ENE TECHNOLOGY INC.

KB3920

Keyboard Controller Datasheet

Revision 0.8
Jun. 2006

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

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

CONTENT

1. Features

1.1 Feature Summary

- **Low Pin Count Host Interface (LPC)**
  - SIRQ supporting IRQ1, IRQ12, SCI or SMI# interrupt
  - I/O Address Decoding:
    - KBC IO Port 60h/64h
    - Programmable EC IO Port 62h/66h
    - Programmable 4-byte Index I/O ports to access internal registers
    - One Programmable I/O write byte-address decoding (for example, it can be used to decode Port_80h)
  - Support CLKRUN# as a slave device

- **X-Bus Bus Interface (XBI)**
  - Interfaces to ISA-Type memory and I/O devices
  - Addressable memory range up to 2MB
  - The 64K code memory of 8051 can be mapped onto 2 independent 16KB pages
  - 2 SELIO# for expended IO (XIO) by directly connecting to TTL (74373, 74244)

- **8051 Microprocessor**
  - Industry 8051 Instruction set complaint with 3~5 cycles per instruction.
  - Programmable 4/8/16 MHz clock
  - Fast instruction fetching from XBI Interface
  - 128 bytes and 2KB tightly-coupled SRAM
  - Twenty-four extended interrupt sources
  - Two 16-bit tightly-coupled Timers

- **8042 Keyboard Controller**
  - 8 Standard keyboard commands processed by hardware
  - Each hardware command can be optionally processed by firmware

- **PS/2 Controller**
  - External PS/2 device operation in firmware mode operation (no hardware mode available)
  - Support three external PS/2 devices
  - Optimized hardware for firmware processing

Ø Internal Keyboard Encoder (IKB)

- **n** 18x8 keyboard scan matrix
- **n** Optimized hardware for firmware processing
- **n** Support W2K Internet / multimedia keys
- **n** Support hot-key events
- **n** Ghost key phenomenon cancellation

Ø Embedded Controller (EC)

- **n** Five EC Standard Commands can be processed by hardware
- **n** ACPI Specification 2.0 compliant
- **n** each hardware command can be optionally switched to be processed by firmware
- **n** Programmable EC I/O port addressing (default 62h/66h)

Ø SMBus Host Controller (SMBUS)

- **n** 2 SMBus Interfaces multiplexed with one internal SMBus host controller
- **n** SMBus Specification 2.0 compliant
- **n** Wake-up from low-power modes

Ø Digital To Analog Converter (DAC)

- **n** Four built-in DACs with 8-bit resolution
- **n** The DAC pins can be alternatively configured as General Purpose Outputs (GPOs)

Ø Analog To Digital Converter (ADC)

- **n** Four built-in ADCs with 8-bit resolution
- **n** The ADC pins can be alternatively configured as General Purpose Inputs (GPIs)

Ø Pulse Width Modulator (PWM)

- **n** Four built-in PWMs (2 are FANPWM)
- **n** Selectable clock sources: 1MHz/64KHz/4KHz/256Hz
- **n** Configurable cycle time (up to 1 sec) and duty cycle

Ø Watchdog Timer (WDT)

- **n** 32.768KHz input clock with 20-bit time scale
- **n** 8-bit watchdog timer interrupt and reset setting

- **General Purpose Timer (GPT)**
  - Two 16-bit, two 8-bit general purpose timers with 32.768KHz resolution

- **General Purpose Wake-Up (GPWU)**
  - All General Purpose Input pins can be configured to generate interrupts or wake-up events

- **General Purpose Input/Output (GPIO)**
  - All outputs can be optionally tri-stated
  - All inputs equipped with pull-up, high/low active, edge/level trigger selection
  - DAC pins can only be configured as GPO pins (no GPI function embedded)
  - ADC pins can only be configured as GPI pins (no GPO function embedded)

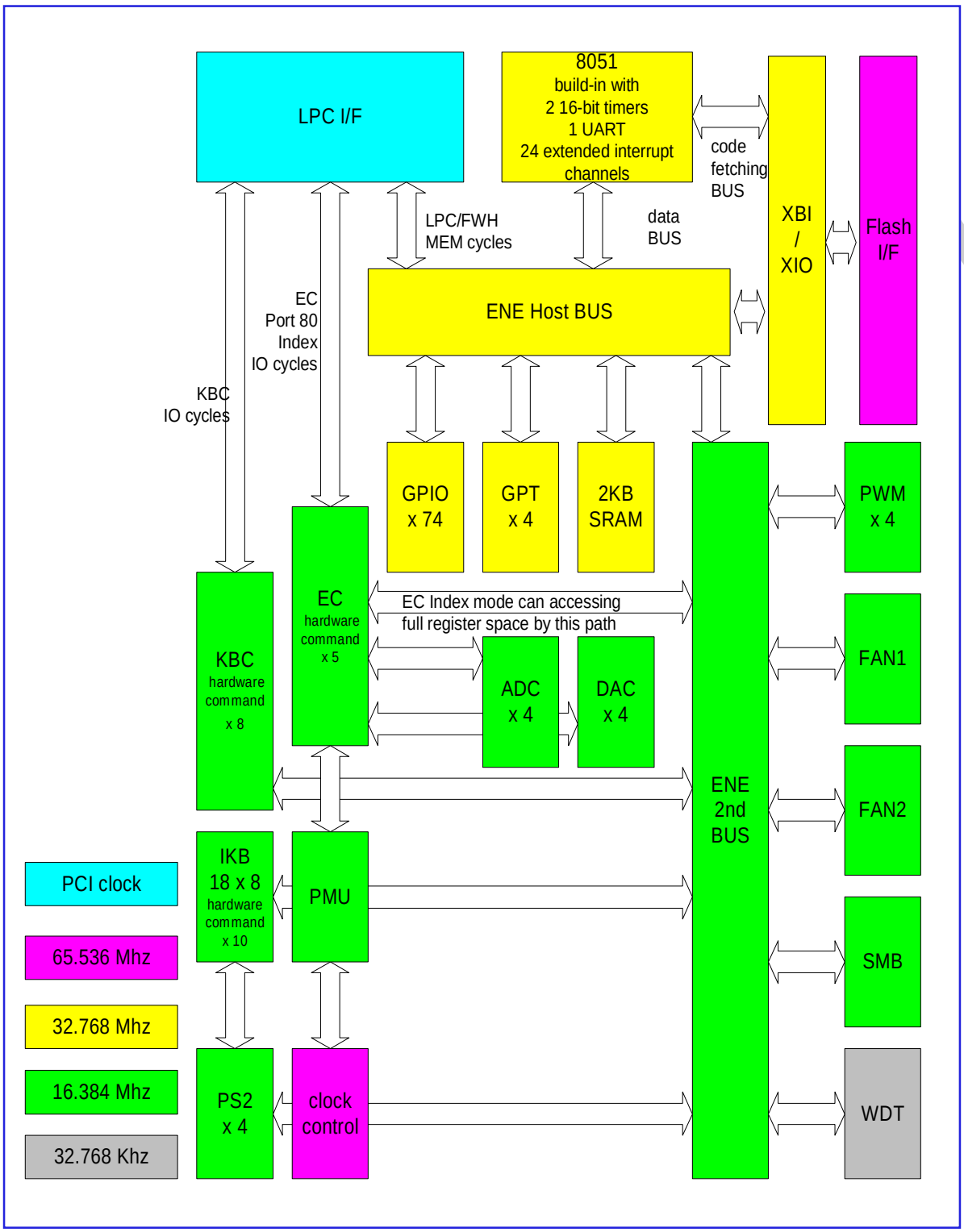
- **Fan Controller (FAN)**
  - Two fan controllers with tachometer inputs
  - Automatic FAN speed control

- **Power Management**
  - Sleep State: 8051 Program Counter (PC) stopped
  - Deep Sleep State: Stop all internal clocks. Target power consumption ~10uA.

1.2 Comparison on KB3910 176-pin vs. KB3920 144-pin

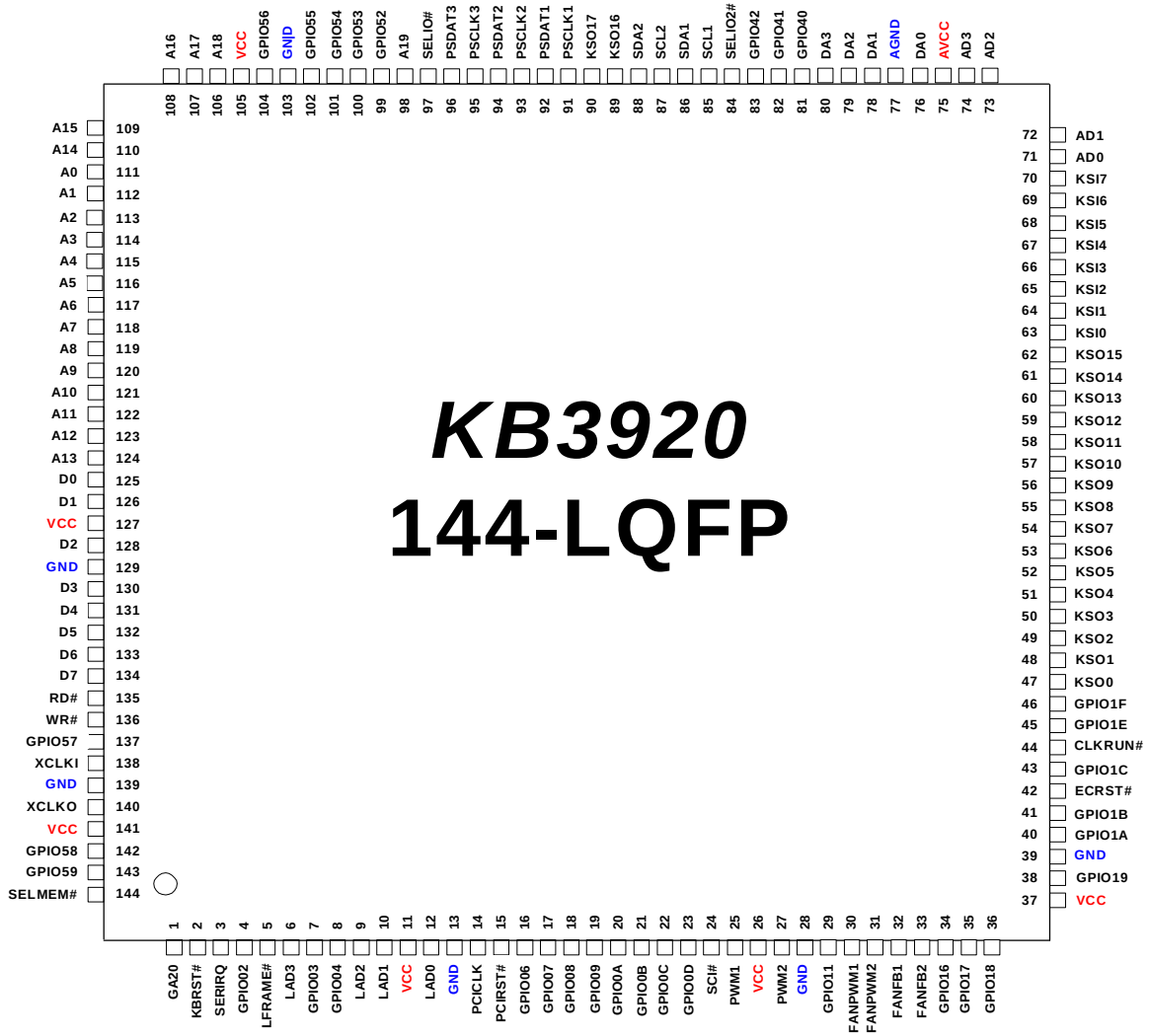
	KB3910 176-pin LQFP	KB3920 144-pin LQFP
Chip Dimension	24mmx 24mm	20mm x 20mm
Microprocessor	8051	8051
Built-in SRAM	2048 bytes for 8051	2048 bytes for 8051
	96 bytes for EC RAM	(EC RAM included in 2048 bytes)
	256 bytes for KB Matrix Table	Matrix Table doesn't exist in SRAM
X-Bus Memory Range	2M bytes	2M bytes
RTC	Optional	No
ADC	8	4
DAC	8	4
Watch Dog Timer	1	1
PWM	8	4
External PS/2 devices	3	3
GPIOs	Maximum 106 pins	Maximum 90 pins (+27 in SPI mode)
KB matrix scan	18x8	18x8
FAN Controller	3	2
General Purpose Timer	6	6
SM Bus	2 Interfaces	2 Interfaces
	2 Internal Controllers	1 Internal Controller
HW KBC Standard	24	8
HW IKB Standard	10	10
HW EC Standard	5	5
Power Consumption	25mA typ. in operating mode	TBD
	10uA in STOP mode	TBD

1.3 Block Diagram

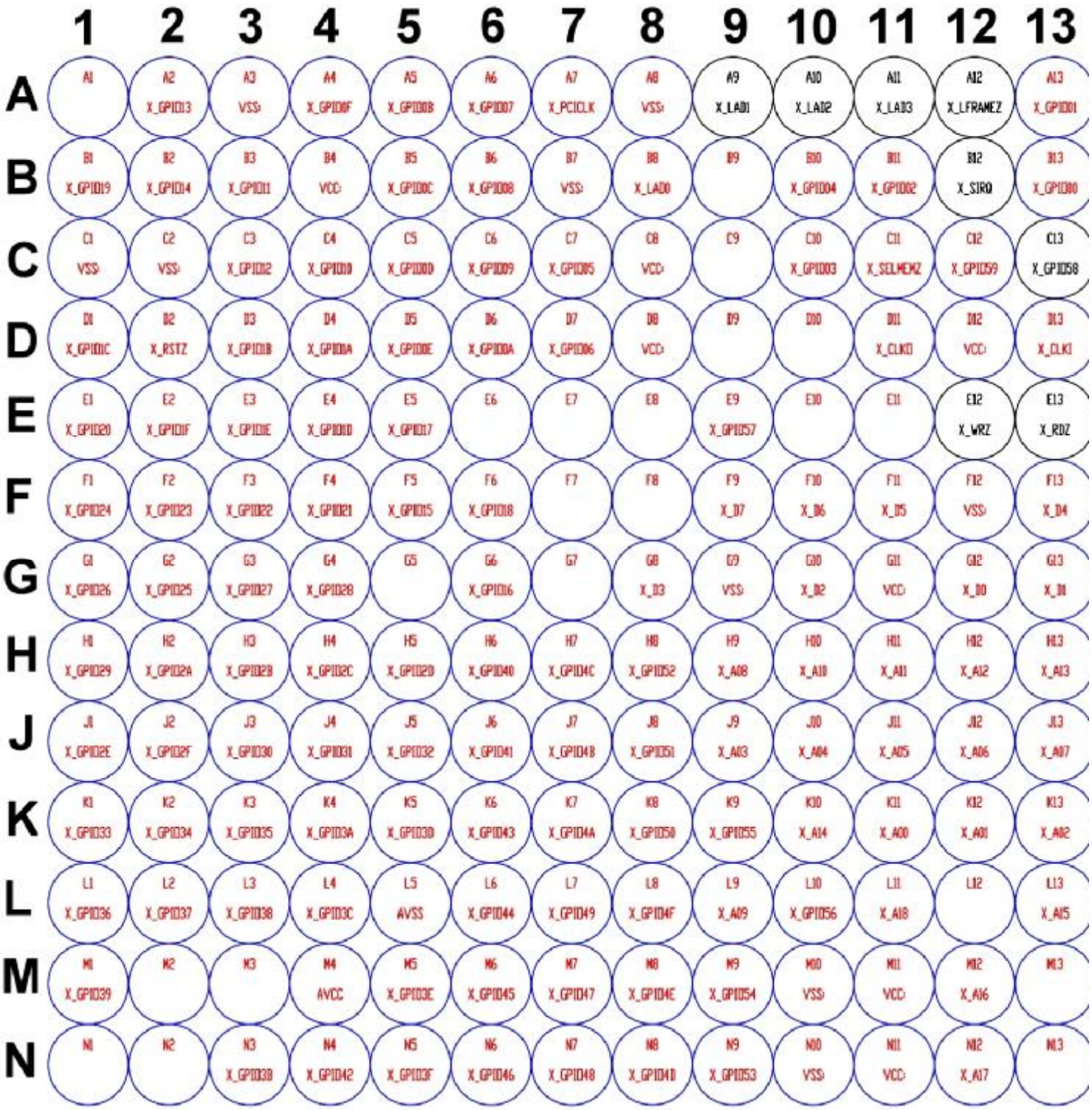


2. Pin Assignment and Description

2.1 144 pin LQFP Diagram Top View



2.2 169 GBA Diagram Top View



2.3 Pin Assignment Side A

No.	Pin Name	BGA	GPIO	OEM Function	Alt. Output	Alt. Input	Default	ECRST# Low/High	IO CELL
1	GA20	B13	GPO_00		GA20	Input is N.A.	GA20	OL / OL	BQC04HU
2	KBRST#	A13	GPO_01		KBRST#	Input is N.A.	KBRST#	OH/ OH	BQC04HU
3	SERIRQ	B12		SERIRQ				HiZ / HiZ	BCC16H
4	GPIO02	B11	GPO_02			Input is N.A.	GPO_02	HiZ / HiZ	BQC04HU
5	LFRAME#	A12		LPC_FRAME#				HiZ / HiZ	BCC16H
6	LAD3	A11		LPC_AD3				HiZ / HiZ	BCC16H
7	GPIO03	C10	GPO_03			Input is N.A.	GPO_03	HiZ / HiZ	BQC04HU
8	GPIO04	B10	GPI_04		Output is N.A.	GPWU	GPI_04	HiZ / HiZ	BQC04HU
9	LAD2	A10		LPC_AD2				HiZ / HiZ	BCC16H
10	LAD1	A9		LPC_AD1				HiZ / HiZ	BCC16H
11	VCC	C8/D8		VCC					VCC
12	LAD0	B8		LPC_AD0				HiZ / HiZ	BCC16H
13	GND	A8/B7		GND					GND
14	PCICLK	A7		CLK_PCI_EC				IE / IE	BCC16H
15	PCIRST#	C7	GPI_05		Output is N.A.	PCIRST# GPWU	GPI_05	HiZ / HiZ	BCC16H
16	GPIO06	D7	GPO_06			Input is N.A.	GPO_06	HiZ / HiZ	BQC04HU
17	GPIO07	A6	GPIO_07		I_clk_8051	GPWU	GPI_07	HiZ / HiZ	BQC04HU
18	GPIO08	B6	GPO_08		I_clk_peri	Input is N.A.	GPO_08	HiZ / HiZ	BQC04HU
19	GPIO09	C6	GPI_09		Output is N.A.	SDIDI GPWU	GPIO_09	HiZ / HiZ	BQC04HU
20	GPIO0A	D6	GPI_0A		Output is N.A.	GPWU	GPI_0A	HiZ / HiZ	BQC04HU
21	GPIO0B	A5	GPO_0B		I_clk_32Khz	Input is N.A.	GPO_0B	HiZ / HiZ	BQC04HU
22	GPIO0C	B5	GPO_0C			Input is N.A.	GPO_0C	HiZ / HiZ	BQC04HU
23	GPIO0D	C5	GPI_0D		Output is N.A.	GPWU	GPI_0D	HiZ / HiZ	BQC04HU
24	SCI#	D5	GPO_0E		SCI#	Input is N.A.	SCI#	OH / OH	BQC04HU
25	PWM1	A4	GPO_0F		PWM1	Input is N.A.	GPO_0F	HiZ / HiZ	BQC04H
26	VCC	B4		VCC					VCC
27	PWM2	C4	GPIO_10		PWM2	GPWU	GPIO_10	HiZ / HiZ	BQC04H
28	GND	A3		GND					GND
29	GPIO11	B3	GPIO_11				GPIO_11	HiZ / HiZ	BQC04HU
30	FANPWM1	C3	GPIO_12		FANPWM1		GPIO_12	HiZ / HiZ	BQC04H
31	FANPWM2	A2	GPO_13		FANPWM2	Input is N.A.	GPO_13	HiZ / HiZ	BQC04H
32	FANFB1	B2	GPIO_14			FANFB1 GPWU	GPIO_14	HiZ / HiZ	BQC04H
33	FANFB2	F5	GPIO_15			FANFB2 GPWU	GPIO_15	HiZ / HiZ	BQC04H
34	GPIO16	G6	GPIO_16		E51TXD		GPIO_16	HiZ / HiZ	BCC16H
35	GPIO17	E5	GPIO_17		E51CLK	E51RXD GPWU	GPIO_17	HiZ / HiZ	BCC16H
36	GPIO18	F6	GPIO_18			GPWU	GPIO_18	HiZ / HiZ	BQC04HU

2.4 Pin Assignment Side B

No.	Pin Name	BGA	GPIO	OEM Function	Alt. Output	Alt. Input	Default	ECRST# Low/High	IO CELL
37	VCC	B4		VCC					VCC
38	GPIO19	B1	GPO_19			Input is N.A.	GPO_19	HiZ / HiZ	BCC16H
39	GND	C1/C2		GND					GND
40	GPIO1A	D4	GPO_1A		NUMLED#	Input is N.A.	GPO_1A	HiZ / HiZ	BCC16H
41	GPIO1B	D3	GPIO_1B				GPIO_1B	HiZ / HiZ	BQC04H
42	ECRST#	D2		EC_RST#				IE / IE	BQC04HU
43	GPIO1C	D1	GPI_1C		Output is N.A.	GPWU	GPI_1C	HiZ / HiZ	BQC04HU
44	CLKRUN#	E4	GPIO_1D		CLKRUN#	CLKRUN# GPWU	GPIO_1D	HiZ / HiZ	BCC16H
45	GPIO1E	E3	GPIO_1E			GPWU	GPIO_1E	HiZ / HiZ	BQC04H
46	GPIO1F	E2	GPIO_1F			GPWU	GPIO_1F	HiZ / HiZ	BQC04H
47	KSO0	E1	GPIO_20		KSO0	TP_TEST	GPIO_20	IE / IE	BQC04HU
48	KSO1	F4	GPIO_21		KSO1	TP_PLL	GPIO_21	IE / IE	BQC04HU
49	KSO2	F3	GPIO_22		KSO2	TP_SPI	GPIO_22	IE / IE	BQC04HU
50	KSO3	F2	GPIO_23		KSO3	TP_ISP	GPIO_23	IE / IE	BQC04HU
51	KSO4	F1	GPO_24		KSO4	Input is N.A.	GPO_24	HiZ / HiZ	BQC04HU
52	KSO5	G2	GPO_25		KSO5	Input is N.A.	GPO_25	HiZ / HiZ	BQC04HU
53	KSO6	G1	GPO_26		KSO6	Input is N.A.	GPO_26	HiZ / HiZ	BQC04HU
54	KSO7	G3	GPO_27		KSO7	Input is N.A.	GPO_27	HiZ / HiZ	BQC04HU
55	KSO8	G4	GPO_28		KSO8	Input is N.A.	GPO_28	HiZ / HiZ	BQC04HU
56	KSO9	H1	GPO_29		KSO9	Input is N.A.	GPO_29	HiZ / HiZ	BQC04HU
57	KSO10	H2	GPO_2A		KSO10	Input is N.A.	GPO_2A	HiZ / HiZ	BQC04HU
58	KSO11	H3	GPO_2B		KSO11	Input is N.A.	GPO_2B	HiZ / HiZ	BQC04HU
59	KSO12	H4	GPO_2C		KSO12	Input is N.A.	GPO_2C	HiZ / HiZ	BQC04HU
60	KSO13	H5	GPO_2D		KSO13	Input is N.A.	GPO_2D	HiZ / HiZ	BQC04HU
61	KSO14	J1	GPO_2E		KSO14	Input is N.A.	GPO_2E	HiZ / HiZ	BQC04HU
62	KSO15	J2	GPO_2F		KSO15	Input is N.A.	GPO_2F	HiZ / HiZ	BQC04HU
63	KSI0	J3	GPI_30		Output is N.A.	KSI0, GPWU	GPI_30	HiZ / HiZ	BQC04HU
64	KSI1	J4	GPI_31		Output is N.A.	KSI1, GPWU	GPI_31	HiZ / HiZ	BQC04HU
65	KSI2	J5	GPI_32		Output is N.A.	KSI2, GPWU	GPI_32	HiZ / HiZ	BQC04HU
66	KSI3	K1	GPI_33		Output is N.A.	KSI3, GPWU	GPI_33	HiZ / HiZ	BQC04HU
67	KSI4	K2	GPI_34		Output is N.A.	KSI4, GPWU	GPI_34	HiZ / HiZ	BQC04HU
68	KSI5	K3	GPI_35		Output is N.A.	KSI5, GPWU	GPI_35	HiZ / HiZ	BQC04HU
69	KSI6	L1	GPI_36		Output is N.A.	KSI6, GPWU	GPI_36	HiZ / HiZ	BQC04HU
70	KSI7	L2	GPI_37		Output is N.A.	KSI7, GPWU	GPI_37	HiZ / HiZ	BQC04HU
71	AD0	L3	GPI_38		Output is N.A.	AD0, GPWU	GPI_38	HiZ / HiZ	IQA
72	AD1	M1	GPI_39		Output is N.A.	AD1, GPWU	GPI_39	HiZ / HiZ	IQA

2.5 Pin Assignment Side C

No.	Pin Name	BGA	GPIO	OEM Function	Alt. Output	Alt. Input	Default	ECRST# Low/High	IO CELL
73	AD2	K4	GPI_3A		Output is N.A.	AD2, GPWU	GPI_3A	HiZ / HiZ	IQA
74	AD3	N3	GPI_3B		Output is N.A.	AD3, GPWU	GPI_3B	HiZ / HiZ	IQA
75	AVCC	M4		AVCC					AVCC
76	DA0	L4	GPO_3C			Input is N.A.	GPO_3C	HiZ / HiZ	OC04A
77	AGND	L5		AGND					AGND
78	DA1	K5	GPO_3D			Input is N.A.	GPO_3D	HiZ / HiZ	OC04A
79	DA2	M5	GPO_3E			Input is N.A.	GPO_3E	HiZ / HiZ	OC04A
80	DA3	N5	GPO_3F			Input is N.A.	GPO_3F	HiZ / HiZ	OC04A
81	GPIO40	H6	GPI_40		Output is N.A.	GPWU	GPI_40	HiZ / HiZ	BQC04H
82	GPIO41	J6	GPIO_41				GPIO_41	HiZ / HiZ	BQC04HU
83	GPIO42	N4	GPO_42			Input is N.A.	GPO_42	HiZ / HiZ	BQC04H
84	SELIO2#	K6	GPO_43		SELIO2#	Input is N.A.	GPO_43	HiZ / HiZ	BQC04HU
85	SCL1	L6	GPIO_44		SCL1	GPWU	GPIO_44	HiZ / HiZ	BCC16H
86	SDA1	M6	GPIO_45		SDA1	GPWU	GPIO_45	HiZ / HiZ	BCC16H
87	SCL2	N6	GPIO_46		SCL2	GPWU	GPIO_46	HiZ / HiZ	BQC04H
88	SDA2	M7	GPIO_47		SDA2	GPWU	GPIO_47	HiZ / HiZ	BQC04H
89	KSO16	N7	GPIO_48		KSO16	GPWU	GPIO_48	HiZ / HiZ	BQC04HU
90	KSO17	L7	GPIO_49		KSO17	GPWU	GPIO_49	HiZ / HiZ	BQC04HU
91	PSCLK1	K7	GPIO_4A		PSCLK1	GPWU	GPIO_4A	HiZ / HiZ	BQC04H
92	PSDAT1	J7	GPIO_4B		PSDAT1	GPWU	GPIO_4B	HiZ / HiZ	BQC04H
93	PSCLK2	H7	GPIO_4C		PSCLK2	GPWU	GPIO_4C	HiZ / HiZ	BCC16H
94	PSDAT2	N8	GPIO_4D		PSDAT2	GPWU	GPIO_4D	HiZ / HiZ	BCC16H
95	PSCLK3	M8	GPIO_4E		PSCLK3	GPWU	GPIO_4E	HiZ / HiZ	BCC04H
96	PSDAT3	L8	GPIO_4F		PSDAT3	GPWU	GPIO_4F	HiZ / HiZ	BCC04H
97	SELIO#	K8	GPIO_50		SELIO#	GPWU	GPIO_50	HiZ / HiZ	BQC04H
98	A19	J8	GPIO_51		A19		A19	OL / OL	BQC04H
99	GPIO52	H8	GPIO_52		E51CS#		GPIO_52	HiZ / HiZ	BCC16H
100	GPIO53	N9	GPO_53		CAPSLED#	Input is N.A.	GPO_53	HiZ / HiZ	BCC16H
101	GPIO54	M9	GPIO_54			E51TMR0 GPWU	GPIO_54	HiZ / HiZ	BCC16H
102	GPIO55	K9	GPIO_55		SCORLED#	E51INT0	GPIO_55	HiZ / HiZ	BCC16H
103	GND	M10/N10		GND					GND
104	GPIO56	L10	GPO_56			Input is N.A.	GPO_56	HiZ / HiZ	BQC04HU
105	VCC	N11/M11		VCC					VCC
106	A18	L11	GPXA18		A18			OL / OL	BQC04HU
107	A17	N12	GPXA17		A17			OL / OL	BQC04HU
108	A16	M12	GPXA16		A16			OL / OL	BQC04HU

2.6 Pin Assignment Side D

No.	Pin Name	BGA	GPIO	OEM Function	Alt. Output	Alt. Input	Default	ECRST# Low/High	IO CELL
109	A15	L13	GPXA15					OL / OL	BQC04HU
110	A14	K10	GPXA14					OL / OL	BQC04HU
111	A0	K11	GPXA00		SDICS#			OL / OL	BQC04HU
112	A1	K12	GPXA01		SDICLK			OL / OL	BQC04HU
113	A2	K13	GPXA02		SDIDO			OL / OL	BQC04HU
114	A3	J9	GPXA03					OL / OL	BQC04HU
115	A4	J10	GPXA04					OL / OL	BQC04HU
116	A5	J11	GPXA05					OL / OL	BQC04HU
117	A6	J12	GPXA06					OL / OL	BQC04HU
118	A7	J13	GPXA07					OL / OL	BQC04HU
119	A8	H9	GPXA08					OL / OL	BQC04HU
120	A9	L9	GPXA09					OL / OL	BQC04HU
121	A10	H10	GPXA10					OL / OL	BQC04HU
122	A11	H11	GPXA11					OL / OL	BQC04HU
123	A12	H12	GPXA12					OL / OL	BQC04HU
124	A13	H13	GPXA13					OL / OL	BQC04HU
125	D0	G12	GPXD0		SDIDI			HiZ / HiZ	BQC04HU
126	D1	G13	GPXD1					HiZ / HiZ	BQC04HU
127	VCC	G11		VCC					VCC
128	D2	G10	GPXD2					HiZ / HiZ	BQC04HU
129	GND	G9		GND					GND
130	D3	G8	GPXD3					HiZ / HiZ	BQC04HU
131	D4	F13	GPXD4					HiZ / HiZ	BQC04HU
132	D5	F11	GPXD5					HiZ / HiZ	BQC04HU
133	D6	F10	GPXD6					HiZ / HiZ	BQC04HU
134	D7	F9	GPXD7					HiZ / HiZ	BQC04HU
135	RD#	E13				SPIDI	RD#	OH / OH	BQC04H
136	WR#	E12			SPIDO		WR#	OH /OH	BQC04H
137	GPIO57	E9	GPIO_57		XCLK32K	GPWU	GPIO_57	HiZ / HiZ	BQC04HU
138	XCLKI	D13		CRY1					
139	GND	F12		GND					GND
140	XCLKO	D11		CRY2					
141	VCC	G11/D12		VCC					VCC
142	GPIO58	C13	GPIO_58		SPICLK	GPWU	GPIO_58	HiZ / HiZ	BQC16HU
143	GPIO59	C12	GPIO_59			TEST_CLK SPICLKI GPWU	GPIO_59	HiZ / HiZ	BQC04HU
144	SELMEM#	C11			SPICS#		SELMEM#	OH /OH	BQC04H

2.7 I/O Cell Descriptions

2.7.1 I/O Buffer Table

IO Name	Descriptions	Applications
BQC04HU	Schmitt trigger, 2~4mA Output / Sink Current, with , Input / Output / Pull Up Enable	GPIO
BQC04H	Schmitt trigger, 2~4mA Output / Sink Current, 5 V Tolerance, Input / Output Enable	GPIO
BCC16H	8~16mA Output / Sink Current, 5 V Tolerance, Input / Output Enable	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
BQC16HU	Schmitt trigger, 8~16mA Output / Sink Current, with , Input / Output / Pull Up Enable	

*** IQA I/O buffer without Schmitt trigger.**

2.7.2 I/O Buffer Characteristic Table

Port / IO Name	IO	I	O	OE	IE	AE	5VTor	PE	Output / Sink Current
BQC04HU	V	V	V	V	V			80K	2~4mA
BQC04H	V	V	V	V	V		V		2~4mA
BCC16H	V	V	V	V	V		V		8~16mA
IQA	V		V		V	V			
OC04A	V	V		V		V			2~4mA
BQC16HU	V	V	V	V	V			80K	8~16mA

2.7.3 I/O 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
AE IO Buffer Analog mode Enable (AE > OE)
Q Schmitt Trigger
H 5V Tolerance

3. Pin Descriptions

3.1 Hardware Strap

Hardware strap pins are used to 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.

For normal application, there is no application component required for selecting the normal mode because KB3920 has built-in internal pull-up resistor for selecting the right operation mode.

Pin Name	Pin No.	HW Strap Value
TP_TEST* ***(GPIO20, KSO0)***	**47**	***TP_TEST***: Clock Test Mode **LOW**: Test Mode. **HIGH**: 32KHz clock in normal running **(MUST, Power-On Default)**
TP_PLL* ***(GPIO21, KSO1)***	**48**	***TP_PLL***: DPLL Test Mode **LOW**: Test Mode. **HIGH**: Normal operation **(MUST, Power-On Default)**
TP_SPI* ***(GPIO22, KSO2)***	**49**	***TP_SPI*** *:* Default flash access **LOW** : Boot from SPI flash part. **HIGH** : Boot from ISA flash part (**Power-On Default**)
TP_ISP* ***(GPIO23, KSO3)***	**50**	***TP_ISP :*** In system programming mode **LOW** : ISP mode. **HIGH** : Normal Mode (**Power-On Default**)

* For these hardware straps, there is a pull-up resistor embedded in the chip. For hardware strap value at high, no pull-up resistor is needed. Pull-low resistor is only required if hardware strap value is at low.

3.2 Pin Descriptions by Functionality

3.2.1 Low Pin Count Interface Pin Descriptions

Pin Name	Pin No	Direction	Description
LAD[3:0]	***6,9,10, 12***	I/O	LPC ***LAD[3:0]***
LFRAME#	***5***	I	LPC: The ***LFRAME#*** signal
PCIRST#	***15***	I	The ***PCIRST#*** signal used to reset the embedded LPC module
PCICLK	***14***	I	The 33MHz PCI Clock Input
SERIRQ	***3***	I/OD	***SERIRQ#***
CLKRUN#	***44***	I/OD	***CLKRUN#***

3.2.2 X-BUS Interface Pin Descriptions

Pin Name	Pin No	Direction	Description
RD#	***135***	O	Read pulse
WR#	***136***	O	Write pulse
SELMEM#	***144***	O	Memory cycle chip select pulse
SELIO#	***97***	O	I/O cycle chip select pulse
SELIO2#	***84***	O	I/O cycle chip select pulse. This is a derivative of ***SELIO#***
D[7:0]		I/O	Data Bus

A[19:0]	***98***	O	X-bus Address Bus.

3.2.3 PS2 Interface Pin Descriptions

Pin Name	Pin No	Direction	Description
PSCLK1	***91***	I/OD	PS2 port_1 clock
PSDAT1	***92***	I/OD	PS2 Port_1 Data
PSCLK2	***93***	I/OD	PS2 port_2 clock
PSDAT2	***94***	I/OD	PS2 port_2 data
PSCLK3	***95***	I/OD	PS2 port_3 clock
PSDAT3	***96***	I/OD	PS2 port_3 data

3.2.4 Internal Keyboard Encoder Pin Descriptions

Pin Name	Pin No	Direction	Description
KSO[17:0]		O	Keyboard Scan-Out. The pre-charge time and driving time for KSO0~17 is programmable in **IKB** register set.
KSI[7:0]		I	Keyboard Scan-In

3.2.5 SMBus Pin Descriptions

Pin Name	Pin No	Direction	Description
SCL1	***85***	I/OD	SMBus 1 Clock
SDA1	***86***	I/OD	SMBus 1 Data
SCL2	***87***	I/OD	SMBus 2 Clock
SDA2	***88***	I/OD	SMBus 2 Data

3.2.6 FAN Pin Descriptions

Pin Name	Pin No	Direction	Description
FANFB1	***32***	I	FAN1 Tachometer Input
FANFB2	***33***	I	FAN2 Tachometer Input
FANPWM1	***30***	O	FAN1 PWM output
FANPWM2	***31***	O	FAN2 PWM output

3.2.7 Pulse Width Modulation Pin Descriptions

Pin Name	Pin No	Direction	Description
PWM[1:2]	***25,27***	O	PWM pulse output.

3.2.8 Analog to Digital Pin Descriptions

Pin Name	Pin No	Direction	Description
AD[3:0]	***71-73***	I	Analog to Digital Conversion Input. The internal ADC conversion module is 8-bit in resolution. It polls ***AD0~AD3*** inputs in a round robin way.

3.2.9 Digital to Analog Pin Descriptions

Pin Name	Pin No	Direction	Description
DA[3:0]	***80,79, 78,76***	O	Digital to Analog Conversion Output with 8-bit resolution.

3.2.10 8051 External Interface Pin Descriptions

Pin Name	Pin No	Direction	Description
E51CS#	***99***	O	This pin is used for debug purpose only. The ***E51CS#*** is used to select auxiliary memory region on a development board. The auxiliary memory is usually a SRAM device used to store 8051's firmware code to facilitate real-time debugging such as setting break points. Register **XBICS** enables the mapping of the four 16K segments (SEG3-0) memory spaces individually into auxiliary memories. When a particular segment is enabled, ***E51CS#*** will be asserted instead of ***SELMEM#*** for memory accesses such that auxiliary memory device (SRAM) will be accessed instead of the flash BIOS memory. The assertion of ***E51CS#*** and ***SELMEM#*** is mutually exclusive.
E51TXD	***34***	O	These two pins are for debug purpose only. **E51TXD/ E51RXD** are the Transmit/Receive signals of the 8051's serial port. They can be connected to an external RS232 transceiver and communicate with another PC's COM port by sending debug code
E51RXD	***35***	I	
E51CLK	***35***	I	If 8051 serial port selects **MODE 0** (shift register), ***E51RXD*** becomes
E51TMR0	***101***	I	The Input pulse to the clock input of the 8051's timer_0
E51INT0	***102***	I	The interrupt signal to the embedded 8051

3.2.11 Clock Pin Descriptions

Pin name	Pin No	Direction	Description
XCLKI	***138***	I	32.768KHz Input.
XCLKO	***140***	O	32.768KHz Output

3.2.12 Miscellaneous Pin Descriptions

Pin name	Pin No	Direction	Description
SCI#	***24***	O	The ***SCI#*** signal is output to system chipset to indicate an outstanding SCI
GA20	***1***	O	The ***GA20*** signal is output to system chipset.
KBRST#	***2***	O	The ***KBRST#*** is output to system chipset to generate a system reset.
ECRST#	***42***	I	***ECRST#*** is used as the Global Reset signal for the whole chip.

3.2.13 Power Pin Descriptions

Pin name	Pin No	Direction	Description
VCC	***11,26,37***	-	Power Supply to all internal modules except Analog portions of **ADC** and
GND	***13,28,39***	-	Digital Ground
AVCC	***75***	-	Analog power to analog portions of **ADC** and **DAC**
AGND	***77***	-	Analog Ground (paired with ***VCCA***)

4. Module Descriptions

4.1 Chip Architecture

4.1.1 Power Planes

There are 2 power planes in KB3920. One is to be used for all logic; the other is to be used for Analog parts (ADC/DAC).

4.1.2 Clock Domains

There are 3 clock domains in KB3920.

1. 8051 / XBI use high clock (setting in CLKCFG, FF0Dh), ranges from 4~16Mhz.
2. WDT uses 32.768Khz clock. WDT default is driven by internal 32Khz clock. The WDTCFG bit 7 is used to switch WDT clock to external 32Khz clock oscillator.
3. Other peripherals use low clock (setting in CLKCFG, FF0Dh), ranges from 2~8Mhz.

4.1.3 Reset Domains

KB3920 builds in power-on reset. There is also an input reset signal (***ECRST#***) for global resetting. The ***ECRST#*** may be tied to VCC directly.

WDT Reset will reset almost all logic, except WDT and GPIO modules. The WDT reset can be configured to only reset 8051 by EC register (PXCFG, FF14h).

There is an additional 8051-reset source from EC register (PXCFG, FF14h).

4.1.4 Internal Memory Map

No.	Abbreviation	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	
9	**SDI**	SPI Device Interface	FE70h~FE7Fh	16	

10	**WDT**	Watchdog Timer	FE80h~FE8Fh	16	
11	**LPC**	Low Pin Count	FE90h~FE9Fh	16	
12	**XBI**	X-BUS Interface	FEA0h~FECFh	48	
13	**XIO**	IO Expender	FED0h~FEDFh	16	
14	**PS2**	PS2	FEE0h~FEFFh	32	
15	**EC**	Embedded Controller **(hardware EC Space)**	FF00h~FF1Fh	32	
16	**GPWU**	General Purpose Wake-up **(hardware EC Space)**	FF20h~FE7Fh	96	
17	**SMBus**	System Management BUS **(hardware EC Space)**	FF80h~FFFFh	128	

4.2 GPIO

4.2.1 GPIO Functional Description

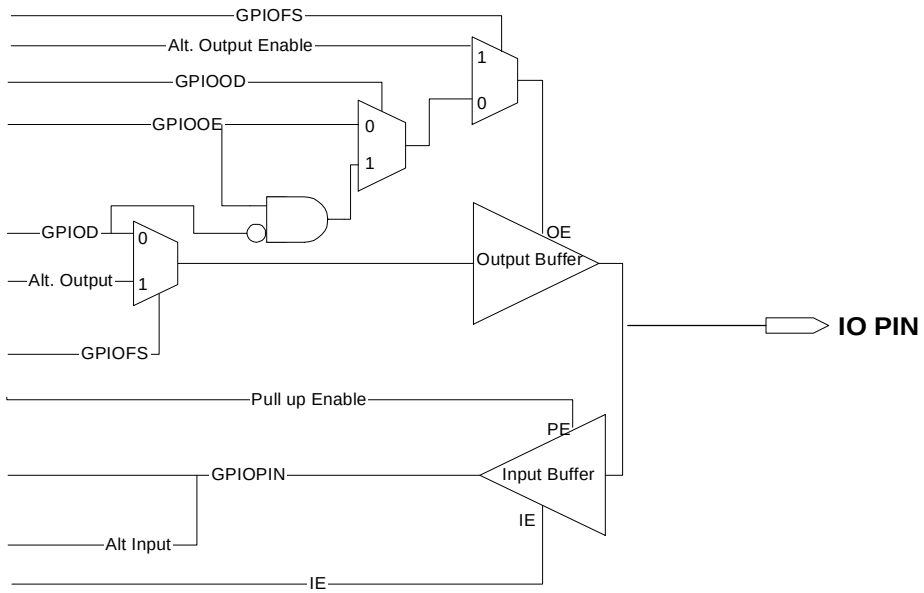
Multi-function pin **Output Function Selection** (FS) bit = 0, is set for GPIO Output Function, and FS bit = 1, is set for Alternative Output. The alternative input function is enabled by Input Enable register (IE), and is not affected by FS register.

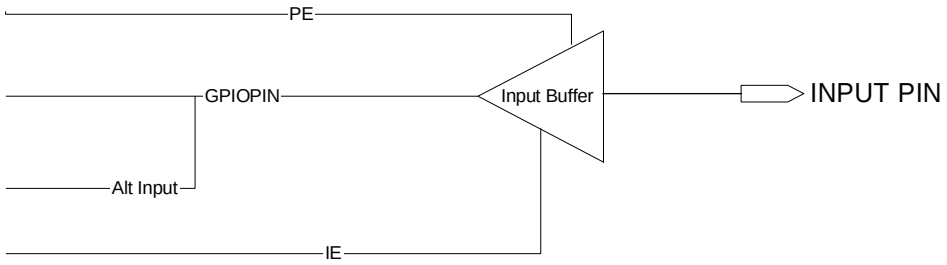
GPIO No	Alt. Output	Alt. Input	Default Alt. Output	Alt. Sel. Reg.	User Define	User Alt. define
GPIO00	***GA20***		***GA20***	GPIOFS00.0		
GPIO01	***KBRST#***		***KBRST#***	GPIOFS00.1		
GPIO02			*GPO_02*	GPIOFS00.2		
GPIO03			*GPO_03*	GPIOFS00.3		
GPIO04			*GPI_04*	GPIOFS00.4		
GPIO05		***PCIRST#***	*GPI_05*	GPIOFS00.5		
GPIO06			*GPO_06*	GPIOFS00.6		
GPIO07	***i_clk (8051)***		*GPI_07*	GPIOFS00.7		
GPIO08	***i_clk (peripheral)***		*GPO_08*	GPIOFS08.0		
GPIO09			*GPO_09*	GPIOFS08.1		
GPIO0A			*GPI_0A*	GPIOFS08.2		
GPIO0B	***i_clk 32khz (XCLKI)***		*GPO_0B*	GPIOFS08.3		
GPIO0C			*GPO_0C*	GPIOFS08.4		
GPIO0D			*GPI_0D*	GPIOFS08.5		
GPIO0E	***SCI#***		***SCI#***	GPIOFS08.6		
GPIO0F	***PWM1***		*GPO_0F*	GPIOFS08.7		
GPIO10	***PWM2***		*GPIO_10*	GPIOFS10.0		
GPIO11			*GPIO_11*	GPIOFS10.1		
GPIO12	***FANPWM1***		*GPIO_12*	GPIOFS10.2		
GPIO13	***FANPWM2***		*GPO_13*	GPIOFS10.3		
GPIO14		***FANFB1***	*GPIO_14*	GPIOFS10.4		
GPIO15		***FANFB2***	*GPIO_15*	GPIOFS10.5		

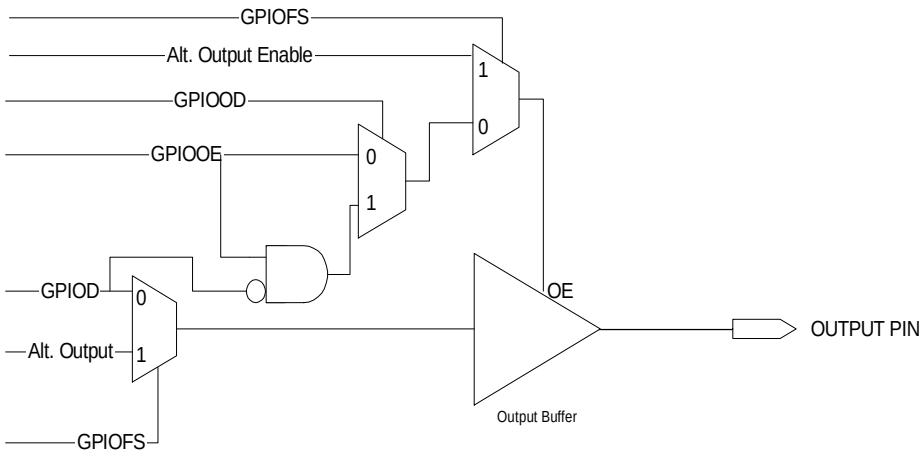
GPIO16	***E51TXD***		*GPIO_16*	GPIOFS10.6		
GPIO17	***E51CLK***	***E51RXD***	*GPIO_17*	GPIOFS10.7		
GPIO18			*GPIO_18*	GPIOFS18.0		
GPIO19			*GPO_19*	GPIOFS18.1		
GPIO1A	***NUMLED#***		*GPO_1A*	GPIOFS18.2		
GPIO1B			*GPIO_1B*	GPIOFS18.3		
GPIO1C			*GPI_1C*	GPIOFS18.4		
GPIO1D	***CLKRUN#***	***CLKRUN#***	*GPIO_1D*	GPIOFS18.5		
GPIO1E			*GPIO_1E*	GPIOFS18.6		
GPIO1F			*GPIO_1F*	GPIOFS18.7		
GPIO20~2F	***KSO0~15***	***TP_DPLL***	*GPIO_20~GPIO_23 GPO_24~GPO_2F*	GPIOFS20		
		TP_TEST		GPIOFS28		
GPIO30		***KSI0***	*GPI_30*	GPIOFS30.0		
GPIO31		***KSI1***	*GPI_31*	GPIOFS30.1		
GPIO32		***KSI2***	*GPI_32*	GPIOFS30.2		
GPIO33		***KSI3***	*GPI_33*	GPIOFS30.3		
GPIO34		***KSI4***	*GPI_34*	GPIOFS30.4		
GPIO35		***KSI5***	*GPI_35*	GPIOFS30.5		
GPIO36		***KSI6***	*GPI_36*	GPIOFS30.6		
GPIO37		***KSI7***	*GPI_37*	GPIOFS30.7		
GPI38			***GPI_38***	GPIOFS38.0		
GPI39			***GPI_39***	GPIOFS38.1		
GPI3A			***GPI_3A***	GPIOFS38.2		
GPI3B			***GPI_3B***	GPIOFS38.3		
GPO3C			***GPO_3C***	GPIOFS38.4		
GPO3D			***GPO_3D***	GPIOFS38.5		
GPO3E			***GPO_3E***	GPIOFS38.6		
GPO3F			***GPO_3F***	GPIOFS38.7		
GPIO40			*GPI_40*	GPIOFS40.0		
GPIO41			*GPIO_41*	GPIOFS40.1		
GPIO42			*GPO_42*	GPIOFS40.2		
GPIO43	***SELIO2#***		*GPO_43*	GPIOFS40.3		
GPIO44	***SCL1***	***SCL1***	*GPIO_44*	GPIOFS40.4		
GPIO45	***SDA1***	***SDA1***	*GPIO_45*	GPIOFS40.5		
GPIO46	***SCL2***	***SCL2***	*GPIO_46*	GPIOFS40.6		
GPIO47	***SDA2***	***SDA2***	*GPIO_47*	GPIOFS40.7		
GPIO48	***KSO16***		*GPIO_48*	GPIOFS48.0		

GPIO49	***KSO17***		*GPIO_49*	GPIOFS48.1		
GPIO4A	***PSCLK1***	***PSCLK1***	*GPIO_4A*	GPIOFS48.2		
GPIO4B	***PSDAT1***	***PSDAT1***	*GPIO_4B*	GPIOFS48.3		
GPIO4C	***PSCLK2***	***PSCLK2***	*GPIO_4C*	GPIOFS48.4		
GPIO4D	***PSDAT2***	***PSDAT2***	*GPIO_4D*	GPIOFS48.5		
GPIO4E	***PSCLK3***	***PSCLK3***	*GPIO_4E*	GPIOFS48.6		
GPIO4F	***PSDAT3***	***PSDAT3***	*GPIO_4F*	GPIOFS48.7		
GPIO50	***SELIO#***		*GPIO_50*	GPIOFS50.0		
GPIO51	***A19***		***A19***	GPIOFS50.1		
GPIO52	***E51CS#***		*GPIO_52*	GPIOFS50.2		
GPIO53	***CAPLED#***		*GPO_53*	GPIOFS50.3		
GPIO54		***E51TMR0***	*GPIO_54*	GPIOFS50.4		
GPIO55	***SCROLED#***	***E51INT0***	*GPIO_55*	GPIOFS50.5		
GPIO56			*GPO_56*	GPIOFS50.6		
GPIO57			*GPIO_57*	GPIOFS50.7		
GPIO58			*GPIO_58*	GPIOFS58.0		
GPIO59		***TEST_CLK***	*GPIO_59*	GPIOFS58.1		

4.2.2 GPIO Input / Output Control Structure







4.2.3 GPIO Registers Descriptions

(Base Address=FC00h, 128 bytes)

GPIO Output Function Selection

This register is used to select the function of GPO or Alternative Output of each GPIO.

For those GPI only, setting the corresponding bit in register is no effect. The detail alternative functions of each I/O can be referred in Section 2.2~2.4, Pin Assignment Side.

(0:GPO, 1:Alternative Output)

In bitmap, where I for GPI only, O for GPO only, and I/O for GPIO.

Offset	Reg Abbr.	Attr	bitmap			Default	Bank
00	GPIOFS00 (GPIO00~GPIO07)	R/W	7	GPIO07	I	03h	FC
			6	GPIO06	O		
			5	GPIO05	I		
			4	GPIO04	I		
			3	GPIO03	O		
			2	GPIO02	O		
			1	GPIO01	O		
			0	GPIO00	O		
01	GPIOFS08 (GPIO08~GPIO0F)	R/W	7	GPIO0F	O	40h	FC
			6	GPIO0E	O		
			5	GPIO0D	I		
			4	GPIO0C	O		
			3	GPIO0B	O		
			2	GPIO0A	I		
			1	GPIO09	I/O		
			0	GPIO08	O		
02	GPIOFS10 (GPIO10~GPIO17)	R/W	7	GPIO17	I/O	00h	FC
			6	GPIO16	I/O		
			5	GPIO15	I/O		
			4	GPIO14	I/O		
			3	GPIO13	O		
			2	GPIO12	I/O		
			1	GPIO11	I/O		
			0	GPIO10	I/O		
03	GPIOFS18 (GPIO18~GPIO1F)	R/W	7	GPIO1F	I/O	00h	FC
			6	GPIO1E	I/O		
			5	GPIO1D	I/O		
			4	GPIO1C	I		
			3	GPIO1B	I/O		
			2	GPIO1A	O		
			1	GPIO19	O		
			0	GPIO18	I/O		
04	GPIOFS20 (GPIO20~GPIO27)	R/W	7	GPIO27	O	00h	FC
			6	GPIO26	O		
			5	GPIO25	O		
			4	GPIO24	O		
			3	GPIO23	I/O		
			2	GPIO22	I/O		
			1	GPIO21	I/O		
			0	GPIO20	I/O		
05	GPIOFS28 (GPIO28~GPIO2F)	R/W	7	GPIO2F	O	00h	FC
			6	GPIO2E	O		
			5	GPIO2D	O		
			4	GPIO2C	O		
			3	GPIO2B	O		
			2	GPIO2A	O		
			1	GPIO29	O		
			0	GPIO28	O		
06	GPIOFS30 (GPIO30~GPIO37)	R/W	7	GPIO37	I	00h	FC
			6	GPIO36	I		
			5	GPIO35	I		
			4	GPIO34	I		
			3	GPIO33	I		
			2	GPIO32	I		
			1	GPIO31	I		
			0	GPIO30	I		

07	GPIOFS38 (GPIO38~GPIO3F)	R/W	7	GPIO3F	O	00h	FC
			6	GPIO3E	O		
			5	GPIO3D	O		
			4	GPIO3C	O		
			3	GPIO3B	I		
			2	GPIO3A	I		
			1	GPIO39	I		
			0	GPIO38	I		
08	GPIOFS40 (GPIO40~GPIO47)	R/W	7	GPIO47	I/O	00h	FC
			6	GPIO46	I/O		
			5	GPIO45	I/O		
			4	GPIO44	I/O		
			3	GPIO43	O		
			2	GPIO42	O		
			1	GPIO41	I/O		
			0	GPIO40	I		
09	GPIOFS48 (GPIO48~GPIO4F)	R/W	7	GPIO4F	I/O	00h	FC
			6	GPIO4E	I/O		
			5	GPIO4D	I/O		
			4	GPIO4C	I/O		
			3	GPIO4B	I/O		
			2	GPIO4A	I/O		
			1	GPIO49	I/O		
			0	GPIO48	I/O		
0A	GPIOFS50 (GPIO50~GPIO57)	R/W	7	GPIO57	I/O	02h	FC
			6	GPIO56	O		
			5	GPIO55	I/O		
			4	GPIO54	I/O		
			3	GPIO53	O		
			2	GPIO52	I/O		
			1	GPIO51	I/O		
			0	GPIO50	I/O		
0B	GPIOFS58 (GPIO58~GPIO5F)	R/W	7	GPIO5F	N/A	00h	FC
			6	GPIO5E	N/A		
			5	GPIO5D	N/A		
			4	GPIO5C	N/A		
			3	GPIO5B	N/A		
			2	GPIO5A	N/A		
			1	GPIO59	I/O		
			0	GPIO58	I/O		

GPIO Output Enable Register

This register is used to enable or disable output function of GPIO. These registers are no use for those pins which are GPI only. Also if alternative output selected, this register is no effect on corresponding GPIO pin. For those GPXA00~GPAX18 can be utilized as GPO function.

(0:Disable Output 1:Enable Output)

In bitmap, where I for GPI only, O for GPO only, and I/O for GPIO.

Offset	Reg Abbr.	Attr		bitmap		Default	Bank
10	GPIOOE00 (GPIO00~GPIO07)	R/W	7	GPIO07	I	03	FC
			6	GPIO06	O		
			5	GPIO05	I		
			4	GPIO04	I		
			3	GPIO03	O		
			2	GPIO02	O		
			1	GPIO01	O		
			0	GPIO00	O		
11	GPIOOE08 (GPIO08~GPIO0F)	R/W	7	GPIO0F	O	40h	FC
			6	GPIO0E	O		
			5	GPIO0D	I		
			4	GPIO0C	O		
			3	GPIO0B	O		
			2	GPIO0A	I		
			1	GPIO09	I/O		
			0	GPIO08	O		
12	GPIOOE10	R/W	7	GPIO17	I/O	00h	FC

	(GPIO10~GPIO17)		6	GPIO16	I/O		
			5	GPIO15	I/O		
			4	GPIO14	I/O		
			3	GPIO13	O		
			2	GPIO12	I/O		
			1	GPIO11	I/O		
			0	GPIO10	I/O		
13	GPIOOE18 (GPIO18~GPIO1F)	R/W	7	GPIO1F	I/O	00h	FC
			6	GPIO1E	I/O		
			5	GPIO1D	I/O		
			4	GPIO1C	I		
			3	GPIO1B	I/O		
			2	GPIO1A	O		
			1	GPIO19	O		
			0	GPIO18	I/O		
14	GPIOOE20 (GPIO20~GPIO27)	R/W	7	GPIO27	O	00h	FC
			6	GPIO26	O		
			5	GPIO25	O		
			4	GPIO24	O		
			3	GPIO23	I/O		
			2	GPIO22	I/O		
			1	GPIO21	I/O		
			0	GPIO20	I/O		
15	GPIOOE28 (GPIO28~GPIO2F)	R/W	7	GPIO2F	O	00h	FC
			6	GPIO2E	O		
			5	GPIO2D	O		
			4	GPIO2C	O		
			3	GPIO2B	O		
			2	GPIO2A	O		
			1	GPIO29	O		
			0	GPIO28	O		
16	GPIOOE30 (GPIO30~GPIO37)	R/W	7	GPIO37	I	00h	FC
			6	GPIO36	I		
			5	GPIO35	I		
			4	GPIO34	I		
			3	GPIO33	I		
			2	GPIO32	I		
			1	GPIO31	I		
			0	GPIO30	I		
17	GPIOOE38 (GPIO38~GPIO3F)	R/W	7	GPIO3F	O	00h	FC
			6	GPIO3E	O		
			5	GPIO3D	O		
			4	GPIO3C	O		
			3	GPIO3B	I		
			2	GPIO3A	I		
			1	GPIO39	I		
			0	GPIO38	I		
18	GPIOOE40 (GPIO40~GPIO47)	R/W	7	GPIO47	I/O	00h	FC
			6	GPIO46	I/O		
			5	GPIO45	I/O		
			4	GPIO44	I/O		
			3	GPIO43	O		
			2	GPIO42	O		
			1	GPIO41	I/O		
			0	GPIO40	I		
19	GPIOOE48 (GPIO48~GPIO4F)	R/W	7	GPIO4F	I/O	00h	FC
			6	GPIO4E	I/O		
			5	GPIO4D	I/O		
			4	GPIO4C	I/O		
			3	GPIO4B	I/O		
			2	GPIO4A	I/O		
			1	GPIO49	I/O		
			0	GPIO48	I/O		
1A	GPIOOE50 (GPIO50~GPIO57)	R/W	7	GPIO57	I/O	02h	FC
			6	GPIO56	O		
			5	GPIO55	I/O		
			4	GPIO54	I/O		
			3	GPIO53	O		
			2	GPIO52	I/O		
			1	GPIO51	I/O		
			0	GPIO50	I/O		

1B	GPIOOE58 (GPIO58~GPIO5F)	R/W	7	GPIO5F	N/A	00h	FC
			6	GPIO5E	N/A		
			5	GPIO5D	N/A		
			4	GPIO5C	N/A		
			3	GPIO5B	N/A		
			2	GPIO5A	N/A		
			1	GPIO59	I/O		
			0	GPIO58	I/O		
1C	GPXAOE00 (GPXA0~GPXA7)	R/W	7	GPXA07	GPXA / O	00h	FC
			6	GPXA06	GPXA / O		
			5	GPXA05	GPXA / O		
			4	GPXA04	GPXA / O		
			3	GPXA03	GPXA / O		
			2	GPXA02	GPXA / O		
			1	GPXA01	GPXA / O		
			0	GPXA00	GPXA / O		
1D	GPXAOE08 (GPXA8~GPXA15)	R/W	7	GPXA15	GPXA / O	00h	FC
			6	GPXA14	GPXA / O		
			5	GPXA13	GPXA / O		
			4	GPXA12	GPXA / O		
			3	GPXA11	GPXA / O		
			2	GPXA10	GPXA / O		
			1	GPXA09	GPXA / O		
			0	GPXA08	GPXA / O		
1E	GPXAOE16 (GPXA16~GPX18)	R/W	7	N/A		00h	FC
			6	N/A			
			5	N/A			
			4	N/A			
			3	N/A			
			2	GPXA18	GPXA / O		
			1	GPXA17	GPXA / O		
			0	GPXA16	GPXA / O		
1F	Reserved						

GPIO Output Data Register

While GPO function chosen, these registers are utilized to toggle GPO data high or low. If alternative output selected or pin of GPI only, the corresponding bit is no effect.

(0: Data Output Low 1:Data Output High)

In bitmap, where I for GPI only, O for GPO only, and I/O for GPIO.

Offset	Reg Abbr.	Attr		bitmap		Default	Bank
20	GPIOFS00 GPIOD00 (GPIO00~GPIO07)	R/W	7	GPIO07	I	00h	FC
			6	GPIO06	O		
			5	GPIO05	I		
			4	GPIO04	I		
			3	GPIO03	O		
			2	GPIO02	O		
			1	GPIO01	O		
			0	GPIO00	O		
21	GPIOD08 (GPIO08~GPIO0F)	R/W	7	GPIO0F	O	00h	FC
			6	GPIO0E	O		
			5	GPIO0D	I		
			4	GPIO0C	O		
			3	GPIO0B	O		
			2	GPIO0A	I		
			1	GPIO09	I/O		
			0	GPIO08	O		
22	GPIOD10 (GPIO10~GPIO17)	R/W	7	GPIO17	I/O	00h	FC
			6	GPIO16	I/O		
			5	GPIO15	I/O		
			4	GPIO14	I/O		
			3	GPIO13	O		
			2	GPIO12	I/O		
			1	GPIO11	I/O		
			0	GPIO10	I/O		
23	GPIOD18	R/W	7	GPIO1F	I/O	00h	FC

	(GPIO18~GPIO1F)		6	GPIO1E	I/O		
			5	GPIO1D	I/O		
			4	GPIO1C	I		
			3	GPIO1B	I/O		
			2	GPIO1A	O		
			1	GPIO19	O		
			0	GPIO18	I/O		
24	GPIOD20 (GPIO20~GPIO27)	R/W	7	GPIO27	O	00h	FC
			6	GPIO26	O		
			5	GPIO25	O		
			4	GPIO24	O		
			3	GPIO23	I/O		
			2	GPIO22	I/O		
			1	GPIO21	I/O		
			0	GPIO20	I/O		
25	GPIOD28 (GPIO28~GPIO2F)	R/W	7	GPIO2F	O	00h	FC
			6	GPIO2E	O		
			5	GPIO2D	O		
			4	GPIO2C	O		
			3	GPIO2B	O		
			2	GPIO2A	O		
			1	GPIO29	O		
			0	GPIO28	O		
26	GPIOD30 (GPIO30~GPIO37)	R/W	7	GPIO37	I	00h	FC
			6	GPIO36	I		
			5	GPIO35	I		
			4	GPIO34	I		
			3	GPIO33	I		
			2	GPIO32	I		
			1	GPIO31	I		
			0	GPIO30	I		
27	GPIOD38 (GPIO38~GPIO3F)	R/W	7	GPIO3F	O	00h	FC
			6	GPIO3E	O		
			5	GPIO3D	O		
			4	GPIO3C	O		
			3	GPIO3B	I		
			2	GPIO3A	I		
			1	GPIO39	I		
			0	GPIO38	I		
28	GPIOD40 (GPIO40~GPIO47)	R/W	7	GPIO47	I/O	00h	FC
			6	GPIO46	I/O		
			5	GPIO45	I/O		
			4	GPIO44	I/O		
			3	GPIO43	O		
			2	GPIO42	O		
			1	GPIO41	I/O		
			0	GPIO40	I		
29	GPIOD48 (GPIO48~GPIO4F)	R/W	7	GPIO4F	I/O	00h	FC
			6	GPIO4E	I/O		
			5	GPIO4D	I/O		
			4	GPIO4C	I/O		
			3	GPIO4B	I/O		
			2	GPIO4A	I/O		
			1	GPIO49	I/O		
			0	GPIO48	I/O		
2A	GPIOD50 (GPIO50~GPIO57)	R/W	7	GPIO57	I/O	00h	FC
			6	GPIO56	O		
			5	GPIO55	I/O		
			4	GPIO54	I/O		
			3	GPIO53	O		
			2	GPIO52	I/O		
			1	GPIO51	I/O		
			0	GPIO50	I/O		
2B	GPIOD58 (GPIO58~GPIO5F)	R/W	7	GPIO5F	N/A	00h	FC
			6	GPIO5E	N/A		
			5	GPIO5D	N/A		
			4	GPIO5C	N/A		
			3	GPIO5B	N/A		
			2	GPIO5A	N/A		
			1	GPIO59	I/O		
			0	GPIO58	I/O		

2C	GPXAOE00 (GPXA0~GPXA7)	R/W	7	GPXA07	GPXA / O	00h	FC
			6	GPXA06	GPXA / O		
			5	GPXA05	GPXA / O		
			4	GPXA04	GPXA / O		
			3	GPXA03	GPXA / O		
			2	GPXA02	GPXA / O		
			1	GPXA01	GPXA / O		
			0	GPXA00	GPXA / O		
2D	GPXAOE08 (GPXA8~GPXA15)	R/W	7	GPXA15	GPXA / O	00h	FC
			6	GPXA14	GPXA / O		
			5	GPXA13	GPXA / O		
			4	GPXA12	GPXA / O		
			3	GPXA11	GPXA / O		
			2	GPXA10	GPXA / O		
			1	GPXA09	GPXA / O		
			0	GPXA08	GPXA / O		
2E	GPXAOE16 (GPXA16~GPX18)	R/W	7	N/A		00h	FC
			6	N/A			
			5	N/A			
			4	N/A			
			3	N/A			
			2	GPXA18	GPXA / O		
			1	GPXA17	GPXA / O		
			0	GPXA16	GPXA / O		
2F	Reserved						

GPIO Input Status Register

While GPI function chosen, reading these registers reflects the input data. For pins of GPO only, the corresponding register bitmap is no meaning.

(0: Input Data Low 1:Input Data High)

In bitmap, where I for GPI only, O for GPO only, and I/O for GPIO.

Offset	Reg Abbr.	Attr	bitmap			Default	Bank
30	GPIOIN 00 (GPIO00~GPIO07)	R/W	7	GPIO07	I	FFh	FC
			6	GPIO06	O		
			5	GPIO05	I		
			4	GPIO04	I		
			3	GPIO03	O		
			2	GPIO02	O		
			1	GPIO01	O		
			0	GPIO00	O		
31	GPIOIN08 (GPIO08~GPIO0F)	R/W	7	GPIO0F	O	FFh	FC
			6	GPIO0E	O		
			5	GPIO0D	I		
			4	GPIO0C	O		
			3	GPIO0B	O		
			2	GPIO0A	I		
			1	GPIO09	I/O		
			0	GPIO08	O		
32	GPIOIN10 (GPIO10~GPIO17)	R/W	7	GPIO17	I/O	FFh	FC
			6	GPIO16	I/O		
			5	GPIO15	I/O		
			4	GPIO14	I/O		
			3	GPIO13	O		
			2	GPIO12	I/O		
			1	GPIO11	I/O		
			0	GPIO10	I/O		
33	GPIOIN18 (GPIO18~GPIO1F)	R/W	7	GPIO1F	I/O	FFh	FC
			6	GPIO1E	I/O		
			5	GPIO1D	I/O		
			4	GPIO1C	I		
			3	GPIO1B	I/O		
			2	GPIO1A	O		
			1	GPIO19	O		
			0	GPIO18	I/O		
34	GPIOIN20 (GPIO20~GPIO27)	R/W	7	GPIO27	O	FFh	FC
			6	GPIO26	O		
			5	GPIO25	O		

			4	GPIO24	O		
			3	GPIO23	I/O		
			2	GPIO22	I/O		
			1	GPIO21	I/O		
			0	GPIO20	I/O		
35	GPIOIN 28 (GPIO28~GPIO2F)	R/W	7	GPIO2F	O	FFh	FC
			6	GPIO2E	O		
			5	GPIO2D	O		
			4	GPIO2C	O		
			3	GPIO2B	O		
			2	GPIO2A	O		
			1	GPIO29	O		
			0	GPIO28	O		
36	GPIOIN30 (GPIO30~GPIO37)	R/W	7	GPIO37	I	FFh	FC
			6	GPIO36	I		
			5	GPIO35	I		
			4	GPIO34	I		
			3	GPIO33	I		
			2	GPIO32	I		
			1	GPIO31	I		
			0	GPIO30	I		
37	GPIOIN38 (GPIO38~GPIO3F)	R/W	7	GPIO3F	O	FFh	FC
			6	GPIO3E	O		
			5	GPIO3D	O		
			4	GPIO3C	O		
			3	GPIO3B	I		
			2	GPIO3A	I		
			1	GPIO39	I		
			0	GPIO38	I		
38	GPIOIN40 (GPIO40~GPIO47)	R/W	7	GPIO47	I/O	FFh	FC
			6	GPIO46	I/O		
			5	GPIO45	I/O		
			4	GPIO44	I/O		
			3	GPIO43	O		
			2	GPIO42	O		
			1	GPIO41	I/O		
			0	GPIO40	I		
39	GPIOIN48 (GPIO48~GPIO4F)	R/W	7	GPIO4F	I/O	FFh	FC
			6	GPIO4E	I/O		
			5	GPIO4D	I/O		
			4	GPIO4C	I/O		
			3	GPIO4B	I/O		
			2	GPIO4A	I/O		
			1	GPIO49	I/O		
			0	GPIO48	I/O		
3A	GPIOIN50 (GPIO50~GPIO57)	R/W	7	GPIO57	I/O	FFh	FC
			6	GPIO56	O		
			5	GPIO55	I/O		
			4	GPIO54	I/O		
			3	GPIO53	O		
			2	GPIO52	I/O		
			1	GPIO51	I/O		
			0	GPIO50	I/O		
3B	GPIOIN58 (GPIO58~GPIO5F)	R/W	7	GPIO5F	N/A	FFh	FC
			6	GPIO5E	N/A		
			5	GPIO5D	N/A		
			4	GPIO5C	N/A		
			3	GPIO5B	N/A		
			2	GPIO5A	N/A		
			1	GPIO59	I/O		
			0	GPIO58	I/O		
3C~3E	Reserved						
3F	GPXDIN00 (GPXD00~GPXD07)	R/W	7	GPXD07	GPXD / I		FC
			6	GPXD06	GPXD / I		
			5	GPXD05	GPXD / I		
			4	GPXD04	GPXD / I		
			3	GPXD03	GPXD / I		
			2	GPXD02	GPXD / I		
			1	GPXD01	GPXD / I		
			0	GPXD00	GPXD / I		

GPIO Pull-up Enable Register

While GPI function chosen, each GPI or GPIO pin can be programmed to be pulled high 3.3v internally. These registers are no meaning for pins of GPO only.

(0: pull-up disable 1:pull-up enable)

In bitmap, where I for GPI only, O for GPO only, and I/O for GPIO. PU for I/O with internal Pull-up resistor.

Offset	Reg Abbr.	Attr	Bitmap				Default	Bank
40	GPIOPU00 (GPIO00~GPIO07)	R/W	7	GPIO07	I	PU	00h	FC
			6	GPIO06	O			
			5	GPIO05	I			
			4	GPIO04	I	PU		
			3	GPIO03	O			
			2	GPIO02	O			
			1	GPIO01	O			
			0	GPIO00	O			
41	GPIOPU08 (GPIO08~GPIO0F)	R/W	7	GPIO0F	O		00h	FC
			6	GPIO0E	O			
			5	GPIO0D	I	PU		
			4	GPIO0C	O			
			3	GPIO0B	O			
			2	GPIO0A	I	PU		
			1	GPIO09	I/O	PU		
			0	GPIO08	O			
42	GPIOPU10 (GPIO10~GPIO17)	R/W	7	GPIO17	I/O		00h	FC
			6	GPIO16	I/O			
			5	GPIO15	I/O			
			4	GPIO14	I/O			
			3	GPIO13	O			
			2	GPIO12	I/O			
			1	GPIO11	I/O	PU		
			0	GPIO10	I/O			
43	GPIOPU18 (GPIO18~GPIO1F)	R/W	7	GPIO1F	I/O		00h	FC
			6	GPIO1E	I/O			
			5	GPIO1D	I/O			
			4	GPIO1C	I	PU		
			3	GPIO1B	I/O			
			2	GPIO1A	O			
			1	GPIO19	O			
			0	GPIO18	I/O	PU		
44	GPIOPU 20 (GPIO20~GPIO27)	R/W	7	GPIO27	O		03h	FC
			6	GPIO26	O			
			5	GPIO25	O			
			4	GPIO24	O			
			3	GPIO23	I/O	PU		
			2	GPIO22	I/O	PU		
			1	GPIO21	I/O	PU		
			0	GPIO20	I/O	PU		
45	GPIOPU28 (GPIO28~GPIO2F)	R/W	7	GPIO2F	O		00h	FC
			6	GPIO2E	O			
			5	GPIO2D	O			
			4	GPIO2C	O			
			3	GPIO2B	O			
			2	GPIO2A	O			
			1	GPIO29	O			
			0	GPIO28	O			
46	GPIOPU30 (GPIO30~GPIO37)	R/W	7	GPIO37	I	PU	FFh	FC
			6	GPIO36	I	PU		
			5	GPIO35	I	PU		
			4	GPIO34	I	PU		
			3	GPIO33	I	PU		
			2	GPIO32	I	PU		
			1	GPIO31	I	PU		
			0	GPIO30	I	PU		
47	GPIOPU38 (GPIO38~GPIO3F)	R/W	7	GPIO3F	O		00h	FC
			6	GPIO3E	O			
			5	GPIO3D	O			
			4	GPIO3C	O			

			3	GPIO3B	I			
			2	GPIO3A	I			
			1	GPIO39	I			
			0	GPIO38	I			
48	GPIOPU40 (GPIO40~GPIO47)	R/W	7	GPIO47	I/O		00h	FC
			6	GPIO46	I/O			
			5	GPIO45	I/O			
			4	GPIO44	I/O			
			3	GPIO43	O			
			2	GPIO42	O			
			1	GPIO41	I/O	PU		
			0	GPIO40	I			
49	GPIOPU48 (GPIO48~GPIO4F)	R/W	7	GPIO4F	I/O		00h	FC
			6	GPIO4E	I/O			
			5	GPIO4D	I/O			
			4	GPIO4C	I/O			
			3	GPIO4B	I/O			
			2	GPIO4A	I/O			
			1	GPIO49	I/O	PU		
			0	GPIO48	I/O	PU		
4A	GPIOPU50 (GPIO50~GPIO57)	R/W	7	GPIO57	I/O	PU	00h	FC
			6	GPIO56	O			
			5	GPIO55	I/O			
			4	GPIO54	I/O			
			3	GPIO53	O			
			2	GPIO52	I/O			
			1	GPIO51	I/O			
			0	GPIO50	I/O			
4B	GPIOPU58 (GPIO58~GPIO5F)	R/W	7	GPIO5F	N/A		00h	FC
			6	GPIO5E	N/A			
			5	GPIO5D	N/A			
			4	GPIO5C	N/A			
			3	GPIO5B	N/A			
			2	GPIO5A	N/A			
			1	GPIO59	I/O	PU		
			0	GPIO58	I/O	PU		

GPIO Open-Drain Enable Register

While GPO function chosen, each GPO pin can be programmed to be open-drain internally. These registers are no meaning for pins of GPI only. While open-drain enabled, writing 0 to related GPIO output data register cause the corresponding output pin to be low, and writing 1 to be floating. *All output pins can be set to open-drain.*

(0:open-drain disable 1:open-drain enable)

In bitmap, where I for GPI only, O for GPO only, and I/O for GPIO.

Offset	Reg Abbr.	Attr	bitmap			Default	Bank
50	GPIOOD00 (GPIO00~GPIO07)	R/W	7	GPIO07	I	00h	FC
			6	GPIO06	O		
			5	GPIO05	I		
			4	GPIO04	I		
			3	GPIO03	O		
			2	GPIO02	O		
			1	GPIO01	O		
			0	GPIO00	O		
51	GPIOOD08 (GPIO08~GPIO0F)	R/W	7	GPIO0F	O	00h	FC
			6	GPIO0E	O		
			5	GPIO0D	I		
			4	GPIO0C	O		
			3	GPIO0B	O		
			2	GPIO0A	I		
			1	GPIO09	I/O		
			0	GPIO08	O		
52	GPIOOD10 (GPIO10~GPIO17)	R/W	7	GPIO17	I/O	00h	FC
			6	GPIO16	I/O		
			5	GPIO15	I/O		
			4	GPIO14	I/O		
			3	GPIO13	O		

			2	GPIO12	I/O		
			1	GPIO11	I/O		
			0	GPIO10	I/O		
53	GPIOOD18 (GPIO18~GPIO1F)	R/W	7	GPIO1F	I/O	00h	FC
			6	GPIO1E	I/O		
			5	GPIO1D	I/O		
			4	GPIO1C	I		
			3	GPIO1B	I/O		
			2	GPIO1A	O		
			1	GPIO19	O		
			0	GPIO18	I/O		
54	GPIOOD20 (GPIO20~GPIO27)	R/W	7	GPIO27	O	00h	FC
			6	GPIO26	O		
			5	GPIO25	O		
			4	GPIO24	O		
			3	GPIO23	I/O		
			2	GPIO22	I/O		
			1	GPIO21	I/O		
			0	GPIO20	I/O		
55	GPIOOD28 (GPIO28~GPIO2F)	R/W	7	GPIO2F	O	00h	FC
			6	GPIO2E	O		
			5	GPIO2D	O		
			4	GPIO2C	O		
			3	GPIO2B	O		
			2	GPIO2A	O		
			1	GPIO29	O		
			0	GPIO28	O		
56	GPIOOD30 (GPIO30~GPIO37)	R/W	7	GPIO37	I	00h	FC
			6	GPIO36	I		
			5	GPIO35	I		
			4	GPIO34	I		
			3	GPIO33	I		
			2	GPIO32	I		
			1	GPIO31	I		
			0	GPIO30	I		
57	GPIOOD38 (GPIO38~GPIO3F)	R/W	7	GPIO3F	O	00h	FC
			6	GPIO3E	O		
			5	GPIO3D	O		
			4	GPIO3C	O		
			3	GPIO3B	I		
			2	GPIO3A	I		
			1	GPIO39	I		
			0	GPIO38	I		
58	GPIOOD40 (GPIO40~GPIO47)	R/W	7	GPIO47	I/O	00h	FC
			6	GPIO46	I/O		
			5	GPIO45	I/O		
			4	GPIO44	I/O		
			3	GPIO43	O		
			2	GPIO42	O		
			1	GPIO41	I/O		
			0	GPIO40	I		
59	GPIOOD48 (GPIO48~GPIO4F)	R/W	7	GPIO4F	I/O	00h	FC
			6	GPIO4E	I/O		
			5	GPIO4D	I/O		
			4	GPIO4C	I/O		
			3	GPIO4B	I/O		
			2	GPIO4A	I/O		
			1	GPIO49	I/O		
			0	GPIO48	I/O		
5A	GPIOOD50 (GPIO50~GPIO57)	R/W	7	GPIO57	I/O	00h	FC
			6	GPIO56	O		
			5	GPIO55	I/O		
			4	GPIO54	I/O		
			3	GPIO53	O		
			2	GPIO52	I/O		
			1	GPIO51	I/O		
			0	GPIO50	I/O		
5B	GPIOOD58 (GPIO58~GPIO5F)	R/W	7	GPIO5F	N/A	00h	FC
			6	GPIO5E	N/A		
			5	GPIO5D	N/A		
			4	GPIO5C	N/A		

			3	GPIO5B	N/A		
			2	GPIO5A	N/A		
			1	GPIO59	I/O		
			0	GPIO58	I/O		

GPIO Input Enable Register

While input function chosen, each pin can be programmed to be input port. These registers are no meaning for pins of GPO only.

(0:input mode disable 1:input mode enable)

In bitmap, where I for GPI only, O for GPO only, and I/O for GPIO.

Offset	Reg Abbr.	Attr	bitmap			Default	Bank
60	GPIOIE00 (GPIO00~GPIO07)	R/W	7	GPIO07	I	20h	FC
			6	GPIO06	O		
			5	GPIO05	I		
			4	GPIO04	I		
			3	GPIO03	O		
			2	GPIO02	O		
			1	GPIO01	O		
			0	GPIO00	O		
61	GPIOIE08 (GPIO08~GPIO0F)	R/W	7	GPIO0F	O	00h	FC
			6	GPIO0E	O		
			5	GPIO0D	I		
			4	GPIO0C	O		
			3	GPIO0B	O		
			2	GPIO0A	I		
			1	GPIO09	I/O		
			0	GPIO08	O		
62	GPIOIE10 (GPIO10~GPIO17)	R/W	7	GPIO17	I/O	00h	FC
			6	GPIO16	I/O		
			5	GPIO15	I/O		
			4	GPIO14	I/O		
			3	GPIO13	O		
			2	GPIO12	I/O		
			1	GPIO11	I/O		
			0	GPIO10	I/O		
63	GPIOIE18 (GPIO18~GPIO1F)	R/W	7	GPIO1F	I/O	00h	FC
			6	GPIO1E	I/O		
			5	GPIO1D	I/O		
			4	GPIO1C	I		
			3	GPIO1B	I/O		
			2	GPIO1A	O		
			1	GPIO19	O		
			0	GPIO18	I/O		
64	GPIOIE20 (GPIO20~GPIO27)	R/W	7	GPIO27	O	00h	FC
			6	GPIO26	O		
			5	GPIO25	O		
			4	GPIO24	O		
			3	GPIO23	I/O		
			2	GPIO22	I/O		
			1	GPIO21	I/O		
			0	GPIO20	I/O		
65	GPIOIE28 (GPIO28~GPIO2F)	R/W	7	GPIO2F	O	00h	FC
			6	GPIO2E	O		
			5	GPIO2D	O		
			4	GPIO2C	O		
			3	GPIO2B	O		
			2	GPIO2A	O		
			1	GPIO29	O		
			0	GPIO28	O		
66	GPIOIE30 (GPIO30~GPIO37)	R/W	7	GPIO37	I	FFh	FC
			6	GPIO36	I		
			5	GPIO35	I		
			4	GPIO34	I		
			3	GPIO33	I		
			2	GPIO32	I		
			1	GPIO31	I		
			0	GPIO30	I		

67	GPIOIE38 (GPIO38~GPIO3F)	R/W	7	GPIO3F	O	00h	FC
			6	GPIO3E	O		
			5	GPIO3D	O		
			4	GPIO3C	O		
			3	GPIO3B	I		
			2	GPIO3A	I		
			1	GPIO39	I		
			0	GPIO38	I		
68	GPIOIE40 (GPIO40~GPIO47)	R/W	7	GPIO47	I/O	00h	FC
			6	GPIO46	I/O		
			5	GPIO45	I/O		
			4	GPIO44	I/O		
			3	GPIO43	O		
			2	GPIO42	O		
			1	GPIO41	I/O		
			0	GPIO40	I		
69	GPIOIE48 (GPIO48~GPIO4F)	R/W	7	GPIO4F	I/O	00h	FC
			6	GPIO4E	I/O		
			5	GPIO4D	I/O		
			4	GPIO4C	I/O		
			3	GPIO4B	I/O		
			2	GPIO4A	I/O		
			1	GPIO49	I/O		
			0	GPIO48	I/O		
6A	GPIOIE50 (GPIO50~GPIO57)	R/W	7	GPIO57	I/O	00h	FC
			6	GPIO56	O		
			5	GPIO55	I/O		
			4	GPIO54	I/O		
			3	GPIO53	O		
			2	GPIO52	I/O		
			1	GPIO51	I/O		
			0	GPIO50	I/O		
6B	GPIOIE58 (GPIO58~GPIO5F)	R/W	7	GPIO5F	N/A	03h	FC
			6	GPIO5E	N/A		
			5	GPIO5D	N/A		
			4	GPIO5C	N/A		
			3	GPIO5B	N/A		
			2	GPIO5A	N/A		
			1	GPIO59	I/O		
			0	GPIO58	I/O		
6C~6E	Reserved						
6F	GPXDIE00 (GPXD00~GPXD07)	R/W	7	GPXD07	GPXD / I		FC
			6	GPXD05	GPXD / I		
			5	GPXD05	GPXD / I		
			4	GPXD04	GPXD / I		
			3	GPXD03	GPXD / I		
			2	GPXD02	GPXD / I		
			1	GPXD01	GPXD / I		
			0	GPXD00	GPXD / I		

GPIO Misc Control Register

While input function chosen, each pin can be programmed to issue event by interrupt. These registers are no meaning for pins of GPO only. While interrupt enabled, each event can be programmed to be level or edged trigger. Level or Edged trigger can be set in GPWU register module. For more detail please refer to section 4.14.

(0:interrupt disable 1:interrupt enable)

Offset	Reg Abbr.	Attr	bitmap		Default	Bank
70	GPIO_MISC	R/W	7	Reserved	00h	FC
			6	Reserved		
			5	Reserved		
			4	Reserved		
			3	Reserved		
			2	Select SDI Interface output in GPXA2~GPXA0		
			1	Enable system beep glue logic		
			0	Enable port 80 output to PSCLK1/PSDAT1		

4.3 KBC

4.3.1 KBC Functional Description

a. 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.

b. I/O 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. KB3920 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).

c. I/O 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 generated from PS/2 auxiliary device.
4	**Uninhibited**	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.

d. Hardware Processed Command

Value	Command	Description	
20h	**Read Command Byte**	Read the command byte of KBC	
		Response	Command byte
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
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#***.	

4.3.2 KBC Registers Descriptions

(Base Address = FC80h, 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
80h	**KBCCB**	**KBC Command Byte (KBC command 20h/60h)**			40h	FCh
		7	RSV			
		6	R/W	**Scan Code Conversion**		
		5	R/W	**Auxiliary Device Disable**		
		4	R/W	**Keyboard Device Disable**		
		3	R/W	**Inhibit Override**		
		2	R/W	**System Flag**		
		1	R/W	**IRQ12 Enable**		
		0	R/W	**IRQ1 Enable**		
81h	**KBCCFG**	**KBC Configuration**			00h	FCh
		7	R/W	**Keyboard Lock Enable**		
		6	R/W	**Fast Gate A20 Control**		
		5~4	RSV			
		3	R/W	**Keyboard Lock Flag**		
		2	RSV			
		1	R/W	**IBF Interrupt Enable** This bit enables KBC to generate interrupt to the 8051 at the rising edge of IBF,		
		0	R/W	**OBF Interrupt Enable** This bit enables KBC to generate interrupt to the 8051 at the falling edge of **OBF**.		
82h	**KBCIF**	**KBC Interrupt Pending Flag**			00h	FCh
		7~3	RSV			
		2	R/WC1	**KBC firmware mode in processing flag** Exit KBC firmware mode and re-enable the hardware mode by writing 1		
		1	R/WC1	**IBF interrupt pending flag**		
		0	R/WC1	**OBF interrupt pending flag**		
83h	**KBCHWEN**	**KBC Hardware Command Enable**			00h	FCh
		7	R/W	Feh: KB Reset command processed by hardware		
		6	R/W	E0h: read test input command processed by hardware		
		5	R/W	D3h: write AUX output buffer		
		4	R/W	D2h: write KB output buffer		
		3	R/W	D1h: write P2 command processed by hardware		
		2	R/W	D0h: read P2 command processed by hardware		
		1	R/W	C0h: read P0 command processed by hardware		
		0	R/W	20h: read command byte processed by hardware		
84h	**KBCCMD**	**KBC Command Buffer**			00h	FCh
		7~0	RO	The data written to I/O port 64h will be stored in this register.		
85h	**KBCDAT**	**KBC Data Input / Output Buffer**			00h	FCh
		7~0	R/W	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.		
86	**KBCSTS**	**KBC Host Status**			00h	FCh
		7	R/W	**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	R/W	**Time-Out.** 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	R/W	Auxiliary Data Flag		
		4	RO	Uninhibited		
		3	RO	Address (A2)		
		2	RO	System Flag		
		1	R/WC1	**IBF**, write **IBF** = 1 to clear **IBF**		
		0	R/WC1	**OBF**, write **KBCDAT** will set **OBF** to 1. Write **OBF** = 1 to clear **OBF**		

4.4 IKB

4.4.1 IKB Functional Description

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.

Command	Description	
Edh	**Set LED.** Modify the status of LED by the following argument byte. The specification of the argument byte is: Normal sequence: ED FA WW FA (WW is ED setting to IKBLED bit 2~0)	
	Bit 7 – 3	All must be zero
	Bit 2	Caps Lock LED status
	Bit 1	Num Lock LED status
	Bit 0	Scroll Lock LED status
Eeh	**Echo.** Send Eeh back to the host after receiving this command. Normal sequence: Eeh Eeh .	
F0h	**Access Scan Code Set.** The host uses the first argument to specify the read/write operation. If the first argument equals 00h, it represents a read operation and KB3920 returns two bytes, Fah and 41h. If the first argument is not equal to 00h, it represents a write operation and KB3920 ignores the argument because KB3920 only supports Set 2 scan code. Normal Sequence: F0h Fah 00h Fah 02h, or F0h Fah 0Wh Fah (W is setting new scan code set).	
F2h	**Get Device ID.** Normal Sequence: F2h Fah Abh 41h .	
F3h	**Set Typematic Rate.** Normal sequence, F3h Fah xxh Fah (xx is the typematic Rate to IKBTYPEC register)	
F4h	**Enable.** Begin scanning the key matrix and sending the scan code to the host. Note that KB3920 is in disable mode after hardware reset. The system BIOS should configure all options of KB3920 and enable it at last. Normal sequence : F4h Fah .	
F5h	**Disable.** Stop scanning the key matrix and flush the scan code buffers. Normal sequence: F5h Fah.	
F6h	**Set Default.** Restore the default setting of typematic rate and LED status. Normal sequence: F6h Fah.	
Feh	**Resend.** Re-transmit the last byte. Normal sequence: Feh XXh (XX is the last byte that KB3920 IKB sent to PS2).	
FFh	**Reset.** Generate an internal soft-reset signal to reset the PS/2 interface, clear all internal flags of scan controller, and flush the scan code buffers. Normal sequence: FFh Fah Aah.	

4.4.2 IKB Registers Descriptions

(Base address = FCA0h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
A0h	**IKBCFG**	**IKB Configuration**			00h	FCh
		7	R/W	IKB Scan Controller test mode enable		
		6	R/W	IKB PS/2 Interface waiting time. (0: 8 us, 1: 64 us)		
		5	R/W	**IKB automatic Make and Break key enable.** **IKBMK, IKBBK** will output the latest make and break key of the scanning. The bit 0 and 1 in IKBIE and IKBPF will be changed.		
		4	WO	Force Scan Controller to scan key matrix. Write '1' to start scanning		
		3	R/W	IKB 1st repeat key delay typematic repeat time be general timer enable		
		2	R/W	IKB scan repeat timer enable, write '1' will enable and reset timer. If enable (set 1), each 30ms Scan Controller will repeat to scan after a make key is issued, until the write **IKBPF.7** (Force Scan Controller to enter idle state) '1' after a scan finished interrupt without any make key interrupt.		
		1	R/W	IKB PS/2 KB command hardware enable		
		0	R/W	IKB Scan Controller enable		
A1h	**IKBLED**	**IKB LED Control**			00h	FCh
		7~5	RSV			
		4	R/W	LED output polarity: 0: output as register setting, 1: output level is inversed.		
		3	RSV			
		2	R/W	CapLock, if **IKBCFG.1 =1**, this bit will be Read Only and only IKB PS2 hardware command can override it.		
		1	R/W	NumLock, if **IKBCFG.1 =1**, this bit will be Read Only and only IKB PS2 hardware command can override it.		
		0	R/W	ScrLock, if **IKBCFG.1 =1**, this bit will be Read Only and only IKB PS2 hardware command can override it.		
A2h	**IKBTYPEC**	**IKB Typematic Control, if IKBCFG.3 is set(1), the bit 6~0 is general timer base on 4ms.**			00h	FCh
		7	RSV	Reserved.		
		6~5	R/W	1st repeat key delay time, 00: 250ms, 01: 500ms, 10: 750ms, 11: 1 second		
		4~0	R/W	Typematic repeat time. 1Fh: 2 cps, 1Bh: 3 cps, 18h: 4 cps, 17h: 5 cps, 15h: 6 cps, 13h: 8 cps, 10h: 10 cps, 0Dh: 12 cps, 0Bh: 15cps, 08h: 16cps, 05h: 20cps, 00h: 30 cps.		
A3h	**IKBIE**	**IKB Interrupt Enable**			00h	FCh
		7~6	RSV			
		5	R/W	This interrupt will issue while the following commands receiving. (IKB PS2 KB Reset / Disable command / none standard PS/2 command.)		
		4	R/W	IKB PS2 RX finished interrupt enable		

		3	R/W	IKB PS2 TX finished interrupt enable		
		2	R/W	IKB Typematic repeat timeout interrupt enable. And timer will be reset and start to count by writing '1' to this enable bit. The repeat timer will count from "delay" time first then the "repeat" time in the **IKBTYPEC** register setting. This bit will be cleared by Disable and Reset command.		
		1	R/W	**IKB automatic make and break key enable** is not set(=0): IKB Key scan finished interrupt enable **IKB automatic Make and Break key enable** is set(=1): IKB Break Key interrupt enable.		
		0	R/W	IKB Key make interrupt enable.		
A4h	**IKBPF**	**IKB Interrupt Pending Flag**			00h	FCh
		7	WO	Force Scan Controller to enter idle state. After a scan finished interrupt without any make key, this bit should be written '1' to force Scan Controller to enter idle state and wait for a new key to be pressed. This action will prevent from 30ms timer in Scan Controller to scan key matrix again.		
		6	R/WC1	***Scan Address is valid pending flag*.(IKBCFG.5 = 0)** For current scanning row, write 1 before read next column at the same row. If this bit is still '1', the **IKBSADR** is valid for appeared scan address. ***Ghost Key Identified Flag* (IKBCFG.5 = 1)** If a ghost key is identified, this flag will be set for firmware to know the last make key and it should not be repeated again. This flag will be cleared by WC1 and when the ghost key is gone. If the ghost key still exists, the flag will be set in next clock.		
		5	R/WC1	IKB PS2 KB Reset or Disable hardware command interrupt pending		
		4	R/WC1	**IKB PS2 RX finished interrupt pending** If IKB PS2 RX interrupt is not enabled, and this pending flag is set, it means a non-hardware is coming from PS/2 host controller.		
		3	R/WC1	IKB PS2 TX finished interrupt pending		
		2	R/WC1	Typematic repeat timeout interrupt pending		
		1	R/WC1	**IKB automatic make and break key enable** is clear(=0): **Key scan finished interrupt pending.** The Scan Controller finished a matrix scanning. **IKB automatic make and break key enable** is set(=1): The flag means a valid and the latest break key is in **IKBBK.** If the bit is still 1 after cleared, the additional make key is in **IKBBK** reg.		

		0	R/WC1	**IKB automatic make and break key enable** is clear(=0): **Ikey make interrupt pending.** If the **Scan Address is valid** bit is not set, the write '1' will also cause Scan Controller to scan next row address with make key.		
				IKB automatic make and break key enable is set(=1): The flag means a valid and the latest make key is in **IKBMK.** If the bit is still 1 after being cleared, the additional make key is in **IKBMK** reg.		
A5h	**IKBTXDAT**	**IKB PS2 TX Data Byte**			00h	FCh
		7 ~ 0	R/W	IKB sends data to PS2 controller data port. The data written to this register will be transferred to the PS/2 controller. After the end of data transmission, the PS2 TX finished interrupt can be asserted to firmware.		
A6h	**IKBRXDTA**	**IKB PS2 RX Data Byte**			00h	FCh
		7 ~ 0	RO	The received data byte from PS2 controller. This byte is valid if PS2 RX finished interrupt pending flag is set.		
A7h		**Reserved**				
A8h	**IKBKSI**	**IKB Scan Inputs**			00h	FCh
		7~0	RO	Scan Input buffer		
A9h	**IKBSADR**	**IKB Scan Address**			00h	FCh
		7 ~ 0	RO	Scan Address of current scanned pressed key.		
Aah	**IKBSDB**	**Scan De-bounce Control**			F7h	FCh
		7 ~ 4	R/W	Scan output release wait time (8 us unit)		
		3 ~ 0	R/W	Scan output driving low time (8 us unit)		
Abh	**IKBMK**	**IKB Make Key**			00h	FCh
		7 ~ 0	RO	The latest make key of the latest scanning by Scan Controller if **IKB automatic make and break key enable** is set in **IKBCFG.5**.		
Ach	**IKBBK**	**IKB Break Key**			00h	FCh
		7 ~ 0	RO	The latest break key of the latest scanning by Scan Controller if **IKB automatic make and break key enable** is set in **IKBCFG.5**.		
AD-Afh		**Reserved**			00h	FCh

4.5 PWM

4.5.1 PWM Functional Description

There are two PWM channels with 8-bit resolution.

There are 2 FANPWN, FANPWM1 and FANPWM2, with 12-bit resolution.

The PWM Cycle Length defines the PWM cycle time in setting clock source. The PWM High Period Length defines the PWM pulse high period length, should be less than Cycle Length.

Here is the formula of PWM duty cycle.

*Duty Cycle = (PWM High Period Length+1)/(PWM Cycle Period Length+1) *100%*

Please note the following case:

Condition	PWM Output
H > C	Always 1 (High)
H and C=0x00	Always 1 (High)
H=0x00, C=0xFF	A short pluse
H=0xFF, C=0x00	Always 1 (High)
1. Where **H** means High Period Length (PWMHIGH) ; **C** means Cycle Period Length (PWMCYCL) Please refer to the following PWM register description. 2. To force PWM output Low, please force this pin to be GPIO mode and output low.	

4.5.2 PWM Registers Descriptions

(Base address_FE00h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
00h	**PWMCFG**	**PWM Configuration**			00h	Feh
		7~6	R/W	PWM1 clock source selection 0: 1us 1: 64us 2: 256us 3: 4ms		
		5	RSV			
		4	R/W	PWM1 Enable		
		3~2	R/W	PWM0 clock source selection 0: 1us 1: 64us 2: 256us 3: 4ms		
		1	RSV			
		0	R/W	PWM0 Enable		
01h	**PWMHIGH0**	**PWM0 High Period Length**			00h	Feh
		7~0	R/W	The High Period Length of PWM should be small than Cycle Length.		
02h	**PWMCYCL0**	**PWM0 Cycle Length**			00h	Feh
		7~0	R/W	The Cycle Length of a PWM cycle, includes high and low Length.		
03h	**PWMHIGH1**	**PWM1 High Period Length**			00h	Feh
		7~0	R/W	The High Period Length of PWM should be small than Cycle Length.		
04h	**PWMCYCL1**	**PWM1 Cycle Length**			00h	Feh
		7~0	R/W	The Cycle Length of a PWM cycle, includes high and low Length.		

4.6 Fan

4.6.1 FAN Functional Description

There are three identical FAN controllers in KB3920. 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).

4.6.2 Functions in a FAN controller:

- FAN tachometer monitor
- FAN PWM control

4.6.3 FAN Tachometer Monitor

FAN speed is calculated by a 12-bit counter with **30us** resolution by counting the duty cycle length of ***FANFB1/2*** 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***. The corresponding RPM vs. counter value is listed as follows.

RPM	Round / 1 min	Round / sec	u sec / Round	counter value for 1PPR FAN
6000	6000	100(**6000** / 60 sec)	10000	333 (10000/30)
5000	5000	83.33(**5000** / 60)	12000	400 (12000/30)
4000	4000	66.667 (**4000** / 60)	15000	500 (5000/30)
3000	3000	50 (**4000** / 60)	20000	666 (20000/30)
2000	2000	33.333 (**2000** / 60)	30000	1000 (30000/30)
1000	1000	16.667 (**1000** / 60)	60000	2000 (60000/30)
500	500	8.3 (**500** / 60)	120000	4000 (120000/30)

RPM (round / min) = 60,000,000 / (FANMON*30)

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 when monitoring update interrupt and timeout interrupt. The update flag is in **FANSTS** bit 0.

4.6.4 FAN PWM Control

The Fan PWM is a 12-bit counter clocked by system clock with a resulting 4.096~0.256ms 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**, it indicates 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.

4.6.5 FAN Registers Descriptions

(Base Address = FE20h, 32 bytes)

(FAN1 = FE200h~FE2Fh / FAN2 = FE300h~FE3Fh)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
20h 30h	**FANCFG1** **FANCFG2**	**Fan Controller Configuration**			00h	Feh
		7	R/W	=0, the monitor base clock will be system clock. =1, the monitor base clock will be 30us clock (normal FAN setting).		
		6	R/W	Enable FAN feedback input detect edge changing. FANMON value will become a low and high duration counter.		
		5	R/W	**Select FANPWM** 0= 12-bit **FANPWM** 1= **FANPWM** frequency be 25KHz.The **FANPWM** will not be 25KHz after **CLKCFG**.0,1 (FF0Dh bit 0, 1) are set, but be 12.5KHz (2Mhz clock) and 6.125KHz (1MHz clock) The **FANPWM** frequency will become the following clock count by different clock setting: (The resolution will be lower if low clock is selected). 16Mhz : 28Fh 8Mhz : 147h 4Mhz : A0h		
		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	**FANSTS1** **FANSTS2**	**Fan Controller Status**			00h	Feh
		7~2	R/W	Reserved.		
		1	R/WC1	Flag bit for Fan speed monitor timeout error event. This bit can be cleared by writing a '1'.		
		0	R/WC1	Flag bit for Fan speed monitor update event. This bit can be cleared by writing a '1'.		
23h 22h 33h 32h	**FANMONL1** **FANMONH1** **FANMONL2** **FANMONH2**	**Fan Speed Monitor Counter Value**			FFh 0Fh FFh 0Fh	Feh
		7-0	R/W	The latest fan speed monitor counter value. This value is the pulse width in monitoring clock count (30.. **FANMONH[7:4] is not cared.**		
25h 24h 35h 34h	**FANSETL1** **FANSETH1** **FANSETL2** **FANSETH2**	**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). **FANSETH[7:4] is not cared.**		
27h 26h 37h 36h	**FANPWML1** **FANPWMH1** **FANPWML2** **FANPWMH2**	**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 programmed in the sequence of high-byte (FANPWMH) to low-byte (FANPWML).		
29h 28h 39h 38h	**FANCPWML1** **FANCPWMH1** **FANCPWML2** **FANCPWMH2**	**Fan Current PWM High Pulse Width Bits [11:0]**			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.		

FANMON and FANSET are now 12 bits.

4.7 GPT

4.7.1 GPT Functional Description

There are 4 **G**eneral **P**urpose **T**imers in KB3920. 2 GPTs are 16-bit, and the other 2 are 8-bit. All these timers are based on 30.516 us (32.768Khz) clock, and are independent on clock setting in EC register.

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

4.7.2 GPT Register Descriptions

(Base address = FE50h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
50h	**GPTCFG**	**GPT Configuration**			00h	Feh
		7~5	RSV			
		4	R/W	GPT test mode, the GPT base clock will be system clock		
		3	R/W	Enable GPT3 counting and GPT3 interrupt		
		2	R/W	Enable GPT2 counting and GPT2 interrupt		
		1	R/W	Enable GPT1 counting and GPT1 interrupt		
		0	R/W	Enable GPT0 counting and GPT0 interrupt		
51h	**GPTPF**	**GPT Pending Flag**			00h	Feh
		7	WO	GPT3 write 1 to restart		
		6	WO	GPT2 write 1 to restart		
		5	WO	GPT1 write 1 to restart		
		4	WO	GPT0 write 1 to restart		
		3	R/WC1	GPT3 Interrupt Pending Flag		
		2	R/WC1	GPT2 Interrupt Pending Flag		
		1	R/WC1	GPT1 Interrupt Pending Flag		
		0	R/WC1	GPT0 Interrupt Pending Flag		
53h	**GPT0**	**GPT0 Count Value**			00h	Feh
		7~0	R/W	After GPT0 reach this value and interrupt will occur and GPT0 reset and counting from zero again.		
55h	**GPT1**	**GPT1 Count Value**			00h	Feh
		7~0	R/W	After GPT1 reach this value and interrupt will occur and GPT1 reset and counting from zero again.		
56h 57h	**GPT2H** **GPT2L**	**GPT2 Count Value**			00h 00h	Feh
		7~0	R/W	After GPT2 reach this value and interrupt will occur and GPT2 reset and counting from zero again.		
58h 59h	**GPT3H** **GPT3L**	**GPT3 Count Value**			00h 00h	Feh
		7~0	R/W	After GPT3 reach this value and interrupt will occur and GPT3 reset and counting from zero again.		

4.8 SPI Device Interface

4.8.1 SDI Functional Description

The KBC / EC is a SPI Master and use **SDI Device Interface (SDI)** to interface the SPI devices.

The initial state of ***SDICS#*** and ***SDICLK*** is high.

4.8.2 SDI Registers Descriptions

(Base address = FE70h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
70h	**SDICSR**	**SDI Control Status Register**			00h	Feh
		7	RO	**SDI IDLE Flag**. As this flag is 1, the **SDIBO** and **SDIBI** are permitted to be read/write. Otherwise, the result will unpredictable.		
		6~4	RSV	Reserved.		
		3~2	R/W	***SDICLK* Divider.** The ***SDICLK*** is **Peripheral Clock / ((*SDICLK* Divider + 1) x 2)**.		
		1	R/W	***SDIDO* / *SDIDI* Timing.** =0, ***SDIDO*** change data bit at ***SDICLK*** rising (device latch at ***SDICLK*** falling). =0, ***SDIDI*** latch data bit at ***SDICLK*** rising (device change at ***SDICLK*** falling). =1, ***SDIDO*** change data bit at ***SDICLK*** falling (device latch at ***SDICLK*** rising). =1, ***SDIDI*** latch data bit at ***SDICLK*** falling (device change at ***SDICLK*** rising).		
		0	R/W	**Enable *SDICS#* and SDI Data Port.** The ***SDICS#*** will output low state as set this bit to 1 and the **SDIDAT** is able to be accessed according the **SDI Busy Flag**.		
71h	**SDIBO**	**SDI Data Byte Output Port**			00h	Feh
		0	R/W	Write this port will output 8 bits ***SDICLK*** with input/output ***SDIDI*/*SDIDO***. The **SDI Busy Flag** will be set immediately, and will be reset again after 8 ***SDICLK*** completed. A write **SDIBO** should be done before reading **SDIBI** with correct data byte.		
72h	**SDIBI**	**SDI Data Byte Input Port**				
				Read this port will get the byte data from with input/output ***SDIDI* signal**.		

4.9 WDT

4.9.1 WDT Functional Description

WDT timer clock uses 32.768 Khz oscillator clock and base unit is 64ms.

WDT register set will be reset only by power on reset and ECRST#.

WDT range is between 64 ms to 16 seconds.

WDT reset time is between 128ms to 32 seconds.

WDT reset can reset all KB3920 logic, except GPIO registers. Thus, the GPIO setting can be preserved after WDT reset occurred. The WDT reset can optionally be set only to reset 8051 logic in EC register.

4.9.2 WDT Registers Descriptions (Base address = FE80h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
80h	**WDTCFG**	**WDT Configuration**			00h	Feh
		7	R/W	Enable WDT clock source from 32.768Khz OSC This should be set for normal setting after 32Khz OSC being stable.		
		6~3	R/W	Force to disable WDT by writing 1001b to this field		
		2	R/W	Enable WDT test mode 1: WDT clock is peripheral clock divided from 32Mhz 0: WDT clock is 32Khz (normal setting)		
		1	R/W	Enable WDT interrupt (WDT reset warning)		
		0	R/W	Enable WDT reset, and reset WDT timer, the WDT timer and 2 pending flags will be reset and count from zero again. If the WDT and reset after a interrupt occurred, the next interrupt will occurred after 16 seconds.		
81h	**WDTPF**	**WDT Pending Flag**			00h	Feh
		7~5	RSV			
		1	R/WC1	WDT interrupt pending, WDT half timeout flag. If this bit is set, the following WDT timeout event will cause a WDT reset signal to system.		
		0	R/WC1	WDT reset event pending flag (the last WDT reset was ever happened), WDT reset will assert if WDT count to WDT and WDT interrupt is pending.		
82h	**WDT**	**WDT 8-bit Count Value (for Watch Dog Timer reset system)**			00h	Feh
		7~0	R/W	After WDT counts to this value the half of WDT/2, the interrupt will occur. The WDT timer unit is 64ms.		

4.10 LPC

4.10.1 LPC / FWH Functional Description

There are 5 address ranges on LPC/FWH interface will be responded by KB3920 EC.

1. Keyboard controller I/O ports: 60h, 64h
2. Embedded controller I/O ports: 2 programmable I/O ports (default 62h, 66h)
3. EC I/O Index and Data Ports: Through which the system host can access KB3920 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.
4. LPC/FWH memory access
5. Extended LPC write byte: can be programmed to port 80 and generate interrupt to 8051.

4.10.1.1 LPC Decoding IO Ports

The keyboard I/O ports are 60h/64h, while the EC I/O ports are programmable in **LPCEBA** (FE98h, FE99h). The enable/disable of I/O ports decoding on LPC bus can be configured individually via register **LPCCFG** (FE95h).

4.10.1.2 LPC Decoding Memory Space

Memory Setting (LPCFWH bit 7,6)	Memory Size	Decoded BIOS Address
00	256k (default)	000C_0000h – 000F_FFFFh FFFC_0000h – FFFF_FFFFh
01	512k	000C_0000h – 000F_FFFFh FFF8_0000h – FFFF_FFFFh
10	1M	000C_0000h – 000F_FFFFh FFF0_0000h – FFFF_FFFFh
11	2M	000C_0000h – 000F_FFFFh FFE0_0000h – FFFF_FFFFh

4.10.2 LPC Registers Descriptions

(Base address = FE90h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
90h		Reserved				
91h	**LPCSIRQ**	**LPC SIRQ Configuration**			00h	Feh
		7	R/W	Enable don't care A22 of FWH memory cycle		
		6	R/W	Enable SCI SIRQ		
		5	R/W	Enable IRQ12 SIRQ		
		4	R/W	Enable IRQ1 SIRQ		
		3 ~ 0	R/W	SCI Serial IRQ channel 0: no 1:IRQ1 2:SMI# 3:IRQ3 … 15:IRQ15		
92h 93h	**LPCIBAH** **LPCIBAL**	**LPC Index IO Base Address**			FFh 2Ch	Feh
		7 ~ 0	R/W	EC index mode IO port base address. The address should be 4 bytes align.		
94h	**LPCFWH**	**LPC FWH Configuration**			00h	Feh
		7 ~ 6	R/W	Memory Size (both for LPC Memory and FWH) 00: 256KB 01: 512KB 10: 1MB 11: 2MB		
		5	R/W	Enable FWH memory cycle		
		4	R/W	Enable FWH IDSEL check		
		3 ~ 0	R/W	FWH ID		
95h	**LPCCFG**	**LPC Configuration**			80h	Feh
		7	R/W	Enable LPC memory write protection (including FWH)		
		6	R/W	Enable index IO port		
		5	R/W	Enable KBC IO port: 60h, 64h		
		4	R/W	Enable Extended IO port (IO write only)		
		3	R/W	Enable EC IO port		
		2	R/W	Enable LPC memory cycle (not including FWH)		
		1	R/W	Enable SIRQ fixed in continuous mode. This bit should be forced to 1.		
		0	R/W	Enable LPC CLKRUN#		
96h	**LPCXBAH**	**LPC Extended IO Base Address**			00h	Feh
97h	**LPCXBAL**	OnlyLPC byte write is supported			80h	
98h	**LPCEBAH**	**LPC EC IO Base Address**			00h	Feh
99h	**LPCEBAL**	LPCEBAL bit 0 and bit 1 are not ignored for decoding			62h	
9A~9Ch		Reserved				
9Dh	**LPC68CFG**	**LPC 68/6Ch IO Configuration**				Feh
		7	R/W	Enable LPC I/F decode IO 68h, 6Ch		
		6~2	RSV			
		1	R/W	IBF Interrupt Enable		
		0	R/W	OBF Interrupt Enable		
9Eh	**LPC68CSR**	**LPC 68h IO Command Status Register**				Feh
		7	RO	**IO68/6Ch Busy Flag.** This flag reflects the status while the system host is accessing the 68/6Ch IO. If this bit is set, the software cannot access the 68/6Ch port. This flag can be clear by writing 6Ch = FFh.		
		6	RO	A2 (address bit 2) of the last write 68/6C IO.		
		5~4	RSV			
		3	R/WC1	IBF Interrupt Pending Flag		
		2	R/WC1	OBF Interrupt Pending Flag.		
		1	R/WC1	IBF		
		0	R/WC1	OBF		
9Fh	**LPC68DAT**	**LPC 6Ch IO Data Register**			00h	Feh
		7~0	R/W	The data byte of current memory cycle.		

4.11 XBI / XIO

4.11.1 XBI / XIO / ISP Functional Description

XBI includes several functions, as fellows,

- Ø 4 8051 code segments
- Ø external SRAM accessing
- Ø performance improvement: instruction sustain fetch, pre-fetch
- Ø flash write protection
- Ø XIO
- Ø ISP should be enabled by hardware trap pin during hardware reset.

  The ISP packet format: If bit 7th of 1st byte is 1 means write packet, otherwise means read.

  a. ISP write : [1000_XXXX] [WDATA]

  b. ISP read : [0000_XXXX] [RDATA]

4.11.2 XBI Registers Descriptions

(Base address = FEA0h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
A0h	**XBISEG0**	**8051 Address Segment 0 (0000h-3FFFh) Mapping Configuration**			00h	Feh
		7	R/W	Enable 8051 Code Space SEG0 Remapping		
		6~0	R/W	XBI address = **XBISEG0***16k + 8051 address [13:0]		
A1h	**XBISEG1**	**8051 Address Segment 1 (4000h-7FFFh) Mapping Configuration**			00h	Feh
		7	R/W	Enable 8051 Code Space SEG1 Remapping		
		6~0	R/W	XBI address = **XBISEG1***16k + 8051 address [13:0]		
A3~A2h		Reserved				
A4h	**XBIXIOEN**	**XBI XIO Enable**			00h	Feh
		7~0	R/W	**Bit[0:3] correspond to XIO0-XIO3 enable respectively.**		
A5h	**XBICFG**	**XBI Configuration**			07h	Feh
		7	R/W	Enable XBI BUS IO buffer pull up		
		6	R/W	Enable 8051 sustain instruction fetch		
		5		**Reserved**		
		4	RO	Enable WR# to flash		
		3	R/W	Enable extend SELMEM# and SELE51# 1 clock for RD# and WR# setup and hold time.		
		2~0	R/W	RD# and WR# command clock count= [2:0]		
A6h	**XBICS**	**XBI E51CS# Configuration**			00h	Feh
		7	R/W	Enable E51CS# address		
		6~5	RSV	Reserved		
		4	R/W	While this bit is set to be 1 and 8051 reset, register XBISEG0 and XBISEG1 are reset to be default vale. The 8051 can be reset by the following sources: WDT / Wakeup / EC Register.		
		3	RSV			
		2	R/W	After enable this bit, XBI interface will be set to initial state while the chip is in STOP and IDLE mode.		
		1	R/W	**EHB Fast Access** will be enabled while this bit is set to 1**.** (An enhanced option to speed up EHB performance.)		
		0	R/W	Select XIO select to SELIO# (set 1) or SELIO2# (set 0).		
A7h	**XBIWE**	**XBI Write Enable**			00h	Feh
		7~0	WO	Write 00h to reset all rest mode. Write A3h to enable flash write cycles. Write C5h to SRAM test.		

A8h A9h Aah	**SPIA0** **SPIA1** **SPIA2**	**XBI SPI Flash Address**			00h	Feh
		7~0	R/W	SPIA0 = A7~0 SPIA1 = A15~8 SPIA2 = A22~16		
Abh	**SPIDAT**	**XBI SPI Flash Output / Input Data**			00h	Feh
		7~0	R/W	Output(write SPIDAT) / Input(read SPIDAT) data to/from SPI flash interface.		
Ach	**SPICMD**	**XBI SPI Flash Command**			00h	Feh
		7~0	R/W	The issued SPI command to SPI flash chip. The write to this register will start the SPI accessing, so that the SPIA2~0 and SPIDAT should be ready before SPICMD is written. **SPICMD support command :** 01h Write Status Register 02h Byte Program 03h Read 04h Write Disable 05h Read Status Register 06h Write Enable 0Bh High Speed Read 20h Sector Erase (SST) 50h Enable Write Status Register (SST) 52h Block Erase (SST) 60h Chip Erase (SST) C7h Chip Erase (PCM, NexFlash) D7h Sector Erase (PCM) D8h Block Erase (PCM, NexFlash)		
Adh	**SPICFG**	**SPI Flash Configuration / Status**			00h	Feh
		7	R/W	**Enable DPLL for ISP mode**		
		6	R/W	**reserved**		
		5	R/W	**reserved**		
		4	R/W	**SPICS# force output low.** After set this bit, the protocol will control by firmware. The SPICMD will output to SPI BUS each time the write operation to SPICMD. The SPIDAT will store the read operation data from SPI BUS.		
		3	R/W	**SPICMD write enable.** Enable SPICMD write action to start SPI flash protocol accessing.		
		2	R/W	**Enable SPI Flash Dummy Byte for Read Command.** Enable SPI flash read by 8051 instruction by **Fast Mode (High Speed Read) 0Bh command**.		
		1	RO	**SPI flash accessing in progress status.** Use this bit to check if the SPI accessing is finished or not.		
		0	R/W	Enable SPICMD follow with a SPI status check until Busy flag cleared.		
Aeh	**SPIDATR**	**SPI Flash Output Data for Read Compare**			00h	Feh
		7~0	RO	Output data to SPI flash interface.		
Afh	**SPICFG2**	**SPI Flash Configuration 2**			00h	Feh
		7~4	RSV	Reserved.		
		3~0	R/W	SPI Offset / Short Read Command high nibble.		

4.11.3 ISP Space Registers Descriptions (8 bytes)

In ISP mode, the register EC_REG[3:2] (0xFFOD) determines the 8051/peripheral clock. The default value of EC_REG[3:2] is 00b. That is, the 8051_clk will be forced to 8MHz.

The following is the formula for baud rate setting:

(1/baud_rate) = (8051_clk_period) * (SCON2, SCON3)

The default value of (SCON2, SCON3) is 0x89. 8051_clk is 8MHz in ISP mode.

Therefore, the default baud_rate = 1 / ((125ns) * (0x89)) = 1 / (17,125ns) ~= 57600bps

In ISP mode, SCON2 always keep 0x00, if baud rate change needed, just modify the value of SCON3.

Please note that, before accessing ISP register in ISP mode, writing 0x5A (default baud rate 57600) to 8051_RX to enable ISP module.

Offset	Register Abbreviation	Register Full Name			Def
		Bit	Attr	Description	
00h 01h 02h	**ISPIA0** **ISPIA1** **ISPA2**	**ISP SPI Flash Address**			00h
		7~0	R/W	SPIA0 = A7~0 SPIA1 = A15~8 SPIA2 = A22~16	
03h	**ISPDAT**	**ISP SPI Flash Output / Input Data**			00h
		7~0	R/W	Output(write SPIDAT) / Input(read SPIDAT) data to/from SPI flash interface.	
04h	**ISPCMD**	**ISP SPI Flash Command**			00h
		7~0	R/W	The issued SPI command to SPI flash chip. The write to this register will start the SPI accessing, so that the SPIA2~0 and SPIDAT should be ready before SPICMD is written. **SPICMD support command :** 01h Write Status Register 02h Byte Program 03h Read 04h Write Disable 05h Read Status Register 06h Write Enable 0Bh High Speed Read 20h Sector Erase (SST) 50h Enable Write Status Register (SST) 52h Block Erase (SST) 60h Chip Erase (SST) C7h Chip Erase (PCM, NexFlash) D7h Sector Erase (PCM) D8h Block Erase (PCM, NexFlash)	
05h	**ISPCFG**	**ISP SPI Configuration / Status**			00h
		7 ~ 5	R/W		
		4	R/W	SPICS# force output low. After set this bit, the protocol will control by firmware. The SPICMD will output to SPI BUS each time the write operation to SPICMD. The SPIDAT will store the read operation data from SPI BUS.	

		3	R/W	SPICMD write enable. Enable SPICMD write action to start SPI flash protocol accessing.	
		2	R/W	SPI flash read by 8051 instruction by **Fast Mode (High Speed Read) 0Bh** protocol.	
		1	RO	SPI flash accessing in progress status. Use this bit to check if the SPI accessing is finished or not.	
		0	R/W	Enable SPICMD follow with a SPI status check until Busy flag cleared.	
06h	**ISPDATR**	**ISP SPI Flash Output Data for Read Compare**			00h
		7~0	RO	Output data to SPI flash interface.	
07h	**ISPSCON3**	**ISP SPI ISP RS232 Baud Rate Setting**			00h
		7~0	WO	In ISP mode, 8051_SFR(SCON2) always 0 write this reg to program 8051_SFR(SCON3) Default: SCON3 = 0x89, baud-rate = 57600 (while 8051 clk = 8Mhz) set SCON3 = 0x45 for baud-rate =115200 (while 8051 clk = 8Mhz) (1/baud_rate) = (8051_clk_period) * (SCON2, SCON3)	

4.11.4 XIO Registers Descriptions

(Base Address = FED0h, 16 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
D0h ~ D3h	**XIO0~3**	**XIO Read / Write Port**			00h	Feh
		7~0	R/W	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[7:0]** register bits [15:0]. Data transfer is through ***D[7:0]***.		

4.12 PS / 2 Interface

4.12.1 PS/2 Functional Description

The PS2 Controller supports byte-level programming interface to PS2 devices, including IKB module. A PS/2 TX action will be pending if a PS/2 RX is active. After PS/2 RX is completed (received a byte), the TX will start transmitting to the specified port. PS2 Controller will maintain the PS2 channel's integrity in byte level. But the input signal should not be floating not drive low if the PS2 channel is not used (MUST set correct **GPIOFS** and **PS2CFG** Enable PS2 ports).

4.12.2 PS2 Registers Descriptions

(Base Address = FEE0h, 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
E0h	**PS2CFG**	**PS2 Configuration**			0	Feh
		7	R/W	Enable PS/2 port 3(TX/RX), disable will let PSCLK3 in low state	0	
		6	R/W	Enable PS/2 port 2(TX/RX), disable will let PSCLK2 in low state	0	
		5	R/W	Enable PS/2 port 1(TX/RX), disable will let PSCLK1 in low state	0	
		4	R/W	Enable PS/2 port 0 (IKB)(TX/RX), disable will let IKBCLK(internal) in low state	0	
		3	R/W	Enable interrupt of PS/2 parity error	0	
		2	R/W	Enable interrupt of PS/2 TX timeout (clock >180us or request > 100ms)	0	
		1	R/W	Enable interrupt of PS/2 transmitted byte	0	
		0	R/W	Enable interrupt of PS/2 received byte	0	
E1h	**PS2PF**	**PS2 Interrupt Pending Flag**			0	Feh
		7	RO	Received Byte Port is PS/2 port 3	0	
		6	RO	Received Byte Port is PS/2 port 2	0	
		5	RO	Received Byte Port is PS/2 port 1	0	
		4	RO	Received Byte Port is PS/2 port 0 (IKB)	0	
		3	R/WC1	Interrupt Pending Flag of PS/2 parity error	0	
		2	R/WC1	Interrupt Pending Flag of PS/2 TX timeout	0	
		1	R/WC1	Interrupt Pending Flag of PS/2 transmitted byte	0	
		0	R/WC1	Interrupt Pending Flag of PS/2 received byte, Clear this bit will also clear bit 4~7 and RX timeout flag for receiving next byte from a PS/2 device.	0	
E2h	**PS2CTRL**	**PS2 Transmitter / Receiver Control**			0	Feh
		7	R/W	Transmit Byte Port is PS/2 port 3	0	
		6	R/W	Transmit Byte Port is PS/2 port 2	0	
		5	R/W	Transmit Byte Port is PS/2 port 1	0	
		4	R/W	Transmit Byte Port is PS/2 port 0 (IKB)	0	
		3	WO	Write 1 to force reset PS/2 transmitter state, for emergency usage.	0	
		2	WO	Write 1 to force reset PS/2 receiver state, for emergency usage.	0	
		1	RO	Flag of PS/2 RX timeout	0	
		0	R/W	Enable PS/2 transmit data port, This bit should be set for transmitting a byte (write to PS2DATA) to a device.	0	
E3h	**PS2DATA**	**PS/2 DATA**			0	Feh
		7~0	R/W	Write to start a byte transmitting to a PS/2 device, and clear previous state. Read to get the data of received byte for a PS/2 device.	0	
E4h	**PS2CFG2**	**PS/2 Configuration 2**			0	Feh
		7~2	RSV	Reserved.		
		1	R/W	**PS/2 Protocol Waiting Time Enable** This bit include two functions: **1.** After PS/2 bus remains idle (clock-high,data-high) for 16 μs, PS/2 module will issue the commannd to device. **2.** In the PS/2 protocol, while PS/2 module issues command to the device, the clock is low first then the data waits for 16 μs to be low.	0	
		0	R/W	**PS2CLK / PS2DAT input de-bounce enable.** (0: 1 us; 1: 2 us)	0	
E5h	**PS2PINS**	**PS/2 Pin Input Status**				Feh

		7	RO	PS/2 Port 3 Clock		
		6	RO	PS/2 Port 2 Clock		
		5	RO	PS/2 Port 1 Clock		
		4	RO	PS/2 Port 0 Clock (IKB)		
		3	RO	PS/2 Port 3 Data		
		2	RO	PS/2 Port 2 Data		
		1	RO	PS/2 Port 1 Data		
		0	RO	PS/2 Port 0 Data (IKB)		
E6h	**PS2PINO**	**PS/2 Pin Output**				Feh
		7	R/W	PS/2 Port 3 Clock		
		6	R/W	PS/2 Port 2 Clock		
		5	R/W	PS/2 Port 1 Clock		
		4	R/W	PS/2 Port 0 Clock (IKB)		
		3	R/W	PS/2 Port 3 Data		
		2	R/W	PS/2 Port 2 Data		
		1	R/W	PS/2 Port 1 Data		
		0	R/W	PS/2 Port 0 Data (IKB)		

4.13 EC

4.13.1 EC Functional Description

There are 6 parts in EC:

- Hardware EC Commands
- EC Index IO mode
- EC Extended IO Write
- SMBus host controller (described in SMBus sub-chapter)
- SCI Generation
- Miscellaneous functions

4.13.1.1 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.

4.13.1.2 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) 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.

4.13.1.3 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
80h	**EC Read**	Read operation for an internal register in EC Space.
81h	**EC Write**	Write operation for an internal register in EC Space.
82h	**EC Burst Enable**	Enable EC burst operation mode.
83h	**EC Burst Disable**	Disable EC burst operation mode.
84h	**EC Query**	Query the SCI event queue.
Others	**Firmware Command**	No responded from hardware EC. Firmware EC commands.

4.13.1.4 EC Command Program Sequence

Command Byte	Command Name	Programming Sequence
80h	**Read EC**	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 EC**	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).

4.13.1.5 EC Index IO Mode

You may use EC Index IO mode to access the KB3920 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 KB3920 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).

4.13.1.6 SCI Generation

Most interrupts generated from KB3920 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 **SCICFG** (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 KB3920 internal hardware logic, 8051 firmware or system BIOS can also generate a SCI event by writing the desired SCI ID into **SCID** register (0Bh) in EC space. The **SCID** should be first enabled in **ECCFG** bit3. The SCI IDs are defined as follows.

4.13.1.7 SCI ID Table

SCI ID	Name	PxI	Description	Priority
0	**Nothing**	**N.A.**	Indicates a EC Command is received from the Host. Alternatively also means nothing happens	**Highest**
01h		**N.A.**		
02h		**N.A.**		
03h		**N.A.**		
04h		**N.A.**		
05h		**N.A.**		
06h~07		**N.A.**	Not used.	
08h	**WDT**	**P0I.0**	Indicates a Watchdog Timer event.	
09h		**N.A.**		
0Ah	**PS2**	**P0I.2**	Indicates a PS2 event.	
0Bh	**KBC**	**P0I.3**	Indicates a KBC Host Interface event.	
0Ch	**IKB**	**P0I.4**	Indicates a Internal Keyboard event.	
0Dh	**LPC**	**P0I.5**	LPC cycle interrupt	
0Eh	**ECFW**	**P0I.6**	EC firmware mode SCI (IBF/OBF SCI).	
SCID	**SCID**	**P0I.7**	Write **SCI** ID, Query value is SCID.	
10h	**FAN1**	**P1I.0**	Indicates a FAN Controller 1 event.	
11h	**FAN2**	**P1I.1**	Indicates a FAN Controller 2 event.	
12h	**SMBus**	**P1I.2**	Indicates a SMBus Host Controller event.	
13h		**P1I.3**	Not used.	
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	**EXTWIO**	**P3I.0**	Indicates a Write Extended IO interrupt (Port80).	
19h	**GPIO00~0F**	**P3I.1**	Indicates a **GPIO00~0F** event.	
1Ah	**GPIO10~1F**	**P3I.2**	Indicates a **GPIO10~1F** event.	
1Bh	**GPIO20~2F**	**P3I.3**	Indicates a **GPIO20~2F** event.	
1Ch	**GPIO30~3F**	**P3I.4**	Indicates a **GPIO30~3F** event.	
1Dh	**GPIO4F~4F**	**P3I.5**	Indicates a **GPIO40~4F** event.	
1Eh	**GPIO50~59**	**P3I.6**	Indicates a **GPIO50~59** event.	
1Fh	**ADC**	**P3I.7**	Indicates an **ADC** updated event.	

4.13.2 EC Register Descriptions

(Base Address = FF00h, 32 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
00h	**ECHV**	**EC Hardware Revision ID**			B0h	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. ADC test data input when ADC test enable.		
02h	**ECHA**	**EC High Address**			0Fh	FFh
		7~4	RSV			
		3 – 0	R/W	High-byte address of the 64KB EC address space. Used for standard EC commands to access F000~FFFFh internal space. The default setting will let host accessing the FF00~FFFFh (EC, GPWU, SMBus) space.		
03h	**SCICFG**	**SCI Configuration**			90h	FFh
		7	R/W	Enable the generation of SCI by standard EC commands (default enable)		
		6	R/W	Enable SCID port (Firmware generated SCI).		
		5	R/W	EC SCI pulse polarity; =0, low active; (default); =1, high active.		
		4	R/W	Enable EC SCI (set 1) from SCIIFx, default is enabled.		
		3 – 0	R/W	SCI pulse width = SCIPW x 64us, max length = 1 ms, as no width=0. IF width = 0, the pulse width will be a system clock.		
04h	**ECCFG**	**EC Configuration**			00h	FFh
		7	R/W	**Enable EPB Fast Access** An enhanced option to speed up EPB performance during accessing.		
		6	R/W	Test mode. Must be programmed to 0 for normal operation.		
		5	R/W	Enable hardware EC Read/Write command		
		4	R/W	Enable hardware EC Burst Enable/Disable command		
		3	R/W	Enable hardware EC Query command		
		2	R/W	Enable Extended IO port Interrupt to 8051		
		1	R/W	**IBF** Interrupt Enable, also be the Firmware Mode 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.		
		0	R/W	**OBF** Interrupt Enable. This bit enables KBC to generate interrupt to the core processor at the falling edge of OBF.		
05h 06h 07h	**SCIE0** **SCIE1** **SCIE3**	**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	**SCIE0** **SCIE1** **SCIE3**	**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	**SCID**	**EC SCI ID Write Port for 8051 firmware to generate SCI event**			00h	FFh
		7 – 0		8051 firmware can write to this port with SCI_ID value to generate a SCI event. The host can use EC Query command to read this specified value.		

0Ch	**PMUCFG**	**PMU Control / Configuration**			2Fh	FFh
		7		Enter STOP mode by writing this bit = 1, the same as 8051 PCON STOP		
		6		Enter IDLE mode by writing this bit = 1, the same as 8051 PCON IDLE		
		5		Enable auto return to normal clock as async-wakeup event coming in Ultra Low clock state. Async-wakeup including LPC cycles and GPWU.		
		4		MUST be set to '1' to enable wakeup feature.		
		3		Enable SCI to be one of wake up interrupt source 8051 interrupt source will always exit Ultra Low clock to normal clock		
		2		Enable LPC cycle and Watchdog interrupt wake up from STOP mode		
		1		Enable GPWU wake up from STOP mode		
		0		Enable Interrupt wake up from IDLE mode		
0Dh	**CLKCFG**	**Clock Configuration**			00h	FFh
		7	R/W	**Enable Flash (ISA / SPI) Interface Clock From External Pin (GPIO59)** The Flash I/F clock is default from internal PLL (66Mhz). The internal PLL 66Mhz clock duty cycle is not 50-50%, so that the clock high and low limit of SPI flash should be noted. A 66Mhz SPI flash may not be usable in the PLL 64Mhz configuration. But a 100Mhz SPI flash may be usable.		
		6	R/W	**Flash (ISA / SPI) Interface Clock Control** 1: full speed (Internal clock is 66(+-%25) Mhz) 0: half speed (default, ½ of supplied clock) SPI clock is 16Mhz if CLKCFG set to 8/4 Mhz; SPI clock is stopped when 8051 in IDLE if CLKCFG.0 is set.		
		5	R/W	Enable PLL to generate a good 32.768Mhz. (default reset PLL)		
		4	R/W	Enable PLL enter low power state in STOP mode		
		3~2	R/W	**8051 / Peripherals Normal Run Clock Selection.** 10: 22 / 8 Mhz 01: 16 / 8 Mhz (default) 00: 8 / 4 Mhz Clock rate is fixed at 2/1Mhz when 8051 is in IDLE mode if CLKCFG.0 is set. The flash interface (SPI or ISA) is fixed at 32.768 Mhz or higher by **CLKCFG.6** setting.		
		1	R/W	**Enable Peripheral Auto Slow Clock Control to be 1 Mhz.** The Peripheral's clock will be running at 1-Mhz when no host is accessing.		
		0	R/W	**Enable 8051 IDLE Mode Slow Clock Control to be 2 / 1 Mhz.** When 8051 enters IDLE state, the clock of 8051 and peripherals will be changed automatically to 2 / 1 Mhz. And the flash interface clock will be stopped if this bit is set.		
0Eh	**EXTIO**	**EC Extended Write IO data**			00h	FFh
		7~0	R/W	Read this byte to get the host write extended IO data.		
10h	**DAC0**	**DAC0 output value**			00h	FFh
		7 – 0	R/W	Output DAC0 data		
11h	**DAC1**	**DAC1 output value**			00h	FFh
		7 – 0	R/W	Output DAC1 data		
12h	**DAC2**	**DAC2 output value**			00h	FFh
		7 – 0	R/W	Output DAC2 data		
13h	**DAC3**	**DAC3 output value**			00h	FFh

		7 – 0	R/W	Output DAC3 data		
14h	**PXCFG**	**8051 on-chip Control**			00h	FFh
		7~2	R/W	Reserved		
		1	R/W	**Enable WDT timeout only reset 8051** 1: WDT timeout event only resets 8051. 0: The WDT timeout event resets whole chip (not including GPIO module)		
		0	R/W	**Reset 8051 and 8051 internal peripherals** (8051 serial port, timer, interrupt controller) . After reset, the 8051 will restart from reset vector if this bit is reset to '0'. Write '1' to reset 8051. Write '0' to restart 8051.		
15h	**ADDAEN**	**ADC/DAC Enable**			00h	FFh
		7~4	R/W	Enable DAC3~0		
		3~0	R/W	Enable ADC3~0		
16~17h		Reserved				
18h	**ADCTRL**	**ADC Control Register**			00h	FFh
		7~4	RSV			
		3~2	R/W	Select converting ADC channel (AD0~3)		
		1	R/W	ADC test mode		
		0	R/W	Start (write 1 action) ADC converter and Enable ADC converted interrupt		
19h	**ADCDAT**	**ADC Data output port**			00h	FFh
		7 – 0	RO	After ADC converted, the value is hold here.		
1Ah	**ECIF**	**EC Interrupt Pending Flag**			00h	FFh
		7~3	RSV			
		2	R/WC1	**EC firmware mode in processing flag**, Exit EC firmware mode and re-enable hardware mode by writing 1		
		1	R/WC1	**IBF** interrupt pending flag, as **ECSTS IBF** is set by host write		
		0	R/WC1	**OBF** interrupt pending flag, as **ECSTS OBF** is clear by host read **ECDAT**		
1Bh	**ECDAT**	**EC Data port**			00h	FFh
		7 – 0	R/W	The EC Data Port serves as the window between system host and EC. Write **ECDAT** will set **ECSTS** bit 0 (**OBF**) at the same time.		
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	Free r/w bit for host interface		
		6	R/W	Free r/w bit for host interface		
		5	RO	SCI pending flag		
		4	R/W	Burst Enable Status		
		3	RO	**A2** (Command or Data Flag) =0, previous host write is Data =1, previous host write is Command		

		2	RSV			
		1	R/WC1	**IBF**, write **IBF** = 1 to clear **IBF**		
		0	R/WC1	**OBF**, write port **ECDAT** will set **OBF** to 1. Write **OBF** = 1 to clear **OBF**		
others		7~0	RSV	Reserved.	00h	FFh

4.14 GPWU

4.14.1 GPWU Functional Description

Each GPIO with GPI pin can generate events (interrupt or wakeup). The GPI input can be set as **Level** or **Edge** trigger or **Change** trigger. **Polarity** bit setting will affect Level and Edge trigger, but it poses no meaning to Change trigger.

4.14.2 GPWU Register Descriptions)

(Base Address = FF20h, 96 bytes

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr.	Description		
30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 3Ah 3Bh	**GPWUEN00** **GPWUEN08** **GPWUEN10** **GPWUEN18** **GPWUEN20** **GPWUEN28** **GPWUEN30** **GPWUEN38** **GPWUEN40** **GPWUEN48** **GPWUEN50** **GPWUEN58**	**GPIO 00~59 Event Enable and Asynchronous Wake Up Enable**			00h	FFh
		7~0	R/W	Enable bit to generate event (interrupt, and wakeup) for an active input. Also Enable bit for waking up from Deep Sleep mode. This bit is not cared by GPO-only channel.		
40h ~ 4Bh	**GPWUPF00** ~ **GPWUPF58**	**GPIO 00~59 Event Pending Flag**			00h	FFh
		0	R/WC1	GPIO 00~59 Event Pending Flag		
50h ~ 5Bh	**GPWUPS00** ~ **GPWUPS58**	**GPIO 00~59 Polarity Selection**			00h	FFh
		0	R/W	GPIO 00~59 input active polarity selection		
60h ~ 6Bh	**GPWUEL00** ~ **GPWUEL58**	**GPIO 00~59 Edge / Level Trigger Selection**			00h	FFh
		0	R/W	GPIO 00~59 input is edge or level trigger		

4.15 SMB

4.15.1 SMB Functional Description

There is only a SMBus host controller. 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, SMBus clock period setting, etc.

4.15.1.1 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) is acting as their data byte for reading or writing to SMBus transactions. For Read Word, WriteWord protocols, the **SMBDAT0** (offset 9Ch for SMB1) and **SMBDAT1** (offset 9Dh for SMB1) are used for their first data byte and 2nd data byte. As 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) to **SMBDAT31** (BBh). For the Write Block, the **SMBCNT** (offset BCh for SMB1) should be set before the protocol is issued. For Read Block, it is not necessary to set the **SMBCNT** before the protocol is issued, but read by software for knowing the real read data byte count.

Before a protocol is completed, the **SMBPRTCL** will remain the same value as written. After completion, the **SMBPRTCL** will be cleared to zero. Besides, the **SMBSTS** will return the completed status. Bit 7 of **SMBSTS** will be set to indicate the protocol is DONE. Bits 4~0 return the status indicating whether 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.

4.15.1.2 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); and the 2 written data bytes are placed in **SMBADAT0** (offset Beh) and **SMBADAT1** (offset BFh). 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.

4.15.1.3 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) bit 0 and 1 are used for host protocol completed and alarm received interrupt enabling.

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) 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.

4.15.2 SMB Register Descriptions

(Base Address = FF80h, 64 bytes)

Offset	Register Abbreviation	Register Full Name			Def	Bnk
		Bit	Attr	Description		
80~91h			NA	Reserved	00h	FFh
92h	**SMBTCRC**	**SMBus Transmit CRC Value**			00h	FFh
		7 – 0	RO	The CRC value transmit to SMBus		
93h	**SMBPIN**	**SMBus PIN Control**			00h	FFh
		7	R/W	**Write 0 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 0 to 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	RSV	Reserved		
		2	R/W	**SCL / SDA input debounce enable.** 0: de-bounce circuit disable 1: de-bounce circuit enable		
		1	R/W	**SMB controller uses SCL2 / SDA2 pins** 93h_bit_1: Map **SCL2 / SDA2** to SMB Controller		
		0	R/W	**SMB controller uses SCL1 / SDA1 pins** 93h_bit_0: Map **SCL1 / SDA1** to SMB Controller		
94h	**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	**SMBEN**	**SMBus Interrupt Enable**			00h	FFh
		7	RO	SMBus Host Controller is operating.		
		6~3	R/W	Reserved		
		2	R/W			
		1	R/W	SMBus alarm received interrupt enable		
		0	R/W	SMBus protocol completion interrupt enable		
96h			RSV	Reserved	00h	FFh
97h	**SMBRCRC**	**SMBus Received CRC Value**			00h	FFh
		7 – 0	R/W	The latest CRC value received from SMBus slave device		
98h	**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	**SMBSTS**	**EC SMBus Status**				00h	FFh
		7	R/WC0	SMBus command done flag			
		6	R/WC0	SMBus alarm received flag			
		5	R/WC0	SMBus Block Protocol Data Array flag, clear this bit to allow Block Protocol to continue if the Block Protocol is not finished.			
		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	Device data no acknowledge		
				17h	Device access denied		
				18h	SMBus timeout		
				19h	Unsupported protocol		
				1Ah	SMBus busy		
				1Fh	PEC error		
				Others	Reserved		
9Ah	**SMBADR**	**EC SMBus Address Field**				00h	FFh
		7 – 0	R/W	EC SMBus address. The bit 0 is don't care.			
9Bh	**SMBCMD**	**EC SMBus Command Field**				00h	FFh
		7 – 0	R/W	EC SMBus command. The KB3920 SMBus Host controller is SMBus 2.0 compliant. The Send Byte data does not use Command field again. This is different from KB3886 (The previous version)			
9Ch ~ A3h	**SMBDAT**	**EC SMBus Data Array**				00h	FFh
		7 – 0	R/W	EC SMBus data array (total 8 bytes)			
				SMBDAT0 (9Ch) are used for Send/ Receive/ Read Byte/ Write Byte protocol data byte and Read Word / Write Word protocol first data byte			
				SMBDAT1 (9Dh) is used for Read Word / Write Word protocol 2nd data byte.			
				Others are used for block mode protocols. If the block protocol byte count is larger than 8 bytes, the SMBSTS bit 5 will be set and an interrupt will be issued to firmware for receiving the data array. After firmware stores the received bytes, the SMBSTS bit 5 should be cleared to continue the block protocol.			

BCh	**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	**SMBAADR**	**EC SMBus Alarm Address**			00h	FFh
		7 – 0	R/W	EC SMBus alarm address from SMBus master device		
Beh	**SMBADAT0**	**EC SMBus Alarm Data 0**			00h	FFh
		7 – 0	R/W	EC SMBus alarm data0 (Low byte) from SMBus master device		
BFh	**SMBADAT1**	**EC SMBus Alarm Data 1**			00h	FFh
		7 – 0	R/W	EC SMBus alarm data1 (High byte) from SMBus master device		

4.16 8051

4.16.1 8051 Functional Description

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

KB3920 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 circumstances the 2 timers will not be used for baud-rate generation but for other purposes.

The 8051 uses **MOVX** and **MOVC** instructions to read or write KB3920 peripherals, i.e., **EC, SMBus, GPIO, GPWU, KBC, IKB, GPT, PWM, PS2, XBI, LPC, XRAM**…etc.

Hereunder lists the differences between the KB3920's embedded 8051 and that of the industrial standard 8051:

4.16.1.1 Interrupt priorities for each channel is fixed, and no nested 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 KB3920 doesn't use multi-level priority interrupt.

4.16.1.2 Improved Instruction Execution Time

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

4.16.1.3 8051 Structure

The original 8051 adopts Harvard Architecture (independent Code and Data paths), while the 8051 in KB3920 is using 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 KB3920 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. But the internal 2KB SRAM can't be used as code space.

4.16.1.4 Execution Time

The execution time of instruction **MUL** and **DIV** are the same as normal instructions.

4.16.2 Interrupt Vectors Table

Interrupt Source	Vector Address	KB3920 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 0 (***GPIO1B***)	03h	Highest
TF1	001Bh	8051 Timer 1	04h	Highest
RI & TI	0023h	8051 Serial Port	05h	Highest
P0I.0	0043h	**WDT**	08h	High
P0I.1	004Bh	**N.A.**	09h	
P0I.2	0053h	**PS/2**	0Ah	
P0I.3	005Bh	**KBC** Host Interface interrupt	0Bh	
P0I.4	0063h	**IKB**	0Ch	
P0I.5	006Bh	LPC interrupt	0Dh	
P0I.6	0073h	**EC** Host Interface interrupt	0Eh	
P0I.7	007Bh	N.A.	0Fh	
P1I.0	0083h	**FAN1**	10h	
P1I.1	008Bh	**FAN2**	11h	
P1I.2	0093h	**SMBus**	12h	
P1I.3	009Bh	N.A.	13h	
P1I.4	00A3h	**GPT0**	14h	
P1I.5	00Abh	**GPT1**	15h	
P1I.6	00B3h	**GPT2**	16h	
P1I.7	00BBh	**GPT3**	17h	
P3I.0	00C3h	***GPIO00~0F***	18h	
P3I.1	00CBh	***GPIO10~1F***	19h	
P3I.2	00D3h	***GPIO20~2F***	1Ah	
P3I.3	00DBh	***GPIO30~3F***	1Bh	
P3I.4	00E3h	***GPIO40~4F***	1Ch	
P3I.5	00Ebh	***GPIO50~59***	1Dh	
P3I.6	00F3h	***GPIO50~59***	1Eh	
P3I.7	00FBh	**ADC** updated	1Fh	Lowest

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

4.16.3 SFR Map

Color Information

XXX	Original Industrial standard 8051 features
XXX	KB3920's embedded 8051 new features
XXX	XXX Changed 8051 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	SCON3					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 KB3920, the 3 ports are used for interrupt input (always rise pulses) extensions. Totally there are 24 interrupt events.

P3IF, P1IF, P0IF are Interrupt Flag(IF) corresponding to the 24 interrupt inputs. The Ifs are set by external interrupt event (always a rising pulse, one clock width), and are cleared by software (execute IRET instruction for active interrupt).

The original alternate 8051 port 3 functions are not related with **P3IE** and **P3IF**.

4.16.4 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 (KB3920 should set to 0)	
		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
		7~6	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.	
		5~4	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: 8-bit shift register mode, E51RX should be E51CLK shift clock. 01: 8-bit Serial Port (variable) 10: 9-bit Serial Port (variable) 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 mode 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 / 3 Register**				00h
9Bh	**SCON3**	7~0	R/W	**SCON2** is high byte, **SCON3** is low byte to be a 16-bit counter for baud rate based on 8051 clock.		
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 Rating

Symbol	Parameter	Condition	Ratings	Unit
VCC	Power source voltage	All voltages are referenced to 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 Condition

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	3.3	3.6	V
AGND	Analog ground voltage		0		V

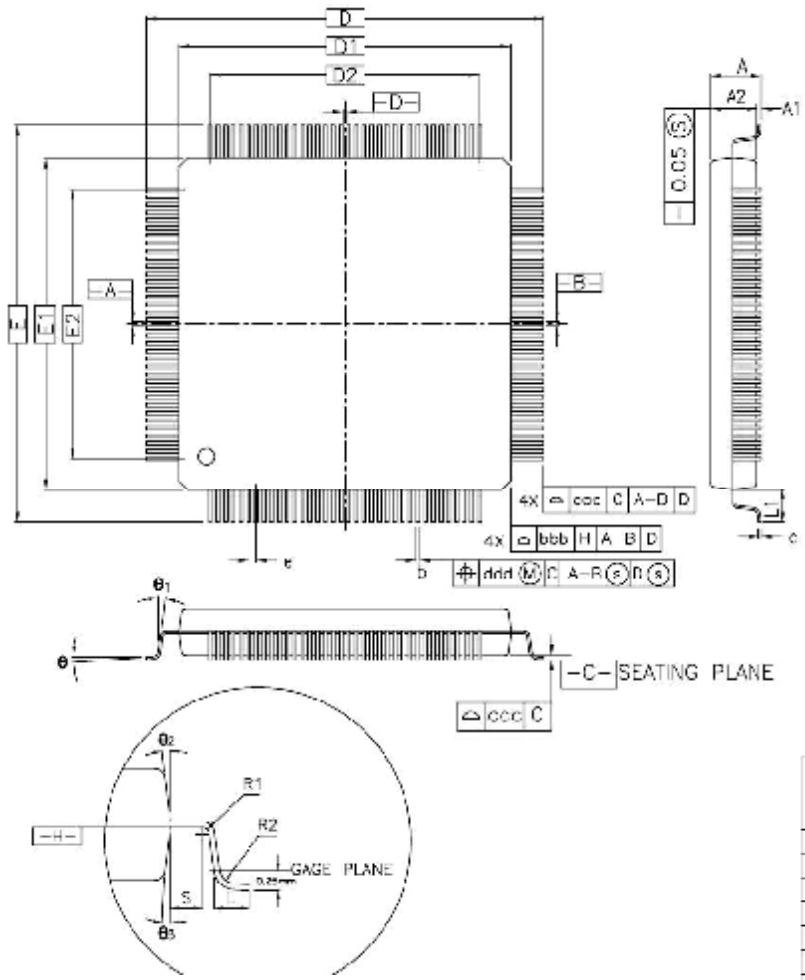
5.3 Operating Current

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
I_{CC}	Typical current consumption in operating state under Windows environment: all clock domains are running, no PS2/KB/mouse actions		TBD		mA

6. Packaging Information

6.1 144 LQFP

(20mm x 20mm x 1.4mm)

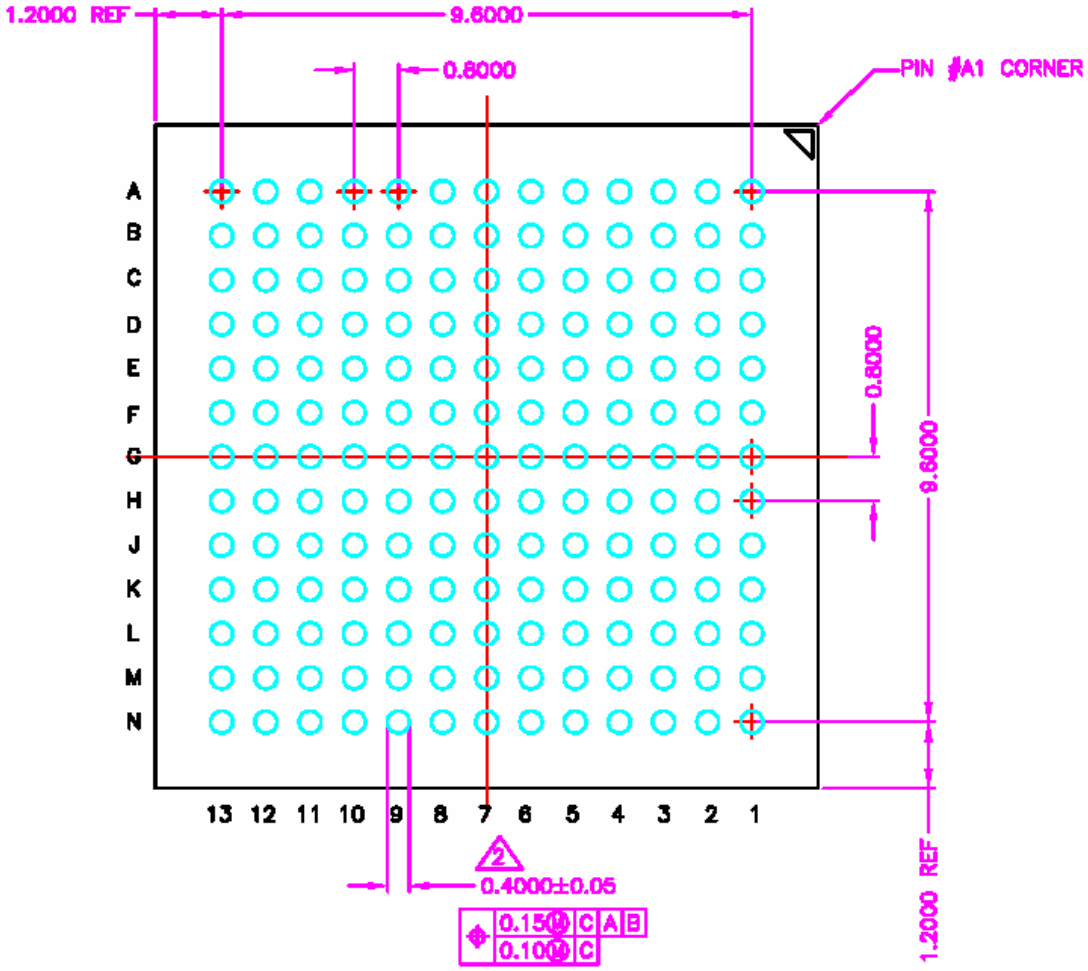


CONTROL DIMENSIONS ARE IN MILLIMETERS.

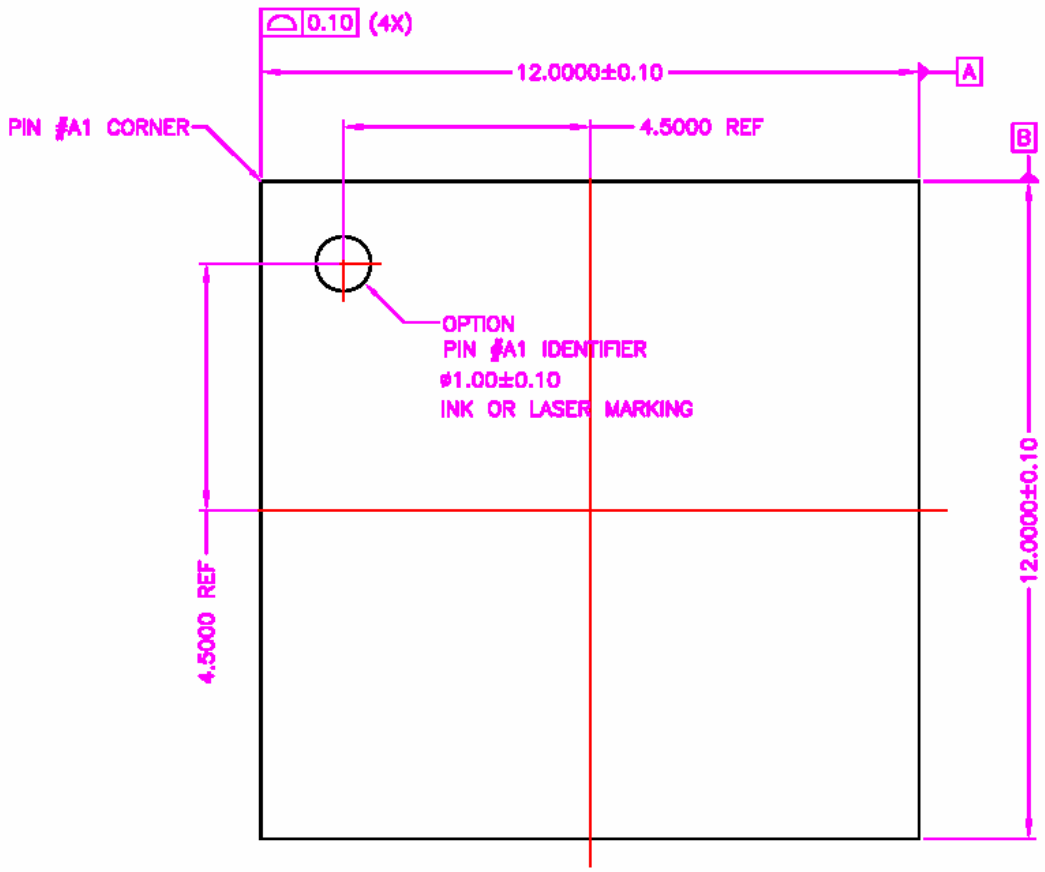
SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	22.00 BSC.			0.866 BSC.		
D1	20.00 BSC.			0.787 BSC.		
E	22.00 BSC.			0.866 BSC.		
E1	20.00 BSC.			0.787 BSC.		
R2	0.08	—	0.20	0.003	—	0.008
R1	0.08	—	—	0.003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ.	0°	—	—	0°	—	—
θ2	11°	12°	13°	11°	12°	13°
θ3	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
L1	1.00 REF			0.039 REF		
S	0.20	—	—	0.008	—	—

SYMBOL	144L					
	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	17.50			0.689		
E2	17.50			0.689		
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		

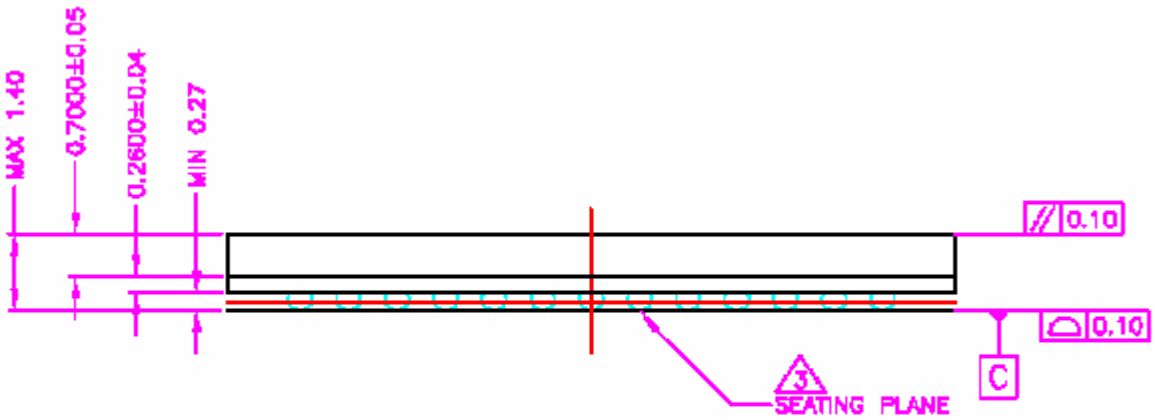
6.2 169 BGA



BOTTOM VIEW



TOP VIEW



SIDE VIEW

7. Revision History

Rev.	Preliminary/Changes	Date
0.10	• Initial Release	Oct. 25, 2005
0.11	• Format Change	Nov. 2, 2005
0.12	• Rename the one-way GPIO pins as GPI or GPO • Refine the description of hardware strap pins	Nov. 10, 2005
0.7	• Fix naming mistake in 2.3 Pin Assignment Side B, pin 65. • Fix I/O description in 2.2 Pin Assignment Side A, pin17,18,19 • Fix max. GPIO support number to be 90 in 1.2 • BGA definition added • Remove Alt. Output of pin76/78/79/80 to avoid misunderstanding. • Section 4.2.3 GPIO Register Description rewrite	Apr. 4. 2006
0.8	• In section 4.10.2, modify LPC SIRQ mode selection. Now LPC SIRQ is forced to continuous mode. • BGA coordinates information added in 6.2 • SCI# reset value mistake corrected in 2.3. • Correct some BGA information also update the ball-map.	June. 16. 2006