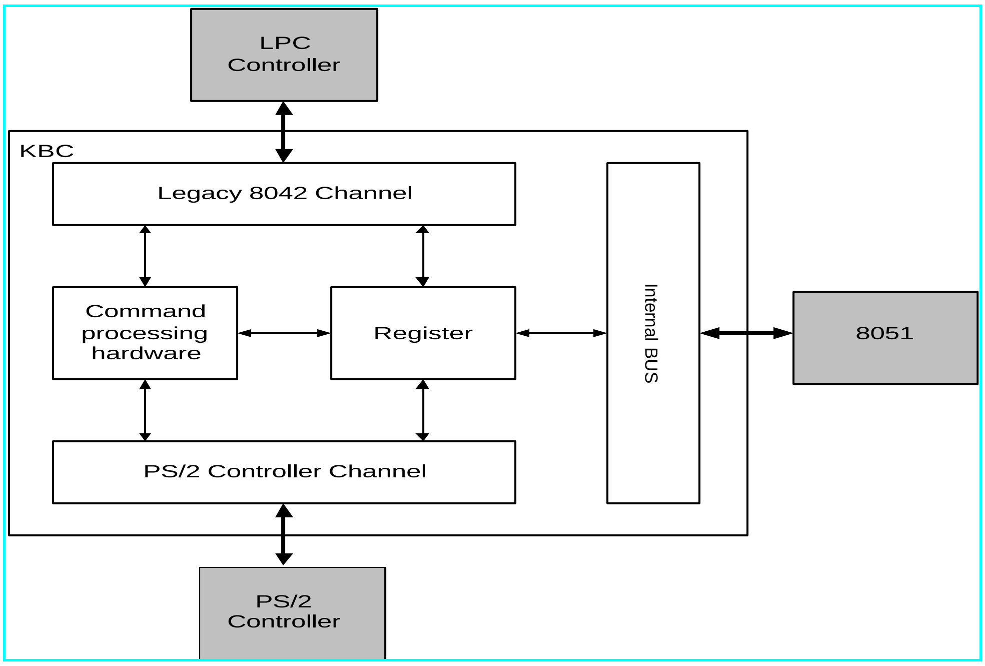
The main functions of KBC are as follows:

--Process most 8042 commands by hardware

--Forward extended commands to 8051

--Forward PS/2 commands to the PS/2 controller

--Forward PS/2 data packets to the host



Legacy 8042 Channel

The interface between KBC and the LPC controller is the legacy 8042 I/O ports. It is based on two I/O ports: command/status port (64h) and data port (60h).

IO 60h: KBC Data Input Register (KBDIN)

When the host writes I/O ports 60h and 64h, the data is stored in **KBDIN**. At the same time, the input buffer full flag (**IBF** bit in **KBSTS**) is set. The input data stored in **KBDIN** is directly fetched by the command processing logic and IBF is also cleared automatically.

IO 60h: KBC Data Output Register (KBDOUT)

The data responded to the host is generated by the hardware circuit. The data is pushed into **KBDOUT** and the output buffer full flag (**OBF** bit in **KBSTS**) is set automatically. KB3910 can be configured to generate interrupts to the host when **OBF** is set. **OBF** is automatically cleared after that the host reads **KBDOUT** (through I/O port 60h).

IO 64h: KBC Status Register (KBSTS)

The host read it through I/O port 64h. The bit format of this register is as follows:

Status Bit	Name	Description
7	**Parity Error**	PS/2 Bus parity error.
6	**General Timeout**	PS/2 Bus timeout.
5	**Aux OBF**	KBDOUT data is from PS/2 auxiliary device.
4	**UnInhibit**	Keyboard is not inhibited.
3	**A2**	Address of the previous write cycle.
2	**System Flag**	POST of the system is finished.
1	**IBF**	Input Buffer Full flag.
0	**OBF**	Output Buffer Full flag.

Hardware Processed Commands

The following standard commands are processed by hardware directly.

<table>
<tr><th>Value</th><th>Command</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">20h</td><td rowspan="2"><b>Read Command Byte</b></td><td colspan="2">Read the command byte of KBC</td></tr>
<tr><td>Response</td><td>Command byte</td></tr>
<tr><td rowspan="2">60h</td><td rowspan="2"><b>Write Command Byte</b></td><td colspan="2">Write the command byte of KBC</td></tr>
<tr><td>Argument</td><td>Command byte</td></tr>
<tr><td rowspan="3">61h</td><td rowspan="3"><b>Set EC Command/Status/Data Ports Base Address</b></td><td colspan="2">Set the base address of EC ports. Arguments for base address written through this command will be stored in Registers FC02h and FC03h</td></tr>
<tr><td rowspan="2">Argument</td><td>Byte1 = base address low byte</td></tr>
<tr><td>Byte2 = base address high byte</td></tr>
<tr><td>90h</td><td><b>Aux Port0 Prefix</b></td><td colspan="2">In PS/2 multiplexing mode, the write cycle to 60h that follows this command is transferred to PS/2 port 0.</td></tr>
<tr><td>91h</td><td><b>Aux Port1 Prefix</b></td><td colspan="2">In PS/2 multiplexing mode, the write cycle to 60h that follows this command is transferred to PS/2 port 1.</td></tr>
<tr><td>92h</td><td><b>Aux Port2 Prefix</b></td><td colspan="2">In PS/2 multiplexing mode, the write cycle to 60h that follows this command is transferred to PS/2 port 2.</td></tr>
<tr><td>93h</td><td><b>Aux Port3 Prefix</b></td><td colspan="2">In PS/2 multiplexing mode, the write cycle to 60h that follows this command is transferred to PS/2 port 3.</td></tr>
<tr><td rowspan="2">A4h</td><td rowspan="2"><b>Test Password</b></td><td colspan="2">This command checks if a password is previously stored in the KB3910.</td></tr>
<tr><td>Response</td><td>FAh if the password is loaded.
F1h if the password is not loaded.</td></tr>
<tr><td>A6h</td><td><b>Enable Password</b></td><td colspan="2">This command enables the lock function of the keyboard controller. In lock mode, KBC forwards the scan code received from PS/2 controller to the 8051 to compare against with the pre-loaded password.</td></tr>
<tr><td>A7h</td><td><b>Disable Aux Device</b></td><td colspan="2">This command inhibits receiving any data packet from the auxiliary device.</td></tr>
<tr><td>A8h</td><td><b>Enable Aux Device</b></td><td colspan="2">This command enables the data packet transfer from the auxiliary device.</td></tr>
<tr><td rowspan="2">A9h</td><td rowspan="2"><b>Test Aux Port</b></td><td colspan="2">Test the signals of the auxiliary port.</td></tr>
<tr><td>Response</td><td>Always return 00h</td></tr>
<tr><td rowspan="2">AAh</td><td rowspan="2"><b>Self Test</b></td><td colspan="2">KBC self test.</td></tr>
<tr><td>Response</td><td>55h</td></tr>
<tr><td rowspan="2">ABh</td><td rowspan="2"><b>Test KB Port</b></td><td colspan="2">Test the signals of the keyboard port.</td></tr>
<tr><td>Response</td><td>Always return 00h</td></tr>
</table>

Value	Command	Description	
ADh	**Disable KB Device**	This command inhibits receiving any data packet from the keyboard device.	
AEh	**Enable KB Device**	This command enables the data packet transfer from the keyboard device.	
C0h	**Read P1**	Read the input port of 8042 P1. Because there is no real 8042 in the chip, this command just emulates the function.	
		Response	Always return 00h
D0h	**Read P2**	Read the output port of 8042 P2. Because there is no real 8042 in the chip, this command just emulates the function.	
		Response	Bit1 is the status of GA20
D1h	**Write P2**	Write the output port of 8042 P2. Because there is no real 8042 in the chip, this command will just emulate the function and set/clear GA20 based on data bit 1.	
		Argument	Bit1 is the status of ***GA20***
D2h	**Write KB Output Buffer**	Write data into **KBDOUT** as if it comes from the keyboard.	
		Argument	Keyboard data
D3h	**Write AUX Output Buffer**	Write data into **KBDOUT** as if it comes from the auxiliary device.	
		Argument	Auxiliary data
D4h	**AUX Port Prefix**	The write cycle to 60h that follows this command is transferred to auxiliary ports.	
E0h	**Read Test Input**	Read the test inputs T0 and T1 of 8042. Because there is no real 8042 in the chip, this command will just emulate the function.	
		Response	Always return 00h
FEh	**KB Reset**	This command generates a 6us low pulse on ***KBRST#***.	

PS/2 Control Channel

There is a PS/2 control channel for data transmission between KBC and the PS/2 Controller. KBC directly forwards the PS/2 device command (received from the host) to PS/2 Controller. KBC also receives the PS/2 data packet from PS/2 Controller and transfers it to the host.

4.6.2 KBC Register Descriptions (Base Address = FC00h, Space 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
80h	KBCCFG	**KBC Command Byte**			40h	FCh
		7	RSV	Reserved		
		6	RW	Scan Code Conversion		
		5	RW	Auxiliary Device Disable		
		4	RW	Keyboard Devic e Disable		
		3	RW	Inhibit Override		
		2	RW	Syatem Flag		
		1	RW	***IRQ12*** Enable		
		0	RW	***IRQ1*** Enable		
81h	KBCLOCK	**Keyboard Lock Control**			00h	FCh
		7	RW	Keyboard Lock Enable		
		6	RW	This bit controls the PS/2 device command/data (through port 60h) w ill be processed by Hardware (=0) or by Firmware(=1). It is recommended that 1: if FEE0h_bit5=1 (PS/2 Bytemode Enable) 0: if FEE0_bit6=1 (PS/2 Hareware mode Enable)		
		5	RW	This bit enables individual KBC standard commands to be processed by firmware. Note that the processing of KBC extended commands will always be firmware-processed and is irrelevant to this bit, 1: KBC standard commands can be programmed to be HW-processed or FW-processed individually in registers FC91h~FC93h 0: No KBC standard commands will be processed by firmware		
		4	RSV	Reserved		
		3~1	RSV	Reserved		
		0	RW	Keyboard Lock		
82h	KBCECAL	**EC Command/Status/Data Ports Base Address Low Byte**			00h	FCh
		7 ~ 0	RW	EC I/O Ports Base Address Low Byte. Bit 2 must be set to "1" to enable the EC Cmd/Sts/Data ports. This register is usually written as "66h", thereby enables the KB3910 to respond to I/O addresses 62h/66h on the LPC bus. This register can be Read/Write directly through EC address space, or it can be written indirectly via KBC command 61h		
83h	KBCECAH	**EC Command/Status/Data Ports Base Address High Byte**			00h	FCh
		7 ~ 0	RW	EC I/O Ports Base Address High Byte. This register is usually written as 00h, and it can be Read/Write directly through EC address space, or written indirectly via KBC command 61h		
84h	KBCDIN	**KBC Input Buffer**			00h	FCh
		7 ~ 0	RO	The data written to I/O port 60h will be stored in this register.		
85h	KBCDOUT	**KBC Output Buffer**			00h	FCh
		7 ~ 0	RW	Writing to this register will cause the output buffer full flag **OBF** to be set. The host can read this register through I/O port 60h.		
86h	KBCSTS	**KBC Status**			00h	FCh

		7	RW	Parity Error. When PS/2 protocol has a parity error, this bit will be set to high. This bit is also used as port indicator for PS/2 active multiplexing mode.		
		6	RW	TimeOut. When PS/2 protocol has a timeout error, this bit will be set to high. This bit is also used as port indicator for PS/2 active multiplexing mode.		
		5	RW	Auxiliary Data Flag.		
		4	RO	Uninhibited.		
		3	RO	Address.		
		2	RO	System Flag.		
		1	RO	**IBF**		
		0	RO	**OBF**.		
87h	**KBCINTC**	**Interrupt Control**			00h	FCh
		7	RW	**IBF** Interrupt Enable. This bit enables KBC to generate interrupt to the 8051 at the rising edge of IBF, when the KBC command being received will be bypassed to firmware for processing. **This bit is recommended to set to 1.**.		
		6	RW	**OBF** Interrupt Enable. If this bit is set, KBC generates interrupt to the core processor at the falling edge of **OBF**.		
		5	RW	PS/2 Request Interrupt Enable. **f** this bit is set, KBC generates interrupt to the core processor when PS/2 controller request to transfer data.		
		4	RW	PS/2 Transfer Ready Interrupt Enable. If this bit is set, KBC generates interrupt to the core processor when PS/2 controller sends the ready signal.		
		3	R/WC	**IBF** Event Status. Write “1” to clear this bit. This bit indicates that there is a firmware-processed KBC command is currently being processed by the firmware		
		2	R/WC	**OBF** Event Status. Write “1” to clear this bit.		
		1	R/WC	PS/2 Request Event Status. Write “1” to clear this bit.		
		0	R/WC	PS/2 Ready Event Status. Write “1” to clear this bit.		
88h	**KBCPS2C**	**PS/2 Interface Control**			00h	FCh
		7	WO	Revert to Legacy Mode, write 1 to this bit will revert the AUX devices to legacy mode, write 0 has no effect		
		6	R/W	Set PS/2 Request, write 1 to this bit will set the PS/2 request, write 0 has no effect PS/2 Request Status can be read from this bit		
		5	R/W	Clear the Firmware Processing Status, write 1 to this bit will clear the firmware processing status, write 0 has no effect Firmware Processing Status can be read from this bit		
		4	R/W	0: D4h command is processing. 1: D4h command has been finished.		
		3	WO	Bit4 update Enable		
		2	WO	Write “1” to this bit to notify the PS/2 Controller that the data transfer request has been serviced.		
		1	WO	Write “1” to this bit to clear **IBF**.		
		0	WO	Write “1” to this bit to clear **OBF**.		
89h	**KBCPS2F**	**PS/2 Interface Flag**			00h	FCh
		7	RO	PS/2 Status Bit 7		
		6	RO	PS/2 Status Bit 6		

		5	RO	PS/2 Auxiliary Device Error		
		4	RO	PS/2 Command Error		
		3 ~ 2	RSV	Reserved		
		1	RO	PS/2 Keyboard/Auxiliary Flag		
		0	RO	PS/2 Data/Response Flag		
8Ah	**KBCPS2D**	**PS/2 Interface Data**			00h	FCh
		7 ~ 0	RO	PS/2 data which will be transferred to the host.		
8Bh	**KBCGA20**	**Fast Gate A20 Control**			00h	FCh
		7 ~ 2	RSV	Reserved		
		1	RW	Fast Gate A20 Control		
		0	RSV	Reserved		
8C~8Eh	**Reserved**	**Reserved**			00h	FCh
8Fh	**KBCPS2M**	**PS/2 Multiplexing Mode**			00h	FCh
		7	RW	PS/2 multiplexing mode enable		
		6	RW	PS/2 port 2 enable		
		5	RW	PS/2 port 1 enable		
		4	RW	PS/2 port 0 enable		
		3	RW	Target port is PS/2 port 3		
		2	RW	Target port is PS/2 port 2		
		1	RW	Target port is PS/2 port 1		
		0	RW	Target port is PS/2 port 0		
90h	**KBCFSM**	**Command Finite State Machine**			01h	FCh
		7 ~ 0	RO	Hardware Command logic State Machine		
91h	**KBC_FWCMDEN0**	**Enable Firmware Processing of KBC Standard Command Set_0**			00h	FCh
		7	RW	Standard KBC commands are default to be hardware-processed directly. It is possible to enable standard commands individually to be firmware-processed by setting the corresponding enable bits in this register. When a standard KBC command is enabled to be firmware-processed, the hardware logic will bypass this command and inform the 8051 upon receiving this command. Note that Register **FC81h_bit5** must be set to 1 for this register to become effective. **KBC Command A4h: Test Password** 1: Firmware processed 0: Hardware processed		
		6	RW	**KBC Command 93h: AUX Port3 Prefix** 1: Firmware processed 0: Hardware processed		
		5	RW	**KBC Command 92h: AUX Port2 Prefix** 1: Firmware processed 0: Hardware processed		
		4	RW	**KBC Command 91h: AUX Port1 Prefix** 1: Firmware processed 0: Hardware processed		
		3	RW	**KBC Command 90h: AUX Port0 Prefix** 1: Firmware processed 0: Hardware processed		
		2	RW	**KBC Command 61h: Set EC CMD/Status/Data Base address** 1: Firmware processed 0: Hardware processed		

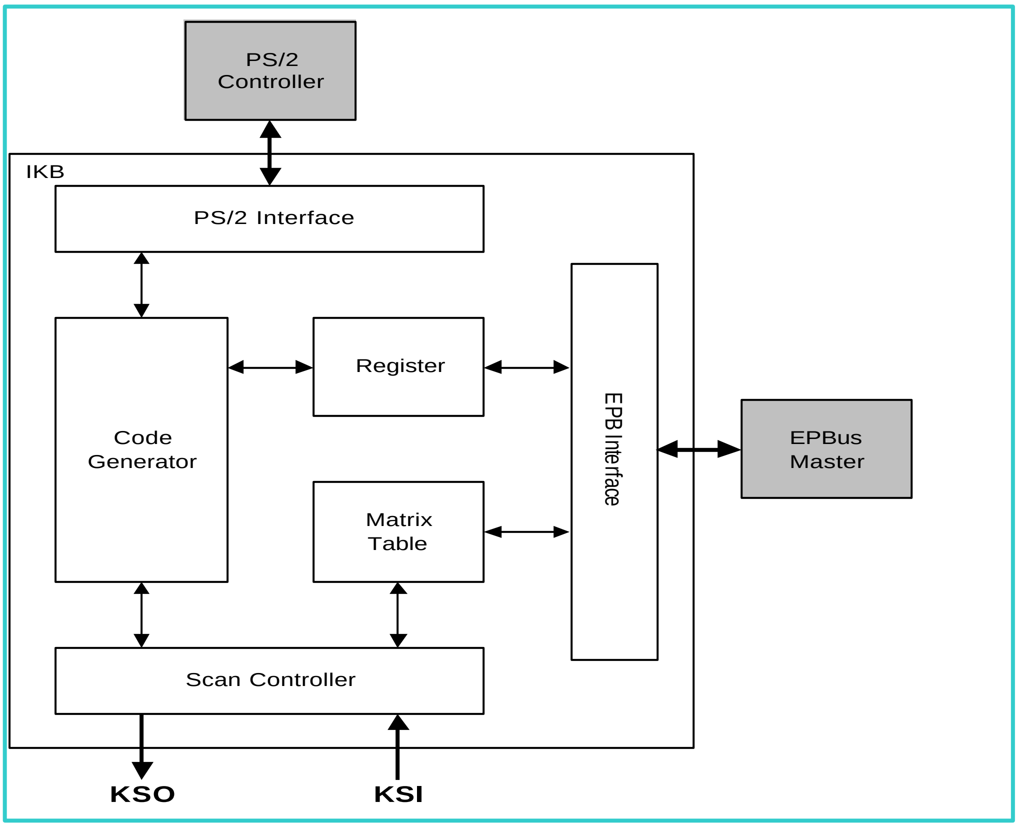
		1	RW	**KBC Command 60h: Write Command Byte** 1: Firmware processed 0: Hardware processed		
		0	RW	**KBC Command 20h: Read Command Byte** 1: Firmware processed 0: Hardware processed		
92h	KBC_FWCMDEN1			**Enable Firmware Processing of KBC Standard Command Set_1**	00h	FCh
		7	RW	Standard KBC commands are default to be hardware-processed directly. It is possible to enable standard commands individually to be firmware-processed by setting the corresponding enable bits in this register. When a standard KBC command is enabled to be firmware-processed, the hardware logic will bypass this command and inform the 8051 upon receiving this command. Note that Register **FC81h_bit5** must be set to 1 for this register to become effective. **KBC Command AEh: Enable KB Device** 1: Firmware processed 0: Hardware processed		
		6	RW	**KBC Command ADh: Disable KB Device** 1: Firmware processed 0: Hardware processed		
		5	RW	**KBC Command ABh: Test KB Port** 1: Firmware processed 0: Hardware processed		
		4	RW	**KBC Command AAh: Self Test** 1: Firmware processed 0: Hardware processed		
		3	RW	**KBC Command A9h: Test AUX Port** 1: Firmware processed 0: Hardware processed		
		2	RW	**KBC Command A8h: Enable AUX device** 1: Firmware processed 0: Hardware processed		
		1	RW	**KBC Command A7h: Disab le AUX device** 1: Firmware processed 0: Hardware processed		
		0	RW	**KBC Command A6h: Enable Password** 1: Firmware processed 0: Hardware processed		
93h	KBC_FWCMDEN2			**Enable Firmware Processing of KBC Standard Command Set_2**	00h	FCh
		7	RW	Standard KBC commands are default to be hardware-processed directly. It is possible to enable standard commands individually to be firmware-processed by setting the corresponding enable bits in this register. When a standard KBC command is enabled to be firmware-processed, the hardware logic will bypass this command and inform the 8051 upon receiving this command. Note that Register **FC81h_bit5** must be set to 1 for this register to become effective. **KBC Command FEh: KB Reset** 1: Firmware processed 0: Hardware processed		
		6	RW	**KBC Command EDh: Read Test Input** 1: Firmware processed 0: Hardware processed		
		5	RW	**KBC Command D4h: AUX Port Prefix** 1: Firmware proc essed 0: Hardware processed		
		4	RW	**KBC Command D3h: Write AUX Output Buffer** 1: Firmware processed 0: Hardware processed		

		3	RW	**KBC Command D2h: Write KB Output Buffer** 1: Firmware processed 0: Hardware processed		
		2	RW	**KBC Command D1h: Write Port 2** 1: Firmware processed 0: Hardware processed		
		1	RW	**KBC Command D0h: Read Port 2** 1: Firmware processed 0: Hardware processed		
		0	RW	**KBC Command C0h: Read Port 1** 1: Firmware processed 0: Hardware processed		
94h	KBC_PS2CMDBUF	**Command Command Buffer**			00h	FCh
		7 ~ 0	RW	PS/2 Command written in this register will be sent to PS/2 module		
95h~9Fh	Not Used	7 ~ 0	RSV	Reserved	00h	FCh

4.7 IKB Module

4.7.1 IKB Funtional Descriptions

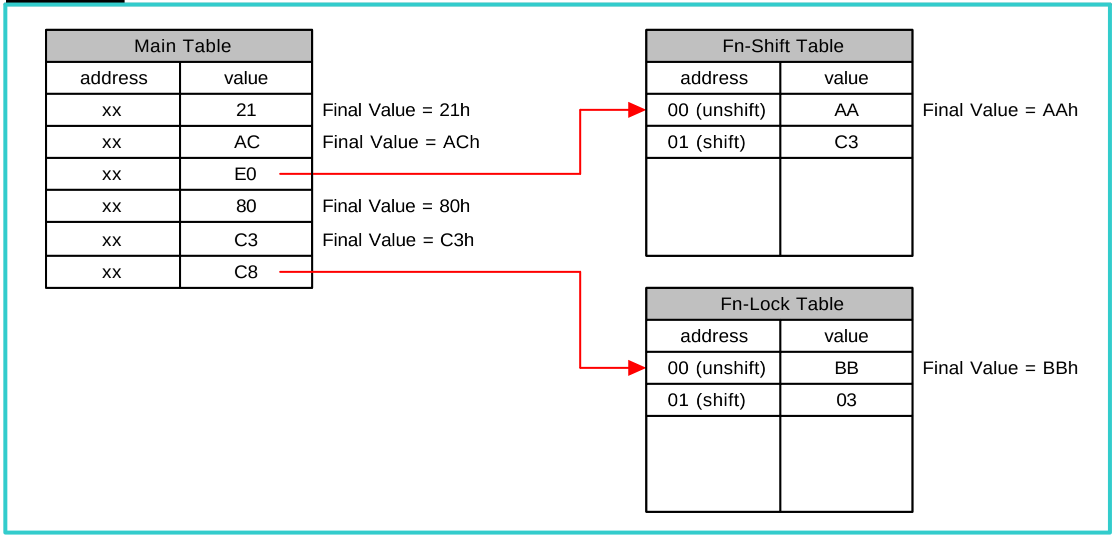
The Internal Keyboard **(IKB)** is full hardware implemented. It includes a scan controller, a scan code generator, a PS/2 interface, and an internal memory used to store the matrix value and the configuration setting. It can support all of the Windows2000™ WWW/Multimedia Key, 12 functional hot keys, and 16 user-defined hot keys. The block diagram of IKB is shown as follows:



Keyboard Matrix Scan Controller

The scan controller supports 18 scan output line and 8 scan input lines. All of the scan outputs are driven to low when there is no key pressed. If any key is pressed, the scan controller disables all of the scan outputs and then drives these outputs one by one. According to the index of the driven scan output and the value of the scan inputs; the controller can address the matrix table to get the matrix value associated with the pressed key.

The matrix table is a two levels structure: the main table and Fn-Shift/Fn-Lock table. The controller first accesses the main table. If the value is greater than C7h, it will use the value as an index to access Fn-Shift/Fn-Lock table. The final value is pushed into a first-in-first-out (FIFO) buffer, from which output is connected to the scan code generator.



Main Table		
address	value	
xx	21	Final Value = 21h
xx	AC	Final Value = ACh
xx	E0	
xx	80	Final Value = 80h
xx	C3	Final Value = C3h
xx	C8	

Fn-Shift Table		
address	value	
00 (unshift)	AA	Final Value = AAh
01 (shift)	C3	

Fn-Lock Table		
address	value	
00 (unshift)	BB	Final Value = BBh
01 (shift)	03	

Scan Code Generator

The code generator gets the matrix value from the FIFO buffer and decodes it to generate the scan code of the key. If the matrix value is less than 80h, the value is treated as **set-2 matrix value** and the conversion from **set-2 martrix value** to the corresponding **set-1 scan code** is done by the hardware. If the matrix value is greater than 7Fh and less than B4h, a special function associated with the value is executed. The function may be sending a sequence of scan codes, updating some internal flags. If the matrix value is greater than B3h, a hot key event is generated. The detail description is as follows.

Matrix Value (set 2)	Description	Scan Code (set 1)	Matrix Value (set 2)	Description	Scan Code (set 1)
00h	Error(overrun)	FFh	40h	Reserved	6Bh
01h	F9	43h	41h	< ,	33h
02h	F7	41h	42h	K	25h
03h	F5	3Fh	43h	I	17h
04h	F3	3Dh	44h	O	18h
05h	F1	3Bh	45h	) 0	0Bh
06h	F2	3Ch	46h	(9	0Ah
07h	F12	58h	47h	Reserved	60h
08h	Reserved	64h	48h	Reserved	6Ch
09h	F10	44h	49h	> .	34h
0Ah	F8	42h	4Ah	? /	35h
0Bh	F6	40h	4Bh	L	26h
0Ch	F4	3Eh	4Ch	: ;	27h
0Dh	Tab	0Fh	4Dh	P	19h
0Eh	~	29h	4Eh	_ -	0Ch
0Fh	Reserved	59h	4Fh	Reserved	61h
10h	Reserved	65h	50h	Reserved	6Dh
11h	Left Alt	38h	51h	Reserved	73h
12h	Left Shift	2Ah	52h	“ ‘	28h
13h	Reserved	70h	53h	Reserved	74h
14h	Left Ctrl	1Dh	54h	{ [	1Ah
15h	Q	10h	55h	+ =	0Dh
16h	! 1	02h	56h	Reserved	62h
17h	Reserved	5Ah	57h	Reserved	6Eh
18h	Reserved	66h	58h	Caps Lock	3Ah
19h	Reserved	71h	59h	Right Shift	36h
1Ah	Z	2Ch	5Ah	Return	1Ch
1Bh	S	1Fh	5Bh	}]	1Bh
1Ch	A	1Eh	5Ch	Reserved	75h
1Dh	W	11h	5Dh	|\(USonly) ~#(102-key)	2Bh

1Eh	@ 2	03h	5Eh	Reserved	63h
1Fh	Reserved	5Bh	5Fh	Reserved	76h
20h	Reserved	67h	60h	Fn (PTL)	55h
21h	C	2Eh	61h	|\(102-key)	56h
22h	X	2Dh	62h	Reserved	77h
23h	D	20h	63h	Reserved	78h
24h	E	12h	64h	Reserved	79h
25h	$ 4	05h	65h	Reserved	7Ah
26h	# 3	04h	66h	Backspace	0Eh
27h	Reserved	05h	67h	Reserved	7Bh
28h	Reserved	68h	68h	Reserved	7Ch
29h	Space	39h	69h	1 End	4Fh
2Ah	V	2Fh	6Ah	Reserved	7Dh
2Bh	F	21h	6Bh	4 Left Arrow	4Bh
2Ch	T	14h	6Ch	7 Home	47h
2Dh	R	13h	6Dh	Reserved	7Eh
2Eh	% 5	06h	6Eh	Reserved	7Fh
2Fh	Reserved	5Dh	6Fh	Reserved	6Fh
30h	Reserved	69h	70h	0 Ins	52h
31h	N	31h	71h	. Del	53h
32h	B	30h	72h	2 Down Arrow	50h
33h	H	23h	73h	5	4Ch
34h	G	22h	74h	6 Right Arrow	4Dh
35h	Y	15h	75h	8 Up Arrow	48h
36h	^ 6	07h	76h	ESC	01h
37h	Reserved	5Eh	77h	Num Lock	45h
38h	Reserved	6Ah	78h	F11	57h
39h	Reserved	72h	79h	+	4Eh
3Ah	M	32h	7Ah	3 PgDn	51h
3Bh	J	24h	7Bh	-	4Ah
3Ch	U	16h	7Ch	*	37h
3Dh	& 7	08h	7Dh	9 PgUp	49h
3Eh	* 8	09h	7Eh	Scroll Lock	46h
3Fh	Reserved	5Fh	7Fh	Sys Req (84-key)	54h

Matrix Value	Description	Scan Code	Matrix Value	Description	Scan Code
00h – 7Fh	Standard Keys	See table above	9Ah	ACPI Sleep	E0 5F
80h	Left Shift	2Ah	9Bh	ACPI Wake	E0 63
81h	Left Ctrl	1Dh	9Ch	Left Window	E0 5B
82h	Left Alt	38h	9Dh	Right Window	E0 5C
83h	F7	41h	9Eh	Windows App	E0 5D
84h	SysReq	54h	9Fh	Break	1D E0 46
85h	Right Shift	36h	A0h	Volume Up	E0h 30h
86h	Right Ctrl	E0h 1Dh	A1h	Volume Down	E0h 2Eh
87h	Right Alt	E0h 38h	A2h	Next	E0h 19h
88h	Print Screen	E0h 2Ah E0h 37h	A3h	Previous	E0h 10h
89h	Pause	E1h 1Dh 45h	A4h	Stop	E0h 24h
8Ah	Insert	E0h 52h	A5h	Play/Pause	E0h 22h
8Bh	Home	E0h 47h	A6h	Mute	E0h 20h
8Ch	Page Up	E0h 49h	A7h	Media Select	E0h 6Dh
8Dh	Delete	E0h 53h	A8h	Email Reader	E0h 6Ch
8Eh	End	E0h 4Fh	A9h	Calculator	E0h 21h
8Fh	Page Down	E0h 51h	AAh	My Computer	E0h 6Bh
90h	Up Arrow	E0h 48h	ABh	WWW Search	E0h 65h
91h	Left Arrow	E0h 41h	ACh	WWW Home	E0h 32h
92h	Down Arrow	E0h 50h	ADh	WWW Back	E0h 6Ah
93h	Right Arrow	E0h 4Dh	AEh	WWW Forward	E0h 69h
94h	/	E0h 35h	AFh	WWW Stop	E0h 68h
95h	Enter	E0h 1Ch	B0h	WWW Refresh	E0h 67h
96h	Fn Shift	No scan code	B1h	WWW Favor	E0h 66h
97h	Fn Lock	No scan code	B2h	OADG	45h/46h
98h	Num/Fn Lock	45h	B3h	Empty Key	No scan code
99h	ACPI Power	E0h 5Eh	B4h - FFh	Hot Key	

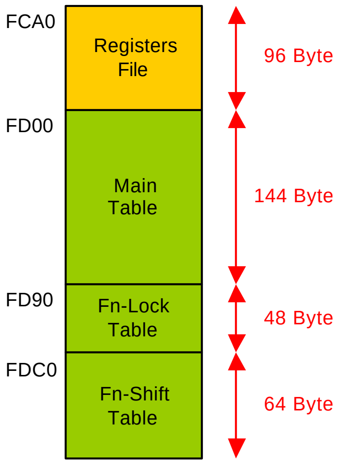
Internal PS/2 Interface

IKB implements a PS/2 interface and uses it to communicate with the PS/2 controller through an internal pseudo-PS/2 bus. The communication protocol is compliant with the PS/2 specification. IKB accepts and responds all of the standard PS/2 keyboard commands. Other reserved commands are also acknowledged, but IKB will not be affected.

<table>
<tr><th>Command</th><th colspan="2">Description</th></tr>
<tr><td rowspan="5">EDh</td><td colspan="2"><b>Set LED.</b> modify the status of LED by the following argument byte. The specification of the argument byte is:</td></tr>
<tr><td>Bit 7 – 3</td><td>All must be zero</td></tr>
<tr><td>Bit 2</td><td>Caps Lock LED status</td></tr>
<tr><td>Bit 1</td><td>Num Lock LED status</td></tr>
<tr><td>Bit 0</td><td>Scroll Lock LED status</td></tr>
<tr><td>EEh</td><td colspan="2"><b>Echo.</b> send EEh back to the host after receiving this command.</td></tr>
<tr><td>F0h</td><td colspan="2"><b>Access Scan Code Set.</b> The host uses the first argument to specify the read/write operation. If the first argument equals 00h, it represents a read operation and KB3910 returns two bytes, FAh and 41h. If the first argument is not equal to 00h, it represents a write operation and KB3910 ignores the argument because KB3910 only supports Set 2 scan code.</td></tr>
<tr><td>F2h</td><td colspan="2"><b>Get Device ID.</b> return 3 bytes, FAh, ABh, and 41h.</td></tr>
<tr><td>F3h</td><td colspan="2"><b>Set Typematic Rate.</b></td></tr>
<tr><td>F4h</td><td colspan="2"><b>Enable.</b> begin scanning the key matrix and sending the scan code to the host. Note that KB3910 is in disable mode after hardware reset. The system BIOS should configure all options of KB3910 and enable it at last.</td></tr>
<tr><td>F5h</td><td colspan="2"><b>Disable.</b> stop scanning the key matrix and flush the scan code buffers.</td></tr>
<tr><td>F6h</td><td colspan="2"><b>Set Default.</b> restore the default setting of typematic rate and LED status.</td></tr>
<tr><td>FEh</td><td colspan="2"><b>Resend.</b> re-transmit the last byte.</td></tr>
<tr><td>FFh</td><td colspan="2"><b>Reset.</b> generates an internal soft-reset signal to reset the PS/2 interface, clear all internal flags of scan controller, and flush the scan code buffers.</td></tr>
</table>

IKB Register Mapping in EC and 8051 Address Space

IKB implements two blocks of memory, the register file and the matrix table.



4.7.2 IKB Register Descriptions (Base address = FC00h, Space = 352 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
A0h	IKBCFG	**Internal Keyboard Control**			03h	FCh
		7	RW	Internal Keyboard Enable.		
		6	RW	Code Generator Disable. If this bit is set, the code generator will NOT generate scan code but generate an interrupt to the core processor.		
		5 ~ 0	RW	PS/2 Cycle Time.		
A1h	IKBLEDC	**LED Control**			90h	FCh
		7	RW	PadLock equal to NumLock.		
		6 ~ 5	RSV	Reserved.		
		4 ~ 3	RO	Scan Code Set.		
		2	RW	CapLock Status.		
		1	RW	NumLock Status.		
		0	RW	ScrLock Status		
A2h	IKBTYPEC	**Typematic Control**			2Bh	FCh
		7	RSV	Reserved.		
		6 ~ 5	RW	Initial Delay.		
		4 ~ 0	RW	Repeat Rate.		
A3h	IKBCGFLG	**Code Generator Flag**			00h	FCh
		7	RO	Left Shift.		
		6	RO	Left Ctrl.		
		5	RO	Left Alt.		
		4	RO	Right Shift.		
		3	RO	Right Ctrl.		
		2	RO	Right Alt.		
		1	RO	Fn Shift.		
		0	RW	Fn Lock.		
A4h	IKBCGCTL	**Code Generator Control**			00h	FCh
		7	RW	TX Request. The core processor set this bit to assert a request to transfer one byte of data to the PS/2 controller.		
		6	RWC1	Scan Request Disable. When the code generator generates an interrupt to the core processor, this bit will be set automatically to prevent from fetching the next matrix value. At the end of the interrupt subroutine, this bit should be cleared by the firmware.		
		5	RO	TX Busy. If this bit is set, the internal PS/2 bus is busy for data transfer.		
		4 ~ 0	RSV	Reserved.		
A5h	IKBTXDAT	**Transfer Data**			00h	FCh
		7 ~ 0	RW	TX Data. The data written to this register will be transferred to the PS/2 controller.		
A6h	IKBHKVAL	**Hot Key Matrix Value**			00h	FCh
		7 ~ 0	RO	The matrix value of the hot key is stored in this register.		
A7h	IKBHKFLG	**Hot Key Flag**			00h	FCh
		7 ~ 2	RSV	Reserved.		
		1	RO	Make Flag.		
		0	RO	Break Flag.		
A8h	IKBSADBC	**Scan Address Buffer Control**			00h	FCh
		7	RW	Scan Address Buffer Enable. If this bit is set to "1", the scan address of the pressed key is stored to the scan address buffer.		
		6	RWC1	Scan Address Buffer Valid. When a new address is pushed into the buffer, this bit will be set. This bit will be cleared to 0 by writing a "1".		

		5	RW	Enable the function of waking up from deep-sleep mode by pressing any key on Keyboard. 0: disable 1: enable		
		4~0	RSV	Reserved		
A9h	**IKBSADB**	**Scan Address Buffer**			00h	FCh
		7 ~ 0	RO	Scan Address.		
AAh	**IKBSTCTL**	**Scan Timing Control**			F7h	FCh
		7 ~ 4	RW	Scanning Pre-charge Time.		
		3 ~ 0	RW	Scanning Driving Time		
ABh	**IKBSDBC**	**Scan Debounce Control**			73h	FCh
		7	RSV	Reserved.		
		6 ~ 4	RW	Break Debounce.		
		3	RSV	Reserved.		
		2 ~ 0	RW	Make Debounce.		
ACh	**IKBINTC**	**Interrupt Control**			00h	FCh
		7 ~ 2	RSV	Reserved.		
		1	RW	Hot Key Interrupt Enable.		
		0	RW	Transfer Ready Interrupt Enable.		
ADh	**IKBINTS**	**Interrupt Status**			00h	FCh
		7 ~ 2	RSV	Reserved.		
		1	R/WC1	Hot Key Interrupt Status.Write “1” to clear this bit.		
		0	R/WC1	Transfer Ready Interrupt Status. Write “1” to clear this bit.		
Other		**Reserved**			00h	FCh

Scan Matrix Tables (Base Address = FD00h, Space = 256 bytes)

Offset	Table	Size	Description
FD00h ~ FD8Fh	**Main Table**	144	Each register is associated a key located on the cross point of the scan output lines and the scan input line. IKB can support up to 144 keys.
FD90h ~ FDBFh	**Fn-Lock Table**	48	These registers are divided into 24 words which are associated with the Fn-Lock keys. The even address byte is stored the un-locked matrix value. The odd address byte is stored the locked matrix value. IKB can support up to 24 Fn-Lock keys.
FDC0h ~ FDFFh	**Fn-Shift Table**	64	These registers are divided into 32 words which are associated with the Fn-Shift keys. The even address byte is stored the un-shifted matrix value. The odd address byte is stored the shifted matrix value. IKB can support up to 32 Fn-Lock keys.

4.8 PWM Module

4.8.1 PWM Funtional Description

Each PWM can be configured to select one of 1us/64us/256us/4ms clocks as the reference clock by setting **PWMCLK03** and **PWMCLK47** registers. Every PWM output has its Cycle Time register (**PWMCYTRi**, i = 0 to 7) and Duty Cycle register (**PWMDCRMSTi**, i = 0 to 7). The PWM cycle is calculated with the selected reference clock. It has (**PWMCYTR** + 1) cycles, out of which the signal will be high for **PWMDCRMSTi** cycles. Polarity bits may be used to inverse the behavior.

The PWM output signal cycle time is:

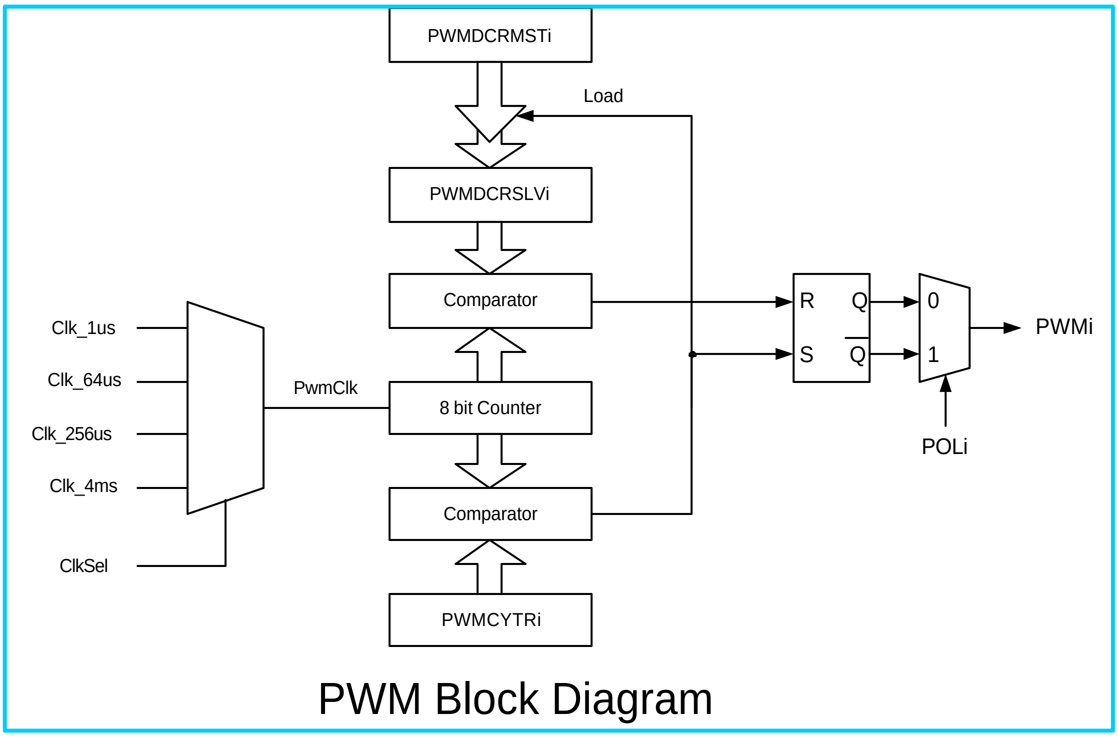
(**PWMCYTRi** + 1) x Tclk   Tclk: selected reference clock

The duty cycle (in %, when polarity bit is 0) of PWM output signal is

(**PWMDCRMSTi** + 1)/ (**PWMCYTR** + 1) x 100

The 8-bit counter starts counting after a PWM is enabled The Duty Cycle Register can be written to at any time, but the duty cycle value is not latched to the slave register until after a match between the 8-bit counter and the cycle time register occurs. The slave register is read only.

The following is a functional block diagram of the PWM module.



PWM Block Diagram

4.8.2 PWM Register Descriptions (Base address = FE00h, Space 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
00h 01h 02h 03h 04h 05h 06h 07h	PWMDCRMST0 PWMDCRMST1 PWMDCRMST2 PWMDCRMST3 PWMDCRMST4 PWMDCRMST5 PWMDCRMST6 PWMDCRMST7	**Duty Cycle Master Register of PWM0~7 output**			00h	FEh
		7 – 0	R/W			
08h 09h 0Ah 0Bh 0Ch 0Dh 0Eh 0Fh	PWMDCRSLV0 PWMDCRSLV1 PWMDCRSLV2 PWMDCRSLV3 PWMDCRSLV4 PWMDCRSLV5 PWMDCRSLV6 PWMDCRSLV7	**Duty Cycle Slave Register of PWM0~7 output**			00h	FEh
		7 – 0	RO			
10h 11h 12h 13h 14h 15h 16h 17h	PWMCYTR0 PWMCYTR1 PWMCYTR2 PWMCYTR3 PWMCYTR4 PWMCYTR5 PWMCYTR6 PWMCYTR7	**Cycle Time Register of PWM0~7 output**			00h	FEh
		7 – 0	R/W			
18h	PWMPOL	**PWM Output Polarity of PWM0~7**			00h	FEh
		7 – 0	R/W			
19h	PWMCTR	**PWM Control Register of PWM0~7**			00h	FEh
		7~0	R/W	0: Disable ***PWM0~7*** 1: Enable ***PWM0~7***		
1Ah 1Bh	PWMCLK03 PWMCLK47	**PWM 0~7 Clock Source Selection**			00h	FEh
		7~6	R/W	***PWM3*** / ***PWM7*** Clock Source Selection		
		5~4	R/W	***PWM2*** / ***PWM6*** Clock Source Selection		
		3~2	R/W	***PWM1*** / ***PWM5*** Clock Source Selection		
		1~0	R/W	***PWM0*** / ***PWM4*** Clock Source Selection		
		Clock source: 00: 1us clock 01: 64us clock 10: 256us clock 11: 4ms clock				

4.9 FAN Module

4.9.1 FAN Funtional Descriptions

There are three identical FAN controllers in KB3910. Each FAN controller includes 2 major parts—the fan tachometer monitor and a FAN PWM controller. The FAN PWM controller may be controlled automatically, by setting the Automatic FAN PWM Control bit (**FANCFG** bit_1).

Functions in a FAN controller:

- FAN tachometer monitor
- FAN PWM control

FAN Tachometer Monitor

Fan speed is calculated by a 16-bit counter with 1us resolution by counting the duty cycle length of ***FANFB1/2/3*** signals. The monitor logic disregards whether the fan is 1-pulse or 2-pulse per rotation cycle. It just calculates the time between rising edges on the Fan feedback signals ***FANFB1/2/3***. The corresponding RPM vs counter value is listed as follows.

RPM	Round / 1 min	Round / sec	Usec / Round	counter value
6000	6000	100 (**6000** / 60 sec)	10000 (10^6 / **100**)	10000* 5000**
5000	5000	83.333 (**5000** / 60)	12000 (10^6 / **83.333**)	12000* 6000*
4000	4000	66.667 (**4000** / 60)	15000 (10^6 / **66.667**)	15000* 7500**
3000	3000	50 (**4000** / 60)	20000 (10^6 / **50**)	20000* 10000*
2000	2000	33.333 (**2000** / 60)	30000 (10^6 / **33.33**)	30000* 15000**
1000	1000	16.667 (**1000** / 60)	60000 (10^6 / **16.667**)	60000* 30000**

* value applicable to 1-pulse per rotation fan

** value applicable to 2-pulse per rotation fan

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

To enable the monitor function of fan feedback speed, the **FANCFG** bit_0 should be set. If fan speed is too slow to calculate, the fan timeout error flag (**FANSTS** bit_1) will be set. The interrupt should be set for monitoring update interrupt and timeout interrupt. The update flag is in **FANSTS** bit 0.

FAN PWM Control

The Fan PWM is a 12-bit counter clocked by 4MHz with a resulting 1.024ms cycle time. If **FANCFG** bit_1 is set, the PWM is controlled automatically by comparing the difference between **FANMON** (FAN monitor speed) and **FANSET** (set FAN target speed). If **FANMON** is larger than **FANSET**, indicating the current fan speed is too slow, the PWM should increase its duty cycle to increase the fan speed, and vice versa. The PWM will remain still if (**FANMON** – **FANSET**) is less than four.

If **FANCFG** bit_1 is clear, the PWM is controlled manually by firmware. Please note that the **FANPWM** value should be set by low-byte then high-byte sequence.

4.9.2 FAN Register Descriptions (Base Address = FE00h, Space = 48 bytes)

(FAN1 = FE200h~FE2Fh / FAN2 = FE300h~FE3Fh / FAN3 = FE400h~FE4Fh)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
20h 30h 40h	FANCFG1 FANCFG2 FANCFG3	**Fan Controller Configuration**			00h	FEh
		7	R/W	Test Mode; Do not enable.		
		6~5	R/W	FAN Initialized Time The FAN Initialized Time defines a period of time during which the FAN monitor event will not be asserted, but the automatic FAN PWM will function normally to control the FAN rotation speed, provided bit_4 of this register is enabled. 00: 1~2 x 256 ms 01: 3~4 x 256 ms 10: 7~8 x 256 ms 11: 14~15 x 256 ms		
		4	R/W	Enable PWM		
		3	R/W	Enable Interrupt caused by fan speed monitor event		
		2	R/W	Enable Interrupt caused by fan speed monitor timeout error event.		
		1	R/W	Enable Automatic Fan PWM control. It is required to set Bit_4 and Bit_0 along with Bit_1.to make this function work.		
		0	R/W	Enable fan tachometer monitor function		
21h 31h 41h	FANSTS1 FANSTS2 FANSTS3	**Fan Controller Status**			00h	FEh
		7~2	R/W	Reserved.		
		1	R/W	Flag bit for Fan speed monitor timeout error event. This bit can be cleared bywriting a '1'.		
		0	R/W	Flag bit for Fan speed monitor update event. This bit can be cleared bywriting a '1'.		
23h 22h 33h 32h 43h 42h	FANMONL1 FANMONH1 FANMONL2 FANMONH2 FANMONL3 FANMONH3	**Fan Speed Monitor Counter Value**			FFh FFh FFh FFh	FEh
		7-0	R/W	The latest fan speed monitor counter value. This value is the pulse width in monitoring clock count.		
25h 24h 35h 34h 45h 44h	FANSETL1 FANSETH1 FANSETL2 FANSETH2 FANSETL3 FANSETH3	**Fan Speed Set Counter Value**			00h	FEh
		7-0	R/W	The target counter value for the fan to achieve in PWM automatic mode (**FANCFG**.1 = 1).		
27h 26h 37h 36h 47h 46h	FANPWML1 FANPWMH1 FANPWML2 FANPWMH2 FANPWML3 FANPWMH3	**Fan PWM High Pulse Width Bits [11:0]**			00h	FEh
		7-0	R/W	If the fan is not in automatic mode (**FANCFG**.1 = 0), this byte will be the PWM high period in a PWM cycle. PWM high period = ((PWM value) + 1) x system clock. These two bytes should be written by low-byte(PWML) then high-byte(PWMH) sequence. The write to PWMH will really update FAN controller PWM value.		
29h 28h 39h 38h 49h 48h	FANCPWML1 FANCPWMH1 FANCPWML2 FANCPWMH2 FANCPWML3 FANCPWMH3	**Fan Current PWM High Pulse Width Bits [11:0]**			00h 00h 00h 00h	FEh
		7-0	RO	If the fan is not in automatic mode (**FANCFG**.1 = 0), this byte will be the PWM high period in a PWM cycle. If the fan is in automatic mode (**FANCFG**.1 = 1), read to these registers will get the current PWM high pulse width.		

4.10 GPT Module

4.10.1 GPT Funtional Description

The KB3910 includes four independent General Purpose Timer modules. Each GPT module consists of a 16-bit counter. The clock source of 16-bit counter can be one of the following:

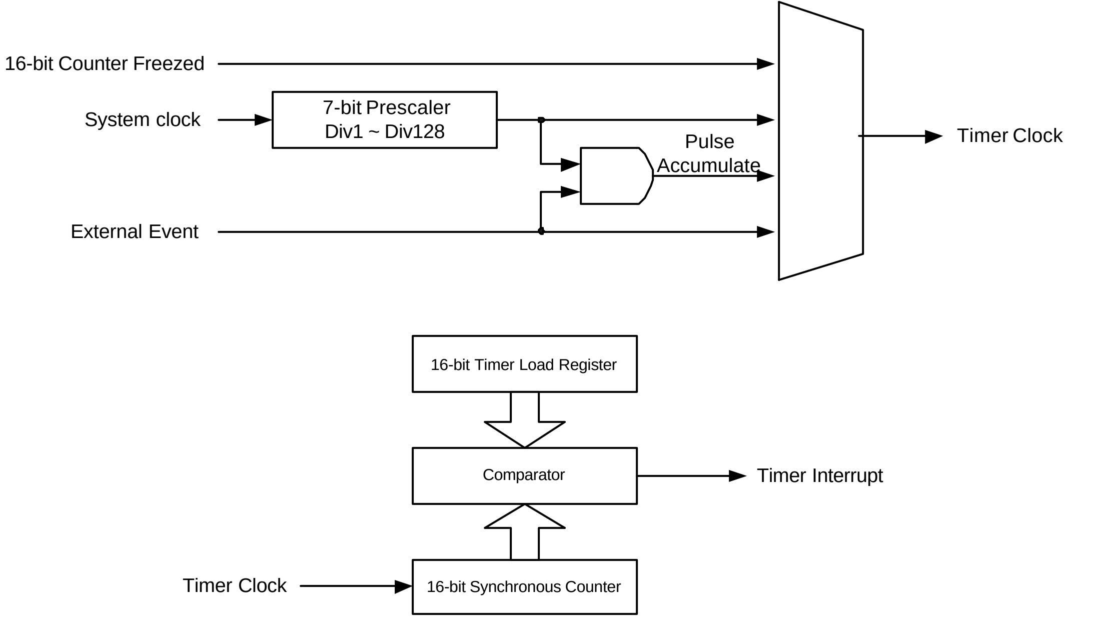
1. Prescaled system clock

(GPT0 and GPT1 share one common prescaler, GPT2 and GPT3 share another common prescaler).

2. External event on ***TIN1*** or ***TIN2***

The timer toggles ***TOUT1*** or ***TOUT2*** pin upon a match GPT0 or GPT2 load register, and the resulting output on ***TOUT1*** or ***TOUT2*** is 50% duty cycle PWM signal. The initial value of ***TOUT1*** and ***TOUT2*** output can be configured as either high or low. Each GPT timer can generate a interrupt flag. If the interrupt enable bit is set, a timer interrupt can be generated as counter count to timer setting.

The following is a functional block diagram of a GPT module.



a GPT Block Diagram

4.10.2 GPT Register Descriptions (Base Address = FE00h, Space = 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
50h	GPT01PRSC	**GPT0/1 Clock Prescaler Register**			000b	FEh
		7	NA	Reserved		
		6-0	R/W	DIV 1:1 to 1:128	00h	FEh
51h	GPT23PRSC	**GPT2/3 Clock Prescaler Register**			000b	FEh
		7	NA	Reserved		
		6-0	R/W	DIV 1:1 to 1:128	00h	FEh
52h	GPT01CLKSEL	**GPT0/1 Clock Selection Register**			00b	FEh
		7-6	NA	Reserved		
		5-3	R/W	GPT0 Clock Source Selector 000: Freeze GPT0 001: Prescaled System Clock 010: External Event on ***TIN1*** 011: Pulse Accumulate Others: Reserved	000b	FEh
		2-0	R/W	GPT1 Clock Source Selector 000: Freeze GPT1 001: Prescaled System Clock 010: External Event on ***TIN1*** 011: Pulse Accumulate Others: Reserved	000b	FEh
53h	GPT23CLKSEL	**GPT2/3Clock Selection Register**			00b	FEh
		7-6	NA	Reserved		
		5-3	R/W	GPT2 Clock Source Selector 000: Freeze GPT2 001: Prescaled System Clock 010: External Event on ***TIN2*** 011: Pulse Accumulate Others: Reserved	000b	FEh
		2-0	R/W	GPT3 Clock Source Selector 000: Freeze GPT3 001: Prescaled System Clock 010: External Event on ***TIN2*** 011: Pulse Accumulate Others: Reserved	000b	FEh
54h	GPT01CSR	**GPT0/1 Control Status Register**			00h	FEh
		7-6	NA	Reserved	00b	
		5	R/W	GPT0 Interrupt Enable	0b	
		4	R/W	GPT1 Interrupt Enable	0b	
		3-2	NA	Reserved	00b	
		1	R/WC1	GPT0 Interrupt Pending Flag (must be cleared by software)	0b	
		0	R/WC1	GPT1 Interrupt Pending Flag (must be cleared by software)	0b	
55h	GPT23CSR	**GPT2/3 Control Status Register**			00h	FEh
		7-6	NA	Reserved	00b	
		5	R/W	GPT2 Interrupt Enable	0b	
		4	R/W	GPT3 Interrupt Enable	0b	
		3-2	NA	Reserved	00b	
		1	R/WC1	GPT2 Interrupt Pending Flag (must be cleared by software)	0b	

		0	R/WC1	GPT3 Interrupt Pending Flag (must be cleared by software)	0b	
56h 57h	**GPT0H** **GPT0L**	**GPT0 Counter Register**			N.D.	FEh
		7 – 0	R/W			
58h 59h	**GPT1H** **GPT1L**	**GPT1 Counter Register**			N.D.	FEh
		7 – 0	R/W			
5Ah 5Bh	**GPT0LDH** **GPT0LDL**	**GPT0 Counter Load Register**			N.D.	FEh
		7 – 0	R/W			
5Ch 5Dh	**GPT1LDH** **GPT1LDL**	**GPT1 Counter Load Register**			N.D.	FEh
		7 – 0	R/W			
5Eh 5Fh	**GPT2H** **GPT2L**	**GPT2 Counter Register**			N.D.	FEh
		7 – 0	R/W			
60h 61h	**GPT3H** **GPT3L**	**GPT3 Counter Register**			N.D.	FEh
		7 – 0	R/W			
62h 63h	**GPT2LDH** **GPT2LDL**	**GPT2 Counter Load Register**			N.D.	FEh
		7 – 0	R/W			
64h 65h	**GPT3LDH** **GPT3LDL**	**GPT3 Counter Load Register**			N.D.	FEh
		7 – 0	R/W			
66h	**GPTCTR**	**GPT Control Register**			N.D.	FEh
		7	R/W	Reserved		
		6	R/W	TOUT1EN: Enable ***TOUT1*** to function as a PWM output.	0b	
		5	R/W	TOUT1POL: Contains the value of ***TOUT1*** output when ***TOUT1*** is used as a PWM output. 0: TOUT1 is low; 1: TOUT1 is High.	0b	
		4	R/W	0: high-to-low on ***TIN1*** or count enabled if ***TIN1*** is low in pulse acc. 1: low -to-high on ***TIN1*** or count enabled if ***TIN1*** is high in pulse acc.	0b	
		3	R/W	Reserved	N.D.	
		2	R/W	TOUT2EN: Enable ***TOUT2*** to function as a PWM output.	0b	
		1	R/W	TOUT2POL: Contains the value of ***TOUT2*** output when ***TOUT2*** is used as a PWM output. 0: ***TOUT2*** is low; 1: ***TOUT2*** is High.	0b	
		0	R/W	0: high-to-low on ***TIN2*** or count enabled if ***TIN2*** is low in pulse acc. 1: low -to-high on ***TIN2*** or count enabled if ***TIN2*** is high in pulse acc.	0b	

4.11 WDT Module

4.11.1 WDT Funtional Description

The watchdog timer generates a watchdog reset upon the expiration of a defined period of time:

(**WDTCNT** + 1) x Tprescaled

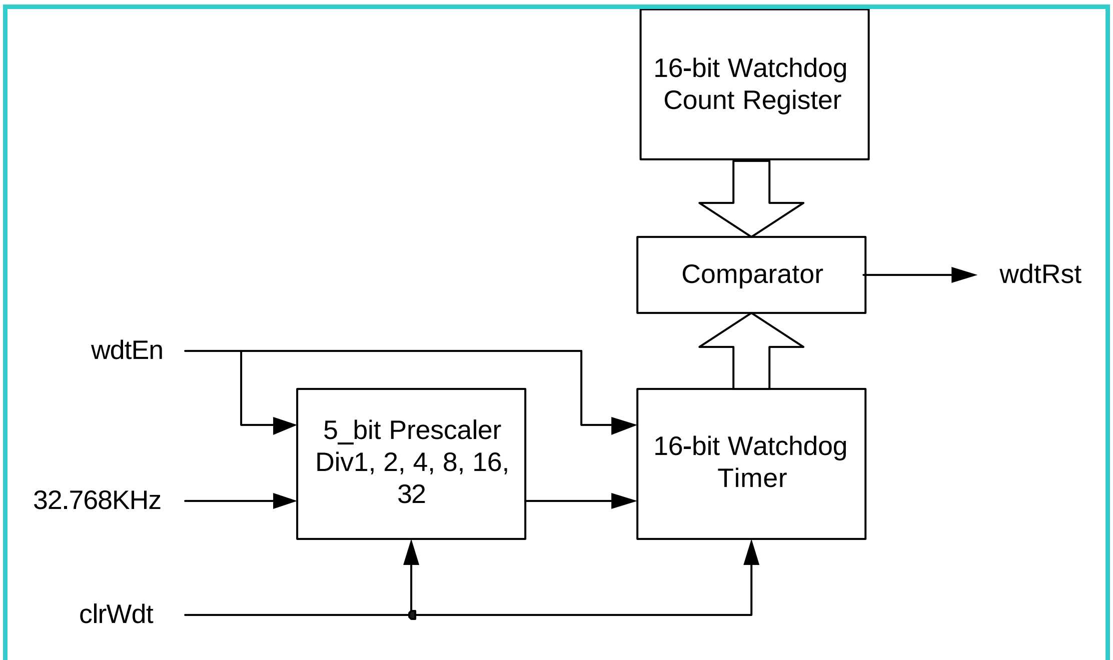
where

Tprescaled is the reciprocal of 32KHz divided by 1, 2, 4, 8, 16, or 32.

The watchdog timer can be programmed to generate a warning flag when the counter is half-way to timeout by settting bit_6 of **WDTCTR (0xFE85h)**. The warning flag can render an interrupt to 8051 or a SCI to inform the system host, or both, depending on register setting in **P0IE (80h in 8051 SFR)** and **ECSCIEN0 (0xFF05h)** respectively.

When the **WDT** timeout event occurs, is an option to reset the 8051 only, or to reset the whole chip. This option can be set in bit_1 of **ECPXCFG 0xFF14).** There is no SCI nor interrupt to 8051 will take place when the WDT times out.

The following is a functional block diagram of the **WDT** module.


16-bit Watchdog Count Register
Comparator
wdtRst
wdtEn
5_bit Prescaler Div1, 2, 4, 8, 16, 32
16-bit Watchdog Timer
32.768KHz
clrWdt

WDT Block Diagram

4.11.2 WDT Register Descriptions (Base Address = FE00h, Space = 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
80h	WDTPRSC	**Watchdog Clock Prescaler Register**			00h	FEh
		7 – 3	NA	Reserved		
		2-0	R/W	DIV bit[2:0] Clock Ratio 000 1:1 001 1:2 010 1:4 011 1:8 100 1:16 101 1:32 others Reserved	000b	FEh
81h	WDTTMRH	**Watchdog Timer Register High Byte**			00h	FEh
		7 – 0	R/O			
82h	WDTTMRL	**Watchdog Timer Register High Byte**			00h	FEh
		7 – 0	R/O			
83h	WDTCNTH	**Watchdog Count Register High Byte**			00h	FEh
		7 - 0	R/W			
84h	WDTCNTL	**Watchdog Count Register Low Byte**			00h	FEh
		7 - 0	R/W			
85h	WDTCTR	**Watchdog Control Register**				FEh
		7	R/W	WDTE: Watchdog Enable Bit 1: Enable 0: Disable. Write "0" disable when **WDTDIS**=0x5Ch	0b	
		6	R/W	TOWE: Enable WDT Timeout Warning flag to generate an 8051 interrupt or SCI, when the SDT is counting halfway to timeut. 1: Enable 0: Disable	0b	FEh
		5-3	R/W	Reserved		FEh
		2	R/W	WDT Clock Source 1 : External Crystal (Take care of Internal oscillator full-swing time) 0 : Internal DPLL circuit	0b	FEh
		1-0	R/W	Reserved		FEh
86h	WDTSTU	**Watchdog Status Register**			00h	FEh
		7-2	NA	Reserved		
		2	RO	WDTWF: WDT Writing flag. 1: indicats a write cycle to WDT is taking place 0: indicats no write cycle is taking place to WDT		
		1	R/WC1	TOWF: WDT Timeout Warning Flag 1: Timeout Warning occur	0b	FEh
		0	RO	**TOZ**: Timeout Bit 0: A Watchdog timeout event occurs **TOZ** can be de-asserted by writing 0x5C to **WDTCLR**	1b	FEh
87h	WDTCLR	**Watchdog Clear Register**			00h	FEh
		7 - 0	R/W	Write the specific value "5Ch" to **WDTCLR** will clear the timer and **TOZ**		
88h	WDTDIS	**Watchdog Disable Register**			00h	FEh
		7 - 0	R/W	Write 5Ch to **WDTDIS**, and then write "0" to bit_7 of **WDTCTR** will disable the watchdog timer.		

4.12 LPC Module

4.12.1 LPC/FWH Funtional Description

There are several address ranges on LPC/FWH interface will be responded by KB3910.

- Keyboard controller I/O ports: 60h, 64h
- Embedded controller I/O ports: 2 programmable I/O ports (default 62h, 66h)
- EC I/O Index and Data Ports: Through which the system host can access KB3910 internal registers more efficiently than through EC commands F0h/F1h. The EC I/O Index and Data Ports are two 8-bit registers with base address defined in FE92h and FE93h. Default Index Port ={002Dh, 002Eh}, Data port =002Fh.
- LPC/FWH memory access

LPC Decoding IO Ports

The keyboar I/O ports are 60h/64h, while the EC I/O ports are programmable by KBC Command 61h, and are default to 62h/66h. The enable/disable of I/O ports decoding on LPC bus can be configured individually via register **LFEN**(FE90h).

LPC Decoding Memory Space

There are two memory address segments that are decoded by KB3910.

1.000Y_0000h ~ 000F_FFFFh, where Y is selectable to C/E/Fh in register **LFMSM**(FE91h)

2.FFXX_0000h~FFFF_FFFFh, where XX is selectable to FE/FC/F8/F0/E0/C0/80h in register **LFMSM**(FE91h).

The legal values of **LFMSM** are listed below.

Table for LPC memory decoding space

Register FE91h Bit [7:6]	Segment 1 Size	Decoded Address
00	64k	000F_0000 – 000F_FFFF
01	128k	000E_0000 – 000F_FFFF
11	256k	000C_0000 – 000F_FFFF

Register FE91h Bit [5:0]	Segment 2 Size	Decoded Address
000000	128k	FFFE_0000 – FFFF_FFFF
000001	256k	FFFC_0000 – FFFF_FFFF
000011	512k	FFF8_0000 – FFFF_FFFF
000111	1M	FFF0_0000 – FFFF_FFFF
001111	2M	FFE0_0000 – FFFF_FFFF
011111	4M	FFC0_0000 – FFFF_FFFF
111111	8M	FF80_0000 – FFFF_FFFF

4.12.2 LPC Register Descriptions (Base Address = FE00h, Space = 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
90h	LFEN	**LPC/FWH Decoder Enable**			2Xh	FEh
		7	RSV	Reserved	0	
		6	R/W	Enable EC Index/Data IO Port Address Decoding on LPC bus. There are 3 EC Index IOPorts – 2 Index address ports and a Data IO port. The IO base address defined in [FE92h: FE93h]. Once enabled, KB3910 will respond to I/O cycles on the LPC bus with address appeared to be the EC Index address Port and EC index Data port. For example: If FE92h=00h, FE93h=2Ch, then EC Index address IO Port high = 002Dh, EC Index address IO Port low = 002Eh, EC index Data IO Port=002Fh 1: Enable 0: Disable	0b	FEh
		5	R/W	Enable decode KBC I/O port: 60h, 64h	1	
		4	R/W	Reserved. Must be written a “0”	0	
		3	R/W	Enable LPC decode range1 (000X0000h-000FFFFFh)	Emwb	
		2	R/W	Enable LPC decode range2 (FFXX0000h-FFFFFFFFh)	Emwb	
		1	R/W	Enable FWH decode range1 (00X0000h-00FFFFFh)	Emwb	
		0	R/W	Enable FWH decode range2 (FXX0000h- FFFFFFFh)	Emwb	
91h	LFMSM	**LPC/FWH Memory Size Mask**			7Fh	FEh
		7 – 6	R/W	Memory Range1 Size = (value+1)*64k		
		5 – 0	R/W	Memory Range2 Size= (value+1)*128K		
92h 93h	LFECBH LFECBL	**EC Index mode IO port base address**			00h 2Ch	FEh
		7 – 0	R/W	EC index mode IO port base address. The Register FE90h Bit 6 should be enabled for decoding the set IO port.		
94h	LFWHID	**FWH ID select & Check enable**			X0h	FEh
		7 – 5	RSV	Reserved	0h	
		4	R/W	Disable check FWH IDSEL	emwb	
		3 – 0	R/W	FWH ID select register	0h	
95h	LFCSM	**Control Status of Memory access from LPC/FWH to XBI**			X0h	FEh
		7	R/W	Enable the redirecting of the address sent from LPC/FWH to XBI interface 1: Enable (recommended) 0: Disable	1b	
		6	R/W	This bit is used to block the LPC memory R/W cycles sent from system host to the X-bus. 1: Wait-states will be inserted indefinitely for LPC memory cycles toward X-bus. 0: LPC memory cycles will be accomplished normally (Recommended) The initial value of this bit is the HW strap signal **DMRP** being inverted.	DMRP inverted	
		5 – 0	RSV	Reserved.	0h	
96h	LFRMA	**LPC/FWH Redirect Memory Access to XB**			00h	FEh
		7	R/W	Enable bit used to re-map the address of LPC Memory cycle to different memory locations on XBI by replacing LPC address A[20:17] to bits [4:1] of this register		

		4 – 1	R/W	XBI address [20:17] used to replace LPC address A[20:17]		
		0	RSV	Reserved.		
97h	**SIRQEN**	**Serial IRQ Enable**			0Ch	FEh
		7 – 4	RSV	Reserved.		
		3	R/W	Enable IRQ1 for keyboard via serial IRQ		
		2	R/W	Enable IRQ12 for auxiliary device via serial IRQ		
		1	R/W	Reserved. Must be 0.		
		0	RSV	Enable SCI via Serial IRQ		
98h	**SIRQRCFG**	**Serial IRQ Routing Configuration**			00h	FEh
		7 – 4	RSV	Reserved		
		3 – 0	R/W	SCI routing table 01: IRQ1 02: IRQ2 … 15: IRQ15		
99~9Ah		Reserved				
9Bh	**LFCFG**	**LPC/FWH Configuration**			25h	FEh
		7	RSV	Reserved.		
		6	R/W	Enable LPC CLOCK RUN.		
		5-4	R/W	Input Enable of general LPC pins. 1X: Controlled by lpc reset#. If lpc reset# =1, input is enabled. 01: Input Enable 00:Input disabled		
		3-2	R/W	Input Enable of lpc clock .1x: Controlled by lpc reset#. If lpc reset#=1, input is enabled. 01: input enable. 00: input disable.		
		1	R/W	LPC reset input is enabled. User shoudld set GPIO2C registers to control the input enable and set this register.		
		0	R/W	Always keep ***LCLK*** clock via ***CLKRUN#***.		
9C~9Fh		Reserved				

4.13 XBI Module

4.13.1 XBI Funtional Description

The X-BUS Interface (XBI) module is used to access ISA-like external memory devices on the X-Bus. The memory space of 8051 is 64KB, divided into four 16KB segments: 0000h-3FFFh (SEG0), 4000h-7FFFh (SEG1), 8000h-BFFFh(SEG2) and C000h-FFFFh (SEG3). Each of which can be mapped to different memory spaces by concatenating ***A[20:14]*** defined in registers: **XBISA0** – **XBISA3**. The resulting address space area is as follows:

8051 Address Space 16K Segment	XBI Address A[20:0]
0000h-3FFFh (SEG0)	***A[20:14]***= **XBISA0**[6:0], ***A[13:0]***=8051 address[13:0]
4000h-7FFFh (SEG1)	***A[20:14]***= **XBISA1**[6:0], ***A[13:0]***=8051 address[13:0]
8000h-BFFFh (SEG2)	***A[20:14]***= **XBISA2**[6:0], ***A[13:0]***=8051 address[13:0]
C000h-FFFFh (SEG3)	***A[20:14]***= **XBISA3**[6:0], ***A[13:0]***=8051 address[13:0]

There are two possible memory ranges decoded by KB3910:
Range1: 000X0000h to 000FFFFFh
Range2: FFXX0000h to FFFFFFFFh

There are totally 4 possible settings supported: LPC/Range1, FWH/Range1, LPC/Range2, FWH/Range2, and can be enabled/disabled individually by setting registers **LPC_M1E, LPC_M2E, FWH_M1E, FWH_M2E**. After a system reset, the values of **LPC_M1E, LPCM2E, FWH_M1E** and **FWH_M2E** are defined by an internal signal called **EMWB,** which comes from hardware strap pin (***A5***). If **EMWB** is High during system reset, all memory regions are enabled initially.

The ***E51CS#*** is used to select memory device for debug purpose. When a particular memory space is being configured to map into debug-memory, ***E51CS#*** will be asserted instead of ***MEMCS#***..Register **XBIE51CS** (FEA7h) controls the mapping of the four segments SEG3-0 memory spaces individually into debug-memory space. The assertion of ***E51CS#*** and ***MEMCS#*** is mutual exclusive.

4.13.2 XBI Register Descriptions (Base Address = FE00h, Space = 48 bytes)

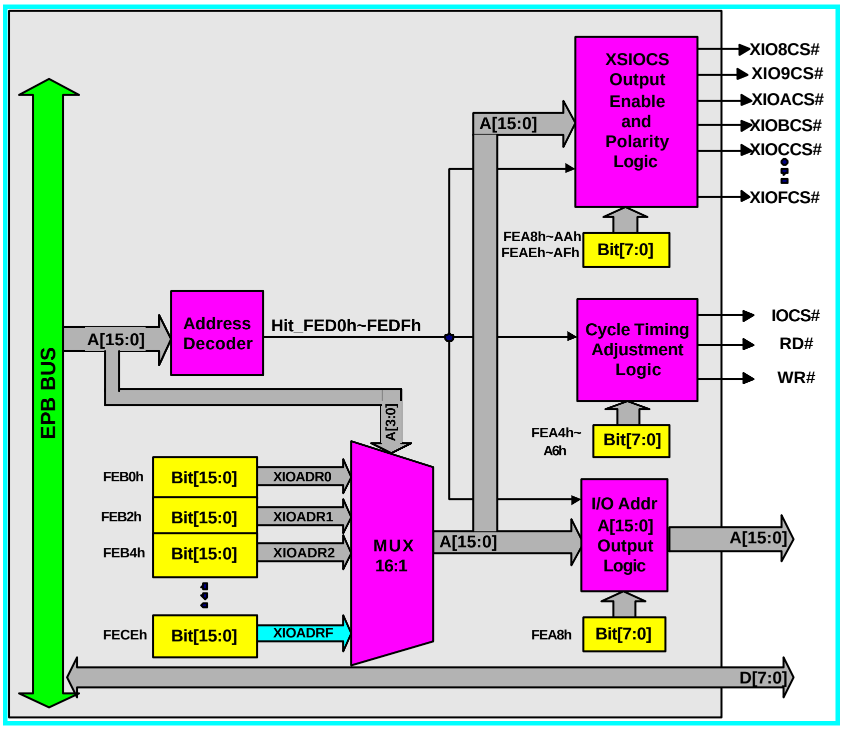
Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
A0h	XBISA0	**Segment address of XBI for 8051 address 0000h-3FFFh**			00h	FEh
		7	R/W	Reserved		
		6-0	R/W	XBI address = XBI_SA0*16k + 8051 address [13:0]		
A1h	XBISA1	**Segment address of XBI for 8051 address 4000h-7FFFh**			01h	FEh
		7	R/W	Reserved		
		6-0	R/W	XBI address = XBI_SA1*16k + 8051 address [13:0]		
A2h	XBISA2	**Segment address of XBI for 8051 address 8000h-BFFFh**			02h	FEh
		7	R/W	Reserved		
		6-0	R/W	XBI address = XBI_SA2*16k + 8051 address [13:0]		
A3h	XBISA3	**Segment address of XBI for 8051 address C000h-FFFFh**			03h	FEh
		7	R/W	Reserved		
		6-0	R/W	XBI address = XBI_SA3*16k + 8051 address [13:0]		
A4h	XBIIOCFG	**XBI Interface I/O Cycle Timing Configuration**			62h	FEh
		7-6	R/W	Setup time in unit of XBI clock		
		5-2	R/W	Command time in unit of XBI clock		
		1-0	R/W	Hold time in unit of XBI clock		
A5h	XBIRMCFG	**XBI Interface Memory Read Cycle Timing Configuration**			F8h	FEh
		7	R/W	Read cycle setup time in unit of XBI clock		
		6-4	R/W	Read cycle command time in unit of XBI clock		
		3	R/W	Read Hold time in unit of XBI clock		
		2	R/W	Enable the extension of active state of ***RD#*** in setup-time phase and hold-time phase. If disabled, RD# will only be asserted during command-time phase.		
		1-0	R/W	Define the length of idle-time to be appended at the end of memory cycle. 00: No idle time 01: one clock 10: two clocks 11: three clocks		
A6h	XBIWMCFG	**XBI Interface Memory Write Cycle Timing Configuration**			C8h	FEh
		7	R/W	Write cycle setup time in unit of XBI clock		
		6-4	R/W	Write cycle command time in unit of XBI clock		
		3	R/W	Write Hold time in unit of XBI clock		
		2	R/W	Enable the extension of active state of WR# to setup-time phase and hold-time phase. If disabled, WR# will only be asserted during command-time phase.		
		1-0	R/W	Define the length of idle-time to be appended at the end of memory cycle. 00: No idle time 01: one clock 10: two clocks 11: three clocks		
A7h	XBIE51CS	**XBI E51CS# Configuration**			00h	FEh
		7-4	R/W	Reserved		
		3-0	R/W	Enable ***E51CS#*** instead of ***MEMCS#*** for 8051 accessing memory space segment 3-0		
A8h	XIOADREN	**XIO Redirect Address to XBI Enable**			00h	FEh
		7	R/W	Map XIO_AHx to ***XIOCSF~8*** for IO Expander[F:C]		

		6	RSV	Reserved		
		5	R/W	Map XIO_AHx to ***XIOCSF~8*** for IO Expander [B:8]		
		4	RSV	Reserved		
		3	R/W	Map XIO_AHx to ***XIOCSF~8*** for IO Expander [7:4]		
		2	RSV	Reserved		
		1	R/W	Map XIO_AHx to ***XIOCSF~8*** for IO Expander [3:0]		
		0	RSV	Reserved		
A9h	**XIOCSH**	**Active level for XIOCS[F:8]**			00h	FEh
		7-0	R/W	Active level for ***XIOCS[F:8]***		
AAh		Reserved			00h	FEh
ABh	**XBIE51CFG**	**XBI 8051 Interface Configuration**			18h	FEh
		7	RSV	Reserved		
		6	R/W	Enable 8051 instruction prefetching		
		3	R/W	=1, XBI defaut accessed by LPC memory cycles. =0, XBI defaut accessing by 8051 or EC.		
		2	R/W	XIO enable. If not set this bit, The accessing to XIO space will not get any respone on XBI to external X-BUS.		
		1-0	RSV	Reserved		
ACh	**XIOCFG**	**XIO Configuration**			00h	FEh
		7-6	R/W	Define the length of idle-time to be appended at the tail of I/O cycle 00: No idle time 01: one clock 10: two clocks 11: three clocks		
		5	R/W	Enable the extension of active state of ***WR#***, ***RD#*** to setup-time phase and hold-time phase for I/O cycle. If disabled, ***WR# RD#*** will only be asserted during command-time phase for IO cycle		
		4	R/W	Enable the extension of active state of ***IOCS#*** to setup-time phase and hold-time phase for I/O cycle. If disabled, ***IOCS#*** will only be asserted during command-time phase for IO cycle.		
		3	R/W	Enable the extension of active state of ***MEMCS#*** to setup-time phase and hold-time phase for memory read cycles. If disabled, ***MEMCS#*** will only be asserted during command-time phase for memory read cycles.		
		2	R/W	Enable the extension of active state of ***MEMCS#*** to setup-time phase and hold-time phase for memory write cycles. If disabled, ***MEMCS#*** will only be asserted during command-time phase for memory write cycles.		
		1-0	RSV	Reserved		
ADh		Reserved			00h	FEh
AEh	**XIOEXSH0**	**Extend XIOCS[F:8] setup-time and hold-time**			00h	FEh
		7-0	R/W	Enable the extension timing of ***XIOCS[f:8]***		
AFh		Reserved			00h	FEh
B0-BFh		Reserved			00h	FEh
C0h C1h	**XIOAH8** **XIOAL8**	**XBI Address Bus Output when XIO8 port is being accessed**			01h 00h	FEh
		7-0	W/R	Output to external ***A15:8]*** upon XIO8 port accessed Output to external ***A[7:0]*** upon XIO8 port accessed		
C2h C3h	**XIOAH9** **XIOAL9**	**XBI Address Bus Output when XIO9 port is being accessed**			02h 00h	FEh
		7-0	W/R	Output to external ***A[15:8]*** upon XIO9 port accessed Output to external ***A[7:0:]*** upon XIO9 port accessed		
C4h C5h	**XIOAHA** **XIOALA**	**XBI Address Bus Output when XIOA port is being accessed**			04h 00h	FEh
		7-0	W/R	Output to external ***A[15:8]*** upon XIOA port accessed Output to external ***A[7:0]*** upon XIOA port accessed		

C6h C7h	**XIOAHB** **XIOALB**	**XBI Address Bus Output when XIOB port is being accessed**			08h 00h	FEh
		7-0	W/R	Output to external ***A[15:8]*** upon XIOB port accessed Output to external ***A[7:0:]*** upon XIOB port accessed		
C8h C9h	**XIOAHC** **XIOALC**	**XBI Address Bus Output when XIOC port is being accessed**			10h 00h	FEh
		7-0	W/R	Output to external ***A[15:8]*** upon XIOC port accessed Output to external ***A[7:0:]*** upon XIOC port accessed		
CAh CBh	**XIOAHD** **XIOALD**	**XBI Address Bus Output when XIOD port is being accessed**			20h 00h	FEh
		7-0	W/R	Output to external ***A[15:8]*** upon XIOD port accessed Output to external ***A[7:0:]*** upon XIOD port accessed		
CCh CDh	**XIOAHE** **XIOALE**	**XBI Address Bus Output when XIOE port is being accessed**			40h 00h	FEh
		7-0	W/R	Output to external ***A[15:8]*** upon XIOE port accessed Output to external ***A[7:0:]*** upon XIOE port accessed		
CEh CFh	**XIOAHF** **XIOALF**	**XBI Address Bus Output when XIOF port is being accessed**			80h 00h	FEh
		7-0	W/R	Output to external ***A[15:8]*** upon XIOF port accessed Output to external ***A[7:0:]*** upon XIOF port accessed		

4.14 XIO Module

4.14.1 XIO Funtional Description



The **XIO** module interfaces to external IO devices via the XBI interface with programmable addresses defined in registers **XIO_A8~F**(FEC0h~CFh). For example, assuming **XIO_A8** contains “1060h”, when there is a write cycle toward the XIO8 port(FED8h), an IO write cycle will be taking place on XBI bus with ***A[15:0]***=“1060h”, and the host data appearing on ***D[7:0].***

XIOCS[F:8] can be connected directly to the ***CS#*** pins of external IO devices, by reflecting the content of **XIO_A[F:8]** registers which is being addressed. For example, suppose:

FEC0h(**XIOAH8**)=12h; ; FEA8h(**XIOCFG**)=02h

When there is a read or write cycle to XIO ports FED8h, then ***XIOCS[F:8]***=12h

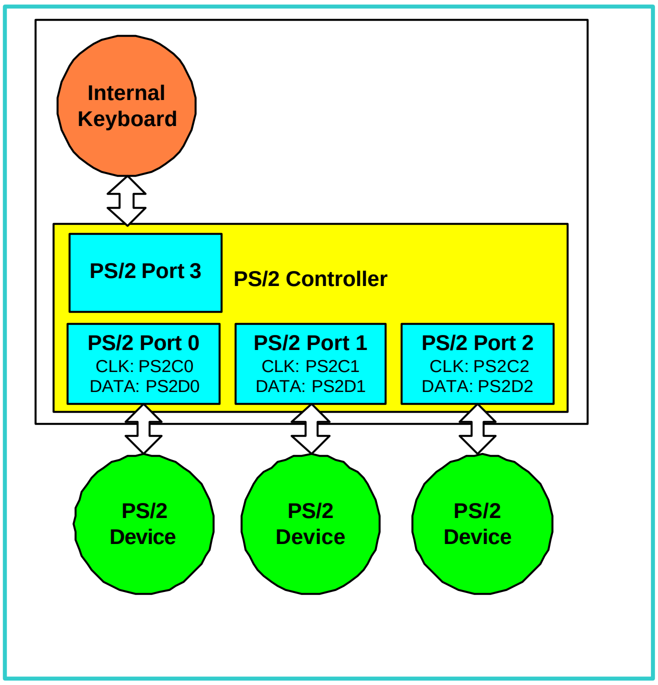
4.14.2 XIO Register Descriptions (Base Address = FE00h, Space = 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
D0-D7h		Reserved			00h	FEh
D8~DFh	XIO8~F	**XIO Read / Write Port**			XXh	FEh
		7-0	W/R	Read / Write to these ports will cause an I/O cycle to take place on the XBI bus, with address ***A[15:0]*** being the same as the corresponding **XIO_A[F:8]** register bits [15:0]. Data transfer is through ***D[7:0]***. Alternatively, read/write to these ports will cause the corresponding **XIO_AH[F:8]** register bits [7:0] to appear on the ***XIOCSF~8*** pins. This function is enabled for individual XIO ports by setting the proper register bits in **XIOCFG** register.		

4.15 PS2 Module

4.15.1 PS2 Funtional Description

The KB3910 PS/2 Controller is just like the KBC module, and is inherited from the KB3886 hardware PS/2 Controller.



KB3910 provides three PS/2 ports, supporting keyboard, pointing devices and other PS/2 compliant devices.

PS2 Host Modes

There are three modes to control PS/2 host.

- Bit Mode
- Full Hardware Mode

The selection of each mode is by setting **PS2MODE** (FEE0h).

Bit Mode

In this mode, the input and output signals of PS/2 ports are controlled by firmware via register **PS2PIN** and **PS2POUT** respectively. The 8051 firmware should manipulate PS/2 data in bit-oriented data structure.

KB3910 supports interrupt from PS/2 hardware for firmware interrupt driving PS/2 host controller. The conditions trigger interrupt are depended in the register **PS2INTC** (FEE5h). The firmware can get interrupt source flag from register **PS2INTS** (FEE2h).

Full Hardware Mode

In this mode, the PS/2 host controller is fully hardware implemented and doesn't require any intervention from firmware. Each PS/2 port can be attached to either a keyboard device or an auxiliary device (mouse). KB3910 also supports hot-plug, hot-removal and hot-switch features for each port. The following sections describe each of these features.

Hot-Plug

This feature allows PS/2 devices to be plugged-in when system is operational by automatically detect device type upon plugging-in and initializes it. There is no need to reboot the system in order to recognize the new device. This feature is default enabled, and can be disabled by setting the **PS2AUXST7** register (FEF0h) bit-6. Of course, devices plugged-in before system is power on will be successfully initialized.

Hot-Removal

This feature allows PS/2 devices to be removed when system is already power-on. KB3910 will automatically detect the device removal event by polling devices and clear the related port status dynamically. System will run normally without being affected by the removal of PS/2 device.

Hot-Swap

This feature allows the swapping of PS2 devices of different types (keyboard or mouse) to take place on the same PS/2 port without affecting system's normal operation. KB3910 will automatically detect the removal/plug-in of PS/2 device and initialize it.

Operation Modes for Auxiliary Devices

There is only one auxiliary device will be chosen as "*enabled*" if multiple auxiliary devices are plugged-in, and all other devices are "*disabled*". The *"enabled"* device will be operational while other *"disabled"* devices are not functional. Two algorithms can be used to decide the *"enabled"* auxiliary device: Hot-Switch mode and Cold-Switch mode. The selection can be done by setting register **PS2AUXST7** bits [4:3].

Hot-Switch Mode

In this mode, the priority of PS/2 auxiliary devices chosen as *"enabled"* will be in the precedence order of their being plugged-in while system is power-on. Last plugged-in device has higher priority than earlier ones. When the current "enabled" device is removed from the system, the next device with highest priority will be chosen as "enabled".

Cold-Switch Mode

In the Cold-Switch mode, the *"enabled"* PS/2 device will be determined while the PS/2 controller is reset. The algorithm to determine the priority order of PS/2 devices being chosen as "enabled" will be determined by the precedence order of their corresponding PS/2 port numbers. Port 0 has highest priority while Port 2 has lowest (Port 3 is the embedded IKB and is always enabled). After an *"enabled"* device is determined, there will be no dynamic change to another device to take place unless the PS/2 controller is reset. This suggests that:

1. A new device being hot-plugged into the system will not be made "enabled" as opposed to the case in Hot-switch mode.
2. If the current "enabled" PS/2 device is being hot-removed, there will be no re-election of "enabled" device.

Subject to the implementation of system hardware circuit, a system warm boot may not reset the PS/2 controller. If this is the case then the PS/2 controller can be reset by writing to register **PS2WRST**. After the PS/2 controller is reset, the new enabled PS/2 device will be chosen according to its priority order.

The Cold-Switch mode is particularly useful when connecting to some special auxiliary devices which cannot be initialized by the standard PS/2 initialization sequence performed by KB3910. These special auxiliary devices are often not completely compatible to standard PS/2 devices and will cause problems if operated under the Hot-Switch mode. In case the auxiliary device is so incompatible to standard PS/2 auxiliary device such that it can not receive polling command from KB3910, the polling feature may be disabled by setting register **PS2SET** Bit_0.

Microsoft Intelli-mode Mouse

All Intelli-mode mouse devices followed this specification are supported including 3-button standard mouse, 3-button/wheel mouse and 5-button/wheel mouse.

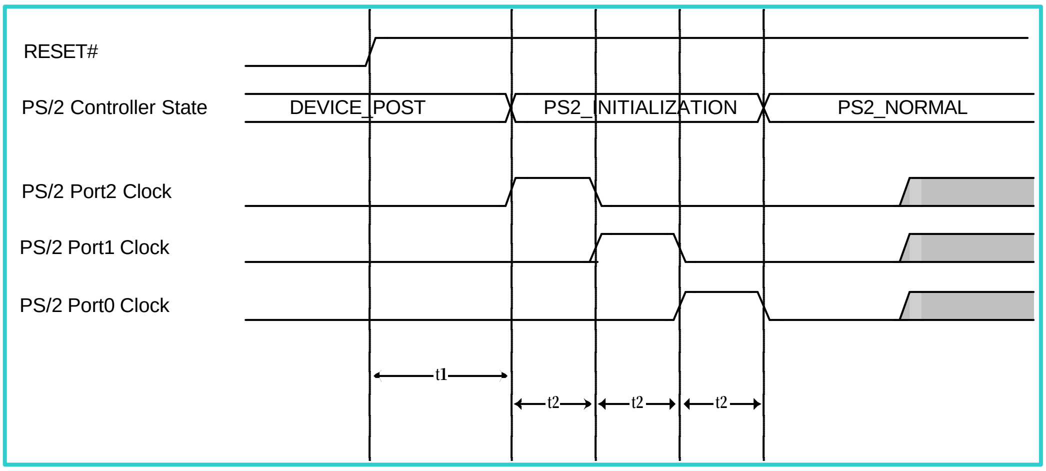
Active PS/2 Multiplexing

The Active PS/2 Multiplexing mechanism is implemented by using bit7/6 of the **Status** (port_64h) bits[7:6] to indicate the source port number. The mapping table is listed as follows.

	CLOCK PIN	DATA PIN	Multiplexed Port Number
PS/2 Port 0	PS2C0	PS2D0	1
PS/2 Port 1	PS2C1	PS2D1	2
PS/2 Port 2	PS2C2	PS2D2	3

PS2 Port Behavior During Chip Initialization Phase

After system is power-on and reset, KB3910 will be in initialization phase. In this phase, KB3910 will go to DEVICE-POST state, during which state the PS/2 devices perform POST (Power-On Self Test) by themselves. After DEVICE-POST state, KB3910 will go to PS/2-INITIALIZATION state, during which state the KB3910 will detect and initialize PS/2 devices that are already plugged-in. Finally, KB3910 will go to PS/2-NORMAL state, during which state KB3910 will be in operational state. The waveforms are demonstrated as follows.



During DEVICE-POST or PS/2-INITIALIZATION states, system initiated commands to PS/2 devices will not be forwarded to the PS/2 device until PS/2-NORMAL state. The time-period of DEVICE-POST, t1, is programmable in register **PS2CFG** Bit_4 (0: 0.5 second, 1: 1 second). The time-period of PS/2-INITIALIZATION for each port, t2, is 28ms.

PS2 Port Behavior During Hot-plug

The Hot-Plug event means a PS/2 device is being plugged into the system when system is power on. KB3910 behaves differently to Hot-plug event under legacy mode and multiplexed mode. In legacy mode (i.e. Non-multiplexed mode), KB3910 will automatically detect and initialize the new PS/2 device without informing the system, i.e., the system will not know a new PS/2 device is being plugged-in. In multiplexed mode, KB3910 only detects device type but doesn't initialize the device. It will forward device data to the system and the device will be initialized by the system.

KB3910 STOP Mode and Wake Up

If the PS/2 controller is selected to reset while the KBC is in STOP mode, the KBC cannot be waken up by PS/2 devices. The PS/2 wake-up enable bits defined in register **PS2WAKE** bits[2:0] are only effective when register **PS2WAKE** bit_7=1 and bit_6=0.

Comments For Sending PS/2 Device Command

To avoid the collision between device command and device data, it is recommended that before device commands are issued to the AUX devices, an AUX disable command (Write A7h to Port 64h) should be sent to KB3910 first. The same recommendation is also applicable to keyboard device.

Internal AUX Device Recommendation

Internal AUX devices are those built-in devices in a notebook, such as a touch pad. Typically there is at least one required. It is recommended that the internal AUX device should be connected to PS/2 Port 2. If there are two internal AUX devices, they should be connected to PS/2 Port2 and then Port1.

4.15.2 PS2 Register Descriptions (Base Address = FE00h, Space = 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
E0h	PS2MODE	PS/2 Mode Configuration			08h	FEh
		7-4	R/W	PS/2 Control Mode bit6: Hardware Mode Enable bit5: Byte mode Enable bit4: Bit Mode Enable		
		3	R/W	PS/2 Port internal Pull-up Enable		
		2-0	RSV	Reserved		
E1h	PS2WRST	PS/2 Write Reset Register			00h	FEh
		7-0	W	Write to this register will reset PS/2 Controller		
E2h	PS2INTS	PS/2 Interrupt Source			00h	FEh
		7-5	RSV	Reserved		
		4	RO	PS/2 Wakep interrupt		
		3	RO	PS/2 data rising edge trigger interrupt		
		2	RO	PS/2 clock rising edge trigger interrupt		
		1	RO	PS/2 data falling edge trigger interrupt		
		0	RO	PS/2 clock falling edge trigger interrupt		
E3h			NA	Reserved	00h	FEh
E4h	PS2WPIN	PS/2 Bit Mode Wakeup Port Number			00h	FEh
		7-4	RSV	Reserved		
		3-0	R/W	Wakeup Port Number		
E5h	PS2INTC	PS/2 Interrupt Control			00h	FEh
		7-4	RSV	Reserved		
		3	R/W	Enable PS/2 data rising edge trigger interrupt		
		2	R/W	Enable PS/2 clock rising edge trigger interrupt		
		1	R/W	Enable PS/2 data falling edge trigger interrupt		
		0	R/W	Enable PS/2 clock falling edge trigger interrupt		
E6~E7h			NA	Reserved	00h	FEh
E8h	PS2POUT	Output value of PS/2 Port			FFh	FEh
		7	R/W	Output value of PS/2 Port3’ s data pin		
		6	R/W	Output value of PS/2 Port3’ s clock pin		
		5	R/W	Output value of PS/2 Port2’ s data pin		
		4	R/W	Output value of PS/2 Port2’ s clock pin		
		3	R/W	Output value of PS/2 Port1’ s data pin		
		2	R/W	Output value of PS/2 Port1’ s clock pin		
		1	R/W	Output value of PS/2 Port0’ s data pin		
		0	R/W	Output value of PS/2 Port0’s clock pin		
E9h	PS2PIN	Input value of PS/2 Port			FFh	FEh
		7	RO	Input value of PS/2 Port3’ s data pin		
		6	RO	Input value of PS/2 Port3’ s clock pin		
		5	RO	Input value of PS/2 Port2’ s data pin		
		4	RO	Input value of PS/2 Port2’ s clock pin		
		3	RO	Input value of PS/2 Port1’ s data pin		
		2	RO	Input value of PS/2 Port1’ s clock pin		
		1	RO	Input value of PS/2 Port0’ s data pin		
		0	RO	Input value of PS/2 Port0’ s clock pin		
EA~EBh			NA	Reserved	00h	FEh
ECh	PS2FPIN	PS/2 Pin Control Regisgter			00h	FEh
		7	RSV	Reserved		
		6	RW	PS/2 ResetZ (0: reset PS/2, 1: no reset)		
		5	RW	Floating PS/2 Ports		
		4	RW	Keyboard Device Type for Scan Code Conversion 0: standard, 1: PS/55		

		3	RW	Force PS/2 Port3 Clock to be LOW. This bit is only effective when PS2 control mode is Byte mode or Bit modeMODE. Refer to **PS2MODE** register (FEE0h)		
		2	RW	Force PS/2 Port2 Clock to be LOW. This bit is only effective when PS2 control mode is Byte mode or Bit modeMODE. Refer to **PS2MODE** register (FEE0h)		
		1	RW	Force PS/2 Port1 Clock to be LOW. This bit is only effective when PS2 control mode is Byte mode or Bit modeMODE. Refer to **PS2MODE** register (FEE0h)		
		0	RW	Force PS/2 Port0 Clock to be LOW. This bit is only effective when PS2 control mode is Byte mode or Bit modeMODE. Refer to **PS2MODE** register (FEE0h)		
ED~EEh			NA	Reserved	00h	FEh
EFh	**PS2CODE**	**Scan Code Conversion Register**			00h	FEh
		7:0	RW	Write this port with the Keyboard Device Scan Code. Read this port to get the converted code.		
F0h	**PS2AUXST7**	**Keyboard Controller AUX State**			13h	FEh
		7	RSV	Reserved		
		6	R/W	Disable hot plug ability		
		5-4	RSV	Reserved		
		3	R/W	0: Hot switch mode 1: Alternative mode (Colde Switch)		
		2	R/WC1	Clear Auxiliary bit3 Write 1 for set the bit to 0, Write 0 for set the bit to 1		
		1	R/W	Disable internal auxiliary bit 3		
		0	R/W	Disable external auxiliary bit 3		
F1h	**PS2WAKE**	**PS/2 Ports Wake-Up Control**			C0h	FEh
		7	R/w	Reset PS/2 controller when STOP mode		
		6	R/w	Flush PS/2 controller when STOP mode.		
		5-3	RSV	Reserved		
		2	R/w	Internal auxiliary device wake up enable		
		1	R/w	External auxiliary device wake up enable		
		0	R/w	External keyboard wake up enable		
F2h	**PS2SET**	**PS/2 Behavior Setting**			00h	FEh
		7-5	R/w	Keyboard-attached port under detect-disable mode		
		4-2	R/w	Auxiliary-attached port under detect-disable mode		
		1	R/w	Enable detect-disable mode		
		0	R/W	Disable polling feature		
F3h	**PS2CFG**	**PS/2 Ports Configuration**				FEh
		7	R/w	Reserved		
		6	R/W	Internal keyboard is primary device.		
		5	R/W	Polling frequency option 0: 2 seconds 1: 1 seconds		
		4	R/W	PS/2 device POST time option 0: 0.5second 1: 1 second		
		3	R/W	Enable Packet Time-out Patch		
		2	R/W	Long packet time 0: 128ms for data packet / 1s for device reset 1: 372ms for data packet / 1s for device reset		
		1	R/W	Enable packet timer		
		0	R/W	Enable polling empty ports .		
F4h	**PS2KBPOL**	**PS/2 Keyboard Polling Command**			10h	FEh

		7-0	R/W	PS/2 polling command for keyboard devices		
F5h	**PS2AUXPOL**	**PS/2 Auxiliary Polling Command**			F1h	FEh
		7-0	R/W	PS/2 polling command for auxiliary devices		
F6h	**PS2PATCH**	**PS/2 Patched Data for Packet Time Out**			00h	FEh
		7-0	R/W	PS/2 Packet Time Out Patched Data		
F7h	**PS2OPTION**	**PS/2 OPTIONS**			04h	FEh
		7-6	R/W	Force setting transparent mode of auxiliary device data 11: Force to enable transparent mode of auxiliary device data 10: Force to disable transparent mode of auxiliary device data 0x: Disable force setting for transparent mode of auxiliary device data		
		5	R/W	Enable transparent mode of auxiliary device data If only one auxiliary device is connected		
		04	R/W	Enable transparent mode of auxiliary device data If non-standard auxiliary device ID is read.		
		3	R/W	Disable transparent mode of auxiliary device data under alternative mode.		
		2	R/W	Disable wait for full data packet		
		1	R/W	Disable data phase time out.		
		0	R/W	Convert kscan code 00 to 00		
F8h	**PS2AUXFLG 1**	**Auxiliary Devices Flag 1**			02h	FEh
		7-6	NA	Reserved		
		5	RO	Auxiliary device is enabled		
		4	RO	Wrap-mode is enabled		
		3	RO	Remote data mode is enabled		
		2	RO	Data scaling		
		1-0	RO	Data resolution		
F9h	**PS2AUXFLG 2**	**Auxiliary Devices Flag 2**			64h	FEh
		7-0	RO	0Ah: 10 Packets/second 14h: 20 Packets/second 28h: 40 Packets/second 3Ch: 60 Packets/second 50h: 80 Packets/second 64h: 100 Packets/second C8h: 200 Packets/second		
FAh	**PS2AUXINF**	**PS/2 Auxiliary Devices Information**			18h	FEh
		7-5	RO	PS/2 POST state		
		4-3	RO	Primary auxiliary port 11: No auxiliary port Others: Primary auxiliary port number		
		2	RO	Legacy/Multiplexing mode for auxiliary devices		
		1-0	RO	Intelligent mouse mode 00: Standard mode 01: 3-button Z-mode 10: 5-button Z-mode 11: Reserved		
FBh	**PS2PINF**	**PS/2 Port Information**			80h	FEh
		7	RO	Reserved		
		6	RO	Keyboard attached on Port2		
		5	RO	Keyboard attached on Port1		
		4	RO	Keyboard attached on Port0		
		3	RO	Reserved		
		2	RO	AUX Device attached on Port2		
		1	RO	AUX Device attached on Port1		
		0	RO	AUX Device attached on Port0		

FCh	**PS2PS55ID**	**PS55 Keypad Device ID**			64h	FEh
		7-0	RW	PS55 Keypad Device ID		
FD~FFh				Reserved		

4.16 EC Module

4.16.1 EC Funtional Description

There are 7 parts in EC:

- Hardware EC Commands
- EC Index IO mode
- 8051 Extended Commands supporting hardware
- SMBus host controller (described in SMBus sub-chapter)
- SCI Generation
- Virtual COM port
- Misc functions

Hardware EC Commands

EC standard commands as described in ACPI 2.0 spec are processed by hardware logic directly without the intervention of firmware. For EC extended commands, EC controller will forward them to 8051 and thereby processed by the firmware. The data and command/status ports are default to 62h and 66h respectively, and can be optionally mapped to other I/O address space by KBC command 61h.

EC Status Register

To read EC Status IO port register is described as follows:

Status Bit	Name	Description
7	Reserved	Not used.
6	Reserved	Not used.
5	**SCI**	This bit is set to 1 by the EC to indicate that there is/are a/more **SCI** event(s) in the **SCI** queue. The system upon detecting this bit being set should thereafter query the **SCI** event queue (by issuing EC command 84h) to obtain the **SCI** ID number. EC standard commands (80h,81h,82h,83h,84h,F0h, F1h) being received and completed by the EC will not cause the **SCI** bit to be set.
4	**Burst Enable**	The Burst Enable flag. 1=Enabled. 0=Disabled.
3	**Command or Data Flag**	1=Previous access port is command port (**EC_Cmd/EC_STS**). 0=Previous access port is data port (**EC_Dat**).
2	Reserved	Not used.
1	**IBF**	Input Buffer Full flag.
0	**OBF**	Output Buffer Full flag.

EC Command Register

There are 7 valid EC Commands for EC command register (write IO 66h); other values are **"don' t care"** by EC if being written.

Value	Command	Description	P.S.
80h	**EC Read**	Read operation for an internal register in EC Space.	Standard
81h	**EC Write**	Write operation for an internal register in EC Space.	Standard
82h	**EC Burst Enable**	Enable EC burst operation mode.	Standard
83h	**EC Burst Disable**	Disable EC burst operation mode.	Standard
84h	**EC Query**	Query the SCI event queue.	Standard
F0h	**EC Read 16-bit Addr**	Read the EC 64KB Space. High Address is **ECHA** (EC Space 02h).	Non-standard

F1h	**EC Write 16-bit Addr**	Write the EC 64KB Space. High Address is **ECHA** (EC Space 02h)	Non-standard
Others	**Firmware Cmd**	No responded from hardware EC. Firmware EC commands.	

EC Command Programming Sequence

Command Byte	Command Name	Programming Sequence
80h	**Read Embedded Controller**	1. Write **EC_CMD**with 80h (66h=80h) 2. Wait **SCI** for **IBF**=0 3. Write address byte to **EC_DAT** (62h=EC address) 4. Wait **SCI** for **OBF**=1 5. Read **EC_DAT** with data in (read data = 62h)
81h	**Write Embedded Controller**	1. Write **EC_CMD**with 81h (66h=81h) 2. Wait **SCI** for **IBF**=0 3. Write address byte to **EC_DAT** (62h=EC address) 4. Wait **SCI** for IBF=0 5. Write data byte to **EC_DAT** (62h = write data) 6. Wait **SCI** for **IBF**=0
82h	**Burst Enable**	1. Write **EC_CMD**with 82h (66h=82h) 2. Wait **SCI** for **OBF**=1 3. Read **EC_DAT** with 90h(Burst ACK)
83h	**Burst Disable**	1. Write **EC_CMD** with 83h (66h=83h) 2. Wait **SCI** for **IBF**=0
84h	**Query EC**	1. Write **EC_CMD**with 84h (66h=84h) 2. Wait **SCI** for **OBF**=1 3. Read **EC_DAT** with SCI ID number (read data = 62h).
F0h	**Read Embedded Controller By 16-bit Address**	1. Write **EC_CMD**with F0h (66h=F0h) 2. Wait **SCI** for **IBF**=0 3. Write low address byte to **EC_DAT** (62h=EC address) 4. Wait **SCI** for **OBF**=1 5. Read **EC_DAT** with data in (read data = 62h)
F1h	**Write Embedded Controller By 16-bit Address**	1. Write **EC_CMD**with F1h (66h=F1h) 2. Wait **SCI** for **IBF**=0 3. Write low address byte to **EC_DAT** (62h=EC address) 4. Wait **SCI** for **IBF**=0 5. Write data byte to **EC_DAT** (62h = write data) 6. Wait **SCI** for **IBF**=0

EC Index IO Mode

You may use EC Index IO mode to access the KB3910 register space (F400h ~FFFFh). The EC Index IO base is set in LPC register FE92h, FE93h. The base address + 1 is index high byte address. The base address + 2 is index low byte address. The base address + 3 is data port for reading from or writing to KB3910 internal register space. For example, set the base address in FE92h=00h, FE93h = 2Ch. The system IO write set 002Dh = FFh, 002Eh = 01h. The read / write to 002Fh will read / write **ECFV** register (FF01h).

8051 Extended Command (Example only)

The 8051 Extended Command provides a way for the 8051 to carry out commands issued from system host. Through 8051 extended commands, system software can control KB3910 and it’ s peripherals indirectly via 8051. This can reduce the potential problems of both masters are accessing the same slave simultaneously with interleaved cycles. In other word, host initiated cycles can be assured to be “Atomic”. The system host communicates with 8051 via

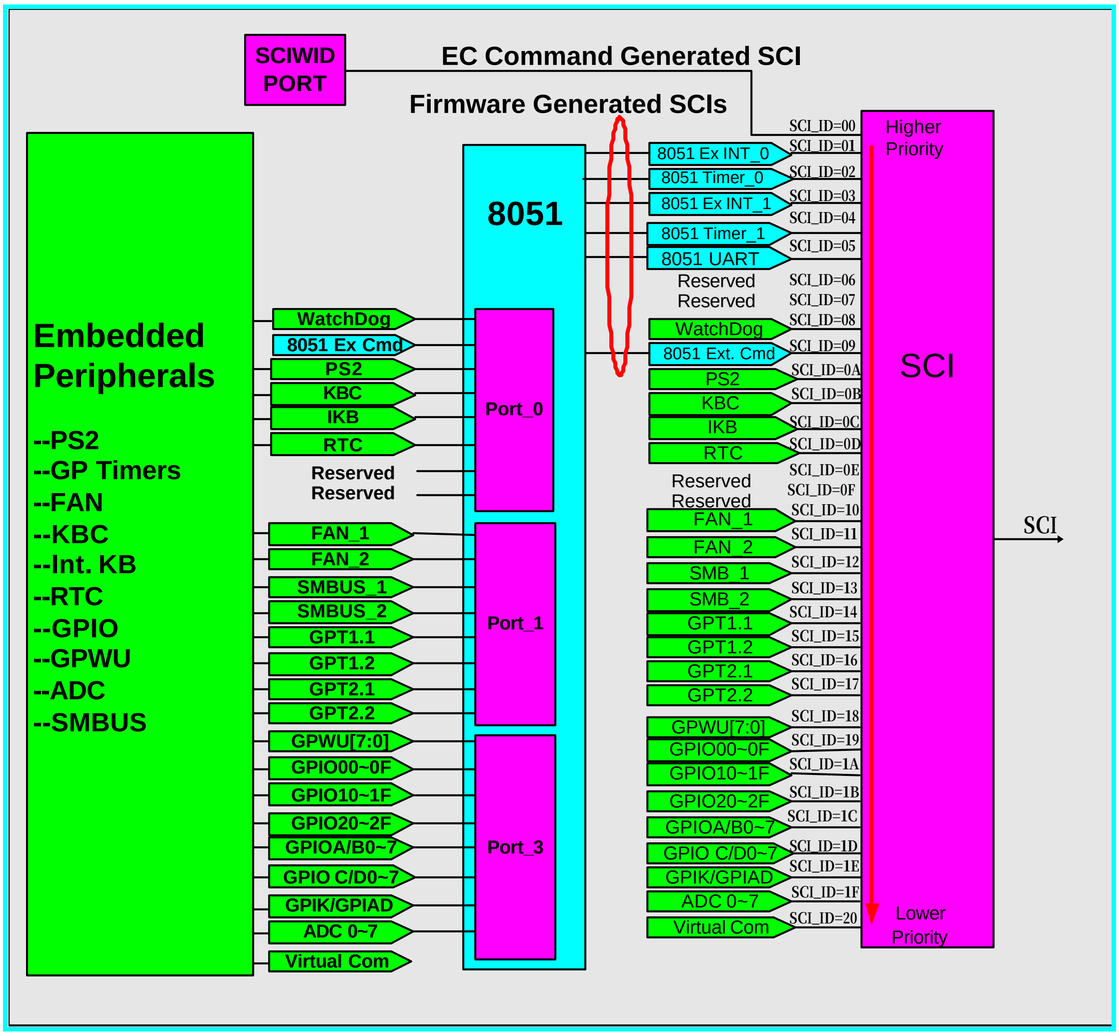
ECPCMD, **ECPARG1**, **ECPARG2**, and **ECPARG3** registers in EC space offset 10h, 11h, 12h and 13h. A non-zero value of **PCMD** will cause the 8051 **P0IE.1** interrupt to take place informing the 8051 firmware, and the 8051 firmware will take the responsibility to carry this command.

PCMD	Command Name	Direction	Parg	Descriptions
01h	**Query 8051 Status**	Entry		Host system checks the 8051 status. If the 8051 firmware is healthy, it should return "Exit" parameter pattern as specified below.
			Parg1	No
			Parg2	No
			Parg3	No
		Exit		00h
			Parg1	00h: Healthy, the Parg1 should = A5h, and Parg2 should = 5Ah. 01h: Error code, 02h: Error code, 03h:
			Parg2	A5h
			Parg3	5Ah
02h	**8051 Go Idle**	Entry		Let KB3910 enter IDLE mode, under which condition the clock is running, no instruction is executing. All interrupt or event can wake-up form IDLE mode
			Parg	No parameter is required.
		Exit	Cmd	00h
			Parg	No returned parameter.
03h	**8051 Go Power Down**	Entry		Let KB3910 enter STOP mode, under which condition the clock is stop for almost all logic, and no instruction is executed. Only asynchronous or WDT event can wake up KB3910 from STOP mode.
			Parg	No parameter is required.
		Exit	Cmd	00h
			Parg	No returned parameter.
04h	**Initialize Flash Write**	Entry		This command notifies 8051 that host will write flash. The 8051 should return OK status to grant host system to write flash. After the completion of this command, the 8051 should enter IDLE mode or reset condition (set EC_PXCFG bit 0 = 1) to prevent from 8051 fetching instruction code.
			Parg1	01h: update all flash content, including 8051 code region. 02h: update system BIOS, not including 8051 code region.
			Parg2	No
			Parg3	No
		Exit	Cmd	00h
			Parg1	Status code: 00h: OK, the host can write flash now. 01h: Error, no AC Adapter (the AC should be plugged during update flash). 02h: Error, unknown condition.

SCI Generation

Most interrupts generated from KB3910 internal modules are connected to the 8051 core and are optionally to generate a SCI event. Each SCI has an associated SCI Enable and SCI Flag bits in EC Space 05h~0Ah. The three extended interrupt ports of 8051, each supporting 8 interrupt channels, can accommodate totally 24 interrupt channels. The pulse-width of SCI is adjustable by setting **ECSCIPW** (default is low-active with 250ns pulse-width). Setting **ECCFG** bit 0=1 (default=0, enabled) to disable the generation of SCI.

In addition to the 24 SCI events generated by KB3910 internal hardware logic, 8051 firmware or system BIOS can also generate a SCI event by writing the desired SCI ID into **ECSCIWID** register (0Bh) in EC space. The **ECSCIWID** should be first enabled in **ECCFG** bit3. The SCI IDs are defined as follows.



SCI ID Table

SCI ID	Name	PxI	Description	Priority
00	**Nothing**	**N.A.**	Indicates a EC Command is received from the Host. Alternatively also means nothing happens	
01h	**IE0**	**N.A.**	Indicates a 8051 external Interrupt IE0 event. The generation of a SCI event will be activated by firmware writing to the **SCIWID**(FF0Bh) port with 01h	
02h	**TR0**	**N.A.**	Indicates a 8051 Timer0 Interrupt TR0. The generation of a SCI event will be activated by firmware writing to the **SCIWID**(FF0Bh) port with 02h	
03h	**IE1**	**N.A.**	Indicates a 8051 external Interrupt IE1. The generation of a SCI event will be activated by firmware writing to the **SCIWI**D(FF0Bh) port with 03h	
04h	**TR1**	**N.A.**	Indicates a 8051 Timer1 Interrupt TR1. The generation of a SCI event will be activated by firmware writing to the **SCIWID**(FF0Bh) port with 04h	
05h	**RI / TI**	**N.A.**	Indicates a 8051 Serial Port Interrupt RI / TI. The generation of a SCI event will be activated by firmware writing to the **SCIWID**(FF0Bh) port with 05h	
06h~07	**N.A.**	**N.A.**	No used.	
08h	**WDT**	**P0I.0**	Indicates a Watchdog Timer event	**Highest**
09h	**E51EC**	**P0I.1**	Indicates a 8051 extended command completed event. After the EC_PCMD wirte with 00h value, this SCI will be generated.	↓
0Ah	**PS2**	**P0I.2**	Indicates a PS2 event	
0Bh	**KBC**	**P0I.3**	Indicates a KBC vent	
0Ch	**IKB**	**P0I.4**	Indicates a Internal Keyboard event	
0Dh	**FAN3**	**P0I.5**	Indicates a FAN Controller 3 event	
0Eh	**PVCOM**	**P0I.6**	Virtual COM port SCI	
SCWIID	**SCWIID**	**P0I.7**	Write **SCI** ID, Query value is SCWIID	
10h	**FAN1**	**P1I.0**	Indicates a FAN Controller 1 event	
11h	**FAN2**	**P1I.1**	Indicates a FAN Controller 2 event	
12h	**SMB1**	**P1I.2**	Indicates a SMBus Host Controller 1 event	
13h	**SMB2**	**P1I.3**	Indicates a SMBus Host Controller 2 event	
14h	**GPT0**	**P1I.4**	Indicates a General Purpose Timer 0 event	
15h	**GPT1**	**P1I.5**	Indicates a General Purpose Timer 1 event	
16h	**GPT2**	**P1I.6**	Indicates a General Purpose Timer 2 event	
17h	**GPT3**	**P1I.7**	Indicates a General Purpose Timer 3 event	
18h	**GPWU0~7**	**P3I.0**	Indicates a General Purpose Wake Up 0~7	
19h	**GPIO00~0F**	**P3I.1**	Indicates a ***GPIO00~0F***	
1Ah	**GPIO10~1F**	**P3I.2**	Indicates a ***GPIO10~1F*** event	
1Bh	**GPIO20~2F**	**P3I.3**	Indicates a ***GPIO20~2F*** event	
1Eh	**GPIAD / GPIK**	**P3I.6**	Indicates a ***GPIAD0~7***, ***GPIK0~7*** event	
1Fh	**ADC**	**P3I.7**	Indicates a ***ADC0~7*** update event	

4.16.2 EC Register Descriptions (Base Address = FF00h, Space = 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
00h	ECHV	**EC Hardware Revision ID**			A0h	FFh
		7 – 0	RO	**ECHV** contains the current hardware version.		
01h	ECFV	**EC Firmware Revision ID**			00h	FFh
		7 – 0	R/W	**ECFV** is written by the 8051 firmware with its current firmware version for system software's recognition		
02h	ECHA	**EC High Address**			00h	FFh
		7 – 0	R/W	High-byte address of the 64KB EC address space. Used for non-standard EC commands to access the full 8051 64KB address space. **ECHA** is not allowed to be programmed as 00h~F3h, since address range 0x0000~0xF3FF is used for the 8051 code memory. When 00h~F3h is accidentently being written, **ECHA** will become FFh.		
03h	ECSCIPW	**SCI Pulse Width**			00h	FFh
		7 – 0	R/W	**SCI** pulse width = (**SCIPW**+1) x 64us		
04h	ECCFG	**EC Configuration**			00h	FFh
		7	R/W	Disable the generation of **SCI** caused by standard EC commands processing 1: Disable 0: Enable (default)		
		6	R/W	Reserved		
		5	R/W	Enable EC Index accessing. 1: Enable 0: Disable (default)		
		4	R/W	Disable EC standard command hardware. 1: Disable 0: Enable (default)		
		3	R/W	Enable **ECSCIWID** port (Firmware generated SCI). 1: Enable 0: Disable (default)		
		2	R/W	EC SCI pulse polarity; =0, low active; =1, high active.		
		1	R/W	Enable EC non-standard commands (F0h, F1h). 1: Enable 0: Disable (default)		
		0	R/W	Disable EC SCI (set 1), default is enabled. 1: Disable 0: Enable (default)		
05h 06h 07h	ECSCIEN0 ECSCIEN1 ECSCIEN3	**EC SCI P0, P1, P3 Interrupt Enable**			00h	FFh
		7 – 0	R/W	Enable extended 8051 Port 0, 1, 3 Interrupt to SCI		
08h 09h 0Ah	ECSCIF0 ECSCIF1 ECSCIF3	**EC SCI P0, P1, P3 Interrupt Flag**			00h	FFh
		7 – 0	R/WC1	Flags for extended 8051 Port 0, 1, 3 Interrupt to SCI. EC Query will clear the query SCI ID flag automatically, or write 1 to clear		
0Bh	ECSCIWID	**EC SCI ID Write Port for 8051 firmware to generate SCI event**			00h	FFh

		7 – 0	R/W	8051 firmware can write to this port with **SCI_ID** value to generate a SCI event. The **SCI_ID** being written into this port can be: 00h: SCI flag in EC_STS will not be set 01h: indicates a 8051 external interrupt IE0 event 02h: indicates a 8051 timer_0 interrupt event 03h: indicates a 8051 external interrupt IE1 event 04h: indicates a 8051 Itimer_1 interrupt event 05h: indicates a 8051 Serial Port interrupt event 09h: indicates a 8051 extended command event 21h~FFh: User custom ID		
0Ch	**EC_PMUTMR**	**PMU Delay Timer before enter STOP mode**			00h	FFh
		7 – 5	R/W	Reserved		
		4 - 0	R/W	Delay time = PMUTMR * 256us. The PMUTMR defines a period of time for all peripherals to stop their transactions before KB3910 go into Deep Sleep mode, where all the clocks generated internally by DPLL are stopped. The Deep sleep mode of KB3910 can be entered by: 1. Program the STOP bit in 8051 SFR PCON 2. If KB3910 was previously waken up (DPLL running) from Deep Sleep mode by an asynchronous wake-up event, but there is no subsequent interrupt event to take place within PMUTMR, then the KB3910 will re-enter Deep Sleep mode		
0Dh	**ECCLKCFG**	**Chip Clock Configuration**			09h	**FFh**
		7	R/W	Test mode. Must be programmed to 0.		
		6	R/W	Ultra Low system clock mode enable, XBI / sytem clock = 1Mhz / 125 KHz, In this mode, all KB3910 peripheral clock will be 125KHz only, so that some device will not work normally under the Ultra Low clock mode, including PS/2, IKB, SMBus, that require faster clock to drive the logic work normally. The timer will run in 125KHz (4Mhz/8) time unit.		
		5	R/W	**Definition for rev. B4 chip** **Enable/Disable DPLL** After the chip is being powered on, the DPLL is default to be disabled. The firmware should enable the internal DPLL after **1 second after power-on** by programming this bit to 1. 0: Disable the internal DPLL (power-on default) 1: Enable the internal DPLL		
		4	R/W	Option to disable the DPLL in deep sleep mode The DPLL module takes the 32.768KHz input and generates 4MHz/32MHz clocks. When in deep sleep mode, all the internal clocks are gated to reduce power consumption. This option further disable the DPLL in deep sleep mode such that no 4MHz/32MHz clocks will be generated at all. This can achieve maximum power saving at the cost that the restarting the DPLL will take longer time. 1: Disable the DPLL in Deep Sleep Mode 0: Do not disable DPLL in deep sleep mode		

		3-2	R/W	XBI Clock Selection: Note that XBI clock must be at least two times faster than the Main Logic Clock 11: 32MHz 10: 16MHz (default) 01: 8MHz 00: 4MHz		
		1-0	R/W	Main Logic Clock Selection 11: 16MHz 10: 8MHz 01: 4MHz (default) 00: 2MHz		
0E~0Fh		Reserved				
10h	**ECPCMD**	**8051 Extended Command**			00h	FFh
		7 – 0	R/W	8051 extended command Port The 8051 Extended Command provides a way for the 8051 to carry out commands issued from system host. Through 8051 extended commands, system software can control KB3910 and it’ s peripherals indirectly via 8051. This can reduce the potential problems of both masters are accessing the same slave simultaneously with interleaved cycles. In other word, host initiated cycles can be assured to be “Atomic”. The system host communicates with 8051 via **ECPCMD**, **ECPARG1**, **ECPARG2**, and **ECPARG3** registers A non-zero value being written to **ECPCMD** will generate a 8051 Extended Interrupt Request (provided that P0IE.0 is enabled). The 8051 firmware should be ready at address 8033h for further processing		
11h	**ECPARG1**	**8051 Extended Parameter 1**			00h	FFh
		7 – 0	R/W	8051 extended parameter 1		
12h	**ECPARG2**	**8051 Extended Parameter 2**			00h	FFh
		7 – 0	R/W	8051 extended parameter 2		
13h	**ECPARG3**	**8051 Extended Parameter 3**			00h	FFh
		7 – 0	R/W	8051 extended parameter 3		
14h	**ECPXCFG**	**8051 Access to off-chip device Control**			00h	FFh
		7	R/W	Reserved.		
		6	R/W	Enable 8051 memory write cycle to off-chip (e.g., BIOS chip) memory spaces. The purpose of this bit is to prevent 8051 firmware from accidentally writing to external memory like BIOS. If not enabled, the 8051 can only access to on-chip peripherals in EC Space, provided register **PCON** Bit_4 is enabled. 1: Enable 0: Dsiable		
		5~2	R/W	Reserved		
		1	R/W	1: **WDT** timout event or wakeup from deep sleep mode to reset 8051 only. 0: **WDT** timeout event or wakeup from deep sleep mode to reset whole chip		
		0	R/W	Reset 8051 and its peripherals The 8051 and its peripherals including timer, interrupt controller, serial port can be reset by setting this bit to one. The 8051 after a reset will restart from reset vector.		
15h	**ECPVSCON**	**8051 Virtual Serial Port Control/Flag**			00h	FFh
		7	R/W	Enable 8051 Virtual Serial Port and SCI.		
		6	R/W	Reserved.		
		5	R/W	**TE** Enable Transmit (TI) SCI.		
		4	R/W	**RE** Enable Reception (RI) SCI.		

		3~2	R/W		Reserved.		
		1	R/WC1	**TI**	8051 received a byte. Clear by EC Query command if **SCI** ID == **PVCOM**.		
		0	R/WC1	**RI**	8051 write SBUF, a byte transmit to host. Clear by EC Query command if **SCI** ID == **PVCOM**.		
16h	**ECPVSBUF**	**8051 Virtual Serial Port Data Buffer**				00h	FFh
		7 – 0	R/W		8051 Virtual Serial Port Data Buffer (SBUF). Read 8051 transmitted byte. Write for 8051 receiving byte.		
17h			NA		Reserved	00h	FFh
18h	**ECMISC**	**EC Miscellanenous Control**				0h	FEh
		7~6	R/W		Reserved		
		5	R/W		EC OBF Interrupt Enable. If this bit is set, EC generates interrupt to the core processor at the falling edge of EC OBF.	0h	
		4~0	R/W		Reserved	0h	
19-1Ah			NA		Reserved	00h	FFh
1Bh	**ECDAT**	**EC Data port**				00h	FFh
		7 – 0	R/W		The EC Data Port serves as the window between system host and EC to convey data.		
1Ch	**ECCMD**	**EC Command port**				00h	FFh
		7 – 0	RO		This register stored the latest EC command from host writing EC command IO port. Normally, standard EC commands will be processed by EC hardware directly. For extended EC commands, 8051 firmware may handle the processing. The port is read-only by the EC.		
1Dh	**ECSTS**	**EC Status port**				00h	FFh
		7	R/W		Enable "write EC command port" interrupt to 8051		
		6	R/W		Enable "write EC data port" interrupt to 8051		
		5	RO		SCI pending flag.		
		4	R/W		Burst Enable.		
		3	RO		Command or data flag. =0, previous host accessing is data port. =1, previous host accessing is command port.		
		2	RSV		Reserved.		
		1	R/WC1		**IBF**, Write **IBF** = 1 to clear **IBF**. (2002/08/15)		
		0	R/WC1		**OBF**, write port **ECDAT** will set **OBF** to 1. Write **OBF** = 1 to clear **OBF**.		
1E~1Fh			NA		Reserved	00h	FFh

4.17 GPWU Module

4.17.1 GPWU Funtional Description

GPWU functions in KB3910 include 2 major groups. One group is ***GPWU0~7***, which occupies one interrupt channel (P3I.0, 8051 port 3 interrupt) and equipped with input de-bounce logic . Another group consists of all GPIOs configured as input mode, including ***GPIO00~2F, ADC0~7(GPIAD0~7), KSI0~7(GPIK0~7)***. This GPIO group shares one interrupt channel (P3I.0~5, 8051 port 3 interrupt extended). Before a specific GPIO pin is configured as GPWU, it should be set to GPIO/Input mode first.

Groups of GPWU:

- ***GPWU0~7*** (equipped with input de-bounce logic)
- ***GPIO00~2F*** (should be set to input mode)
- ***GPIAD0~7*** (ADC pin should be set to digital mode),
- ***GPIK0~7*** (KSI not used for Internal keyboard scan controller).

Each GPIO pin served as GPWU has an associated event enable bit (in **GPWUEN**, **GPIOENxx**, **GPIOxEN**, **GPIKEN** or **GPIADEN**), an event pending flag (**GPWUPF**, **GPIOPFxx**, **GPIOxPF**, **GPIKPF**, **GPIADPF**), an edge/level trigger selection bit (**GPWUEL**, **GPIOELxx, GPIOxEL**, **GPIKEL**, **GPIADEL**) and a high-/low-active or rising/ falling polarity selection bit (**GPWUPS, GPIOPSxx**, **GPIOxPS**, **GPIKPS**, **GPIADPS**). ***GPWU0~7*** use **GPWUDB0~1** to select the source of debounce clock (16us, 256us, 4ms) to accommodate different types of input signals with various noise patterns, such like power button, ring input…etc.

GPWU SCI and 8051 Interrupt

A GPWU event may cause the assertion of a **SCI** or 8051 interrupt, depending on the respective enable bits. The corresponding enable bits for **SCI** or 8051interrupt controller should also be enabled.

GPWU Name	SCI ID	8051 INTR	8051 INTR Vector	Description	Priority
GPWU0~7	18h	***P3I.0***	00C3h	***GPWU0~7***	**Higher**
GPIO00~0F	19h	***P3I.1***	00CBh	***GPIO00~0F***	
GPIO10~1F	1Ah	***P3I.2***	00D3h	***GPIO10~1F***	
GPIO20~2F	1Bh	***P3I.3***	00DBh	***GPIO20~2F***	
GPIK / GPIAD	1Eh	***P3I.6***	00F3h	***GPIAD0~7, GPIK0~7***	
ADC	1Fh	***P3I.7***	00FBh	ADC updated	**Lower**

Due to the interrupt sharing scheme of 8051, the 8051 interrupt service routine should check the event pending flag first to identify the source of the interrupt .

To wake up system from Power Down mode (all system clock are stopped), the GPWU event enable bit should be set for asynchronous waking up PLL to supply KB3910 main logic clock. If a real event is asserted (set pending flag and event is enabled), the KB3910 will leave Power Down mode to RUN state by an interrupt event. Otherwise, the KB3910 will re-enter Power Down after a period of time defined by register **PMUTMR**. The asynchronous wake up signal will check the pin is in input mode, event is enabled and active state (check level and polarity) before de-bounced (for ***GPWU0~7***).

4.17.2 GPWU Register Descriptions (Base Addr= FF00h, Space = 96 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
20h	**GPWUEN**	**GPWU0~7 Enable Register**			00h	FFh
		7-0	R/W	Enable ***GPWU0~7*** Event. When the **GPWUEN** and **GPWUPF** bit are set, an interrupt or wake-up will be issued to 8051 interrupt controller or to SCI.		
21h	**GPWUPF**	**GPWU Pending Flag Register**			00h	FFh
		7-0	R/WC1	Event Flag for GPWU channel 0~7 Writing ‘1’ can clear these bits.		
22h	**GPWUPS**	**GPWU Polarity Selection**			00h	FFh
		7-0	R/W	Polarity selection for GPWU channel 0~7 0: active-low or falling edge trigger 1: active-high or rising edge trigger		
23h	**GPWUDB0**	**GPWU Debouce Clock Selection 0 (for 0~3 channel)**			00h	FFh
		7~6	R/W	***GPWU0*** Clock Source Selection		
		5~4	R/W	***GPWU1*** Clock Source Selection		
		3~2	R/W	***GPWU2*** Clock Source Selection		
		1~0	R/W	***GPWU3*** Clock Source Selection		
				Clock source:		
				00 No debouce.		
				01 16 us clock x 5		
				10 256 us clock x 5		
				11 8ms clock x 5		
24h	**GPWUDB1**	**GPWU Debouce Clock Selection 1 (for4~7 channel)**			00h	FFh
		7~6	R/W	***GPWU4*** Clock Source Selection		
		5~4	R/W	***GPWU5*** Clock Source Selection		
		3~2	R/W	***GPWU6*** Clock Source Selection		
		1~0	R/W	***GPWU7*** Clock Source Selection		
				Clock source:		
				00 No debounce.		
				01 16 us clock x 5		
				10 256 us clock x 5		
				11 8ms clock x5		
25h	**GPWUEL**	**GPWU Edge / Level Trigger Selection**			00h	FFh
		7-0	R/W	For GPWU channel 0~7: 0: Edge Trigger 1: Level Trigger		
28h~2Fh	**SCRATCH**	**SCRATCH Registers**			00h	FFh
		7-0	R/W			
30h 31h 32h 33h 34h 35h	**GPIOEN00** **GPIOEN08** **GPIOEN10** **GPIOEN18** **GPIOEN20** **GPIOEN28**	**GPIO00~27 Enable Register**			00h	FFh
		7-0	R/W	Enable ***GPIO00~2F*** Input Event. When the **GPIOEN** and **GPIOPF** bit are set, a interrupt or wake-up will be issued to 8051 interrupt controller or EC SCI.		
36h	**GPIKEN**	**GPIK0~7 Enable Register**			00h	FFh
		7-0	R/W	Enable ***GPIK0~7*** Input Event.		
37h	**GPIADEN**	**GPIAD0~7 Enable Register**			00h	FFh
		7-0	R/W	Enable ***GPIAD0~7*** Input Event.		
38~3Fh			NA	Reserved	00h	FFh

40h 41h 42h 43h 44h 45h	**GPIOPF00 GPIOPF08 GPIOPF10 GPIOPF18 GPIOPF20 GPIOPF28**	**GPIO00~27 Pending Flag Register**			00h	FFh
		7-0	R/WC1	Event Flags for ***GPIO00~2F***. Writing a ‘ 1’ can clear these bits.		
46h	**GPIKPF**	**GPIK0~7 Pending Flag Register**			00h	FFh
		7-0	R/WC1	Event Flags for ***GPIK0~7***. Writing a ‘ 1’ can clear these bits.		
47h	**GPIADPF**	**GPIAD0~7 Pending Flag Register**			00h	FFh
		7-0	R/WC1	Event Flags for ***GPIAD0~7***. Writing a ‘ 1’ can clear these bits.		
48~4Fh			NA	Reserved	00h	FFh
50h 51h 52h 53h 54h 55h	**GPIOPS00 GPIOPS08 GPIOPS10 GPIOPS18 GPIOPS20 GPIOPS28**	**GPIO00~27 Polarity Selection Register**			00h	FFh
		7-0	R/W	Polarity selection for ***GPIO00~2F***. 0: active-low or falling edge trigger 1: active-high or rising edge trigger		
56h	**GPIKPS**	**GPIK0~7 Polarity Selection Register**			00h	FFh
		7-0	R/W	0: active-low or falling edge trigger 1: active-high or rising edge trigger		
57h	**GPIADPS**	**GPIAD0~7 Polarity Selection Register**			00h	FFh
		7-0	R/W	0: active-low or falling edge trigger 1: active-high or rising edge trigger		
58~5Fh			NA	Reserved	00h	FFh
60h 61h 62h 63h 64h 65h	**GPIOEL00 GPIOEL08 GPIOEL10 GPIOEL18 GPIOEL20 GPIOEL28**	**GPIO00~27 Edge / Level Trigger Register**			00h	FFh
		7-0	R/W	0: Edge Trigger 1: Level Trigger		
66h	**GPIKEL**	**GPIK0~7 Edge / Level Trigger Register**			00h	FFh
		7-0	R/W	0: Edge Trigger 1: Level Trigger		
67h	**GPIADEL**	**GPIAD0~7 Edge / Level Trigger Register**			00h	FFh
		7-0	R/W	0: Edge Trigger 1: Level Trigger		
68~6Fh			NA	Reserved	00h	FFh
70~7Fh	**SCRATCH**	**Scratch registers**			00h	FFh
		7-0	R/W			

4.18 SMBus Module

4.18.1 SMBus Funtional Description

There are 2 identical SMBus host controllers. The SMBus host controller is ACPI 2.0 and SMBus 2.0 compatible. It contains the 40 registers as specified in ACPI EC SMBus Host Controller Interface, and supports all protocols defined in SMBus 2.0 specification, including PEC (CRC Packet Error Check).

Each SMBus host controller function includes several parts:

- SMBus host controller protocol generator and master
- SMBus host controller slave alarm receiver
- Misc functions for interrupt generation, host controller disable, timeout, wakeup, SMBus clock period setting, etc.

SMBus Host Controller and Master

The SMBus protocol generator is started by writing a non-zero value to **SMBPRTCL** register. If the PEC is required, bit 7 of **SMBPRTCL** should be set at the same time to issue a protocol. Before a SMBus protocol commences, the address (**SMBADR**), command (**SMBCMD**) and data (**SMBDAT0~31**) fields of target device should be ready. For Send Byte, Receive Byte, Read Byte, Write Byte protocols, the **SMBDAT0** (offset 9Ch for SMB1, offset DCh for SMB2) is their data byte for reading or writing to SMBus transactions. For Read Word, WriteWord protocols, the **SMBDAT0** (offset 9Ch for SMB1, offset DCh for SMB2) and **SMBDAT1** (offset 9Dh for SMB1, offset DDh for SMB2) are used for their first data byte and 2nd data byte. Because the SMBus device address is 7-bit, bit 0 of **SMBADR** is meaningless.

For the block protocol (Read / Write Block), the data bytes are sequenced in **SMBDAT0** (9Ch, DCh) to **SMBDAT31** (BBh, FBh). For the Write Block, the **SMBCNT** (offset BCh for SMB1, offset FCh for SMB2) should be set before protocol is issued. For Read Block, the **SMBCNT** is not need to be set before protocol is issued, but read by software for knowing the real read data byte count.

Before a protocol is completed, the **SMBPRTCL** will stay in the same value as written. After a protocol completed, the **SMBPRTCL** will be cleared to zero. Besides, the **SMBSTS** will return the completed status. Bit 7 of **SMBSTS** will be set to represent the protocol is DONE. Bits 4~0 return the status representing the protocol DONE flag is OK or there is something wrong. If Error Code in **SMBSTS** is not zero, it means there is something wrong. Refer to the error code with SMBus register descriptions for detail description. If host protocol completed interrupt is enabled, an interrupt will be generated to SCI or 8051 interrupt controller after a protocol is done.

SMBus Host Controller and Slave

The slave alarm receiver is able to receive a device master issuing Write Word protocol to SMBus host controller (8-bit address 10h, 7-bit address 08h). The master device address is received in **SMBAADR** (offset BDh for SMB1, offset FDh for SMB2); and the 2 written data bytes are placed in **SMBADAT0** (offset BEh for SMB1, offset FEh for SMB2) and **SMBADAT1** (offset BFh for SMB1, offset FFh for SMB2). The received address and data bytes from a master device are called SMBus Alarm. Upon receiving an alarm, the Alarm Status (bit 6) in **SMBSTS** will be set. If the alarm interrupt enable is set, an interrupt will be generated to **SCI** or to the 8051 interrupt controller.

SMBus MISC Functions

To enable SMBus host interrupt to SCI or to 8051, the associated interrupt enable bit should be enabled. **SMBEN** (offset 95h for SMB1, offset D5h for SMB2) bit 0 and 1 are used for host protocol completed and alarm received interrupt enabling.

If the SMBus alarm is used to wake up KB3910 from Power Down or IDLE low power state, the wake up enable bit (bit 0) in **SMBWKEN** (offset 96h for SMB1, offset D6h for SMB2) should be set. The interrupt will be generated to wake up KB3910 from Power Down or IDLE only when wake up enable bit (bit 0) is set.

To wake up from Power Down mode (all clocks are stop.), the Asynchronous Wake Up also should be enabled (set **SMBWKEN** bit 7 to 1). In STOP mode, before logic restart, the PLL should be restarted to supply clocks for all logic.

The SMBus host controller (including host protocol generator/master, and host alarm slave receiver) can be disabled completely. This can be done by programming **SMBCFG** (offset 94h/D4h for **SMB1/SMB2**) bits[7:6].

SMBCFG bit 5 can be used to force SMB CLK/DAT signals to low states, which may be used to signal a timeout to reset SMBus device. **SMBCFG** bits[4:0] may be used to set SMBus clock period within 2us ~ 64 us.

SMB Controller and Interface Pin Mapping Scheme

The two SMBus controllers are referred to as **SMB1** and **SMB2**. The internal wiring scheme allows **SMB1** and **SMB2** individually configured to be associated with ***SDA1/SCL1***, ***SDA2/SCL2***, or both. This allows both SMBs to be active and access to the same SMBus slave device simultaneously without arbitration. For example, Windows driver is using **SMB1** while EC firmware is using **SMB2**. By configuring both **SMB1** and **SMB2** to be associated with ***SCL1/SDA1***, where a smart battery device is located, the Windows driver and EC firmware can access to the same smart battery device without any hardware arbitration or software handshaking mechanism in palce. The underlying **SMB1** and **SMB2** will sense the line status of ***SCL1*** and ***SDA1*** automatically before initiate any transaction. If the line status of ***SCL1/SDA1*** is not idle, the SMB Host Controller will suspend its transaction until the ***SCL1/SDA1*** becomes idle.

	SCL1/SDA1	**SCL2/SDA2**	**SCL1/SDA1 Wired-AND SCL2/SDA2**
SMB1	SMB1PIN (FF93h) bit_0=1	SMB1PIN (FF93h) bit_1=1	SMB1PIN (FF93h) bit_0=1 SMB1PIN (FF93h) bit_1=1
SMB2	SMB2PIN (FFD3h) bit_0=1	SMB2PIN (FFD3h) bit_1=1	SMB2PIN (FFD3h) bit_0=1 SMB2PIN (FFD3h) bit_1=1

4.18.2 SMBus Register Descriptions

SMB1 Base Address = FF80h~FFBFh, Space = 64 bytes (EC Space=80h~BFh)

SMB2 Base Address = FFC0h~FFFFh, Space = 64 bytes (EC Space=C0h~FFh)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
80h~8Fh C0~CFh	SCRATCH	**Scratch registers**			00h	FFh
		7 – 0	R/W			
90~91h D0~D1h			NA	Reserved	00h	FFh
92h D2h	SMBTCRC	**SMBus Transmit CRC Value**			00h	FFh
		7 – 0	R/W	The CRC value transmit to SMBus		
93h D3h	SMBPIN	**SMBus PIN Control**			00h	FFh
		7	R/W	**Write to Force SMBus data line to low state** Depending on the mapping of SMB controller (SMBC) to SMBUS interface pins, the corresponding ***SDA1***, ***SDA2***, or both lines will be forced to be LOW when writing a "0" to this bit.		
		6	R/W	**Write Force SMBus clock line to low state** Depending on the mapping of SMB controller (SMBC) to SMBUS interface pins, the corresponding ***SCL1***, ***SCL2***, or both lines will be forced to be LOW when writing a "0" to this bit.		
		5	RO	Read to this bit will reflect the corresponding line status of ***SDA1***, ***SDA2***, or the wired-AND of both.		
		4	RO	Read to this bit will reflect the corresponding line status of ***SCL1***, ***SCL2***, or the wired-AND of both.		
		3~2	R/W	Reserve		
		1	R/W	**SMB controller uses SCL2/SDA2 pins** 93h_bit_1: Map ***SCL2/SDA2*** to SMB Controller_1 D3h_bit_1: Map ***SCL2/SDA2*** to SMB controller_2		
		0	R/W	**SMB controller uses SCL1/SDA1 pins** 93h_bit_0: Map ***SCL1/SDA1*** to SMB Controller_1 D3h_bit_0: Map ***SCL1/SDA1*** to SMB controller_2		
94h D4h	SMBCFG	**SMBus Configuration**			06h	FFh
		7	R/W	SMBus Master Disable (1: Disable)		
		6	R/W	SMBus Slave Disable (1: Disable)		
		4– 0	R/W	SMBus Clock period = bit [4:0] x 4us		
95h D5h	SMBEN	**SMBus Interrupt Enable**			00h	FFh
		7	RO	SMBus Host Controller is operating.		
		6~2	R/W	Reserved		
		1	R/W	SMBus alarm received interrupt enable		
		0	R/W	SMBus protocol completion interrupt enable		
96h D6h	SMBWKEN	**SMBus Wake up Enable**			00h	FFh
		7	R/W	SMBus Async. wakeup enable: to wake up DPLL and start the clock generation.		
		6~1	R/W	Reserved		
		0	R/W	SMBus Alarm Wakeup Enable		
97h D7h	SMBRCRC	**SMBus Received CRC Value**			00h	FFh
		7 – 0	R/W	The latest CRC value received from SMBus slave device		
98h D8h	SMBPRTCL	**EC SMBus Protocol**			00h	FFh
		7	R/W	Enable following SMBus transaction with PEC.		
		6 – 0	R/W	02h Quick write		
				03h Quick read		

				04h	Send byte		
				05h	Receive byte		
				06h	Write byte		
				07h	Read byte		
				08h	Write word		
				09h	Read word		
				0Ah	Write block		
				0Bh	Read Block		
				0Ch	Word Process		
				0Dh	Block Process		
				others	Reserved		
99h D9h	**SMBSTS**	**EC SMBus Status**				00h	FFh
		7	R/WC0	SMBus command done flag			
		6	R/WC0	SMBus alarm received flag			
		5	NA	Reserved			
		4 – 0	R/WC0	Error code			
				00h	SMBus OK, finished normally.		
				07h	Unknown address failure		
				10h	Device address no acknowledge		
				12h	Command no acknowledge		
				13h	Unknown error		
				17h	Device access denied		
				18h	SMBus timeout		
				19h	Un-supported protocol		
				1Ah	SMBus busy		
				1Fh	PEC error		
				Others	Reserved		
9Ah DAh	**SMBADR**	**EC SMBus Address Field**				00h	FFh
		7 – 0	R/W	EC SMBus address. The bit 0 is don't care.			
9Bh DBh	**SMBCMD**	**EC SMBus Command Field**				00h	FFh
		7 – 0	R/W	EC SMBus command. The KB3910 SMBus Host controller is SMBus 2.0 compliant. The Send Byte data is not use Command field again. This may be different from KB3886 (The previous hardware of KB3910). SMBus Host controller.			
9C-BBh DC~FBh	**SMBDAT**	**EC SMBus Data Array**				00h	FFh
		7 – 0	R/W	EC SMBus data array (total 32 bytes) **SMBDAT0** (9Ch / DCh) are used for Send / Receive / Read Byte / Write Byte protocol data byte. **SMBDAT1** (9Dh / DDh) is are used for Read Word / Write Word protocol 2nd data byte. Others are used for block mode protocols.			
BCh FCh	**SMBCNT**	**EC SMBus Block Count**				00h	FFh
		7 – 0	R/W	EC SMBus block count, 0 for 32 bytes data byte length. The bit 6~7 are don't cared.			
BDh FDh	**SMBAADR**	**EC SMBus Alarm Address**				00h	FFh
		7 – 0	R/W	EC SMBus alarm address from SMBus master device			
BEh FEh	**SMBADAT0**	**EC SMBus Alarm Data 0**				00h	FFh
		7 – 0	R/W	EC SMBus alarm data0 (Low byte) from SMBus master device			
BFh FFh	**SMBADAT1**	**EC SMBus Alarm Data 1**				00h	FFh
		7 – 0	R/W	EC SMBus alarm data1 (High byte) from SMBus master device			

4.20 Power Management

KB3910 is an 8051 based microcontroller. The power management of KB3910 is the legacy of standard 8051. KB3910 support 2 levels of sleep modes: **Sleep** mode and **Deep sleep** mode. The Sleep mode of KB3910 puts the embedded 8051 into IDLE mode, while the Deep Sleep mode puts the embedded 8051 into Power Down mode.

System Power State Table

KB3910 State	LPC State	8051 State	Wakeup Source	KB3910 sub-system Power Consumption	Power
RUN	ON	RUN	Not need	Flash (running) + KB3910 (running)	Higher ↓ Lower
SLEEP	ON / CLKRUN# ACTIVE	IDLE	All Interrupts	Flash (standby) + KB3910 (running)	
DEEP SLEEP	CLKRUN# ACTIVE	Power Down	Async. Wake-up events: 1. Keyboard Scan-in event 2. PS/2 Event 3. LPC cycles accessing KBC/EC) 4. SMBus events 5. All programmable GPI pins 6. WDT	Flash (standby) + KB3910 (standby)	

Firmware Programming Note for Power Management

To enter chip lower power state, the 8051 **PCON SFR** should be programmed to proper state. During **Sleep** mode, the 8051 Program Counter is stopped, all register contents and GPIO states are preserved. All 8051's interrupt events will wake up the chip back to the normal running state. During **Deep Sleep** mode, the KB3910 internal clock is stopped, except the 32KHz clock which supplys **WDT** is running, all interal register contents and GPIO states are preserved. Only external asynchronous wakeup events can wake up the chip from **Deep Sleep** mode, including

1. Keyboard scan-in evet
2. PS/2 event
3. LPC cycles accessing KBC/EC
4. SMBus event
5. WDT timeout event
6. All programmable GPI input event

Upon being waken up from Deep Sleep mode(**wake-up event**), KB3910 will start to fetch instruction from address 0000h. Firmware should take check **ECFV** information in 8051 to see whether a **power_on_reset event** or **wake-up event** has just been taking place. Note that the watchdog timer expiration event will cause a **power_on_reset event** to take place rather than a **wake-up event.**

4.21 8051 Module

4.21.1 8051 Funtional Description

8051 Hardware Description
The embedd 8051 is compatoble with industrial standard 8051(or 8031). There are 3 standard 8051 peripherals, including the Interrupt controller, the Serial port and two 16-bit timers.

KB3910 extends the channels of Interrupt Controller in the original 8051 to 24 channels supporting internal peripheral devices. The Serial port use **SCON2** to achieve high speed serial transmission rate up to 115200 bps. The two 16-bit timers are basically the same as that in the standard 8051's, except when **SCON2** is used to generate high-speed baud rate. Under such ciscumstances the 2 timers will not be used for baud-rate generation and can be used for other purposes.

The 8051 uses **MOVX** and **MOVC** instructions to read or write KB3910 peripherals, i.e., **EC, SMBus, GPIO, GPWU, KBC, IKB, GPT, PWM, PS2, XBI, LPC, XRAM**…etc.

Hereuder lists the differences between the KB3910' s embedded 8051 and that of the industrial standard 8051:

1. Interrupt priorities for each channel is fixed, and no recusive interrupt is supported.
The interrupt service routine will not be interrupted until executing the **RETI** instruction. The original 8051 contains 2 priority levels, while the Interrupt controller in KB3910 doesn't use multi-level priority interrupt.

2. **Improved Instruction Execution Time:** The instruction execution time may be calculated by the formula: **[Instruction Fetch Clock Cycles] + [2 Clock cycles]**.

3. **The original 8051 is Harvard Architecture (independent Code and Data paths), while the 8051 in KB3910 is Von Newman (Code and Data shared paths).** But FF40h~FFFFh is separated because the FF40h~FFFFh is used to be internal peripheral register or SRAM space. The data accessing to those address will hit KB3910 internal peripheral registers. The code fetching to those address will hit external flash. But the MOVC will still hit internal peripheral registers in FF40h~FFFFh.

4. The execution time for instructions **MUL** and **DIV** are the same as normal instructions.

8051 Interrupt Vectors Table

Interrupt Source	Vector Address	KB3910 Application	SCI ID	Priority
IE0	0003h	8051 external interrupt 0 (***GPIO1A***)	01h	Highest
TF0	000Bh	8051 Timer 0	02h	Highest
IE1	0013h	8051 external interrupt 1 (***GPIO1B***)	03h	Highest
TF1	001Bh	8051 Timer 1	04h	Highest
RI & TI	0023h	8051 Serial Port	05h	Highest
P0I.0	0043h	Watchdog Timer	08h	High
P0I.1	004Bh	8051 extended command	09h	
P0I.2	0053h	PS/2	0Ah	
P0I.3	005Bh	KBC	0Bh	
P0I.4	0063h	IKB	0Ch	
P0I.5	006Bh	FAN3	0Dh	
P0I.6	0073h	EC non-hardware command.	0Eh	
P0I.7	007Bh	EC Write SCI finished.	0Fh	
P1I.0	0083h	FAN1	10h	
P1I.1	008Bh	FAN2	11h	
P1I.2	0093h	SMBus 1	12h	
P1I.3	009Bh	SMBus 2	13h	

P1I.4	00A3h	**GPT0**	14h	
P1I.5	00ABh	**GPT1**	15h	
P1I.6	00B3h	**GPT2**	16h	
P1I.7	00BBh	**GPT3**	17h	
P3I.0	00C3h	***GPWU0~7***	18h	
P3I.1	00CBh	***GPIO00~0F***	19h	
P3I.2	00D3h	***GPIO10~1F***	1Ah	
P3I.3	00DBh	***GPIO20~2F***	1Bh	
P3I.4	00E3h	***Reserved***	1Ch	
P3I.5	00EBh	***Reserved***	1Dh	
P3I.6	00F3h	***GPIAD0~7, GPIK0~7***	1Eh	
P3I.7	00FBh	ADC updated	1Fh	Lowest

* The MSB of the Interrupt Vector can be set in **PCON.5** (**IVHV**).

8051 SFR Map

Legend

XXX	Original Industrial standard 8051 features
XXX	KB3910' s embedded 8051 new features
XXX	XXX Changed 8051 feature for ENE 8051
XXX	XXX With add-on feature for ENE 8051

	0	1	2	3	4	5	6	7	
F8	P3IF								FF
F0	B								F7
E8	P1IF								EF
E0	ACC								E7
D8	P0IF								DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3IE								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF	SCON2						9F
90	P1IE								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0IE	SP	DPL	DPH			PCON2	PCON	87
	8	9	A	B	C	D	E	F	

This column registers are bit address-able.

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 contains 8-bits. For the embedded 8051 inside KB3910, the 3 ports are used for interrupt input (always rise pulses) extensions. Totally there are 24 interrupt events.

P3IF, P1IF, P0IF are Interrupt Flags (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**.

4.21.2 8051 SFR Descriptions (Direct Addressing 80h~FFh)

Addr	Register Abbreviation	Register Full Name				Def
		Bit	Attr	Description		
80h	**P0IE**	**Port 0 IE**				00h
		7 – 0	R/W	P0 Interrupt Enable Register		
81h	**SP**	**Stack Pointer**				07h
		7 – 0	R/W	Stack Pointer		
82h	**DPL**	**DPTR Low Byte**				00h
		7 – 0	R/W	DPTR low byte		
83h	**DPH**	**DPTR High Byte**				00h
		7 – 0	R/W	DPTR high byte		
84h-85h			NA	Reserved		00h
86h	**PCON2**	**Processor Control Register 2**				00h
		7	R/W	Enable level trigger interrupt		
		6	R/W	**TTST**, Timer 0/1 test mode, let timer 12 times faster.		
		5	R/W	Reserved		
		4	R/W	Enable external space write.		
		3	R/W	Next Interrupt Coming Flag. The same extended interrupt coming during ISR before IRET. After exit ISR with IRET instruction, the 8051 will re-enter the ISR again if the flag is 1. Write 0 to clear the flag and prevent from 8051 re-entering the interrupt again after exit ISR.		
		2~1	NA	Reserved		
		0	R/W	Enable idle loop no fetching instruction		
87h	**PCON**	**Processor Control Register**				00h
		7	R/W	**SMOD**, Double baud rate, Set 1 to double timer 1 baud rate generator.		
		6	R/W	Reserved		
		5	R/W	**IVHB**, Interrupt vector highest bit. Let interrupt vector to be 00xxh or 80xxh, including standard and extended interrupt.		
		4	R/W	Reserved		
		3	R/W	**GF1**, general purposes flag.		
		2	R/W	**GF0**, general purposes flag.		
		1	WO	**Power Down Mode** Stop all 8051 clock, including all peripherals (timer, interrupt, serial port). An external Async. wake-up event can reset the latch of 8051 gated clock. Write 1 to enter Power Down.		
		0	WO	**IDLE Mode.** Set 1 to stop processor fetching instructions. But the clock will not stop. Peripheral interrupt events will let processor exit IDLE mode. Write 1 to enter IDLE.		
88h	**TCON**	**Timer/Counter Control Register**				00h
		7	R/WC	**TF1**	Timer 1 overflow flag.	
		6	R/W	**TR1**	Timer 1 run control bit.	
		5	R/WC	**TF0**	Timer 0 overflow flag.	
		4	R/W	**TR0**	Time 0 run control bit.	
		3	R/WC	**IE1**	External Interrupt 1 edge detected flag.	
		2	R/W	**IT1**	Interrupt 1 falling edge / low active control bit	
		1	R/WC	**IE0**	External Interrupt 0 edge detected flag.	

		0	R/W	IT0	Interrupt 0 falling edge / low active control bit	
89h	TMOD	Processor Control Register				00h
		7	R/W	GATE1	Gating control of TR1 and INT1.	
		6	R/W	CT1	=0, be a timer 1. =1, be a counter 1.	
		4~5	R/W	TM1	=0, Timer 1 is 13-bit timer (8048 timer). =1, Timer 1 is 16-bit timer =2, Timer 1 is 8-bit auto-reload timer =3, Timer 1 is stop.	
		3	R/W	GATE0	Gating control of TR0 and INT0.	
		2	R/W	CT0	=0, be a timer 0. =1, be a counter 0.	
		0~1	R/W	TM0	=0, Timer 0 is 13-bit timer (8048 timer). =1, Timer 0 is 16-bit timer =2, Timer 0 is 8-bit auto-reload timer =3, Timer 0 TL0, TH0 is two 8-bit timer.	
8Ah	TL0	Timer 0 Low Byte				00h
		7 – 0	R/W		Timer 0 Low Byte	
8Bh	TL1	Timer 1 Low Byte				00h
		7 – 0	R/W		Timer 1 Low Byte	
8Ch	TH0	Timer 0 High Byte				00h
		7 – 0	R/W		Timer 0 High Byte	
8Dh	TH1	Timer 1 Low Byte				00h
		7 – 0	R/W		Timer 1 High Byte	
90h	P1IE	Port 1 IE				00h
		7 – 0	R/W		P1 Interrupt Enable Register	
91h-97h			NA		Reserved	00h
98h	SCON	Serial Port Control Register				00h
		7~6	R/W	SM1 SM0	Serial Port Mode: 00: no used 01: 8-bit Serial Port(variable) 10: no used 11: 9-bit Serial Port (variable)	
		5	NA		Reserved	
		4	R/W	REN	Enable Serial Port reception	
		3	R/W	TB8	The 9th bit of transmitted in mode2 2 & 3.	
		2	R/W	RB8	The 9th bit of received.	
		1	R/WC	TI	Transmit Interrupt flag.	
		0	R/WC	RI	Receive Interrupt flag.	
99h	SBUF	Serial Port Data Buffer				00h
		7 – 0	R/W		Serial port data buffer	
9Ah	SCON2	Serial Port Control 2 Register				00h
		7	R/W	ASEN	Advanced Serial Port Enable bit. The Timer 1 will not used to be clock generator of Serial Port.	
		6~0	R/W	ASHT	Advanced Serial Port counter half count. Serial Port clock cycle time will be ASHT x 2. If main logic clock is 4Mhz set ASHT as fellows, 17 = 115200 bps 35 = 57600 bps 104 = 19200 bps	
A0h	P2	Port 2 Latch Register				00h
		7 – 0	R/W		Port 2, high address of external bank accessing.	
A8h	IE	Interrupt Enable Register				00h
		7	R/W	EA	Disable all interrupt (include extended) if clear to 0. If set to 1, all interrupts should be enabled by individual enable bit.	

		6~5	R/W		Reserved	
		4	R/W	**ES**	Enable Serial Port interrupt	
		3	R/W	**ET1**	Enable Timer 1 Overflow interrupt	
		2	R/W	**EX1**	Enable External Interrupt 1	
		1	R/W	**ET0**	Enable Timer 0 Overflow interrupt	
		0	R/W	**EX0**	Enable External Interrupt 0	
B0h	**P3IE**	**Port 3 Interrupt Enable**				00h
		7 – 0	R/W	P3 Interrupt Enable Register		
B8h	**IP**	**Interrupt Priority Register**				00h
		7~5	NA		Reserved	
		4	R/W	**PS**	Serial Port interrupt priority level	
		3	R/W	**PT1**	Timer 1 interrupt priority level	
		2	R/W	**PX1**	External Interrupt 1 priority level	
		1	R/W	**PT0**	Timer 0 interrupt priority level	
		0	R/W	**PX0**	External Interrupt 0 priority level	
D0h	**PSW**	**Processor Status Word**				00h
		7	R/W	**CY**	Carry flag.	
		6	R/W	**AC**	Auxiliary Carry flag.	
		5	R/W	**F0**	Flag 0, 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.	
D8h	**P0IF**	**Port 0 Interrupt Flag**				00h
		7 – 0	R/W	P0 Interrupt Flag Register		
E0h	**ACC**	**ACC, A**				00h
		7 – 0	R/W	Accumulator		
E8h	**P1IF**	**Port 1 Interrupt Flag**				00h
		7 – 0	R/W	P1 Interrupt Flag Register		
E0h	**B**	**B Register**				00h
		7 – 0	R/W	For MUL and DIV operations.		
F8h	**P3IF**	**Port 3 Interrupt Flag**				00h
		7 – 0	R/W	P3 Interrupt Flag Register		

5 Electronic Characteristics

5.1 Absolute Maximum Ratings

Symbol	Parameter	Condition	Ratings	Unit
VCC	Power source volatge	All voltage are based on VSS.	-0.3 to 3.6	V
V_I	Input voltage		-0.3 to 3.6	V
V_O	Output voltage		-0.3 to 3.6	V
T_{OP}	Operating temperature		-25 to 85	℃

5.2 Recommended Operating Conditions

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
VCC	Power source voltage	3	3.3	3.6	V
VSS	Ground voltage	-0.3	0	0.3	V
VCCA	Analog reference voltage(A/D and D/A converter is used)	2.5		3.6	V
AGND	Analog ground voltge		0		V

5.3 Operating Current

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
I_{CC}	Typical Current consumption when the chip is in normal operating state: Under Windows, all clock domains are running, No PS2/KB/mouse movements		12		mA

5.4 A/D Characteristics

(VCC=2.7V to 3.3V , VREF=2.5V to 5V VSS=AVSS=0V T_A=0 to 85℃)

Symbol	Parameter	Test condition	Limits			Unit
			Min	Typ	Max	
	Resolution				8	bit
	Absolute accuracy	VCC=3.3V Vref=5V or Vref=2.5V			+-2	LSB
T_{CONV}	Conversion time	Clock cycle time=2us	30			clk
R_{LADDER}				60		KOHM
I_{VREF}	Reference power source input current	At A/D converter operated			200	uA
		At A/D converter stopped			5	uA

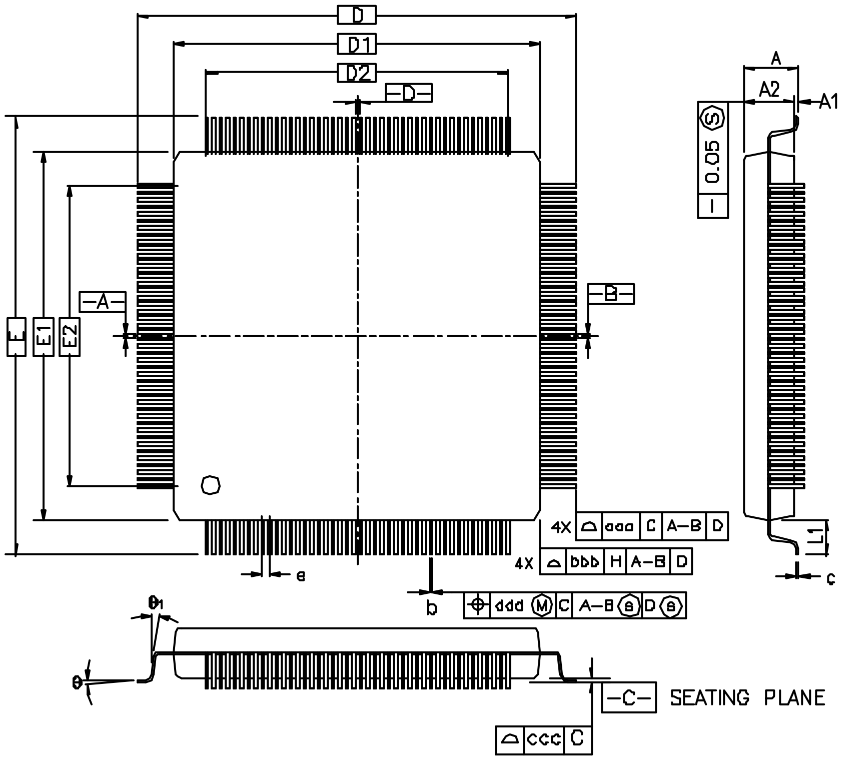
5.5 D/A Characteristics

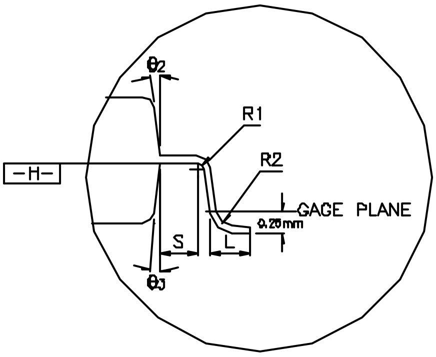
Symbol	Parameter	Test condition	Limits	Unit

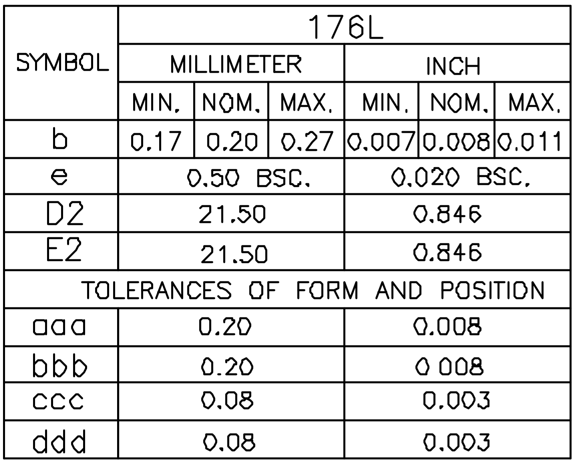
			Min	**Typ**	**Max**	
	Resolution				8	bits
	Absolute accuracy	VCC=3.3V Vref=2.5V or 5V		2.5		%
T_{SET}	Settling time				3	
R_O	Output resistance		1.7	2.5	3.4	KOHM
I_{VREF}	Reference power source input current	Vref=5V VCC=3.3V Conversion bits "1116"			3.2	mA

6. Package Information

6.1 176 LQFP (24x24) Package Dimension

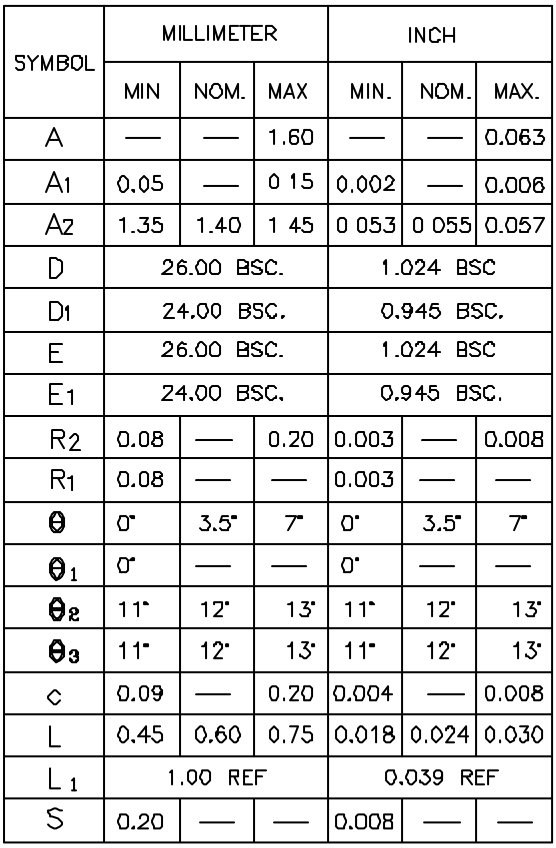






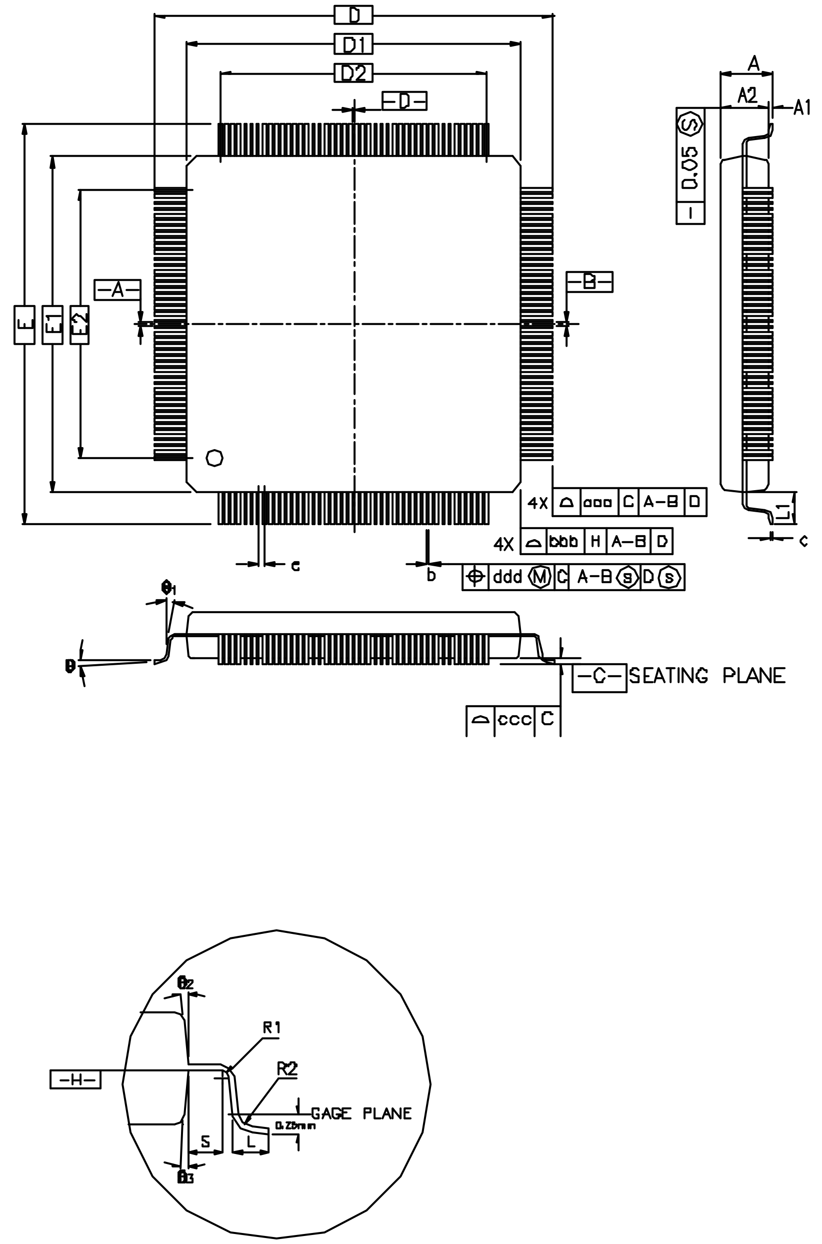
SYMBOL	176L					
	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	21.50			0.846		
E2	21.50			0.846		
TOLERANCES OF FORM AND POSITION						
aaa	0.20			0.008		
bbb	0.20			0 008		
ccc	0.08			0.003		
ddd	0.08			0.003		

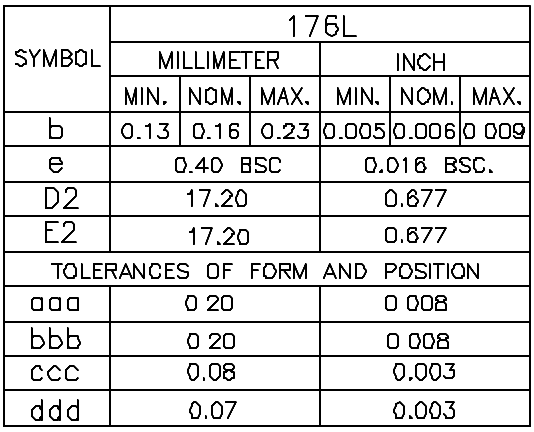
COTROL DIMENSIONS ARE IN MILLIMETERS.



SYMBOL	MILLIMETER			INCH		
	MIN	NOM.	MAX	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A_1	0.05	—	0 15	0.002	—	0.006
A_2	1.35	1.40	1 45	0 053	0 055	0.057
D	26.00 BSC.			1.024 BSC		
D_1	24.00 BSC.			0.945 BSC.		
E	26.00 BSC.			1.024 BSC		
E_1	24.00 BSC.			0.945 BSC.		
R_2	0.08	—	0.20	0.003	—	0.008
R_1	0.08	—	—	0.003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	—	—	0°	—	—
θ_2	11°	12°	13°	11°	12°	13°
θ_3	11°	12°	13°	11°	12°	13°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L_1	1.00 REF			0.039 REF		
S	0.20	—	—	0.008	—	—

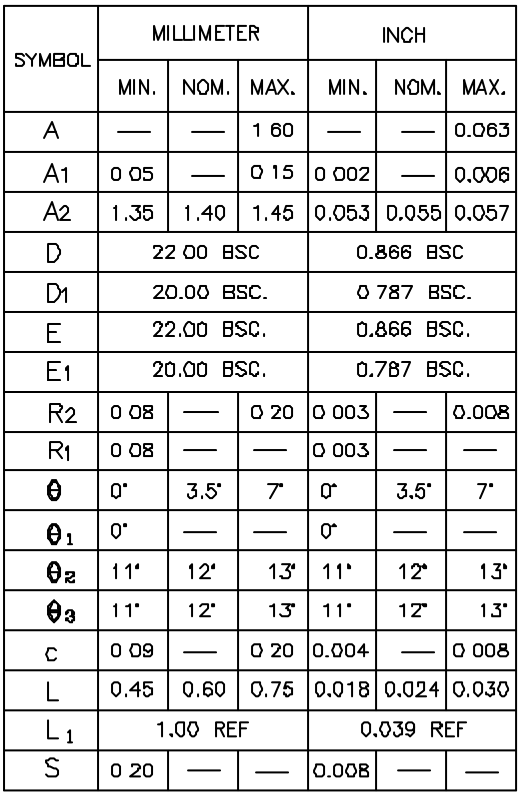
6.2 176 LQFP (20x20) Package Dimension





SYMBOL	176L					
	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
b	0.13	0.16	0.23	0.005	0.006	0 009
e	0.40 BSC			0.016 BSC.		
D2	17.20			0.677		
E2	17.20			0.677		
TOLERANCES OF FORM AND POSITION						
aaa	0 20			0 008		
bbb	0 20			0 008		
ccc	0.08			0.003		
ddd	0.07			0.003		

COTROL DIMENSIONS ARE IN MILLIMETERS.



SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1 60	—	—	0.063
A1	0 05	—	0 15	0 002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	22 00 BSC			0.866 BSC		
D1	20.00 BSC.			0 787 BSC.		
E	22.00 BSC.			0.866 BSC.		
E1	20.00 BSC.			0.787 BSC.		
R2	0 08	—	0 20	0 003	—	0.008
R1	0 08	—	—	0 003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	—	—	0°	—	—
θ_2	11°	12°	13°	11°	12°	13°
θ_3	11°	12°	13°	11°	12°	13°
c	0 09	—	0 20	0.004	—	0 008
L	0.45	0.60	0.75	0.018	0.024	0.030
L_1	1.00 REF			0.039 REF		
S	0 20	—	—	0.008	—	—

6.3 Part Number Descriptions

Part Number	Pack Size	Lead Free Process
KB3910S C1	20mm * 20mm 176 Pins LQFP	
KB3910SF C1	20mm * 20mm 176 Pins LQFP	Lead Free
KB3910Q C1	24mm * 24mm 176 Pins LQFP	
KB3910QF C1	24mm * 24mm 176 Pins LQFP	Lead Free