ENE TECHNOLOGY INC.

IO3731

Mobile IO Controller

Datasheet (OLPC)

V 1.2

Jan. 2011

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

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

Revision	Description	Date
0.1	Draft Release	2010/07
0.2	1. Correct Flash Clock Domain on block diagram 2. Fix GPIO number on comparison table 3. Fix typo on feature description	2010/07
0.3	1. Fix some naming typo	2010/09
0.4	1. Correct chip id related information. Eg: CHIP_ID_L register	2010/09
0.5	1. Update SVT testing data, power-consumption, except AD/DA 2. Fix ADCTRL channel selection typo 3. Fix Rx/Tx pin-out typo 4. Fix PWM bits typo in introduction section. 5. Update recommended power source minimum voltage 6. Update HW trap and MFPCTRL for pins selection notes 7. Update pin related information for precise CLK input and remove pin49/50 related description	2010/10
0.6	1. Naming fix of register 32KCR -> CRY32CR 2. Update DA testing data	2010/10
0.7	1. Update SMB general control in MISC register section 2. Fix typo of SMB in Pin description by function 3. Fix initial value of Reset# L/H of GPIO02/03/04 to correspond related functional trap pins	2010/11
0.8	1. Fix PS2 port, register MFPCTRL, and related ports/pins typo	2010/12
0.9	1. Update thermal information 2. Update IC version into A1 3. Update trap pins, remove GPIO02/03, change GPIO04 4. Update Register CRY32CR description for correctly control	2011/01
1.0	1. Update VER_ID register 2. Update IO cell 5V tolerance information	2011/01
1.1	1. Update naming typo, PS2_INTPF, SMB1DAT0~SMB1DAT31, SMB0ADAT0,SMB0ADAT1,SMB1ADAT0,SMB1ADAT1 2. Arrange 0xF018, 0xF019, 0xF01A, 0xF01C, 0xF01D for IC information	2011/01
1.2	1. Update PS2 port3 and related description in pin-assignment, selection register **MFPCTRL,** and PS2 section 2. Update **XBIEFCFG**[7] description for flag status description	2011/01

1. General Description

1.1 Overview

IO3731 is a customized embedded controller (EC) of IO373x series. It is designed with external crystal device for high precise timing control.

The ENE IO373x is a highly customized embedded controller (EC) for mobile platforms. The embedded controller contains industrial standard 8051 microprocessor and provides function of i8042 keyboard controller basically.

IO373x uses SPI/I2C to communicate with Host. The embedded controller also features rich interfaces for general applications by different parts selection, such as PS/2 interface, Keyboard matrix encoder, PWM controller, A/D converter, D/A converter, SMBus controller, GPIO controller, one wire master, SPI controller, CIR, etc. It also supports resistive touch screen mode and CEC to provide further integration of handheld devices.

IO373x provide cost effective solution for mobile application to improve system integration and customized design. For detail improvement, please refer the related section.

1.2 Features

8051 Microprocessor

- Compatible with industrial 8051 instructions
- 8051 runs at 4/8/16/32 MHz, programmable.
- 256 bytes internal RAM
- 24 extended interrupt sources.
- Two 16-bit timers.
- Full duplex UART integrated.
- Supports idle and stop mode.
- Enhanced ENE debug interface.

PS/2 Controller

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

Internal Keyboard Matrix (IKB)

- 18x8, 10x8, 8x8 keyboard scan matrix by different parts.
- Support hotkey events defined.
- Ghost key cancellation mechanism provided.
- 8 special key (When KSOs are floating, KSIs detects the low signal.)

SMBus Host / Slave Controller

- Support at most 2 SMBus interfaces
- SMBus Spec 2.0 compliant.
- Interrupt to 8051 for transaction completed management.
- Clock rate up to 400Khz.

Digital-to-Analog Converter (DAC)

- 2 DAC channels with 8-bit resolution.
- All pins of DAC can be alternatively configured as GPO.

Analog-to-Digital Converter (ADC)

- 6 ADC channels with 12-bit resolution.
- Support 4-wire resistive touch screen solution
- All pins of ADC can be alternatively configured as GPI.

Pulse Width Modulator (PWM)

- 4 PWM channels are provided.
  - One 32 bits PWM
  - Three 12 bits PWM
- Clock source is programmable.
- Push-Pull mode / Open-Drain PWM support.

Watch Dog Timer (WDT)

- 32.768KHz input clock.
- 10-bit counter with 32ms unit for watchdog reset.
- Two watchdog reset mechanisms.
  - Reset 8051 only
  - Reset whole chip, (Reset GPIO module selectable)

General Purpose Timer (GPT)

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

General Purpose Wakeup (GPWU)

- Those I/O with GPI (general purpose input) configuration can generate interrupts or wakeup events

General Purpose Input/Output (GPIO)

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

Digital Sampler

- Two digital sampler ports
- Clock source is programmable.

Consumer IR (CIR)

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

Consumer Electronic Control (CEC)

- Complies with Consumer Electronic Control (CEC) version 1.3a

ENE Debug Interface (EDI)

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

Serial Peripheral Interface (SPI)

- SPI master / slave mode support
- Flexible design for SPI applications.

One Wire Master Interface

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

Power Management

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

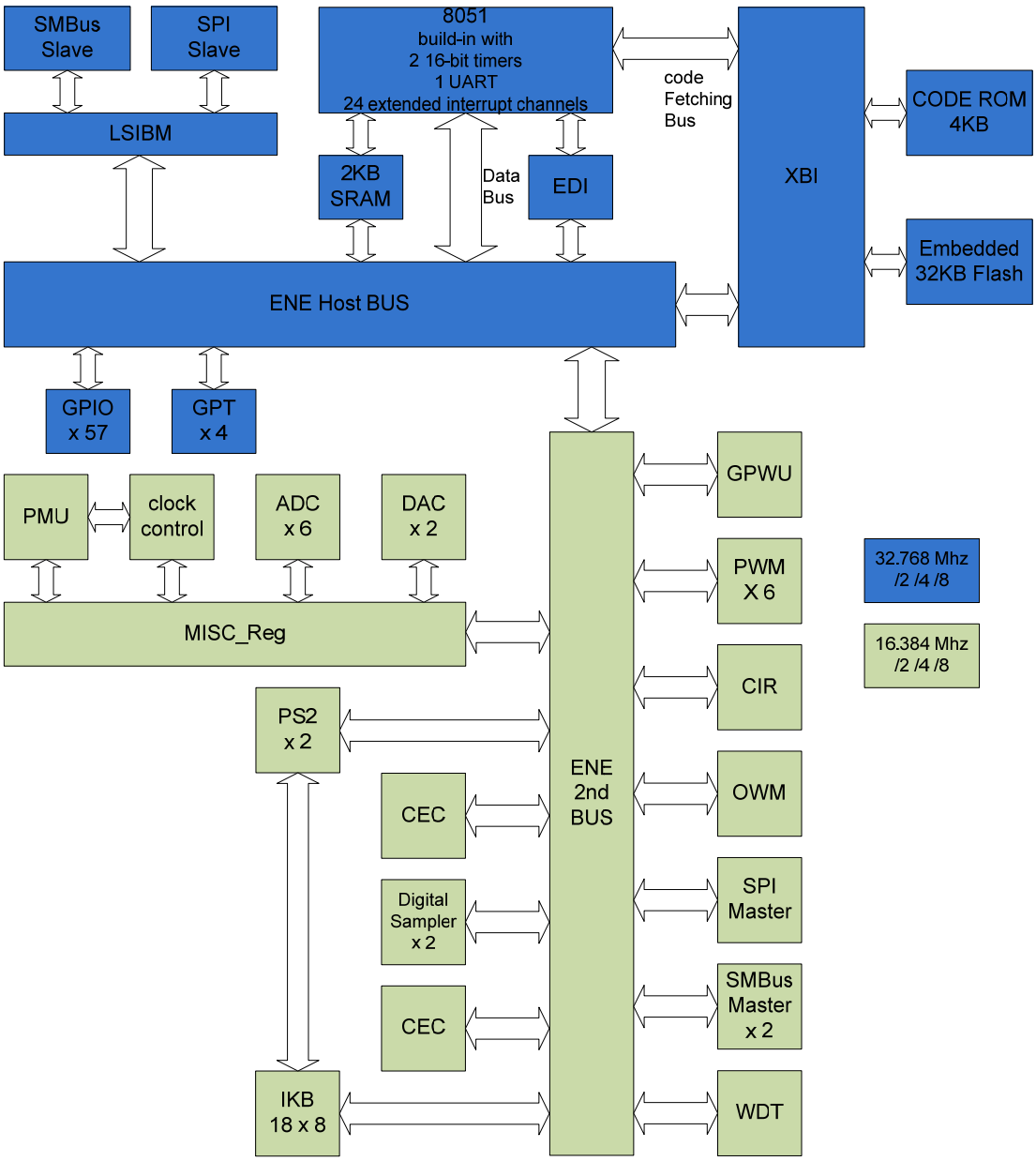
Storage.

- Embedded 32KB flash ROM
- Embedded 4KB mask-ROM

Package

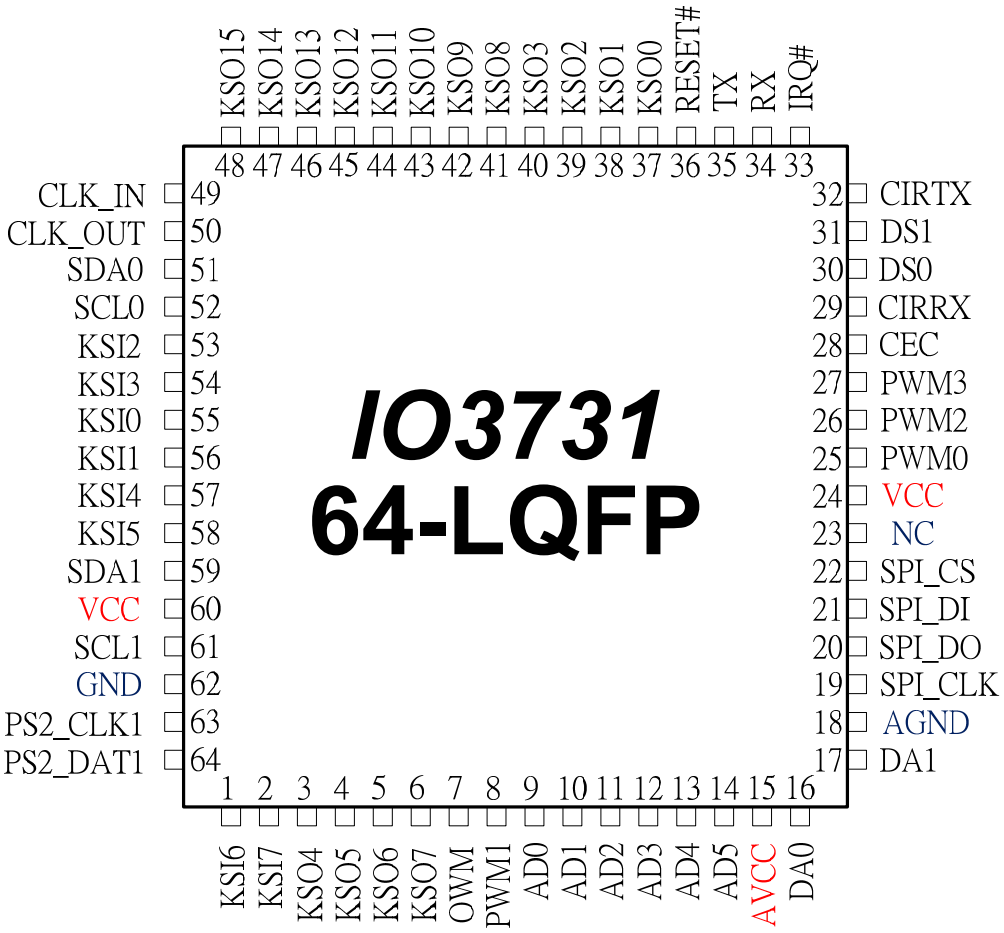
- 64 pins LQFP/ LFBGA package, Lead Free (RoHS).

1.3 Block Diagram



2. Pin Assignment and Description

2.1 IO3731 64-pin LQFP Diagram Top View



2.2.1 IO3731 Pin Assignment Side A

IO3731 Pin No.	Name	GPIO	Default	Reset# L/H	IO CELL
1	KSI6 / EDI_DI	GPIO00	GPIO00	IE(PU) / IE(PU)	BQC04IV
2	KSI7 / EDI_DO	GPIO01	GPIO01	IE(PU) / IE(PU)	BQC04IV
3	KSO4	GPIO02	GPIO02	IE(PU) / IE(PU)	BQC04IV
4	KSO5	GPIO03	GPIO03	IE(PU) / IE(PU)	BQC04IV
5	KSO6	GPIO04	GPIO04	IE(PU) / IE(PU)	BQC04IV
6	KSO7	GPIO05	GPIO05	HiZ / HiZ	BQC04IV
7	OWM	GPIO06	GPIO06	HiZ / HiZ	BQC04IV
8	PWM1	GPIO07	GPIO07	HiZ / HiZ	BQC04IV
9	AD0	**GPI**08	**GPI**08	HiZ / HiZ	IQADCI
10	AD1	**GPI**09	**GPI**09	HiZ / HiZ	IQADCI
11	AD2	**GPI**0A	**GPI**0A	HiZ / HiZ	IQADCI
12	AD3	**GPI**0B	**GPI**0B	HiZ / HiZ	IQADCI
13	AD4	**GPI**0C	**GPI**0C	HiZ / HiZ	IQTI
14	AD5	**GPI**0D	**GPI**0D	HiZ / HiZ	IQTI
15	AVCC				AVCC
16	DA0	**GPO**0E	**GPO**0E	HiZ / HiZ	OCT04

2.2.2 IO3731 Pin Assignment Side B

IO3731 Pin No.	Name	GPIO	Default	ECRST# L/H	IO CELL
17	DA1	**GPO**0F	**GPO**0F	HiZ / HiZ	OCT04
18	AGND				AGND
19	SPI_CLK	GPIO10	GPIO10	HiZ / IE	BQC16IV
20	SPI_DO	GPIO11	GPIO11	HiZ / IE	BQC16IV
21	SPI_DI	GPIO12	GPIO12	HiZ / IE	BQC16IV
22	SPI_CS#	GPIO13	GPIO13	HiZ / IE	BQC16IV
23	NC				NC
24	VCC				VCC
25	PWM0	GPIO14	GPIO14	HiZ / HiZ	BQC04IV
26	PWM2	GPIO15	GPIO15	HiZ / HiZ	BQC04IV
27	PWM3	GPIO16	GPIO16	HiZ / HiZ	BQC04IV
28	CEC	GPIO17	GPIO17	HiZ / HiZ	BQC04IV_26K
29	CIRRX / KSO16	GPIO18	GPIO18	HiZ / HiZ	BQC04IV
30	DS0	GPIO19	GPIO19	HiZ / HiZ	BQC04IV
31	DS1	GPIO1A	GPIO1A	HiZ / HiZ	BQC04IV
32	CIRTX / KSO17	GPIO1B	GPIO1B	HiZ / HiZ	BQC04IV

2.2.3 IO3731 Pin Assignment Side C

IO3731 Pin No.	Name	GPIO	Default	ECRST# L/H	IO CELL
33	IRQ#	GPIO1C	GPIO1C	PU / PU	BQC04IV
34	RX	GPIO1D	GPIO1D	HiZ / HiZ	BQC04IV
35	TX	GPIO1E	GPIO1E	HiZ / HiZ	BQC04IV
36	RESET#			IE / IE	BQC04IV
37	KSO0	GPIO1F	GPIO1F	IE(PU) / IE(PU)	BQC04IV
38	KSO1	GPIO20	GPIO20	IE(PU) / IE(PU)	BQC04IV
39	KSO2	GPIO21	GPIO21	IE(PU) / IE(PU)	BQC04IV
40	KSO3	GPIO22	GPIO22	IE(PU) / IE(PU)	BQC04IV
41	KSO8	GPIO23	GPIO23	HiZ / HiZ	BQC04IV
42	KSO9	GPIO24	GPIO24	HiZ / HiZ	BQC04IV
43	KSO10	GPIO25	GPIO25	HiZ / HiZ	BQC04IV
44	KSO11	GPIO26	GPIO26	HiZ / HiZ	BQC04IV
45	KSO12	GPIO27	GPIO27	HiZ / HiZ	BQC04IV
46	KSO13	GPIO28	GPIO28	HiZ / HiZ	BQC04IV
47	KSO14	GPIO29	GPIO29	HiZ / HiZ	BQC04IV
48	KSO15	GPIO2A	GPIO2A	HiZ / HiZ	BQC04IV

2.2.4 IO3731 Pin Assignment Side D

IO3731 Pin No.	Name	GPIO	Default	ECRST# L/H	IO CELL
49	CLK_IN				
50	CLK_OUT				
51	SDA0	GPIO2D	SDA0	HiZ / IE	BQC04IV
52	SCL0	GPIO2E	SCL0	HiZ / IE	BQC04IV
53	KSI2	GPIO2F	GPIO2F	IE(PU) / IE(PU)	BQC04IV
54	KSI3	GPIO30	GPIO30	IE(PU) / IE(PU)	BQC04IV
55	KSI0	GPIO31	GPIO31	IE(PU) / IE(PU)	BQC04IV
56	KSI1	GPIO32	GPIO32	IE(PU) / IE(PU)	BQC04IV
57	KSI4 / EDI_CS	GPIO33	GPIO33	IE(PU) / IE(PU)	BQC04IV
58	KSI5 / EDI_CLK	GPIO34	GPIO34	IE(PU) / IE(PU)	BQC04IV
59	SDA1 / PS2_DAT3	GPIO35	GPIO35	HiZ / HiZ	BQC04IV
60	VCC				VCC
61	SCL1 / PS2_CLK3	GPIO36	GPIO36	HiZ / HiZ	BQC04IV
62	GND				GND
63	PS2_CLK1	GPIO37	GPIO37	HiZ / HiZ	BQC04IV
64	PS2_DAT1	GPIO38	GPIO38	HiZ / HiZ	BQC04IV

2.3 I/O Cell Descriptions

2.3.1 I/O Buffer Table

Cell	Description	Application
BQC04IV	Schmitt trigger, 4mA Output / Sink Current, Input / Output / Pull Up Enable(40KΩ) , 5V tolerance	GPIO
BQC04IV_26K	Schmitt trigger, 4mA Output / Sink Current, Input / Output / Pull Up Enable(26KΩ) , 5V tolerance	CEC
BQC16IV	Schmitt trigger, 16mA Output / Sink Current, Input / Output / Pull Up Enable(40KΩ) , 5V tolerance	SPI
IQADCI	Mixed mode IO, ADC enable with GPI, Schmitt trigger	ADC / GPI
IQTI	Mixed mode IO, ADC enable, with GPI, Schmitt trigger	ADC / GPI
OCT04	Mixed mode IO, DAC enable, with GPO, 4mA Output Current	DAC / GPO

2.3.2 I/O Buffer Characteristic Table

Cell	Output	Input	Analog Signal	Pull-High Enable(40k)	5V Tolerance	Current (mA)	Application
BQC04IV	V	V		V	V	2~4	GPIO
BQC04IV_26K	V	V		V (With 26K)	V	2~4	CEC
BQC16IV	V	V		V	V	8~16	SPI
IQADCI		V	V				ADC / GPI
IQTI		V	V				ADC / GPI
OCT04	V		V				DAC / GPO

3. Pin Descriptions

Notes for Hardware Trap:

Please be noted that, these four pins are used for testing and flow control. Please reserve these pins to **PULL-HIGH** for proper operation.

Pin Name	Pin No.
GPIO1F	37
GPIO20	38
GPIO21	39
GPIO22	40

Special Functions Hardware Trap:

Trap Name	Pin No.	Description
GPIO04	5	Trap pin for E51_RSTN reset force active **Low:** E51 Reset signal is forced to active **High:** E51 Reset signal is controlled by internal logic. (Default)

Multi- Alt-Function Pins Selection:

Pin Name	Pin No.	Register Field
CIRRX / KSO16	29	MFPCTRL[1] 0 for CIR 1 for IKB
CIRTX / KSO17	32	
SDA1 / PS2_DAT3	59	MFPCTRL[2] 0 for SMB 1 for PS2
SCL1 / PS2_CLK3	61	

Please refer **MFPCTRL** pages for other related register setting.

3.1 Pin Descriptions by Functions

3.1.1 PS/2 I/F Descriptions

Pin Name	Pin No.	Direction	Description
PS2_DAT3	59	I/O	PS/2 port 3 data Muxed with SMBus SDA1
PS2_CLK3	61	I/O	PS/2 port 3 clock Muxed with SMBus SCL1
PS2_CLK1	63	I/O	PS/2 port 1 clock
PS2_DAT1	64	I/O	PS/2 port 1 data

3.1.2 Internal Keyboard Encoder (IKB) Descriptions

Pin Name	Pin No.	Direction	Description
KSI6, KSI7	1-2	I	Keyboard Scan In
KSO4-KSO7	3-6	O	Keyboard Scan Out
KSO16	29	O	Keyboard Scan Out Muxed with CIRRX
KSO17	32	O	Keyboard Scan Out Muxed with CIRTX
KSO0-KSO3	37-40	O	Keyboard Scan Out
KSO8-KSO15	41-48	O	Keyboard Scan Out
KSI2-KSI3	53-54	I	Keyboard Scan In
KSI0-KSI1	55-56	I	Keyboard Scan In
KSI4-KSI5	57-58	I	Keyboard Scan In

3.1.3 SMBus Descriptions

Pin Name	Pin No.	Direction	Description
SDA0	51	I/O	SMBus port0 data (Default as interface with host)
SCL0	52	I/O	SMBus port0 clock (Default as interface with host)
SDA1	59	I/O	SMBus port1 data Muxed with PS2_DAT3
SCL1	61	I/O	SMBus port1 clock Muxed with PS2_CLK3

3.1.4 Digital Sampler

Pin Name	Pin No.	Direction	Description
DS0	30	I	Digital sampler port0
DS1	31	I	Digital sampler port1

3.1.5 Pulse Width Modulation (PWM) Descriptions

Pin Name	Pin No.	Direction	Description
PWM1	8	O	PWM pulse output port1
PWM0	25	O	PWM pulse output port0
PWM2-PWM3	26-27	O	PWM pulse output port2, port3

3.1.6 Analog-to-Digital Converter Descriptions

Pin Name	Pin No.	Direction	Description
AD0-AD5	9-14	I	12bit A/D converter input

3.1.7 Digital-to-Analog Converter Descriptions

Pin Name	Pin No.	Direction	Description
DA0-DA1	16-17	O	8bit D/A converter output

3.1.8 Universal Asynchronous Receiver Transmitter (UART) Descriptions

Pin Name	Pin No.	Direction	Description
RX	34	I	UART data in
TX	35	O	UART data out

3.1.9 One Wire Master (OWM) Description

Pin Name	Pin No.	Direction	Description
OWM	7	I/O	One Wire Master input and output signal

3.1.10 Serial Peripheral Interface (SPI) Description

Pin Name	Pin No.	Direction	Description
SPI_CLK	19	I or O	SPI clock, In SPI slave mode, clock is driven by host, pin as Input In SPI master mode, drive clock to devices, pin as Output
SPI_DO	20	O	SPI serial data output
SPI_DI	21	I	SPI serial data input
SPI_CS#	22	I or O	SPI chip select In SPI slave mode, clock is driven by host, pin as Input In SPI master mode, drive clock to devices, pin as Output

3.1.11 Consumer Infrared Radiation (CIR) Description

Pin Name	Pin No.	Direction	Description
CIRRX	29	I	CIR receive data
CIRTX	32	O	CIR transmit data

3.1.12 Consumer Electronics Control (CEC) Description

Pin Name	Pin No.	Direction	Description
CEC	28	I/O	CEC

3.1.13 Power Pins Descriptions

Pin Name	Pin No.	Direction	Description
AVCC	15		Power supply for analog plane.
AGND	18		Power ground for analog plane.
GND	62		Power ground for digital plane.
VCC	24, 60		Power supply for digital plane.
NC	23		Reserved for IC manufacture

4. Module Descriptions

4.1 Chip Architecture

4.1.1 Power Planes

Power planes are ±10% tolerance for recommend operation condition, IO373x provides are with two power planes for different modules

Power Plane	Description	Power	Ground
Digital Plane	This power provides power for all digital logic no matter what power mode is.	VCC	GND
Analog Plane	This power provides power for all analog logic, such as A/D and D/A converter.	AVCC	AGND

4.1.2 Internal Memory Map

No	Module	Descriptions	Address Range	Size (Byte)	
1	Code ROM	8051 Code Mask-ROM	0x0000~0x0FFF	4K	
2	Flash	Embedded Flash 32KB	0x0000~0x7FFF	32K	
3	XRAM	Embedded SRAM	0x8000~0x87FF	2K	
4	MISC	Misc Registers	0xF000~0xF03F	64	4K
5	RSV	Reserved	0xF040~0xFBFF	3008	
6	GPIO	General purpose I/O	0xFC00~0xFC7F	128	
7	CEC	Consumer Electronics Control Interface	0xFC80~0xFC9F	32	
8	IKB	Internal keyboard matrix	0xFCA0~0xFCAF	16	
9	RSV	Reserved	0xFCB0~0xFCEF	64	
10	OWM	One Wire Master	0xFCF0~0xFCFF	16	
11	SMB	SMBus Device Controller (H/W mode)	0xFD00~0xFD0F	16	
12	SMB	SMBus Device Controller (F/W mode)	0xFD10~0xFD2F	32	
13	RSV	Reserved	0xFD30~0xFD3F	16	
14	SMB	SMBus Slave Controller 0	0xFD40~0xFD5F	32	
15	SMB	SMBus Slave Controller 1	0xFD60~0xFD7F	32	
16	RSV	Reserved	0xFD80~0xFDFF	128	
17	PWM	Pulse width modulation	0xFE00~0xFE1F	32	
18	DS	Digital Sampler	0xFE20~0xFE3F	32	
19	SPID	SPI device Interface (Connected to SPI master)	0xFE40~0xFE4F	16	
20	GPT	General purpose timer	0xFE50~0xFE6F	32	
21	SDIH	SPI host interface (Connected to SPI slave)	0xFE70~0xFE7F	16	
22	WDT	Watchdog timer	0xFE80~0xFE8F	16	
23	RSV	Reserved	0xFE90~0xFE9F	16	
24	XBI	X-bus interface	0xFEA0~0xFEBF	32	
25	CIR	Consumer IR controller	0xFEC0~0xFECF	16	
26	RSV	Reserved	0xFED0~0xFEDF	16	
27	PS2	PS/2 interface	0xFEE0~0xFEFF	32	
28	RSV	Reserved	0xFF00~0xFF2F	48	
29	GPWU	General purpose wakeup event	0xFF30~0xFF7F	80	
30	SMB	SMBus Master Controller 0	0xFF80~0xFFBF	64	
31	SMB	SMBus Master Controller 1	0xFFC0~0xFFFF	64	

4.2 MISC Register Description (General Setting/ADDA/Power/CLK/etc.)

Table for Panel Drive Mode :

PDM	Input Channel	X Driver	Y Driver	Measurement	Description
0	ADC 0~6	OFF	OFF	ADC	Normal ADC
1	0 (X+)	OFF	1 (Y+), VCC 3 (Y-), GND	Y Position	4-Wire
2	0 (X+)	2 (X-) , GND	1 (Y+), VCC	Z1 Position	4-Wire
3	3 (Y-)	2 (X-), GND	1 (Y+), VCC	Z2 Position	4-Wire
4	1 (Y+)	0 (X+) , VCC 2 (X-) , GND	OFF	X Position	4-Wire
5	4 (WIPER)	0 (UL) , VCC 2 (LL), GND	1 (UR), VCC 3 (LR), GND	Y Position	5-Wire
6	4 (WIPER)	0 (UL), VCC 2 (LL), VCC	1 (UR), GND 3 (LR), GND	X Position	5-Wire
7	0 (X+)	0 (X+), PU	3 (Y-), GND	PENIRQ	4-Wire
8	4 (WIPER)	4 (WIPER), PU	3 (Y-), GND	PENIRQ	5-Wire

4-Wire Touch, Note:

X+	ADC0
Y+	ADC1
X-	ADC2
Y-	ADC3

5-Wire Touch, Note:

Upper Left	ADC0
Upper Right	ADC1
Lower Left	ADC2
Lower Right	ADC3
WIPER	ADC4

SMBus Selection Diagram:
IO373x SMBus/SPI interface could be used to communication to host. Some other devices may also be controlled by IO373x. Please refer the following description and diagram.

There is a dedicate SMBus Device (Slave) used to communicate to host, which is configured by SMBD_PC to specific port.
2 SMBus slave controllers are configured by SMB_PC[1:0], SMB_PC[3:2]
2 SMBus master controllers are configured by SMB_PC[5:4], SMB_PC[7:6]
SMB_INTC could be configured for internal connection between SMBus.

In IO3731, there are 2 external SMBus ports: Port 0 (GPIO2D/GPIO2E), Port 1(GPIO35/GPIO36)
In IO3730, there are 3 external SMBus ports: Port 0 (GPIO2D/GPIO2E), Port 1(GPIO35/GPIO36), Port 2 (GPIO2B/GPIO2C) where Port 2 is used for high-precision crystal pads in IO3731.

SMBus Internal Connection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x01	SMB_INTC	7~3	RSV	Reserved	0x00	0xF0
		2	R/W	SMBus 0 is coonected with SMBus 2		
		1	R/W	SMBus 1 is coonected with SMBus 2		
		0	R/W	SMBus 0 is coonected with SMBus 1		

SMBus Port Connection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x02	SMB_PC	7~6	R/W	SMBus Master Controller 1 port connection	0x55	0xF0
		5~4	R/W	SMBus Master Controller 0 port connection		
		3~2	R/W	SMBus Slave Controller 1 port connection		
		1~0	R/W	SMBus Slave Controller 0 port connection		
Please refer 4.2 MISC register section for further description and diagram						

SMBus Device Controller Port Connection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x03	SMBD_PC	7~2	RSV	Reserved	0x00	0xF0
		1-0	R/W	SMBus Device Controller port connection, IO373x communication to system host via this interface **00**: SMBus Port 0 (GPIO2D/GPIO2E) **01**: SMBus Port 1 (GPIO35/GPIO36) **10**: SMBus port 2 (GPIO2B/GPIO2C) (Valid only for IO3730) **11**: Reserved		
Please refer 4.2 MISC register section for further description and diagram						

8051 Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x05	E51_STA	7-1	RSV	Reserved	0x01	0xF0
		0	RO	8051 Status **0**: 8051 is in STOP Mode **1**: 8051 is in Normal Mode		

PMU Control / Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0A	PMUCFG	7	WO	STOP mode enable **1**: Enable & enter STOP mode	0x00	0xF0
		6	WO	IDLE mode enable **1**: Enable & enter IDLE mode		
		5~4	RSV	Reserved		
		3	R/W	Enable reset 8051 while in STOP mode		
		2	R/W	Enable WDT interrupt wake up form STOP mode		
		1	R/W	Enable GPWU wake up from STOP mode		
		0	R/W	Enable Interrupt wake up from IDLE mode		

Clock Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0B	CLKCFG	7~4	RSV	Reserved	0x0C	0xF0
		3~2	R/W	**EHB / EPB** Normal Run Clock Selection **00**: 4 / 2 Mhz **01**: 8 / 4 Mhz **10**: 16 / 8 Mhz **11**: 32 / 16 Mhz (Default)		
		1~0	RSV	Reserved		

DAC0 Output Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0C	DAC0	7~0	R/W	Voltage setting for DA0 output Note : Please program as 0 when DAC0 is disabled.	0x00	0xF0

DAC1 Output Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0D	DAC1	7~0	R/W	Voltage setting for DA1 output Note : Please program as 0 when DAC1 is disabled.	0x00	0xF0

8051 On-Chip Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0E	PXCFG	7-5	RSV	Reserved	0x00	0xF0
		4	R/W	WDT timeout rest setting for GPIO while reset whole chip **0**: GPIO not reset while reset whole chip **1**: GPIO reset while reset whole chip Note : Only valid when PXCFG[1]=0		
		3-2	RSV	Reserved		
		1	R/W	WDT timeout reset setting for 8051, only reset 8051 **0**: WDT reset whole chip (GPIO depends PXCFG[4]) **1**: WDT only reset 8051		
		0	RSV	Reserved		

ADC / DAC Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0F	ADDAEN	7	R/W	Enable bit for DA1	0x00	0xF0
		6	R/W	Enable bit for DA0		
		5	R/W	Enable bit for AD5		
		4	R/W	Enable bit for AD4		
		3	R/W	Enable bit for AD3		
		2	R/W	Enable bit for AD2		
		1	R/W	Enable bit for AD1		
		0	R/W	Enable bit for AD0		

8051 Reset Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x10	E51_RST	7-6	RSV	Reserved	0x00	0xF0
		0	R/W	Control for 8051 0: Run mode 1: Reset / Stop mode		

Code Source selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x11	CODE_SEL	7-6	RSV	Reserved	0x00	0xF0
		0	R/W	Selection for the code memory source 0: Mask ROM 1: FLASH Note : For correctly switch code source, Step as follow: *1. Set E51_RST to reset 8051* *2. Wait XBIEFCFG[7]=1 for XBI IDLE* *3. Set CODE_SEL[0]=0 to change source to mask ROM* *4. Set E51_RST =0 to re-start 8051*		

Host Hand-Shaking						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x12	HOST_HS	7-0	R/W	Reserved for FW handshaking	0x00	0xF0

ADC Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x15	ADCTRL	7-5	RSV	Reserved	0x00	0xF0
		4-2	R/W	Selection for the ADC channel to be converted **000**: Select AD0 **001**: Select AD1 **010**: Select AD2 **011**: Select AD3 **100**: Select AD4 **101**: Select AD5		
		1	RSV	Reserved		
		0	R/W	Write 1 to start ADC converter and enable ADC converted interrupt		

ADC Data Output High Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x16	ADCDAT1	7-0	R/W	High 8 bits of ADC converted data as ADC[11:4]	0x00	0xF0

ADC Data Output Low Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x17	ADCDAT0	7-4	R/W	Low 4 bits of ADC converted data as ADC[3:0]	0x00	0xF0
		3	R/W1C	ADC pending IRQ flag (Enabled by ADCCTL2[0])		
		2	R/W1C	ADC Interrupt flag		
		1-0	RSV	Reserved		

Firmware ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x18	FIRMWARE_ID	7-0	R/W	Firmware version	0x00	0xF0

Version ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x19	VER_ID	7-0	RO	IC hardware version, 00:A0 01:A1 ……	0x01	0xF0

PID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1A	PID	7-2	RSV	Reserved		0xF0
		1-0	RO	PID, reserved for IC testing **00**: Reserved **01**: Reserved **10**: Reserved **11**: IO3730		

Chip ID High Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1C	CHIP_ID_H	7-0	RO	Chip ID high 8 byte	0x37	0xF0

Chip ID Low Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1D	CHIP_ID_L	7-0	RO	Chip ID low 8 byte	0x30	0xF0

Clock Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1F	CLKCFG2	7-0	R/W	1us PLL output clock timing divider The number should be use 1/2 PLL CLK for timing setting Eg: 64MHz as default, 1/2 PLL CLK is 32MHz, for 1us (1MHz), the number should 0x1F (32) Eg: 50Mhz as default, 1/2 PLL CLK is 25MHz, for 1us (1MHz), the number should be 0x18 (25)	0x1F	0xF0

EDI Configuration Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x22	EDICFG	7	R/W	EDI feature enable **0**: Disable **1**: Enable	0x80	0xF0
		6-0	RSV	Reserved		

EDI Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x23	EDISR	7	R/W	EDI Active Status **0**: non-active **1**: Active	0x00	0xF0
		6-0	RSV	Reserved		

EDI Version ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x24	EDIID	7-0	RO	EDI Version ID	0x03	0xF0

ADC Control Register 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x26	ADCCTL2	7-4	R/W	ADC Panel Drive Mode Selection For details pin definition, please refer MISC section table 0 : ADC mode Others : Specific Panel Drive Mdoe	0x04	0xF0
		3	RSV	Reserved		
		2	RO	ADC PENIRQ		
		1	R/W	ADC fast power-on enable		
		0	R/W	ADC Pending IRQ enable		

Multi-Function Pins Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x28	MFPCTRL	7-3	RSV	Reserved	0x00	0xF0
		2	R/W	PS2 port3 and SMBus port1 selection **0**: Select SDA1 in pin 59 Select SCL1 in pin 61 **1**: Select PS2_DAT3 in pin 59 Select PS2_CLK3 in pin 61		
		1	R/W	IKB and CIR selection **0**: Select CIR_RX in pin 29 Select CIR_TX in pin 32 **1**: Select KSO16 in pin 29 Select KSO17 in pin 32		
		0	RSV	Reserved		

32Khz Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2A	CRY32CR	7-5	RSV	Reserved	0x04	0xF0
		4	R/W	32Khz source selection **0**: 32Khz divided from 32Mhz source **1**: Internal 32Khz osc		
		3	R/W	**PS2/GPT/CIR/FAN/FANMON/PWM** clock source selection **0**: clock source from DPLL divider output 32KHz **1**: clock source from external 32Khz crystal (Only valid in IO3731) This should be set for normal setting after 32KHz crystal being stable.		
		2	R/W	Oscillator source selection **0**: external (Only valid in IO3731) **1**: internal		
		1	RSV	Reserved		
		0	R/W	External Oscillator Enable (Only valid in IO3731) **0**: Disable **1**: Enable		

Internal OSC Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2B	IOSCCR	7	R/W	Power down enable **0**: internal clock is gated **1**: normal operation	0xA8	0xF0
		6	R/W	Power down mode setting **0**: Sleep **1**: Coma (longer startup time)		
		5	R/W	Turbo mode enable, Turbo mode is setting for faster start-up time from Power Down Mode **0**: disabled **1**: enable		
		4	R/W	Clock Speed selection **0**: 32KHz **1**: 1Khz		
		3-0	RSV	Reserved		

Clock Enable 1						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x30	CLKEN_1	7	RSV	Reserved	0x7F	0xF0
		6	R/W	Clock enable bit for XRAM (on EHB)		
		5	R/W	Clock enable bit for XBI (on EHB)		
		4	R/W	Clock enable bit for SPI device (on EHB)		
		3	R/W	Clock enable bit for SMBus Device (on EHB)		
		2	R/W	Clock enable bit for GPT (on EHB)		
		1	R/W	Clock enable bit for EPBM		
		0	R/W	Clock enable bit for EDI (on EHB)		

Clock Enable 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x31	CLKEN_2	7	R/W	Clock enable bit for PS2 (on EPB)	0xFF	0xF0
		6	R/W	Clock enable bit for MISC_ADC (on EPB)		
		5	R/W	Clock enable bit for IKB (on EPB)		
		4	R/W	Clock enable bit for GPWU (on EPB)		
		3	R/W	Clock enable bit for Digital Sampler1 (on EPB)		
		2	R/W	Clock enable bit for Digital Sampler0 (on EPB)		
		1	R/W	Clock enable bit for CIR (on EPB)		
		0	R/W	Clock enable bit for CEC (on EPB)		

Clock Enable 3						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x32	CLKEN_3	7	RSV	Reserved	0x7F	0xF0
		6	R/W	Clock enable bit for OWM (on EPB)		
		5	R/W	Clock enable bit for SPI host (on EPB)		
		4	R/W	Clock enable bit for SMBus Slave1 (on EPB)		
		3	R/W	Clock enable bit for SMBus Slave0 (on EPB)		
		2	R/W	Clock enable bit for SMBus1 (on EPB)		
		1	R/W	Clock enable bit for SMBus0 (on EPB)		
		0	R/W	Clock enable bit for PWM (on EPB)		

Clock Enable 4						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x33	CLKEN_4	7-2	RSV	Reserved	0x03	0xF0
		1	R/W	Clock enable bit for CEC (on 32Khz)		
		0	R/W	Clock enable bit for WDT (on 32Khz)		

Low Voltage Detect Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x34	LVD_EN	7-1	RSV	Reserved	0x00	0xF0
		0	R/W	Enable bit for LVD Low voltage detect is used to detect the system power and for protection of system operation. Since embedded Flash and other logic module has different power level requirement. When system power is down than 1.58V, it would prevent 8051 from accessing embedded Flash data		

4.3 GPIO

4.3.1 GPIO Function Description

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

GPIO	IO3731 Alt.	Alt. Selection Reg.
GPIO00	KSI6	GPIOFS00.[0]
GPIO01	KSI7	GPIOFS00.[1]
GPIO02	KSO4	GPIOFS00.[2]
GPIO03	KSO5	GPIOFS00.[3]
GPIO04	KSO6	GPIOFS00.[4]
GPIO05	KSO7	GPIOFS00.[5]
GPIO06	OWM	GPIOFS00.[6]
GPIO07	PWM1	GPIOFS00.[7]
GPI08	AD0	GPIOFS08.[0]
GPI09	AD1	GPIOFS08.[1]
GPI0A	AD2	GPIOFS08.[2]
GPI0B	AD3	GPIOFS08.[3]
GPI0C	AD4	GPIOFS08.[4]
GPI0D	AD5	GPIOFS08.[5]
GPO0E★	DA0	GPIOFS08.[6]
GPO0F★	DA1	GPIOFS08.[7]
GPIO10	SPI_CLK	GPIOFS10.[0]
GPIO11	SPI_DO	GPIOFS10.[1]
GPIO12	SPI_DI	GPIOFS10.[2]
GPIO13	SPI_CS#	GPIOFS10.[3]
GPIO14	PWM0	GPIOFS10.[4]
GPIO15	PWM2	GPIOFS10.[5]
GPIO16	PWM3	GPIOFS10.[6]
GPIO17	CEC	GPIOFS10.[7]
GPIO18	CIRRX / KSO16	GPIOFS18.[0]
GPIO19	DS0	GPIOFS18.[1]
GPIO1A	DS1	GPIOFS18.[2]
GPIO1B	CIRTX / KSO17	GPIOFS18.[3]
GPIO1C	IRQ#	GPIOFS18.[4]
GPIO1D	RX	GPIOFS18.[5]
GPIO1E	TX	GPIOFS18.[6]
GPIO1F	KSO0	GPIOFS18.[7]
GPIO20	KSO1	GPIOFS20.[0]
GPIO21	KSO2	GPIOFS20.[1]
GPIO22	KSO3	GPIOFS20.[2]
GPIO23	KSO8	GPIOFS20.[3]
GPIO24	KSO9	GPIOFS20.[4]

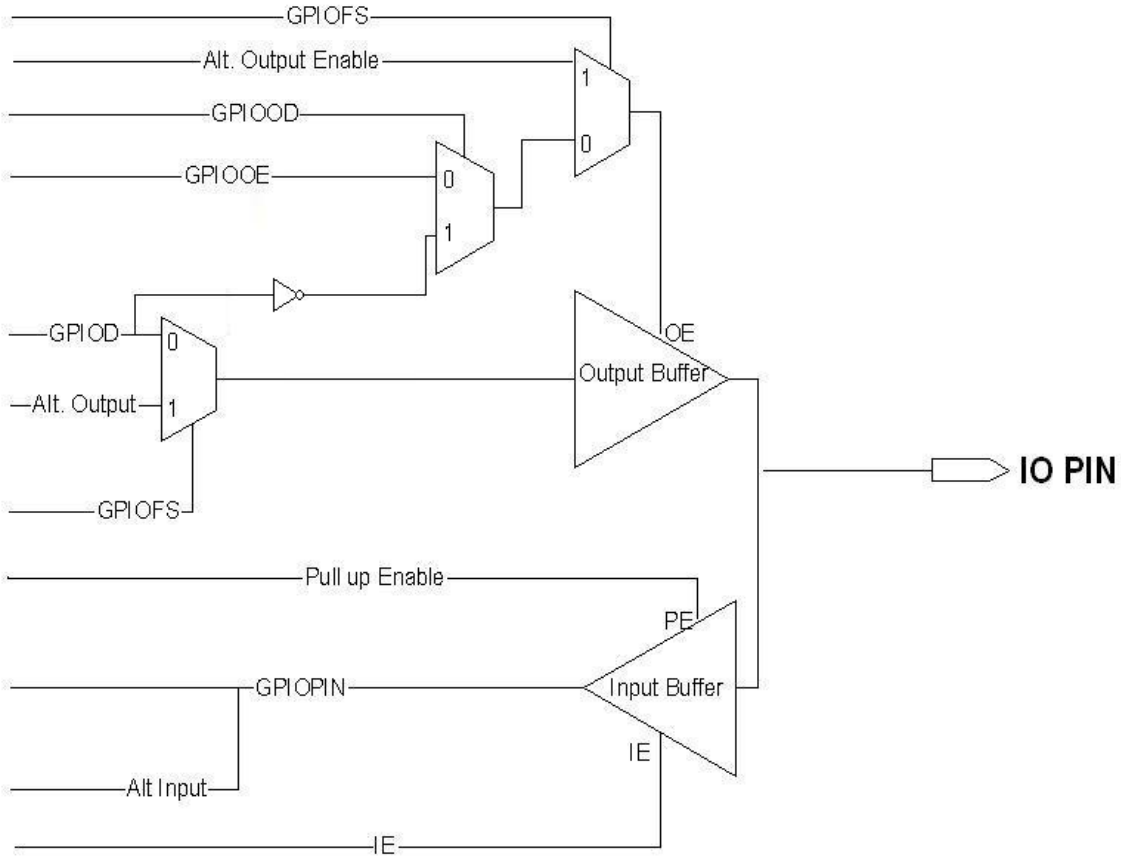
GPIO	IO3731 Alt.	Alt. Selection Reg.
GPIO25	KSO10	GPIOFS20.[5]
GPIO26	KSO11	GPIOFS20.[6]
GPIO27	KSO12	GPIOFS20.[7]
GPIO28	KSO13	GPIOFS28.[0]
GPIO29	KSO14	GPIOFS28.[1]
GPIO2A	KSO15	GPIOFS28.[2]
∗ ~~GPIO2B~~	~~SDA2 / PS2_DAT2~~	~~GPIOFS28.[3]~~
∗ ~~GPIO2C~~	~~SCL2 / PS2_CLK2~~	~~GPIOFS28.[4]~~
GPIO2D	SDA0	GPIOFS28.[5]]
GPIO2E	SCL0	GPIOFS28.[6]
GPIO2F	KSI2	GPIOFS28.[7]
GPIO30	KSI3	GPIOFS30.[0]
GPIO31	KSI0	GPIOFS30.[1]
GPIO32	KSI1	GPIOFS30.[2]
GPIO33	KSI4	GPIOFS30.[3]
GPIO34	KSI5	GPIOFS30.[4]
GPIO35	SDA1 / PS2_DAT3	GPIOFS30.[5]
GPIO36	SCL1 / PS2_CLK3	GPIOFS30.[6]
GPIO37	PS2_CLK1	GPIOFS30.[7]
GPIO38	PS2_DAT1	GPIOFS38.[0]

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

∗Denotes that these pins do not exist in IO3731

4.3.2 GPIO Structures

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



4.3.3 GPIO Attribution Table

GPIO	IO3731 Alt.	Default Alt.	Alt. Reg.	Input Enable	Output Enable	Pull Up (40KΩ)	Open Drain	Output Current
GPIO00	KSI6	GPIO00	GPIOFS00.[0]	v	v	v	v	2~4
GPIO01	KSI7	GPIO01	GPIOFS00.[1]	v	v	v	v	2~4
GPIO02	KSO4	GPIO02	GPIOFS00.[2]	v	v	v	v	2~4
GPIO03	KSO5	GPIO03	GPIOFS00.[3]	v	v	v	v	2~4
GPIO04	KSO6	GPIO04	GPIOFS00.[4]	v	v	v	v	2~4
GPIO05	KSO7	GPIO05	GPIOFS00.[5]	v	v	v	v	2~4
GPIO06	OWM	GPIO06	GPIOFS00.[6]	v	v	v	v	2~4
GPIO07	PWM1	GPIO07	GPIOFS00.[7]	v	v	v	v	2~4
GPI08	AD0	GPIO08	GPIOFS08.[0]	v				
GPI09	AD1	GPIO09	GPIOFS08.[1]	v				
GPI0A	AD2	GPIO0A	GPIOFS08.[2]	v				
GPI0B	AD3	GPIO0B	GPIOFS08.[3]	v				
GPI0C	AD4	GPIO0C	GPIOFS08.[4]	v				
GPI0D	AD5	GPIO0D	GPIOFS08.[5]	v				
GPO0E★	DA0	GPIO0E	GPIOFS08.[6]		v			2~4
GPO0F★	DA1	GPIO0F	GPIOFS08.[7]		v			2~4
GPIO10	SPI_CLK	GPIO10	GPIOFS10.[0]	v	v	v	v	8~16
GPIO11	SPI_DO	GPIO11	GPIOFS10.[1]	v	v	v	v	8~16
GPIO12	SPI_DI	GPIO12	GPIOFS10.[2]	v	v	v	v	2~4
GPIO13	SPI_CS#	GPIO13	GPIOFS10.[3]	v	v	v	v	8~16
GPIO14	PWM0	GPIO14	GPIOFS10.[4]	v	v	v	v	2~4
GPIO15	PWM2	GPIO15	GPIOFS10.[5]	v	v	v	v	2~4
GPIO16	PWM3	GPIO16	GPIOFS10.[6]	v	v	v	v	2~4
GPIO17	CEC	GPIO17	GPIOFS10.[7]	v	v	★ 26KΩ	v	2~4
GPIO18	CIRRX / KSO16	GPIO18	GPIOFS18.[0]	v	v	v	v	2~4
GPIO19	DS0	GPIO19	GPIOFS18.[1]	v	v	v	v	2~4
GPIO1A	DS1	GPIO1A	GPIOFS18.[2]	v	v	v	v	2~4
GPIO1B	CIRTX / KSO17	GPIO1B	GPIOFS18.[3]	v	v	v	v	2~4
GPIO1C	IRQ#	GPIO1C	GPIOFS18.[4]	v	v	v	v	2~4
GPIO1D	RX	GPIO1D	GPIOFS18.[5]	v	v	v	v	2~4
GPIO1E	TX	GPIO1E	GPIOFS18.[6]	v	v	v	v	2~4
GPIO1F	KSO0	GPIO1F	GPIOFS18.[7]	v	v	v	v	2~4
GPIO20	KSO1	GPIO20	GPIOFS20.[0]	v	v	v	v	2~4
GPIO21	KSO2	GPIO21	GPIOFS20.[1]	v	v	v	v	2~4
GPIO22	KSO3	GPIO22	GPIOFS20.[2]	v	v	v	v	2~4
GPIO23	KSO8	GPIO23	GPIOFS20.[3]	v	v	v	v	2~4
GPIO24	KSO9	GPIO24	GPIOFS20.[4]	v	v	v	v	2~4
GPIO25	KSO10	GPIO25	GPIOFS20.[5]	v	v	v	v	2~4
GPIO26	KSO11	GPIO26	GPIOFS20.[6]	v	v	v	v	2~4
GPIO27	KSO12	GPIO27	GPIOFS20.[7]	v	v	v	v	2~4
GPIO28	KSO13	GPIO28	GPIOFS28.[0]	v	v	v	v	2~4
GPIO29	KSO14	GPIO29	GPIOFS28.[1]	v	v	v	v	2~4
GPIO2A	KSO15	GPIO2A	GPIOFS28.[2]	v	v	v	v	2~4
~~＊GPIO2B~~	~~SDA2 / PS2_DAT2~~	~~GPIO2B~~	~~GPIOFS28.[3]~~	~~v~~	~~v~~	~~v~~	~~v~~	~~2~4~~

~~∗GPIO2C~~	~~SCL2 / PS2_CLK2~~	~~GPIO2C~~	~~GPIOFS28.[4]~~	~~v~~	~~v~~	~~v~~	~~v~~	~~2~4~~
GPIO2D	SDA0	GPIO2D	GPIOFS28.[5]	v	v	v	v	2~4
GPIO2E	SCL0	GPIO2E	GPIOFS28.[6]	v	v	v	v	2~4
GPIO2F	KSI2	GPIO2F	GPIOFS28.[7]	v	v	v	v	2~4
GPIO30	KSI3	GPIO30	GPIOFS30.[0]	v	v	v	v	2~4
GPIO31	KSI0	GPIO31	GPIOFS30.[1]	v	v	v	v	2~4
GPIO32	KSI1	GPIO32	GPIOFS30.[2]	v	v	v	v	2~4
GPIO33	KSI4	GPIO33	GPIOFS30.[3]	v	v	v	v	2~4
GPIO34	KSI5	GPIO34	GPIOFS30.[4]	v	v	v	v	2~4
GPIO35	SDA1 / PS2_DAT3	GPIO35	GPIOFS30.[5]	v	v	v	v	2~4
GPIO36	SCL1 / PS2_CLK3	GPIO36	GPIOFS30.[6]	v	v	v	v	2~4
GPIO37	PS2_CLK1	GPIO37	GPIOFS30.[7]	v	v	v	v	2~4
GPOI38	PS2_DAT1	GPOI38	GPIOFS38.[0]	v	v	v	v	2~4

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

∗Denotes that these pins do not exist in IO3731

4.3.4 GPIO Registers Descriptions (0xFC00~0xFC7F)

Function Selection Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x00	GPIOFS00	R/W	GPIO00~GPIO07 Function Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x01	GPIOFS08	R/W	GPIO08~GPIO0F Function Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** General purpose output function selected **1:** Alternative output function selected. Note: **GPI**08/09/0A/0B/0C/0D are AD pins without Alt. output functionality.	0x00	0xFC
0x02	GPIOFS10	R/W	GPIO10~GPIO17 Function Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** General purpose output function selected **1:** Alternative output function selected. Note: Default 0x0F means in IO373x, SPI is default enable,	0x0F	0xFC
0x03	GPIOFS18	R/W	GPIO18~GPIO1F Function Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x04	GPIOFS20	R/W	GPIO20~GPIO27 Function Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x05	GPIOFS28	R/W	GPIO28~GPIO2F Function Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** General purpose output function selected **1:** Alternative output function selected. Note: Default 0x60 means in IO373x, SMBus device is default enable, No GPIO2B/2C in IO3731	0x60	0xFC
0x06	GPIOFS30	R/W	GPIO30~GPIO37 Function Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x07	GPIOFS38	R/W	GPIO38 Function Selection bit[0] stand for GPIO38 **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC

Output Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x10	GPIOOE00	R/W	GPIO00~GPIO07 Output Enable bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x11	GPIOOE08	R/W	GPIO08~GPIO0F Output Enable bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Output Disable **1:** Output Enable Note: **GPI**08/09/0A/0B/0C/0D are AD pins without output enable functionality.	0x00	0xFC
0x12	GPIOOE10	R/W	GPIO10~GPIO17 Output Enable bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x13	GPIOOE18	R/W	GPIO18~GPIO1F Output Enable bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x14	GPIOOE20	R/W	GPIO20~GPIO27 Output Enable bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x15	GPIOOE28	R/W	GPIO28~GPIO2F Output Enable bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Output Disable **1:** Output Enable Note: No GPIO2B/2C in IO3731	0x00	0xFC
0x16	GPIOOE30	R/W	GPIO30~GPIO37 Output Enable bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x17	GPIOOE38	R/W	GPIO38 Output Enable bit[0] stand for GPIO38 **0:** Output Disable **1:** Output Enable	0x00	0xFC

Output Data Port Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x20	GPIOD00	R/W	GPIO00~GPIO07 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO00~GPIO07 separately	0x00	0xFC
0x21	GPIOD08	R/W	GPIO08~GPIO0F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO08~GPIO0F separately Note: **GPI**08/09/0A/0B/0C/0D are AD pins without output data functionality.	0x00	0xFC
0x22	GPIOD10	R/W	GPIO10~GPIO17 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO10~GPIO17 separately	0x00	0xFC
0x23	GPIOD18	R/W	GPIO18~GPIO1F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO18~GPIO1F separately	0x00	0xFC
0x24	GPIOD20	R/W	GPIO20~GPIO27 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO20~GPIO27 separately	0x00	0xFC
0x25	GPIOD28	R/W	GPIO28~GPIO2F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO28~GPIO2F separately Note: No GPIO2B/2C in IO3731	0x00	0xFC
0x26	GPIOD30	R/W	GPIO30~GPIO37 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO30~GPIO37 separately	0x00	0xFC
0x27	GPIOD38	R/W	GPIO38 Output Data Port for output function. Bit[0] stand for GPIO38	0x00	0xFC

Input Data Port Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x30	GPIOIN00	R	GPIO00~GPIO07 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO00~GPIO07 separately	0xFF	0xFC
0x31	GPIOIN08	R	GPIO08~GPIO0F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO08~GPIO0F separately Note: **GPO**0E/0F are DA pins without input data functionality.	0xFF	0xFC
0x32	GPIOIN10	R	GPIO10~GPIO17 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO10~GPIO17 separately	0xFF	0xFC
0x33	GPIOIN18	R	GPIO18~GPIO1F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO18~GPIO1F separately	0xFF	0xFC
0x34	GPIOIN20	R	GPIO20~GPIO27 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO20~GPIO27 separately	0xFF	0xFC
0x35	GPIOIN28	R	GPIO28~GPIO2F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO28~GPIO2F separately Note: No GPIO2B/2C in IO3731	0xFF	0xFC
0x36	GPIOIN30	R	GPIO30~GPIO37 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO30~GPIO37 separately	0xFF	0xFC
0x37	GPIOIN38	R	GPIO38 Input Data Port for input function. Bit[0] stand for GPIO38	0xFF	0xFC

Pull-up Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x40	GPIOPU00	R/W	GPIO00~GPIO07 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x1F	0xFC
0x41	Reserved	RSV	Note: **GPI**08/09/0A/0B/0C/0D are AD pins, **GPO**0E/0F are DA pins without pull-up resistor enable functionality.	0x00	0xFC
0x42	GPIOPU10	R/W	GPIO10~GPIO17 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x43	GPIOPU18	R/W	GPIO18~GPIO1F Internal Pull-Up Resistor Enable for input function bit[0]~ bit[7] stand for GPIO18~GPIO1F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x90	0xFC
0x44	GPIOPU20	R/W	GPIO20~GPIO27 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x07	0xFC
0x45	GPIOPU28	R/W	GPIO28~GPIO2F Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable Note: No GPIO2B/2C in IO3731	0x80	0xFC
0x46	GPIOPU30	R/W	GPIO30~GPIO37 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x1F	0xFC
0x47	GPIOPU38	R/W	GPIO38 Internal Pull-Up Resistor Enable for input function bit[0] stand for GPIO38 **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC

Open Drain Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x50	GPIOOD00	R/W0C	GPIO00~GPIO07 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x51	Reserved	RSV	Note: **GPI**08/09/0A/0B/0C/0D are AD pins, **GPO**0E/0F are DA pins without open drain enable functionality.	0x00	0xFC
0x52	GPIOOD10	R/W0C	GPIO10~GPIO17 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x53	GPIOOD18	R/W0C	GPIO18~GPIO1F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x54	GPIOOD20	R/W0C	GPIO20~GPIO27 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x55	GPIOOD28	R/W0C	GPIO28~GPIO2F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Open drain disable **1:** Open drain enable. Note: No GPIO2B/2C in IO3731	0x00	0xFC
0x56	GPIOOD30	R/W0C	GPIO30~GPIO37 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x57	GPIOOD38	R/W0C	GPIO38 Open Drain Enable for output function bit[0] stand for GPIO38 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC

Input Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x60	GPIOIE00	R/W	GPIO00~GPIO07 Input Enable for input function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x1F	0xFC
0x61	GPIOIE08	R/W	GPIO08~GPIOF Input Enable for input function bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** GPIO input mode disable **1:** GPIO input mode enable. Note: **GPO**0E/0F are DA pins without input enable functionality.	0x00	0xFC
0x62	GPIOIE10	R/W	GPIO10~GPIO17 Input Enable for input function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** GPIO input mode disable **1:** GPIO input mode enable. Note: GPIO10/11/12/13 are SPI interface	0x0F	0xFC
0x63	GPIOIE18	R/W	GPIO18~GPIO1F Input Enable for input function bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x80	0xFC
0x64	GPIOIE20	R/W	GPIO20~GPIO27 Input Enable for input function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x07	0xFC
0x65	GPIOIE28	R/W	GPIO28~GPIO2F Input Enable for input function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** GPIO input mode disable **1:** GPIO input mode enable. Note: GPIO2D/2E are SMBus interface Note: No GPIO2B/2C in IO3731	0xE0	0xFC
0x66	GPIOIE30	R/W	GPIO30~GPIO37 Input Enable for input function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x1F	0xFC
0x67	GPIOIE38	R/W	GPIO38 Input Enable for input function bit[0] stand for GPIO38 **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC

4.4 Consumer Electronics Control Interface (CEC)

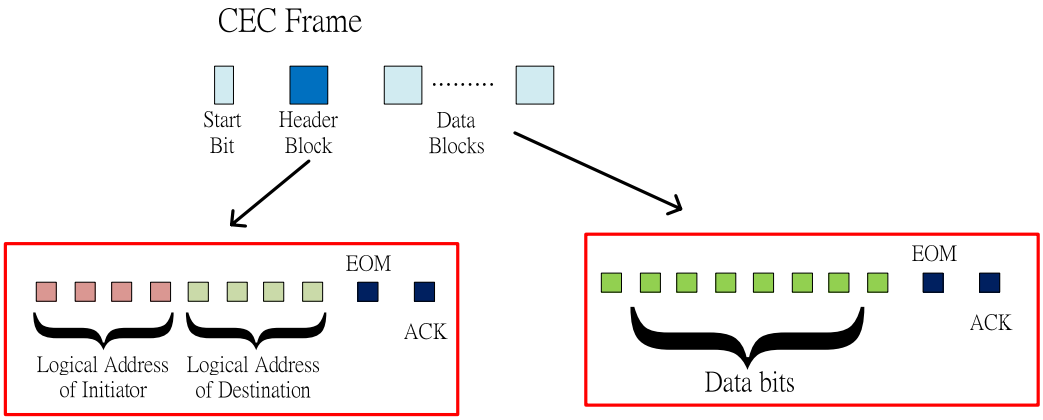
4.4.1 Brief Description

Trade names for CEC are *Anynet (Samsung); Aquos Link (Sharp); BRAVIA Sync (Sony); HDMI-CEC (Hitachi); Kuro Link (Pioneer); CE-Link and Regza Link (Toshiba); RIHD (Remote Interactive over HDMI) (Onkyo); SimpLink (LG); HDAVI Control, EZ-Sync, and VIERA Link (Panasonic); EasyLink (Philips); and NetCommand for HDMI (Mitsubishi).*

All above trademark are reserved by their own manufacturers.

CEC bus is a one-wire / bi-directional signal which connects up to ten (10) AV devices through standard HDMI cabling. The CEC protocol includes automatic mechanisms for *physical address (topology) discovery, (product type based) logical addressing, arbitration, retransmission, broadcasting, and routing control.* When idle, CEC devices pull-up the bus voltage to between 2.5V and 3.63V. Devices pull-down to between 0V and 0.6V would assert bits. Physical rise and fall times need to be less than 250ms and 50ms. The bus rate is approximately 500bits/second.

CEC is using frame based protocol which contains “a start bit”, “a header block”, and “specific data blocks”. The header block contains initiator address and destination. The data block contains transferred information bits.



4.4.2 CEC Registers Description (0xFC80~0xFC9F)

CEC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x80	CEC_CFG	7~4	RSV	Reserved	0x00	0xFC
		3	R/W	Multi-Address Enable **0**: Disable **1**: Enable		
		2	R/W	Line error handling for long bit time enable		
		1	R/W	Follower enable		
		0	R/W	CEC enable		

CEC Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x81	CEC_STS	7~4	RSV	Reserved	0x00	0xFC
		3	RO	RX header type **0**: Data **1**: Header		
		2	RO	End of Message **0**: Not EOM **1**: Enable		
		1	RO	Bus Busy Status **0**: Idle **1**: Busy		
		0	RO	Controller Role **0**: Initiator **1**: Follower		

CEC Error Code Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x82	CEC_ERRSTA	7	RSV	Reserved	0x00	0xFC
		6	RO	TX Arbitration Loss		
		5	RO	CEC line error detected when initiator is active		
		4	RO	CEC no ACK		
		3	RSV	Reserved		
		2	RO	CEC line error due to bit timing period too long		
		1	RO	CEC line error due to bit timing period too short		
		0	RO	CEC error		

CEC Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x83	CEC_INTEN	7~2	RSV	Reserved	0x00	0xFC
		1	R/W	RX completed interrupt enable		
		0	R/W	TX/Start completed interrupt enable		

CEC Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x84	CEC_EPF	7~2	RSV	Reserved	0x00	0xFC
		1	R/W1C	RX completed event pending flag		
		0	R/W1C	TX/Start completed event pending flag		

CEC Function Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x85	CEC_FUNC	7~3	RSV	Reserved	0x00	0xFC
		2	R/W	EOM		
		1~0	R/W	Start bit or data block **00**: Reserved **01**: Issue a data block **10**: Issue start bit **11**: Reserved		

CEC Logical Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x86	CEC_LA	7~4	R/W	Address 1	0xEE	0xFC
		3~0	R/W	Address 0		

CEC TX Data Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x87	CEC_TXDATA	7~0	R/W	TX data byte	0x00	0xFC

CEC RX Data Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x88	CEC_RXDATA	7~0	RO	RX data byte	0x00	0xFC

CEC Start Bit Duration Time						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x89	CEC_SBDT	7~4	R/W	Low Duration Time = 91 + N * 3T (T=32us) Default value 8 = 115T = 3680us	0x88	0xFC
		3~0	R/W	High Duration Time = 17 + N * T (T=32us) Default value 8 = 25T = 800us		

CEC Data Bit Total Duration Time						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8A	CEC_DBTDT	7~4	RSV	Reserved	0x08	0xFC
		3~0	R/W	Data Bit Total Duration Time = 51 + N * T (T=32us) Default value 8 = 75T = 2400us		

CEC Data Bit Low Duration Time						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8B	CEC_DBLDT	7~4	R/W	Logic 1 Low Duration Time = 11+ N * T (T=32us) Default value 8 = 19T = 608us	0x88	0xFC
		3~0	R/W	Logic 0 Low Duration Time = 39 + N * T (T=32us) Default value 8 = 47T = 1504us		

CEC Start Bit Check Low Duration Time						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8C	CEC_SBCLD	7~4	R/W	Minimum Time = 101 + N * T (T=32us) Default value 8 = 109T = 3488us	0x88	0xFC
		3~0	R/W	Maximum Time = 114 + N * T (T=32us) Default value 8 = 122T = 3904us		

CEC Start Bit Check Total Duration Time						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8D	CEC_SBCTD	7~4	R/W	Minimum Time = 126 + N * T (T=32us) Default value 8 = 134T = 4288us	0x88	0xFC
		3~0	R/W	Maximum Time = 139 + N * T (T=32us) Default value 8 = 147T = 4704us		

CEC Sample Time						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8E	CEC_SAMP_T	7~4	RSV	Reserved	0x08	0xFC
		3~0	R/W	Sample Time = 25 + N * T (T=32us) Default value 8 = 33T = 1056us		

CEC Earliest/Latest Time for the Following Bit						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8F	CEC_ELTFFB	7~4	R/W	Earliest Time = 40 + N * 3T (T=32us) Default value 8 = 64T = 2048us	0x88	0xFC
		3~0	R/W	Latest Time = 61 + N * 3T (T=32us) Default value 8 = 85T = 2720us		

CEC Error Indication Period						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x90	CEC_EIP	7~4	R/W	Initiator Error Indication Period (1.3 * bit_period) = 74 + N * 3T (T=32us) Default value 8 = 98T = 3136us	0x88	0xFC
		3~0	R/W	Follower Error Indication Period (1.5 * bit_period) = 89 + N * 3T (T=32us) Default value 8 = 113T = 3616us		

CEC Signal Free Time Plus						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x91	CEC_SFTP	7~6	RSV	Reserved	0x00	0xFC
		5~4	R/W	The idle time need to wait for the “Present Initiator wants to send another frame immediately after previous frame” (Time base is a total data bit time) = 7 + N * T		
		3~2	R/W	The idle time need to wait for the “New Initiator wants to send a frame” (Time base is a total data bit time) = 5 + N * T		
		1~0	R/W	The idle time need to wait for the “Previous attempt to send frame unsuccessfully” (Time base is a total data bit time) = 3 + N * T Default value 0 = 7T		

CEC Software Reset						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	CEC_SWRST	7~1	RSV	Reserved	0x00	0xFC
		0	WO	Software Reset bit When write as 1, will generate a pulse signal to reset hardware state machines, CEC signal, and internal control signals.		

CEC Option						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	CEC_OPT	7~2	RSV	Reserved	0x00	0xFC
		1~0	R/W	IO Buffer Delay Offset The offset to cover the timing delay of output buffer and input buffer **00**: 0T **01**: 1T **10**: 2T **11**: 3T		

4.5 Reserved

4.6 OWM

4.6.1 OWM Functional Description

OWM is called One Wire Bus Master Interface (GPIO0A) which could be used as simple host interface, OWM device ID identification, and device power. OWM interface is featured as 1) Bi-directional; 2) single-master/multi-slave; 3) half-duplex. OWM is physically implemented with single open-drain master connected to one or more open-drain slave devices. Pull-up resistor is commonly used to pull the bus to 3 or 5 V.

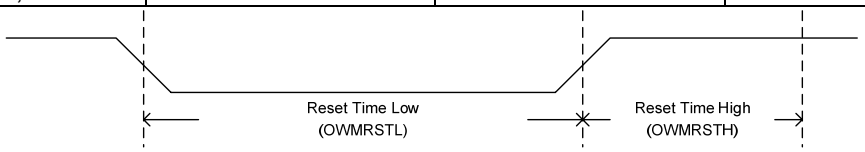
The OWM supports:

1. Dallas One Wire Bus Master and TI HDQ protocol.
2. Interrupt enable for Reset/Break, Read and Write command.
3. Separate 8-bit read and write buffers.
4. Configurable timing registers can be setting by F/W.

4.6.2 OWM Timing Setting Illustration

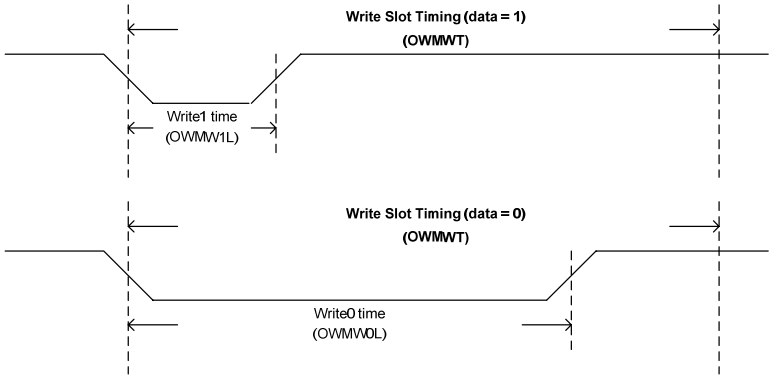
Reset / Break Timing

Register Name	Time Base	Default Value	Default Timing
OWMRSTL, 0xFCF5	8 us	0x40	512 us
OWMRSTH, 0xFCF6	8 us	0x40	512 us



Write Timing

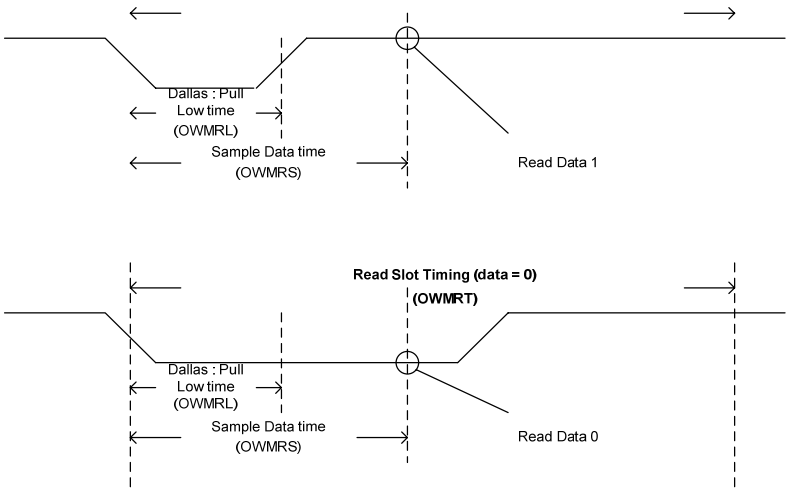
Register Name	Time Base	Default Value	Default Timing
OWMWT, 0xFCF7	2 us	0x2D	90 us
OWMW1L, 0xFCF8	1 us	0x0A	10 us
OWMW0L, 0xFCF9	1 us	0x50	80 us



Read Timing

Register Name	Time Base	Default Value	Default Timing
OWMRT, 0xFCFA	2 us	0x2D	90 us
OWMRL, 0xFCFB	1 us	0x03	3 us
OWMRS, 0xFCFC	1 us	0x14	20 us

Note : OWMRL is for Dallas only



4.6.3 OWM Register Description (0xFCF0~0xFCFF)

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

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

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

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

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

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

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

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

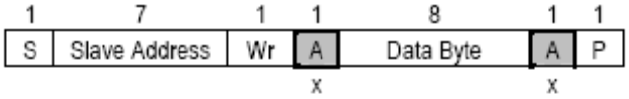
4.7 SMBus Device Controller

SMBus Device Controller is used for host to control the IO373x chip. There is only a SMBus device controller and it's compatible to SMBus 2.0 SPEC including PEC (CRC Packet Error Check). SMBus Device Controller could operate as HW mode or FW mode.

4.7.1 SMBus Device Controller HW mode

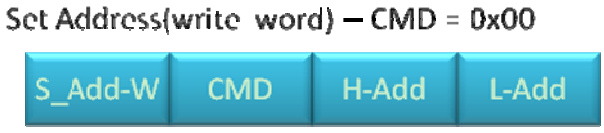
IO373x support several SMBUS protocols including Write Word, Write Byte, Write Block, For SMBus slave addresses, please refer the trapping section and SMBus register description section.

SMBus Bus Protocol :

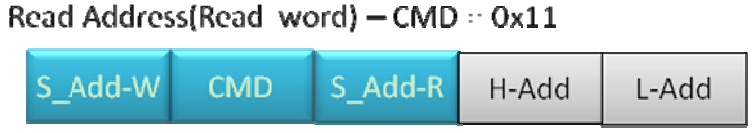


S	Start Condition
Sr	Repeated Start Condition
Rd	Read (bit value of 1)
Wr	Write (bit value of 0)
x	Shown under a field indicates that that field is required to have the value of 'x'
A	Acknowledge (this bit position may be '0' for an ACK or '1' for a NACK)
P	Stop Condition
PEC	Packet Error Code
	Master-to-Slave
	Slave-to-Master
...	Continuation of protocol

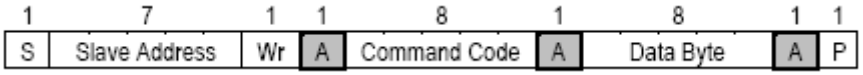
To S**et Address**, use **Write Word** protocol with CMD=**0x00** as following illustration:



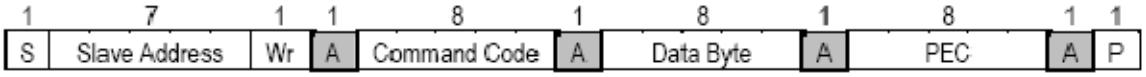
To **Read Address**, use **Read Word** protocol with CMD=**0x11** as following illustration:



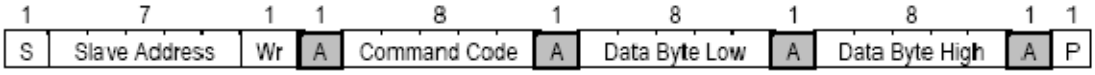
Write Byte : CMD = 0x01



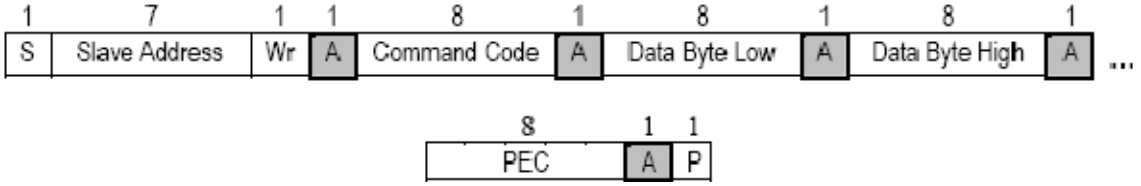
Write Byte with PEC : CMD = 0x01



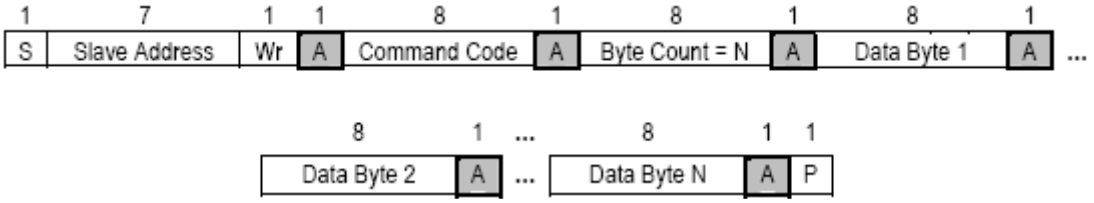
Write Word : CMD = 0x02



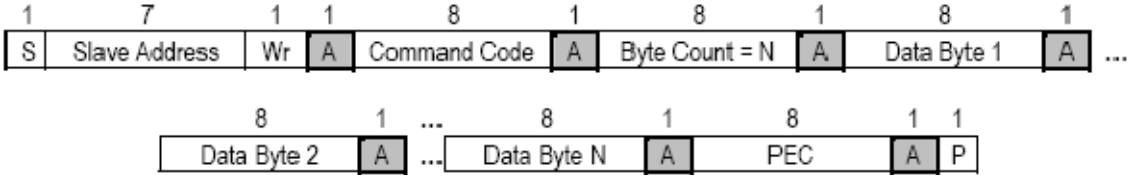
Write Word with PEC : CMD = 0x02



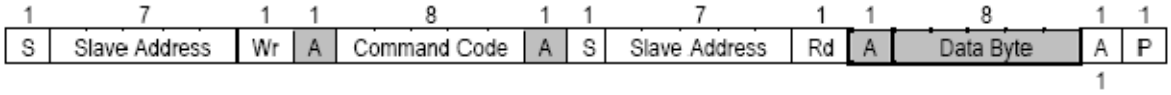
Write Block : CMD = 0x03



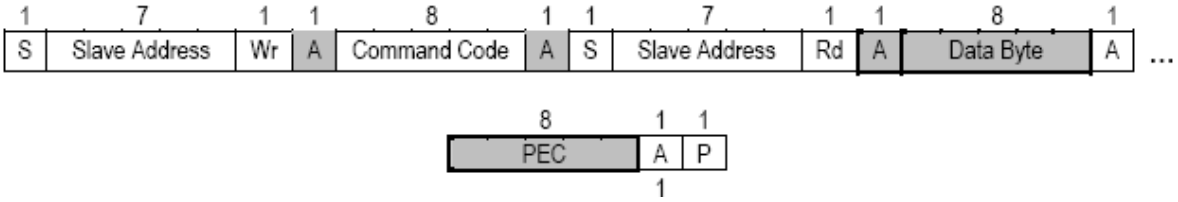
Write Block with PEC : CMD = 0x03



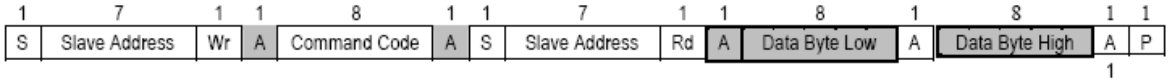
Read Byte : CMD = 0x81



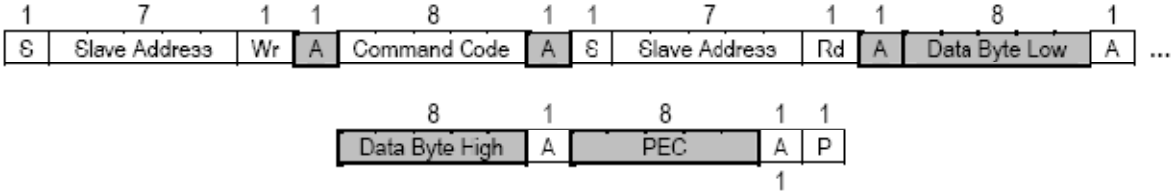
Read Byte with PEC: CMD = 0x81



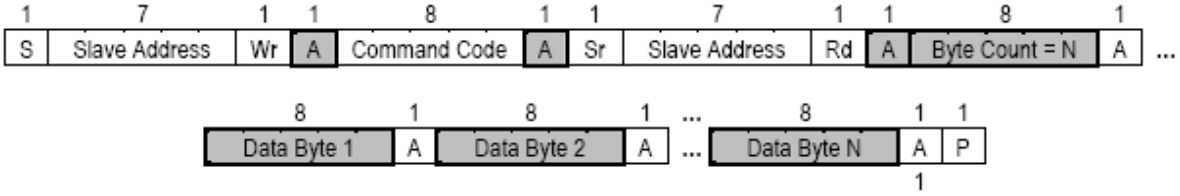
Read Word : CMD = 0x82



Read Word with PEC : CMD = 0x82



Read Block : CMD = 0x80 + Byte Count



Read Block with PEC : CMD = 0x80 + Byte Count

4.7.2 SMBus Device Controller FW mode

In SMBus Device Controller FW mode, the host can only use the 16 bytes data registers for read/write. It supports all protocols in SMBus 2.0 and I2C 2.0 SPEC.

4.7.3 SMBus Device Controller Register Description (0xFD00~0xFD2F)

SMBus Device Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x10	SMBDCFG	7-3	RSV	Reserved	0x03	0xFD
		2	R/W	PEC Enable **0**: Disable **1**: Enable		
		1	R/W	HW mode enable **0**: Disable **1**: Enable		
		0	R/W	SMBus device Function enable **0**: Disable **1**: Enable		

SMBus Device Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x11	SMBDIE	7-2	RSV	Reserved	0x00	0xFD
		1	R/W	Interrupt Enable of Command **0**: Disable **1**: Enable		
		0	R/W	Interrupt Enable of Command Completed **0**: Disable **1**: Enable		

SMBus Device Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x12	SMBDPF	7-2	RSV	Reserved	0x00	0xFD
		1	R/W1C	Pending Flag of Command **0**: Disable **1**: Enable		
		0	R/W1C	Pending Flag of Command Completed **0**: Disable **1**: Enable		

SMBus Device Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x13	SMBDSTS	7-6	RSV	Reserved	0x00	0xFD
		5	RO	Read / Write Status **0**: Write **1**: Read		
		4	RO	Bus Busy **0**: Idle **1**: Busy		
		3~0	RO	**Error Code** **00h:** No Error **01h:** Host no ACK **02h:** SMBus Timeout, CLK/SDA low timeout 25~35ms **03h:** PEC error **04h:** Protocol Error, 1) Count Byte Error, 2) Read/Write State Error, ADR_R address error.		

SMBus Device Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x14	SMBDADR	7-1	R/W	Programmable 7 bits address	0xC0	0xFD
		0	RO	Always 0		

SMBus Device Received Command						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x15	SMBDCMD	7-0	RO	SMBus device received command from host	0x00	0xFD

SMBus Device Byte Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x16	SMBDCNT	7	R/W	Programmable read/write indication **0**: Write **1**: Read	0x00	0xFD
		6	R/W	Byte count slot enable for read/write state **0**: No count value, the first byte starts with data byte **1**: The first byte is the count value.		
		5	RSV	Reserved		
		4-0	R/W	Rx/Tx byte count number for continuously operations		

SMBus Device Received Byte Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x17	SMBDRCNT	7-5	RSV	Reserved	0x00	0xFD
		4-0	RO	Rx/Tx byte count number for continuously operations Only valid when 0xFD16[6] is set.		

SMBus Device PEC						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x18	SMBDPEC	7-0	RO	The PEC value form host Only valid when PEC is enabled.	0x00	0xFD

SMBus Data Array (16 Bytes)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x19	SMBDATA0	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x1A	SMBDATA1	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x1B	SMBDATA2	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x1C	SMBDATA3	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x1D	SMBDATA4	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x1E	SMBDATA5	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x1F	SMBDATA6	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x20	SMBDATA7	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x21	SMBDATA8	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x22	SMBDATA9	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x23	SMBDATA10	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x24	SMBDATA11	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x25	SMBDATA12	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x26	SMBDATA13	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x27	SMBDATA14	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD
0x28	SMBDATA15	7-0	R/W	Data Register for SMB Device Operation	0x00	0xFD

4.8 SMBus Slave Controller

SMBus slave Controller is used to interface with other devices and IO373x serve as slave mode. There are two slave controllers in IO373x.

4.8.1 SMBus Slave0 Controller Register Description (0xFD40~0xFD5F)

SMBus Slave0 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x40	SMS0CFG	7-3	RSV	Reserved	0x03	0xFD
		2	R/W	PEC Enable **0**: Disable **1**: Enable		
		1	R/W	HW mode enable **0**: Disable **1**: Enable		
		0	R/W	SMBus slave0 function enable **0**: Disable **1**: Enable		

SMBus Slave0 Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x41	SMS0IE	7-2	RSV	Reserved	0x00	0xFD
		1	R/W	Interrupt Enable of Command **0**: Disable **1**: Enable		
		0	R/W	Interrupt Enable of Command Completed **0**: Disable **1**: Enable		

SMBus Slave0 Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x42	SMS0PF	7-2	RSV	Reserved	0x00	0xFD
		1	R/W1C	Pending Flag of Command **0**: Disable **1**: Enable		
		0	R/W1C	Pending Flag of Command Completed **0**: Disable **1**: Enable		

SMBus Slave0 Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x43	SMS0STS	7-6	RSV	Reserved	0x00	0xFD
		5	RO	Read / Write Status **0**: Write **1**: Read		
		4	RO	Bus Busy **0**: Idle **1**: Busy		
		3~0	RO	**Error Code** **00h:** No Error **01h:** Host no ACK **02h:** SMBus Timeout, CLK/SDA low timeout 25~35ms **03h:** PEC error **04h:** Protocol Error, 1) Count Byte Error, 2) Read/Write State Error, ADR_R address error.		

SMBus Slave0 Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x44	SMS0ADR	7-1	R/W	Programmable 7 bits address	0x00	0xFD
		0	RO	Always 0		

SMBus Slave0 Received Command						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x45	SMS0CMD	7-0	RO	SMBus device received command from host	0x00	0xFD

SMBus Slave0 Byte Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x46	SMS0CNT	7	R/W	Programmable read/write indication **0**: Write **1**: Read	0x00	0xFD
		6	R/W	Byte count slot enable for read/write state **0**: No count value, the first byte starts with data byte **1**: The first byte is the count value.		
		5	RSV	Reserved		
		4-0	R/W	Rx/Tx byte count number for continuously operations		

SMBus Slave0 Received Byte Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x47	SMS0RCNT	7-5	RSV	Reserved	0x00	0xFD
		4-0	RO	Rx/Tx byte count number for continuously operations Only valid when 0xFD46[6] is set.		

SMBus Slave0 PEC						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x48	SMS0PEC	7-0	RO	The PEC value form host Only valid when PEC is enabled.	0x00	0xFD

SMBus Slave0 Data Array (16 Bytes)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x49	SMS0DAT0	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x4A	SMS0DAT1	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x4B	SMS0DAT2	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x4C	SMS0DAT3	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x4D	SMS0DAT4	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x4E	SMS0DAT5	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x4F	SMS0DAT6	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x50	SMS0DAT7	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x51	SMS0DAT8	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x52	SMS0DAT9	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x53	SMS0DAT10	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x54	SMS0DAT11	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x55	SMS0DAT12	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x56	SMS0DAT13	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x57	SMS0DAT14	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x58	SMS0DAT15	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD

4.8.2 SMBus Slave1 Controller Register Description (0xFD60~0xFD7F)

SMBus Slave1 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x60	SMS1CFG	7-3	RSV	Reserved	0x03	0xFD
		2	R/W	PEC Enable **0**: Disable **1**: Enable		
		1	R/W	HW mode enable **0**: Disable **1**: Enable		
		0	R/W	SMBus slave1 function enable **0**: Disable **1**: Enable		

SMBus Slave1 Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x61	SMS1IE	7-2	RSV	Reserved	0x00	0xFD
		1	R/W	Interrupt Enable of Command **0**: Disable **1**: Enable		
		0	R/W	Interrupt Enable of Command Completed **0**: Disable **1**: Enable		

SMBus Slave1 Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x62	SMS1PF	7-2	RSV	Reserved	0x00	0xFD
		1	R/W1C	Pending Flag of Command **0**: Disable **1**: Enable		
		0	R/W1C	Pending Flag of Command Completed **0**: Disable **1**: Enable		

SMBus Slave1 Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x63	SMS1STS	7-6	RSV	Reserved	0x00	0xFD
		5	RO	Read / Write Status **0**: Write **1**: Read		
		4	RO	Bus Busy **0**: Idle **1**: Busy		
		3~0	RO	**Error Code** **00h:** No Error **01h:** Host no ACK **02h:** SMBus Timeout, CLK/SDA low timeout 25~35ms **03h:** PEC error **04h:** Protocol Error, 1) Count Byte Error, 2) Read/Write State Error, ADR_R address error.		

SMBus Slave1 Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x64	SMS1ADR	7-1	R/W	Programmable 7 bits address	0x00	0xFD
		0	RO	Always 0		

SMBus Slave1 Received Command						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x65	SMS1CMD	7-0	RO	SMBus device received command from host	0x00	0xFD

SMBus Slave1 Byte Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x66	SMS1CNT	7	R/W	Programmable read/write indication **0**: Write **1**: Read	0x00	0xFD
		6	R/W	Byte count slot enable for read/write state **0**: No count value, the first byte starts with data byte **1**: The first byte is the count value.		
		5	RSV	Reserved		
		4-0	R/W	Rx/Tx byte count number for continuously operations		

SMBus Slave1 Received Byte Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x67	SMS1RCNT	7-5	RSV	Reserved	0x00	0xFD
		4-0	RO	Rx/Tx byte count number for continuously operations Only valid when 0xFD66[6] is set.		

SMBus Slave1 PEC						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x68	SMS1PEC	7-0	RO	The PEC value form host Only valid when PEC is enabled.	0x00	0xFD

SMBus Slave0 Data Array (16 Bytes)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x69	SMS1DAT0	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x6A	SMS1DAT1	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x6B	SMS1DAT2	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x6C	SMS1DAT3	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x6D	SMS1DAT4	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x6E	SMS1DAT5	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x6F	SMS1DAT6	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x70	SMS1DAT7	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x71	SMS1DAT8	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x72	SMS1DAT9	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x73	SMS1DAT10	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x74	SMS1DAT11	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x75	SMS1DAT12	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x76	SMS1DAT13	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x77	SMS1DAT14	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD
0x78	SMS1DAT15	7-0	R/W	Data Register for SMB Slave0 Operation	0x00	0xFD

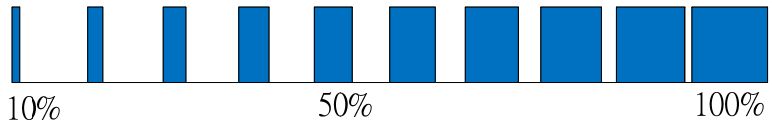
4.9 Pulse Width Modulation (PWM)

4.9.1 PWM Function Description

The PWM supports 4 PWM channels, and can be configured as Push-Pull or Open-Drain:

1. one 32-bits PWM @ PWM0 (4mA)
2. three 12-bits PWM @ PWM1/2/3(4mA)

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



PWM Duty-Cycle range from 0%~100%
(10% per scale in the illustrration)

4.9.2 PWM Duty Cycle Setting Illustration

The following table summarizes the relationship about the applications with the definition in the PWM registers description.

For PWM clock selection, the formula is

F= 16MHz / (N+1) where N is 0~15 independently for 4 channels.

As default N=0, F = 16Mhz, T=62.5 ns.

PWM0 (32 bits):

Definition	Formula
Duty Cycle	(PWM High Period Length+1)/ [(PWM Low Period Length+1) + (PWM High Period Length+1]] * 100%
Cycle Length	[(PWM Low Period Length+1) + (PWM High Period Length+1]] * (PWM clock source)

Term	Register Field
PWM CLK selection as 16MHz / (N+1)	PWMCLK1[3:0] , 0xFE01
PWM enable	PWMCFG[0] , 0xFE00[0]
PWM Push-Pull / Open-drian	PWMCFG[4] , 0xFE00[0] 0:Push-Pull , 1:Open-Drain
PWM Low Period Length High Byte	PWM0LOWH , 0xFE03
PWM Low Period Length Low Byte	PWM0LOWL , 0xFE04 Set high byte before low byte
PWM High Period Length High Byte	PWM0HIGHH , 0xFE05
PWM High Period Length Low Byte	PWM0HIGHL , 0xFE06 Set high byte before low byte

PWM1/2/3 (12 bits):

Definition	Formula
Duty Cycle	(PWM High Period Length+1)/(PWM Cycle Period Length+1) *100%
Cycle Length	(PWMCYC + 1) * (PWM clock source)

PWM Channel	Term	Register Field
PWM1	PWM CLK selection as 16MHz / (N+1)	PWMCLK1[7:4] , 0xFE01[7:4]
	PWM enable	PWMCFG[1] , 0xFE00[1]
	PWM Push-Pull / Open-drian	PWMCFG[5] , 0xFE00[5] 0:Push-Pull , 1:Open-Drain
	PWM High Period Length High Byte	PWM1HIGHH[3:0] , 0xFE07[3:0]
	PWM High Period Length Low Byte	PWM1HIGHL , 0xFE08 Set high byte before low byte
	PWM Cycle Period Length High Byte	PWM1CYCH[3:0] , 0xFE09[3:0]
	PWM Cycle Period Length Low Byte	PWM1CYCL , 0xFE0A Set high byte before low byte

PWM Channel	Term	Register Field
PWM2	PWM CLK selection as 16MHz / (N+1)	PWMCLK2[3:0] , 0xFE02[3:0]
	PWM enable	PWMCFG[2] , 0xFE00[2]
	PWM Push-Pull / Open-drian	PWMCFG[6] , 0xFE00[6] 0:Push-Pull , 1:Open-Drain
	PWM High Period Length High Byte	PWM2HIGHH[3:0] , 0xFE0B[3:0]
	PWM High Period Length Low Byte	PWM2HIGHL , 0xFE0C Set high byte before low byte
	PWM Cycle Period Length High Byte	PWM2CYCH[3:0] , 0xFE0D[3:0]
	PWM Cycle Period Length Low Byte	PWM2CYCL , 0xFE0E Set high byte before low byte

PWM Channel	Term	Register Field
PWM3	PWM CLK selection as 16MHz / (N+1)	PWMCLK2[7:4] , 0xFE02[7:4]
	PWM enable	PWMCFG[3] , 0xFE00[3]
	PWM Push-Pull / Open-drian	PWMCFG[7] , 0xFE00[7] 0:Push-Pull , 1:Open-Drain
	PWM High Period Length High Byte	PWM3HIGHH[3:0] , 0xFE0F[3:0]
	PWM High Period Length Low Byte	PWM3HIGHL , 0xFE10 Set high byte before low byte
	PWM Cycle Period Length High Byte	PWM3CYCH[3:0] , 0xFE11[3:0]
	PWM Cycle Period Length Low Byte	PWM3CYCL , 0xFE12 Set high byte before low byte

4.9.3 PWM Registers Description (0xFE00~0xFE1F)

PWM Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x00	PWMCFG	7	R/W	PWM3 Drive Type Selection **0**: Push-Pull **1**: Open-Drain (LED)	0x00	0xFE
		6	R/W	PWM2 Drive Type Selection **0**: Push-Pull **1**: Open-Drain (LED)		
		5	R/W	PWM1 Drive Type Selection **0**: Push-Pull **1**: Open-Drain (LED)		
		4	R/W	PWM0 Drive Type Selection **0**: Push-Pull **1**: Open-Drain (LED)		
		3	R/W	PWM3 Enable **0**: Disable **1**: Enable		
		2	R/W	PWM2 Enable **0**: Disable **1**: Enable		
		1	R/W	PWM1 Enable **0**: Disable **1**: Enable		
		0	R/W	PWM0 Enable **0**: Disable **1**: Enable		

PWM0, PWM1 CLOCK Source Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x01	PWMCLK1	7~4	R/W	PWM1 Clock Source Selection F = 16MHz / (N+1) where N is 4 bits as decimal 0~15	0x00	0xFE
		3-0	R/W	PWM0 Clock Source Selection F = 16MHz / (N+1) where N is 4 bits as decimal 0~15		

PWM2, PWM3 CLOCK Source Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x02	PWMCLK2	7~4	R/W	PWM3 Clock Source Selection F = 16MHz / (N+1) where N is 4 bits as decimal 0~15	0x00	0xFE
		3-0	R/W	PWM2 Clock Source Selection F = 16MHz / (N+1) where N is 4 bits as decimal 0~15		

PWM0 High Byte of Low Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x03	PWM0LOWH	7-0	R/W	PWM0 high byte of low period length Overall low period is 16 bits length	0x00	0xFE
PWM0 Low Byte of Low Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x04	PWM0LOWL	7-0	R/W	PWM0 low byte of low period length Overall low period is 16 bits length Set high byte before low byte	0x00	0xFE

PWM0 High Byte of High Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x05	PWM0HIGHH	7-0	R/W	PWM0 high byte of high period length Overall High period is 16 bits length	0x00	0xFE
PWM0 Low Byte of High Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x06	PWM0HIGHL	7-0	R/W	PWM0 low byte of high period length Overall High period is 16 bits length Set high byte before low byte	0x00	0xFE

PWM1 High Period Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x07	PWM1HIGHH	4-0	R/W	Higher 4 bits (of 12-bit)	0x00	0xFE
0x08	PWM1HIGHL	7-0	R/W	Lower 8 bits (of 12-bit) Set high byte before low byte	0x00	0xFE

PWM1 Cycle Period Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x09	PWM1CYCH	4-0	R/W	Higher 4 bits (of 12-bit)	0x00	0xFE
0x0A	PWM1CYCL	7-0	R/W	Lower 8 bits (of 12-bit) Set high byte before low byte	0x00	0xFE

PWM2 High Period Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0B	PWM2HIGHH	4-0	R/W	Higher 4 bits (of 12-bit)	0x00	0xFE
0x0C	PWM2HIGHL	7-0	R/W	Lower 8 bits (of 12-bit) Set high byte before low byte	0x00	0xFE

PWM2 Cycle Period Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0D	PWM2CYCH	4-0	R/W	Higher 4 bits (of 12-bit)	0x00	0xFE
0x0E	PWM2CYCL	7-0	R/W	Lower 8 bits (of 12-bit) Set high byte before low byte	0x00	0xFE

PWM3 High Period Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0F	PWM3HIGHH	4-0	R/W	Higher 4 bits (of 12-bit)	0x00	0xFE
0x10	PWM3HIGHL	7-0	R/W	Lower 8 bits (of 12-bit) Set high byte before low byte	0x00	0xFE

PWM3 Cycle Period Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x11	PWM3CYCH	4-0	R/W	Higher 4 bits (of 12-bit)	0x00	0xFE
0x12	PWM3CYCL	7-0	R/W	Lower 8 bits (of 12-bit) Set high byte before low byte	0x00	0xFE

4.10 Digital Sampler

4.10.1 Digital Sampler Description

The IO373x provides 2 digital samplers to count the high pulse width period. The counting period could be used to monitor PWM-type output for operation speed or status.

4.10.2 Fan Registers Description (0xFE20~0xFE3F)

Digital Sampler0 Controller Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x20	DSCC0	7	R/W	Counting behavior of digital sample 0 Latch DS0 input, start sampling, update DSSMON register when : **0**: Rising edge of DS0 input **1**: Rising & Falling of DS0 input	0x00	0xFE
		6-5	R/W	Digital Sampler 0 clock resolution selection **00**: 62.5us (default) **01**: 31.25us **10**: 15.625us **11**: 7.8125us		
		4	R/W1C	Flag of Digital Sampler 0 speed monitor timeout error **0**: no timeout error **1**: timeout error event		
		3	R/W1C	Flag of Digital Sampler 0 speed monitor update event. **0**: no update event. **1**: update event		
		2	R/W	Interrupt enable of Digital Sampler 0 **0**: Disable **1**: Enable		
		1	R/W	Digital Sampler 0 digital noise filter enable. **0**: Disable **1**: Enable		
		0	R/W	Digital Sampler 0 enable **0**: Disable **1**: Enable		

Digital Sampler 0 Speed Monitor Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x21	DSSMONH0	3-0	RO	High 4 bits of Digital Sample 0 speed monitor counter value	0x00	0xFE
0x22	DSSMONL0	7-0	RO	Low 8 bits of Digital Sampler 0 speed monitor counter value	0x00	0xFE

Digital Sampler1 Controller Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x30	DSCC1	7	R/W	Counting behavior of digital sample 1 Latch DS1 input, start sampling, update DSSMON register when : **0**: Rising edge of DS1 input **1**: Rising & Falling of DS1 input	0x00	0xFE
		6-5	R/W	Digital Sampler 1 clock resolution selection **00**: 62.5us (default) **01**: 31.25us **10**: 15.625us **11**: 7.8125us		
		4	R/W1C	Flag of Digital Sampler 1 speed monitor timeout error **0**: no timeout error **1**: timeout error event		
		3	R/W1C	Flag of Digital Sampler 1 speed monitor update event. **0**: no update event. **1**: update event		
		2	R/W	Interrupt enable of Digital Sampler 1 **0**: Disable **1**: Enable		
		1	R/W	Digital Sampler 1 digital noise filter enable. **0**: Disable **1**: Enable		
		0	R/W	Digital Sampler 1 enable **0**: Disable **1**: Enable		

Digital Sampler 1 Speed Monitor Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x31	DSSMONH1	3-0	RO	High 4 bits of Digital Sampler 1 speed monitor counter value	0x00	0xFE
0x32	DSSMONL1	7-0	RO	Low 8 bits of Digital Sampler 1 speed monitor counter value	0x00	0xFE

4.11 SPI Device Interface Controller

4.11.1 SPID Function Description

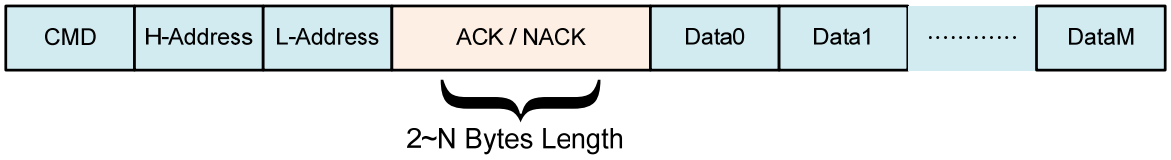
Similar to SMBus Device, SPI device Interface can be used by SPI host to access IO373x internal register file. The SPI protocol should follow specific formats.

Command Byte :

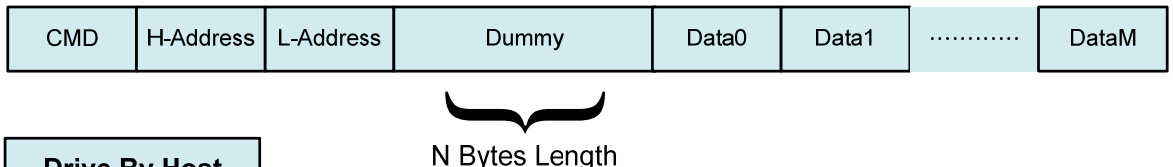
MSB **LSB**

7	6	5	4	3	2	1	0
Dummy Byte numbers, If N=0 the ACK / NACK is used. If N>0, dummy bytes are used.				Reserved Keep Always 0		**0:** Single Byte access **1:** Continuous Access	**0:** Read **1:** Write

Write ACK/NACK Format, when CMD[7:4]=0



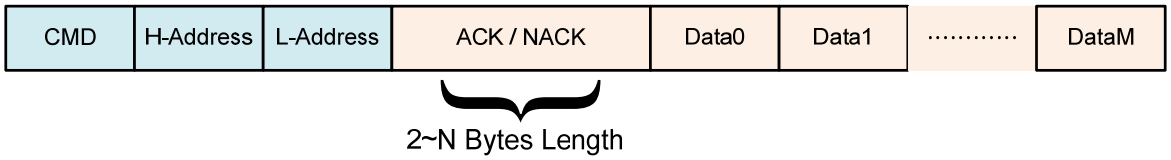
Write Dummy Byte Format, when CMD[7:4]>0



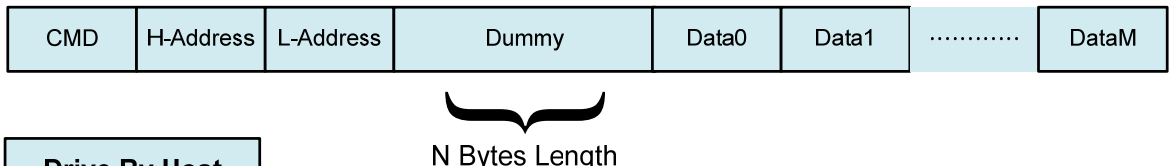
Drive By Host

Drive By Device

Read ACK/NACK Format, when CMD[7:4]=0



Read Dummy Byte Format, when CMD[7:4]>0



Drive By Host

Drive By Device

4.11.2 SPID Register Description (0xFE40~0xFE4F)

SPI Device Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x40	SPID_CFG	7-1	RSV	Reserved	0x01	0xFE
		0	R/W	SPI device interface enable **0**: Disable **1**: Enable		

SPI Device RX FIFO Point						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x41	SPID_RXFIFO	7	RO	RX FIFO full flag	0x40	0xFE
		6	RO	RX FIFO empty flag		
		5-3	RO	RX FIFO write pointer (depth = 8 bytes)		
		2-0	RO	RX FIFO read pointer (depth = 8 bytes)		

SPI Device TX FIFO Point						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x42	SPID_TXFIFO	7	RO	TX FIFO full flag	0x40	0xFE
		6	RO	TX FIFO empty flag		
		5-3	RO	TX FIFO write pointer (depth = 8 bytes)		
		2-0	RO	TX FIFO read pointer (depth = 8 bytes)		

SPI Device Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x43	SPID_STA	7	R/W1C	Pending Flag of Command Fail (For previous command busy, and the host issue Read/Write command)	0x00	0xFE
		6	R/W1C	Pending Flag of Bus Busy (For previous command busy in accessing the system bus)		
		5	R/W1C	Pending Flag of RX FIFO overflow (For IO373x RX FIFO notification when host issues write command fail)		
		4	R/W1C	Pending Flag of TX FIFO overflow (For IO373x TX FIFO notification when host issues read command fail)		
		3-0	RSV	Reserved		

4.12 General Purpose Timer (GPT)

4.12.1 GPT Function Description

The IO373x provides 4 GPTs (General Purpose Timers), two 16-bit timers and two 8-bit timers. These 4 GPTs operate based on 32.768 khz and all timers have the interrupt capability. The GPT is simply a free run counter. While the timer meets the specific value in counter register, for instance, 0xFE53 and 0xFE55, an interrupt issues (if interrupt enabled) and the counter reset to be zero.

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

Since 32.768 khz = 30 us period. For the designed target timer period T us, the required value need to be filled in counter register = ***(T in us) / 30***.

Eg: A 200Hz timer is with timer period of 5ms. The required value is 5000 / 30 = 166 = 0xA6

4.12.2 GPT Registers Description (0xFE50~0xFE6F)

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

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

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

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

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

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

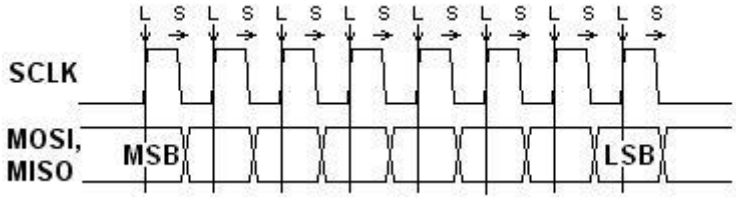
GPT3 Counter Value (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x58	GPT3H	7-0	R/W	High byte of GPT3 counter value. Once GPT3 counter meets this 16-bit value, an interrupt issues. GPT3 restart to count from zero.	0x00	0xFE
0x59	GPT3L	7-0	R/W	Low byte of GPT3 counter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

4.13 SPI Host Interface Controller

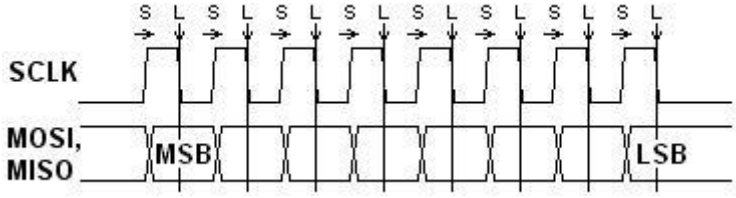
4.13.1 SPI Host Interface Description

The SPI host mode could support the SPI mode 0/1/2/3, and is configurable by SHICSR[5:4]. Where the each mode has different signal latch phase as following:

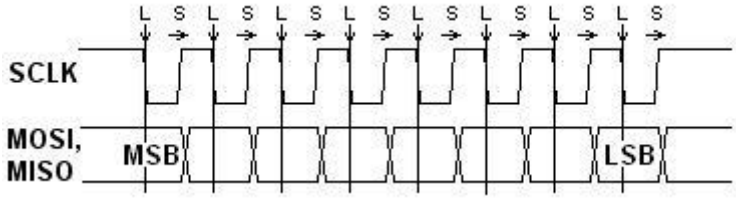
Mode 0



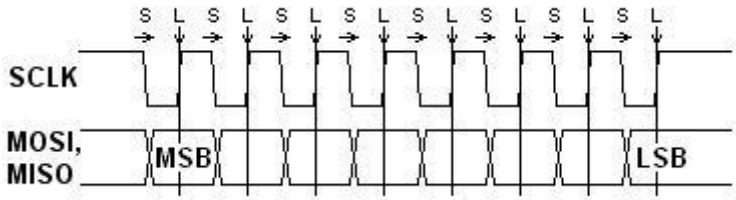
Mode 1



Mode 2



Mode 3



4.13.2 SPI Host Interface Register Description (0xFE70~0xFE7F)

SPI Host Interface Control Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x70	SHICSR	7	RO	SPI host BUSY flag. **0**: Idle **1**: Busy	0x00	0xFE
		6	RSV	Reserved		
		5~4	R/W	SPI Host Signal Phase Mode **00: Mode 0** CLK default 0, Drive data at falling edge and latch data at rising edge **01: Mode 1** CLK default 0, Drive data at rising edge and latch data at falling edge **10: Mode 2** CLK default 1, Drive data at rising edge and latch data at falling edge **11: Mode 3** CLK default 1, Drive data at falling edge and latch data at rising edge		
		3-2	R/W	SDI host CLK divider. **00**: 8 Mhz **01**: 4 Mhz **10**: 2 Mhz **11**: 1 Mhz		
		1	R/W	SPI Host SPI_CS# control **0**: Set SPI_CS# high **1**: Set SPI_CS# low		
		0	R/W	SPI host controller enable **0**: Disable **1**: Enable		

SDI host interface transmit data port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x71	SHITBUF	7-0	R/W	Writing to this register will force data output to SPI_DO in continuously serial 8 bits. MSB first.	0x00	0xFE
SDI host interface receive data port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x72	SHIRBUF	7-0	RO	Reading port for external SPI device.	0x00	0xFE

4.14 Watchdog Timer (WDT)

4.14.1 WDT Function Description

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

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

Here gives the highlight of WDT register field features & setting:

- 20 bit Watchdog (10bit programmable register field with 31.25ms resolution)
- Interrupt support
- 24 bit timer (TMR) support
- System 32khz clock source setting

Timing Example:

With a 32.768 khz WDT clock source, the timing period is about 30.5 us.

In IO373x :

The maximum WDT reset timer is 20bit (2^{20} x 30.5us = 32 seconds)

(Higher 10 bit register field available, 32ms), for real application recommended N≧3.

The maximum WDT interrupt time is half the WDT timer, (16 seconds)

The maximum TMR timer is 24 bit (2^{24} x 30.5us = 512 seconds, about 8 minute)

4.14.2 WDT Registers Description (0xFE80~0xFE8F)

WDT Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x80	WDTCFG	7	R/W	WDT clock source selection **0**: DPLL 32.768KHz source **1**: Internal OSC or External Crystal 32.768KHz source	0x00	0xFE
		6~3	RSV	Reserved		
		2	R/W	WDT test mode enable **0**: normal mode (depend on WDTCFG[7], 0xFE80[7]) **1**: test mode, clock driven by internal 32MHz		
		1	R/W	WDT interrupt enable (WDT reset warning) **0**: Disable **1**: Enable		
		0	R/W	WDT reset enable. Once WDT resets, two WDT pending flags are clear. **0**: Disable **1**: Enable		

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

WDT High 8-bit Counter Value (for WDT reset system of 10 bits counter)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x82	WDT	7-0	R/W	The high 8-bits of WDT counter value. The WDT timer unit is 30.5us * 2^{10} = 32ms. The overall high 10 bits counter is combined from WDT:LEDCFG[7:6] Please note, fill the overall value at least greater than or equal **3** (>=3) for hardware limitation.	0x00	0xFE

WDT Breathing LED Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x83	LEDCFG	7~6	R/W	The following 2 bits after WDT counter value. The WDT timer unit is 30.5us * 2^{10} = 32ms. The overall high 10 bits counter is combined from WDT:LEDCFG[7:6] Please note, fill the overall value at least greater than or equal **3** (>=3) for hardware limitation.	0x00	0xFE
		5~0	RSV	Reserved		

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

WDT TMR (24-bit Timer) Counter Match Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x85	TMR_MATCH	7-0	R/W	The highest 8bit counter match value register Assumed clock source 32.768KHz, the TMR time unit is 2^{16} x 30.5us = 2 second in this register. When timer counter[23:16] is reached this value, timer emits interrupt and TMR_CFG[1] is set to 1 .	0x00	0xFE

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

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

4.15 X-Bus Interface (XBI)

4.15.1 XBI Function Description

IO373x implements XBI module to handle embedded flash or ROM access.

4.15.2 XBI Registers Description (0xFEA0~0xFEBF)

XBI Embedded Flash Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA0	XBIEFCFG	7	RO	XBI Embedded Flash Controller Busy Flag	0xD0	0xFE
		6~4	R/W	Embedded flash read command clock count [2:0]		
		3	R/W	XBI Embedded Flash Controller Fetch Mode **0**: Fetch 4 bytes instruction data **1**: Fetch 1 bytes instruction data		
		2~1	RSV	Reserved		
		0	R/W	Embedded Flash FW mode **0**: Disable **1**: Enable		

XBI Embedded Flash signals 1 in FW mode						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA1	XBIEFSIG1	7	R/W	When F/W mode enable, the control bit of stand-by power 0: Disable 1: Enable	0x00	0xFE
		6	R/W	When F/W mode enable, the control bit of VPOS/VNEG discharge 0: Disable 1: Enable		
		5~4	R/W	When F/W mode enable, Erase/Program sequence control		
		3	R/W	When F/W mode enable, Auxiliary memory address select		
		2	R/W	When F/W mode enable, Page buffer write enable 0: Disable 1: Enable		
		1	R/W	When F/W mode enable, CLK enable for address/mode/ sequence control 0: Disable 1: Enable		
		0	R/W	When F/W mode enable, Address CLK input		

XBI Embedded Flash signals 2 in FW mode						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA2	XBIEFSIG2	7	R/W	PCLK source selection 0: PCLK from OSC output (CLKOUT) 1: PCLK form 32M fix clock	0x00	0xFE
		6	R/W	Pump source selection 0: Pump from flash output 1: Pump from Pe of embedded flash controller		
		5~4	R/W	When F/W mode enable, Data output bit number control		
		3~0	R/W	When F/W mode enable, Operational mode inputs		

XBI Pump IP trimming bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA3	XBIPUMP	7~4	R/W	PDAC[3:0] For independent control of VPOS pump level output DAC	0xD5	0xFE
		3~0	R/W	NDAC[3:0] For independent control of VNEG pump level output DAC		

XBI Flash IP trimming bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA4	XBIFM	7~4	R/W	ITIM[3:0] Trim DAC for trimming SA timing current of Vref	0x16	0xFE
		3~0	R/W	BDAC[3:0] For flash test		

XBI VR IP trimming bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA5	XBIVR	7~4	R/W	TCTRIM[3:0] Trimming bits for temperature coefficient of Vref	0x33	0xFE
		3~0	R/W	ABSTRIM[3:0] Trimming bits for absolute value of Vref		

XBI MISC Reg						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA6	XBIMISC	7	RO	Flash Ready status 0: Flash is not ready 1: Flash is ready	0x00	0xFE
		6	RO	Core logic power level status 0: Core logic power is lower than threshold 1: Core logic power is high than threshold		
		5	RSV	Reserved		
		4~0	R/W	S[4:0] TRIM bits for frequency.		

XBI Embedded Flash Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA7	XBIEFCMD	7-0	R/W	Commands support for embedded flash. Writing this register will force the protocol start. Please note, the address phases must be prior to command phase. **Embedded flash command support:** **02h** Page latch **03h** Read **20h** Erase selected page **60h** Erase whole e-flash **70h** Program selected page **80h** Clear HVPL data **90h** Read Trim data from special rows	0x00	0xFE

XBI Embedded Flash Address (15-bit) = [XBIEFA1(7bit) : XBIEFA0(8bit)]						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA8	XBIEFA0	7-0	R/W	Embedded Flash Address lower 8-bits (A7:A0)	0x00	0xFE
0xA9	XBIEFA1	6-0	R/W	Embedded Flash Address high 4-bits (A14:A8)	0x00	0xFE

XBI Embedded Flash Output Data Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAA	XBIEFDO	7-0	R/W	Output (write) data port of Embedded flash interface.	0x00	0xFE

XBI Embedded Flash Input Data Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAB	XBIEFDI	7-0	RO	Input (read) data port of Embedded flash interface.	0x00	0xFE

XBI Embedded ROM Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB0	XBIERCFG	7-5	RSV	Reserved	0x00	0xFE
		4	R/W	When FW mode enable, Embedded ROM CLK		
		3	R/W	When FW mode enable, Embedded ROM EN		
		2	R/W	When FW mode enable, Embedded ROM OE		
		1	R/W	When FW mode enable, Embedded ROM RST		
		0	R/W	Embedded Flash FW mode **0**: Disable **1**: Enable		

XBI Embedded ROM Trim bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB1	XBIERTRIM	7-4	RSV	Reserved	0x00	0xFE
		3-0	R/W	When FW mode enable, Embedded ROM Trim bits		

XBI Embedded ROM Address (12-bit) = [XBIEFA1(4bit) : XBIEFA0(8bit)]						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB2	XBIERADDR0	7-0	R/W	Embedded ROM Address lower 8-bits (A7:A0)	0x00	0xFE
0xB3	XBIERADDR1	3-0	R/W	Embedded ROM Address high 4-bits (A14:A8)	0x00	0xFE

XBI Embedded ROM Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xB4	XBIERDATA	7-0	R/W	Embedded ROM Data	0x00	0xFE

4.16 Consumer IR Controller (CIR)

4.16.1 CIR Function Description

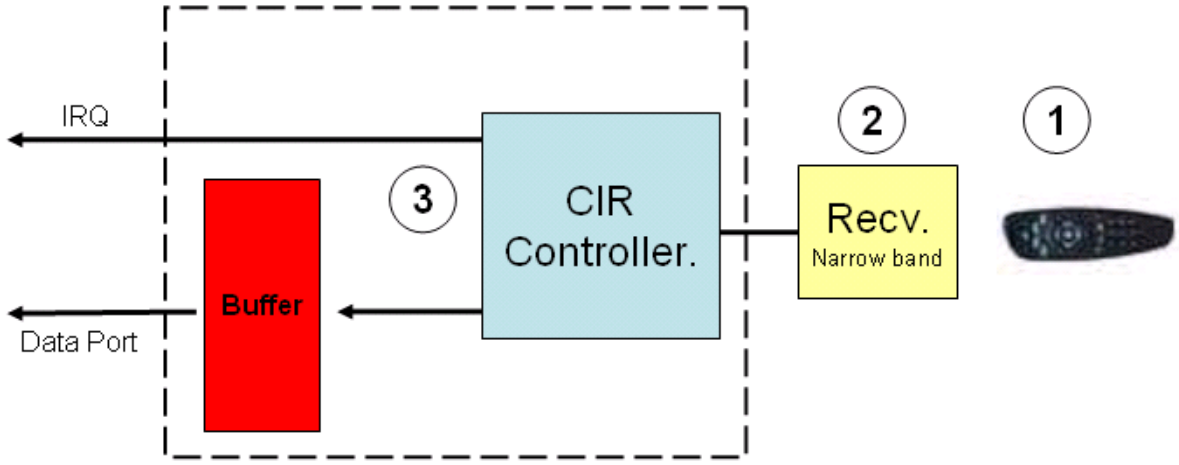
The IO373x embeds with a native hardware Consumer IR controller. Popular protocols are supported, such as RC-5/RC-6/NEC/RLC. The CIR controller handles the protocol of RC-5/RC-6/NEC/RLC for receiving, and only RLC for transmit. IRQ and I/O port are implemented. An extended function is implemented to support learning application. The basic features are list as the following table.

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

Here is the features highlight.

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

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

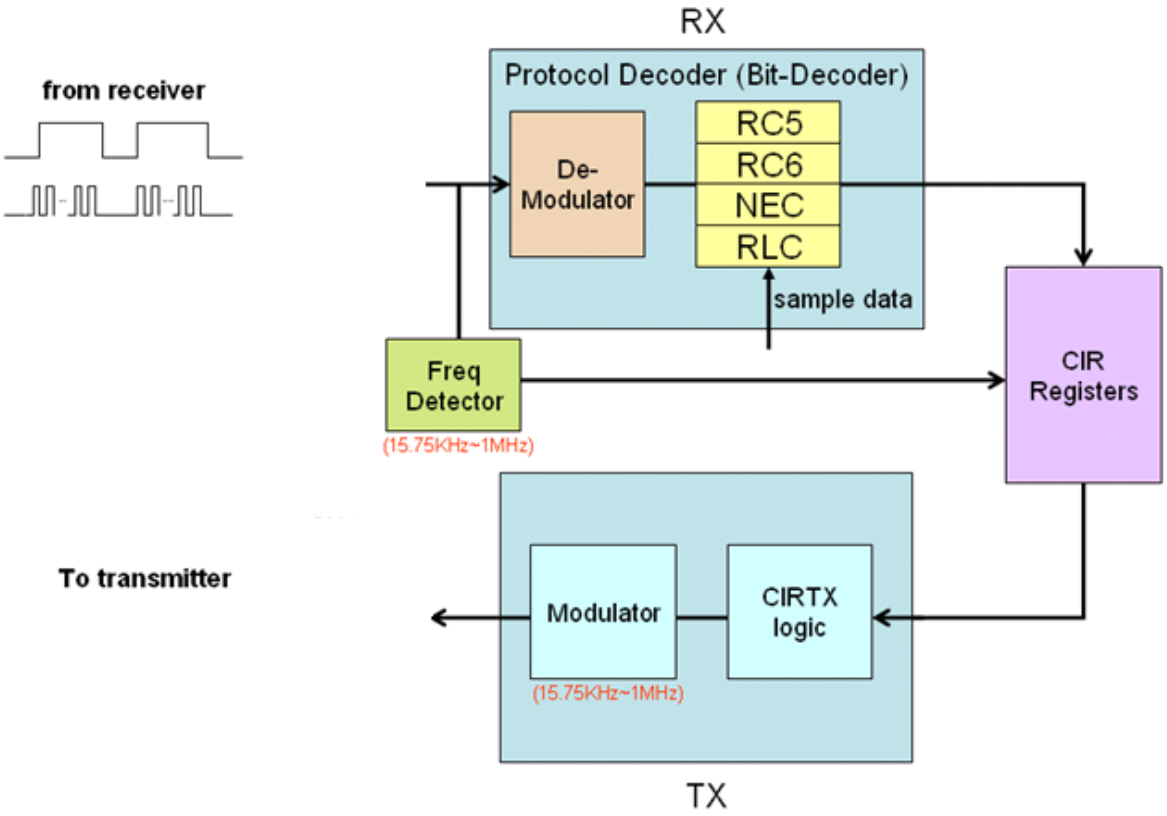


Here gives the guidance for programming CIR.

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

4.16.2 CIR Block Diagram

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



4.16.3 CIR Remote Protocol

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

4.16.3.1 Philips RC5 Protocol

Here highlights the features of Philips RC5 protocol.

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

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

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

4.16.3.2 Philips RC6 Protocol

Here highlights the features of Philips RC6 protocol.

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

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

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

4.16.3.3 NEC Protocol

Here highlights the features of NEC protocol.

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

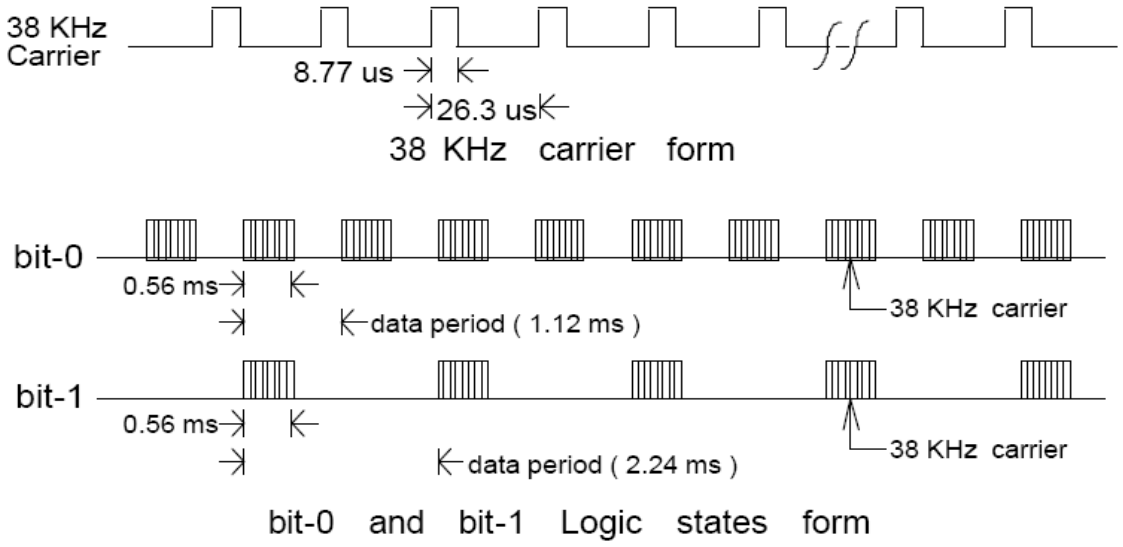
NCE Protocol					
AGC burst	space	Address	～Address	Command	～Command
9ms	4.5ms	8bit	8bit	8bit	8bit

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

4.16.4 CIR Automatic Carrier Frequency Detection and Modulation

To support learning feature, wide-band transmitter and receiver will be used in a system. IO373x introduces a new mechanism to provide carrier frequency information of wide-band receiver to the host.

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



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

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

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

IO373x provides the modulation ability for RLC transmit. The carrier frequency of modulation can be programmable. Before the carrier modulation, the programmer should notice the modulation polarity. That is, if the data bus (TX) is kept low in idle state, only data in high state will be modulated and the bit, **CIRMOD_PRD**[7], should be "**1**".

The related registers for RLC modulation is summarized as below.

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

4.16.5 CIR Registers Description (0xFEC0~0xFECF)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.17 Reserved

4.18 General Purpose Wake-up Controller (GPWU)

4.18.1 GPWU Function Description

The GPIO module provides flexible methods to wake-up the IO373x or to generate interrupt. Once the input function is determined, plenty of features for wakeup can be setup. Here is the table to summarize all the features. Please note DAC(GPO) pads could not served as GPWU.

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

4.18.2 GPWU Registers Description (0xFF30~0xFF7F)

GPIO Wakeup Event Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x30	GPWUEN00	7-0	R/W	GPIO00~GPIO07 Wakeup Event Switch bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x31	GPWUEN08	7-0	R/W	GPIO08~GPIO0F Wakeup Event Switch bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Wakeup event disable **1:** Wakeup event enable Note: GPIO0E/GPIO0F does not have GPI functionality.	0x00	0xFF
0x32	GPWUEN10	7-0	R/W	GPIO10~GPIO17 Wakeup Event Switch bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x33	GPWUEN18	7-0	R/W	GPIO18~GPIO1F Wakeup Event Switch bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x34	GPWUEN20	7-0	R/W	GPIO20~GPIO27 Wakeup Event Switch bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x35	GPWUEN28	7-0	R/W	GPIO28~GPIO2F Wakeup Event Switch bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x36	GPWUEN30	7-0	R/W	GPIO30~GPIO37 Wakeup Event Switch bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF
0x37	GPWUEN38	7-0	R/W	GPIO38 Wakeup Event Switch bit[0] stand for GPIO38 separately **0:** Wakeup event disable **1:** Wakeup event enable	0x00	0xFF

GPIO Wakeup Event Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x40	GPWUPF00	7-0	R/W1C	GPIO00~GPIO07 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x41	GPWUPF08	7-0	R/W1C	GPIO08~GPIO0F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** No wakeup event **1:** Wakeup event pending Note: GPIO0E/GPIO0F does not have GPI functionality.	0x00	0xFF
0x42	GPWUPF10	7-0	R/W1C	GPIO10~GPIO17 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x43	GPWUPF18	7-0	R/W1C	GPIO18~GPIO1F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x44	GPWUPF20	7-0	R/W1C	GPIO20~GPIO27 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x45	GPWUPF28	7-0	R/W1C	GPIO28~GPIO2F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x46	GPWUPF30	7-0	R/W1C	GPIO30~GPIO37 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF
0x47	GPWUPF38	7-0	R/W1C	GPIO38 Wakeup Event Pending Flag bit[0] stand for GPIO38 separately **0:** No wakeup event **1:** Wakeup event pending	0x00	0xFF

GPIO Wakeup Polarity Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x50	GPWUPS00	7-0	R/W	GPIO00~GPIO07 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x51	GPWUPS08	7-0	R/W	GPIO08~GPIO0F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger) Note: GPIO0E/GPIO0F does not have GPI functionality.	0x00	0xFF
0x52	GPWUPS10	7-0	R/W	GPIO10~GPIO17 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x53	GPWUPS18	7-0	R/W	GPIO18~GPIO1F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x54	GPWUPS20	7-0	R/W	GPIO20~GPIO27 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x55	GPWUPS28	7-0	R/W	GPIO28~GPIO2F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x56	GPWUPS30	7-0	R/W	GPIO30~GPIO37 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x57	GPWUPS38	7-0	R/W	GPIO38 Wakeup Polarity Selection bit[0] stand for GPIO38 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:** High active (high trigger) / Rising (edge trigger)	0x00	0xFF

GPIO Wakeup Level/Edge Trigger Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x60	GPWUEL00	7-0	R/W	GPIO00~GPIO07 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x61	GPWUEL08	7-0	R/W	GPIO08~GPIO0F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Edge trigger **1:** Level trigger Note: GPIO0E/GPIO0F does not have GPI functionality.	0x00	0xFF
0x62	GPWUEL10	7-0	R/W	GPIO10~GPIO17 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x63	GPWUEL18	7-0	R/W	GPIO18~GPIO1F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x64	GPWUEL20	7-0	R/W	GPIO20~GPIO27 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x65	GPWUEL28	7-0	R/W	GPIO28~GPIO2F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x66	GPWUEL30	7-0	R/W	GPIO30~GPIO37 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x67	GPWUEL38	7-0	R/W	GPIO38 Wakeup Level/Edge Selection bit[0] stand for GPIO38 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF

GPIO Wakeup Input Change (Toggle) Trigger Selection Note:This setting will ignore the corresponding bit of GPWUELxx.						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x70	GPWUCHG00	7-0	R/W	GPIO00~GPIO07 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x71	GPWUCHG08	7-0	R/W	GPIO08~GPIO0F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Toggle trigger disable **1:** Toggle trigger enable Note: GPIO0E/GPIO0F does not have GPI functionality.	0x00	0xFF
0x72	GPWUCHG10	7-0	R/W	GPIO10~GPIO17 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x73	GPWUCHG18	7-0	R/W	GPIO18~GPIO1F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x74	GPWUCHG20	7-0	R/W	GPIO20~GPIO27 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x75	GPWUCHG28	7-0	R/W	GPIO28~GPIO2F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x76	GPWUCHG30	7-0	R/W	GPIO30~GPIO37 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x77	GPWUCHG38	7-0	R/W	GPIO38 Wakeup Input Change (Toggle) Trigger bit[0] stand for GPIO38 separately **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF

4.19 System Management Bus (SMBus) Master Controller

The SMBus host controller in IO373x supports :

1. SMBus host controller protocol generator and master

2. SMbus host controller slave alarm receiver

3. Misc functions for interrupt generation, host controller disable, timeout, SMBus clock period setting, etc.

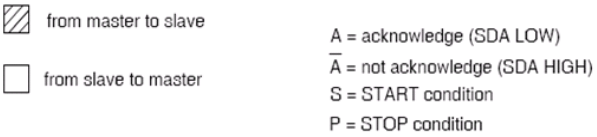
4.19.1 SMBus Master Controller Function Description

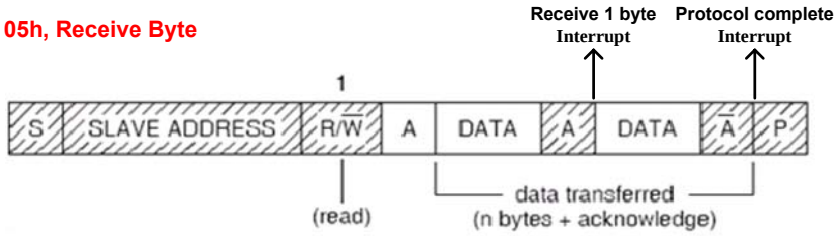
The SMBus controller supports 12 command protocols as following table. For more detail about each command protocol, please refer to the *System Management Bus Specification 2.0.*

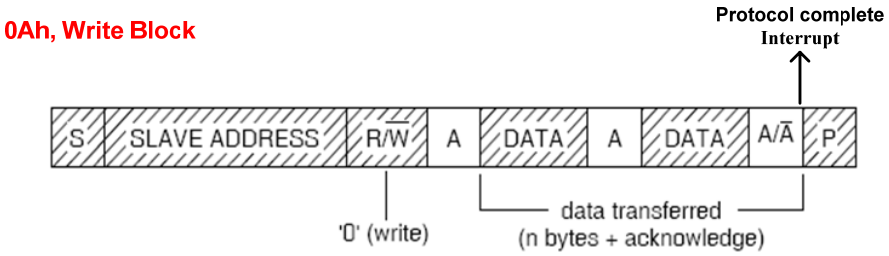
Command Byte	Command	Command Byte	Command
02h	Quick Write	08h	Write Word
03h	Quick Read	09h	Read Word
04h	Send Byte	0Ah	Write Block
05h	Receive Byte	0Bh	Read Block
06h	Write Byte	0Ch	Word Process
07h	Read Byte	0Dh	Block Process

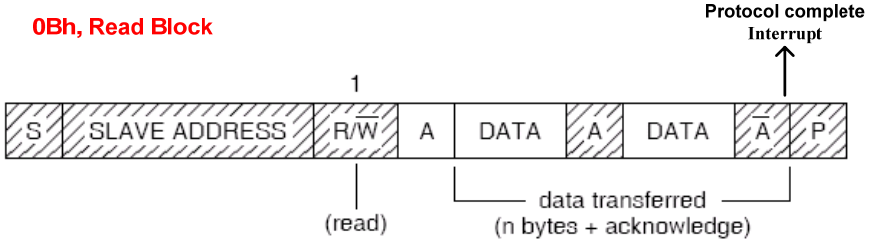
The SMBus supports **I²C mode**. If the SMBus operates in this mode, only 3 protocols are supported, ***05h*** *(Receive Byte)*, ***0Ah*** *(Write Block)* and ***0Bh*** *(Read Block)*. Here gives the brief programming guide of how to use Byte mode as following table.

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









Examples :

A. Write 12 bytes data into SMBus device (address = 0x16)

1. Set SMBADR (0xFF9A) = 0x16. //bit0 = 0 -> write
2. Write 8 bytes data into SMBDAT (0xFF9C~0xFFA3)//the length of data array = 8
3. Set SMBCNT (0xFFBC) = 0x0C. //12 bytes data
4. Set SMBPRTCL (0xFF98) = 0x0A. //start protocol, 0Ah write block.
5. Wait the interrupt of SMBus. //8 bytes data transferred //completely.
6. Write other 4 bytes data into SMBDAT (0xFF9C~0xFF9F).
7. Clear the bit5 of SMBSTS(0xFF99) to notify the block protocol going.
8. Wait the interrupt of SMBus. //protocol completed.

B. Read 12 bytes data from SMBus device (address = 0x16)

1. Set SMBADR (0xFF9A) = 0x17. //bit0 = 1 -> read
2. Set SMBCNT (0xFFBC) = 0x0C. //12 bytes data
3. Set SMBPRTCL (0xFF98) = 0x0B. //start protocol, 0Bh read block
4. Wait the interrupt of SMBus. //8 bytes data transferred //completely
5. Read 8 bytes data from SMBDAT (0xFF9C~0xFFA3)
6. Clear the bit5 of SMBSTS(0xFF99) to notify the block protocol going.
7. Wait the interrupt of SMBus. //completely
8. Read other 4 bytes data from SMBDAT (0xFF9C~0xFF9F)

C. Read n (>1) bytes data from SMBus device (address = 0x16)

1. Set SMBADR (0xFF9A) = 0x17. //bit0 = 1 -> read
2. Set SMBPF (0xFF96.6) = 0. //ACK
3. Set SMBPRTCL (0xFF98) = 0x05. //start protocol, 05h receive byte
4. Wait the interrupt of SMBus. //Received one byte
5. Read 1 byte data from SMBDAT (0xFF9C).
6. This protocol is completed (Is it the last 1 byte data to be read)?
   Yes -> Set SMBPF (0xFF96.6) = 1 (NACK) and clear SMBPF (0xFF96.5). Go to step7.
   No -> Clear SMBPF (0xFF96.5). Go to step4.
7. Wait the interrupt of SMBus. //protocol completed.
8. Read the last byte data from SMBDAT (0xFF9C).

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

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

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

Slave (SMBus device) to Master

Master (SMBus host) to Slave

4.19.2 SMBus Master 0 Register Description (0xFF80~0xFFBF)

SMBus Slave Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x90	SMB0RSA	7-0	R/W	SMBus host slave address (7-bits long), bit0 ignores.	0x00	0xFF

SMBus CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	SMB0TCRC	7-0	RO	CRC value transmits to SMBus.	0x00	0xFF

SMBus Pin Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x93	SMB0PIN	7	R/W	SMBus data line forced to low. Write "**0**" to force **SDA** low.	0xC0	0xFF
		6	R/W	SMBus clock line forced to low. Write "**0**" to force **SCL** low.		
		5	RO	Status of SDA or the wired AND of (SDA0 AND SDA1)		
		4	RO	Status of SCL or the wired AND of (SCL0 AND SCL1)		
		3	R/W	I2C mode function enable 3 protocols support, **Write Block/Read Block/Receive Byte.** Protocols are defined via register SMB0PRTCL[6:0] **0**: Disable **1**: Enable		
		2	R/W	SCL/SDA input debounce enable. **0**: Disable **1**: Enable		
		1	RSV	Reserved		
		0	R/W	SCL/SDA pins connected to SMBus controller. **0**: Disable **1**: Enable		

SMBus Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	SMB0CFG	7	R/W	SMBus master disable **0**: Enable master function. **1**: Disable master function	0x06	0xFF
		6	R/W	SMBus host alarm protocol disable (0xFFBD~0xFFBF disable) **0**: Enable slave function. **1**: Disable slave function		
		5	RSV	Reserved		
		4-0	R/W	SMBus clock period If **SMB0CFG[4:0]>0 and SMB0PIN[2]=1,** the period is SMBus clock period = (SMB0CFG[4:0]+1) * 4 μs If **SMB0CFG[4:0]>0 and SMB0PIN[2]=0**, the period is SMBus clock period = SMB0CFG[4:0] * 4 μs Please **do not** set these bits to "0".		

SMBus Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	SMB0EN	7	RO	SMBus host controller status **0**: not busy **1**: busy	0x00	0xFF
		6-4	RSV	Reserved		
		3	R/W	SMBus slave protocol selection. **0**: word read/write **1**: byte read/write		
		2	R/W	SMBus slave mode enable. **0**: Disable **1**: Enable		
		1	R/W	SMBus alert (host notify protocol) interrupt **0**: Disable **1**: Enable		
		0	R/W	SMBus protocol completion interrupt **0**: Disable **1**: Enable		

SMBus Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	SMB0PF	7	RSV	Reserved	0x00	0xFF
		6	R/W	ACK bit of Receive Byte (I2C Mode) protocol **0**: ACK, the Receive Byte protocol keeps going **1**: NACK, once the F/W ready to obtain the last Receive Byte, F/W set this bit in advance. After this last byte transferred, the controller issues NACK to device and the protocol stop.		
		5	R/W1C	Read data interrupt flag of Receive Byte (I2C Mode) protocol **0**: no event **1**: event occurs		
		4	RO	Read protocol interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		3	R/W1C	Interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		2-0	RSV	Reserved		

SMBus Received CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x97	SMB0RCRC	7-0	RO	The CRC value received from SMBus slave device.	0x00	0xFF

SMBus Protocol						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	SMB0PRTCL	7	R/W	SMBus transaction with PEC (Packet Error Check) **0**: Disable **1**: Enable.	0x00	0xFF
		6-0	R/W	Command protocol. **02h**: Quick Write **03h**: Quick Read **04h**: Send Byte **05h**: Receive Byte / Receive Byte (Byte Mode) **06h**: Write Byte **07h**: Read Byte **08h**: Write Word **09h**: Read Word **0Ah**: Write Block **/** Write Block (Byte Mode) **0Bh**: Read Block / Read Block (Byte Mode) **0Ch**: Word Process **0Dh**: Block Process others: Reserved		

SMBus Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x99	SMB0STS	7	R/W0C	SMBus command done flag **0**: no event (Write 0 to clear) **1**: event occurs	0x00	0xFF
		6	R/W0C	SMBus alarm (host notify protocol) interrupt flag **0**: no event (Write 0 to clear) **1**: event occurs		
		5	RSV	Reserved		
		4-0	R/W	Error code. **00h**: no error **07h**: unknown address failure. **10h**: device address no ACK **12h**: command no ACK **13h**: device data no ACK **17h**: device access deny **18h**: SMBus timeout **19h:** unsupported protocol **1Ah**: SMBus busy **1Fh**: PEC (Packet Error Check) error others: Reserved		

SMBus Address Port (For I2C Mode)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9A	SMB0ADR (SMB0PIN[3]=1)	7-1	R/W	SMBus address (7-bits long).	0x00	0xFF
		0	R/W	Data direction bit **0**: Write **1**: Read		

SMBus Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9B	SMB0CMD	7-0	R/W	SMBus command port	0x00	0xFF

SMBus Data Array (32 Bytes)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9C	SMB0DAT0	7-0	R/W	Data port for ***Send/Receive/Read Byte/Write Byte*** protocol	0x00	0xFF
0x9D	SMB0DAT1	7-0	R/W	Data port for ***Read Word/Write Word*** protocol, 2nd byte data	0x00	0xFF
0x9E	SMB0DAT2	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0x9F	SMB0DAT3	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA0	SMB0DAT4	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA1	SMB0DAT5	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA2	SMB0DAT6	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA3	SMB0DAT7	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xA4	SMB0DAT8	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xA5	SMB0DAT9	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xA6	SMB0DAT10	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xA7	SMB0DAT11	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xA8	SMB0DAT12	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xA9	SMB0DAT13	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xAA	SMB0DAT14	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xAB	SMB0DAT15	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xAC	SMB0DAT16	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xAD	SMB0DAT17	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xAE	SMB0DAT18	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xAF	SMB0DAT19	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB0	SMB0DAT20	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB1	SMB0DAT21	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB2	SMB0DAT22	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB3	SMB0DAT23	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB4	SMB0DAT24	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB5	SMB0DAT25	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB6	SMB0DAT26	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB7	SMB0DAT27	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB8	SMB0DAT28	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xB9	SMB0DAT29	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xBA	SMB0DAT30	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xBB	SMB0DAT31	7-0	R/W	Data port for Block protocol	0x00	0xFF

SMBus Block Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBC	SMB0CNT	7-6	RSV	Reserved	0x00	0xFF
		5~0	R/W	SMBus block count. "0x00", for 32-byte length in a block transfer.		

SMBus Alarm (Host Notify Protocol) Address / SMBus Slave Received Command Code						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBD	SMB0AADR	7-0	R/W	This register is alarm address or SMBus Slave Command Code for Response Slave Address.	0x00	0xFF

SMBus Alarm Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBE	SMB0ADAT0	7-0	R/W	Alarm data (low byte)	0x00	0xFF
0xBF	SMB0ADAT1	7-0	R/W	Alarm data (high byte)	0x00	0xFF

4.19.3 SMBus Master 1 Register Description (0xFFC0~0xFFFF)

SMBus Slave Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD0	SMB1RSA	7-0	R/W	SMBus host slave address (7-bits long), bit0 ignores.	0x00	0xFF

SMBus CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD2	SMB1TCRC	7-0	RO	CRC value transmits to SMBus.	0x00	0xFF

SMBus Pin Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD3	SMB1PIN	7	R/W	SMBus data line forced to low. Write "**0**" to force **SDA** low.	0xC0	0xFF
		6	R/W	SMBus clock line forced to low. Write "**0**" to force **SCL** low.		
		5	RO	Status of SDA or the wired AND of (SDA0 AND SDA1)		
		4	RO	Status of SCL or the wired AND of (SCL0 AND SCL1)		
		3	R/W	I2C mode function enable 3 protocols support, **Write Block/Read Block/Receive Byte.** Protocols are defined via register SMB1PRTCL[6:0] **0**: Disable **1**: Enable		
		2	R/W	SCL/SDA input debounce enable. **0**: Disable **1**: Enable		
		1	RSV	Reserved		
		0	R/W	SCL/SDA pin connected to SMBus controller. **0**: Disable **1**: Enable		

SMBus Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD4	SMB1CFG	7	R/W	SMBus master disable **0**: Enable master function. **1**: Disable master function	0x06	0xFF
		6	R/W	SMBus host alarm protocol disable (0xFFFD~0xFFFF disable) **0**: Enable slave function. **1**: Disable slave function		
		5	RSV	Reserved		
		4-0	R/W	SMBus clock period If **SMB1CFG[4:0]>0 and SMB1PIN[2]=1,** the period is SMBus clock period = (SMB1CFG[4:0]+1) * 4 μs If **SMB1CFG[4:0]>0 and SMB1PIN[2]=0**, the period is SMBus clock period = SMB1CFG[4:0] * 4 μs Please **do not** set these bits to "0".		

SMBus Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD5	SMB1EN	7	RO	SMBus host controller status **0**: not busy **1**: busy	0x00	0xFF
		6-4	RSV	Reserved		
		3	R/W	SMBus slave protocol selection. **0**: word read/write **1**: byte read/write		
		2	R/W	SMBus slave mode enable. **0**: Disable **1**: Enable		
		1	R/W	SMBus alert (host notify protocol) interrupt **0**: Disable **1**: Enable		
		0	R/W	SMBus protocol completion interrupt **0**: Disable **1**: Enable		

SMBus Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD6	SMB1PF	7	RSV	Reserved	0x00	0xFF
		6	R/W	ACK bit of Receive Byte (I2C Mode) protocol **0**: ACK, the Receive Byte protocol keeps going **1**: NACK, once the F/W ready to obtain the last Receive Byte, F/W set this bit in advance. After this last byte transferred, the controller issues NACK to device and the protocol stop.		
		5	R/W1C	Read data interrupt flag of Receive Byte (Byte Mode) protocol **0**: no event **1**: event occurs		
		4	RO	Read protocol interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		3	R/W1C	Interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		2-0	RSV	Reserved		

SMBus Received CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD7	SMB1RCRC	7-0	RO	The CRC value received from SMBus slave device.	0x00	0xFF

SMBus Protocol						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD8	SMB1PRTCL	7	R/W	SMBus transaction with PEC (Packet Error Check) **0**: Disable **1**: Enable.	0x00	0xFF
		6-0	R/W	Command protocol. **02h**: Quick Write **03h**: Quick Read **04h**: Send Byte **05h**: Receive Byte / Receive Byte (Byte Mode) **06h**: Write Byte **07h**: Read Byte **08h**: Write Word **09h**: Read Word **0Ah**: Write Block **/** Write Block (Byte Mode) **0Bh**: Read Block / Read Block (Byte Mode) **0Ch**: Word Process **0Dh**: Block Process others: Reserved		

SMBus Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xD9	SMB1STS	7	R/W0C	SMBus command done flag **0**: no event (Write 0 to clear) **1**: event occurs	0x00	0xFF
		6	R/W0C	SMBus alarm (host notify protocol) interrupt flag **0**: no event (Write 0 to clear) **1**: event occurs		
		5	RSV	Reserved		
		4-0	R/W	Error code. **00h**: no error **07h**: unknown address failure. **10h**: device address no ACK **12h**: command no ACK **13h**: device data no ACK **17h**: device access deny **18h**: SMBus timeout **19h:** unsupported protocol **1Ah**: SMBus busy **1Fh**: PEC (Packet Error Check) error others: Reserved		

SMBus Address Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xDA	SMB1ADR (SMB1PIN[3]=1)	7-1	R/W	SMBus address (7-bits long).	0x00	0xFF
		0	R/W	Data direction bit **0**: Write **1**: Read		

SMBus Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xDB	SMB1CMD	7-0	R/W	SMBus command port	0x00	0xFF

SMBus Data Array (32 Bytes)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xDC	SMB1DAT0	7-0	R/W	Data port for ***Send/Receive/Read Byte/Write Byte*** protocol	0x00	0xFF
0xDD	SMB1DAT1	7-0	R/W	Data port for ***Read Word/Write Word*** protocol, 2nd byte data	0x00	0xFF
0xDE	SMB1DAT2	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xDF	SMB1DAT3	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xE0	SMB1DAT4	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xE1	SMB1DAT5	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xE2	SMB1DAT6	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xE3	SMB1DAT7	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xE4	SMB1DAT8	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xE5	SMB1DAT9	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xE6	SMB1DAT10	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xE7	SMB1DAT11	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xE8	SMB1DAT12	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xE9	SMB1DAT13	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xEA	SMB1DAT14	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xEB	SMB1DAT15	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xEC	SMB1DAT16	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xED	SMB1DAT17	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xEE	SMB1DAT18	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xEF	SMB1DAT19	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF0	SMB1DAT20	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF1	SMB1DAT21	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF2	SMB1DAT22	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF3	SMB1DAT23	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF4	SMB1DAT24	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF5	SMB1DAT25	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF6	SMB1DAT26	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF7	SMB1DAT27	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF8	SMB1DAT28	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xF9	SMB1DAT29	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xFA	SMB1DAT30	7-0	R/W	Data port for Block protocol	0x00	0xFF
0xFB	SMB1DAT31	7-0	R/W	Data port for Block protocol	0x00	0xFF

SMBus Block Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xFC	SMB1CNT	7-6	RSV	Reserved	0x00	0xFF
		5~0	R/W	SMBus block count. "0x00", for 32-byte length in a block transfer.		

SMBus Alarm (Host Notify Protocol) Address / SMBus Slave Received Command Code						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xFD	SMB1AADR	7-0	R/W	This register is alarm address or SMBus Slave Command Code for Response Slave Address.	0x00	0xFF

SMBus Alarm Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xFE	SMB1ADAT0	7-0	R/W	Alarm data (low byte)	0x00	0xFF
0xFF	SMB1ADAT1	7-0	R/W	Alarm data (high byte)	0x00	0xFF

4.20 8051 Microprocessor

4.20.1 8051 Microprocessor Function Description

The Microprocessor inside IO373x is an industrial compatible i8051. The 8051 is featured with 128bytes Special Function Register (SFR), Serial port, 2 16-bit Timers and 3 I/O ports with interrupt capability. The 8051 operates based on external crystal and runs at 8MHz by default. The following figure gives an illustration of the 8051 architecture. Except the standard 128bytes SFR, 8051 in IO373x is designed with overall 256 bytes internal memory.

4.20.2 8051 Microprocessor Instruction

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

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

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

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

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

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

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

4.20.3 8051 Interrupt Controller

In order to support more application, the 8051 in IO373x extends interrupt channel to 24 for internal peripherals, that is, I/O port P0, P1 and P3 are with interrupt capability. The *interrupt priority for each channel is fixed* and no nested interrupt is supported. Here is the table to summarize the implementation of the interrupt controller.

Int. Source	Vector Address	Applications	Priority
IE0	0x0003	8051 external interrupt 0	0(Highest)
TF0	0x000B	8051 Timer 0	1
IE1	0x0013	8051 external interrupt 1	2
TF1	0x001B	8051 Timer 1	3
RI & TI	0x0023	8051 Serial port TX/RX interrupt	4
P0I[0]	0x0043	Watchdog	5
P0I[1]	0x004B	OWM	6
P0I[2]	0x0053	PS/2 event	7
P0I[3]	0x005B		8
P0I[4]	0x0063	IKB	9
P0I[5]	0x006B	SMBus Device	10
P0I[6]	0x0073	SMBus Master 0	11
P0I[7]	0x007B	SMBus Master 1	12
P1I[0]	0x0083	Digital Sampler 0	13
P1I[1]	0x008B	Digital Sampler 0	14
P1I[2]	0x0093		15
P1I[3]	0x009B	CIR events	16
P1I[4]	0x00A3	GPT0 event	17
P1I[5]	0x00AB	GPT1 event	18
P1I[6]	0x00B3	GPT2 event	19
P1I[7]	0x00BB	GPT3 event	20
P3I[0]	0x00C3	CEC	21
P3I[1]	0x00CB	GPIO00~GPIO0F	22
P3I[2]	0x00D3	GPIO10~GPIO1F	23
P3I[3]	0x00DB	GPIO20~GPIO2F	24
P3I[4]	0x00E3	GPIO30~GPIO38	25
P3I[5]	0x00EB	SMBus Slave 0	26
P3I[6]	0x00F3	SMBus Slave 0	27
P3I[7]	0x00FB	ADC update	28(Lowest)

4.20.4 8051 Special Function Register (SFR)

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

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

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

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

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

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

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

4.20.5 8051 Microprocessor Register Description

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

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

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

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

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

Processor Control Register 2					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x86	PCON2	7	R/W	Reserved but this bit should be “0”.	0x20
		6	R/W	Timer0/Timer1 test mode enable. **0**: Disable **1**: Enable	
		5	R/W	Reserved	
		4	R/W	KBC modules write control. Once this bit set, 8051 could issue write access to external modules. **0**: Disable **1**: Enable	
		3	R/WC0	Same interrupt source pending flag. If the 8051 is handling some interrupt event, at the same time, the same source asserting the interrupt again, this flag will be set. If this flag set, the 8051 will re-enter ISR again once executing IRET. Writing “**0**” to clear this flag.	
		2	RSV	Reserved	
		1	R/W	E51 Pipeline Enable **0**: Disable, 8051 executing an instruction as 4T **1**: Enable, 8051 executing an instruction as 2T	
		0	R/W	Enable “not fetching instruction” while in idle loop	

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

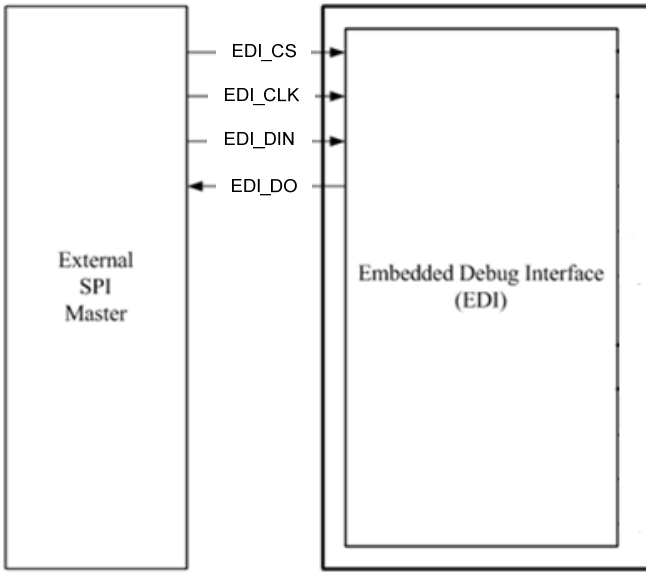
Port1 Interrupt Flag Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xE8	P1IF	7-0	R/W	Port 1 interrupt flag.	0x00

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

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

Application Appendix :

A.1 ENE debug Interface, EDI



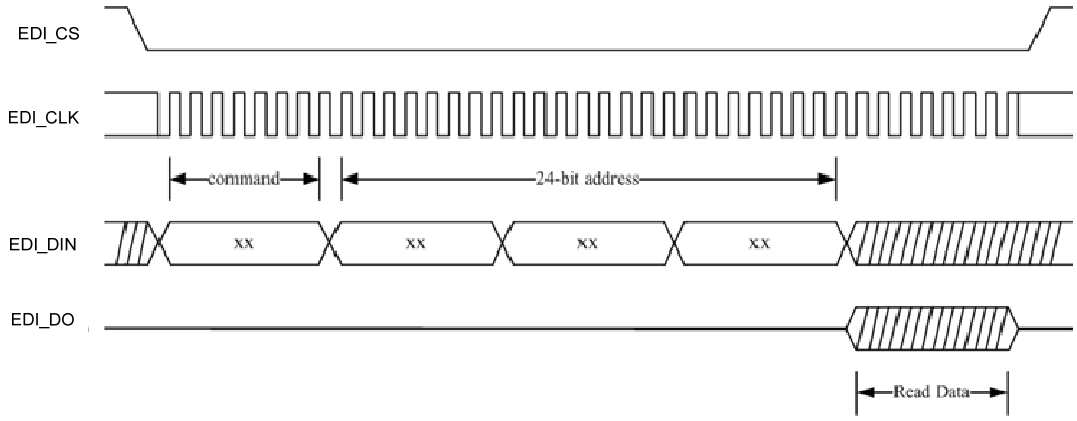
The above picture shows: EDI provide a SPI I/F as a debug interface.

The interface pin number in IO373x is as following :

IO3731 Pin No.	EDI Name	Alt. Function	GPIO	IO CELL
57	EDI_CS#	KSI4	GPIO33	BQC04IV
58	EDI_CLK	KSI5	GPIO35	BQC04IV
1	EDI_DI	KSI6	GPIO00	BQC04IV
2	EDI_DO	KSI7	GPIO01	BQC04IV

A.1.1 Enable EDI

To enable EDI, it is by detecting any SPI command with EDI_CLK frequency between 1MHz to 2MHz. After enabling EDI, the transaction frequency could be up to 16MHz.

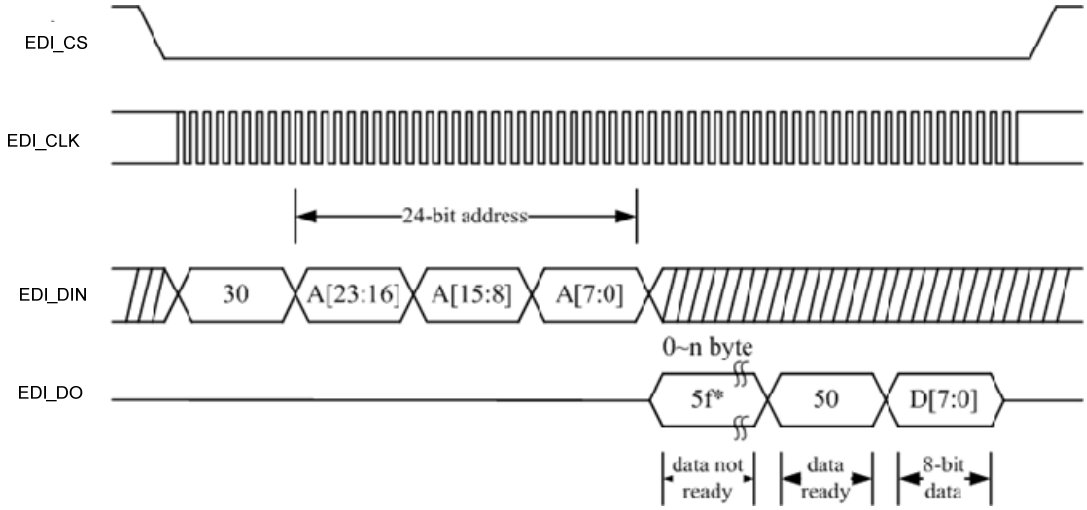


A.1.2 EDI Instructions

Command Name	Command Code	Address	Byte Count
Read	30h	3	
Write	40h	3	
Disable EDI	F3h	0	

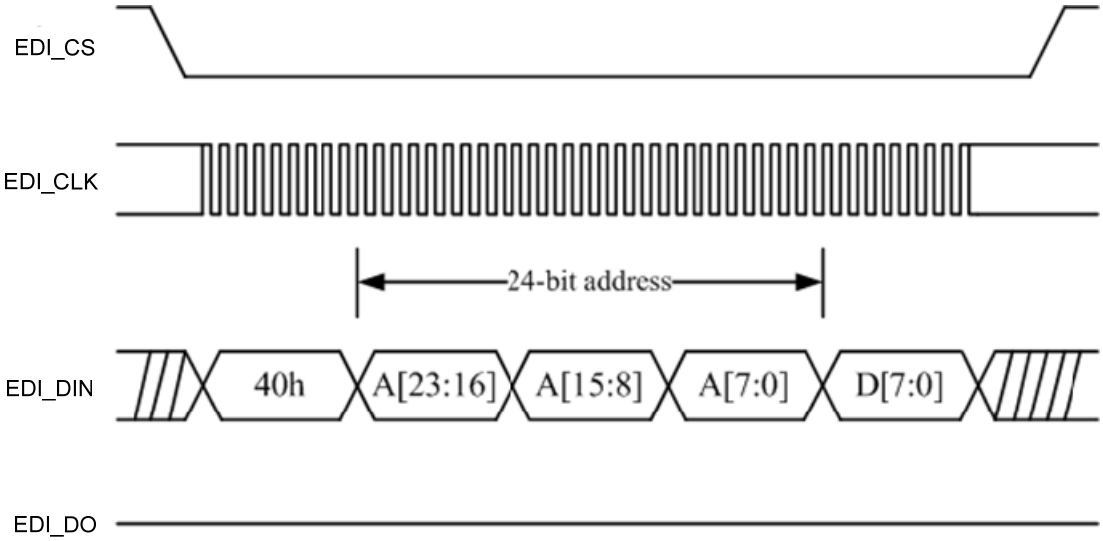
A.1.3 Read Command

Read command is by issuing command code **30h**, which is followed by 3-bytes of the target address. While fetching data, **5Fh** is shown on EDI_DO to indicate BUSY. This could be lasting for n bytes length. After fetching data, **50h** is shown on EDI_DO to indicate the data is ready to be read, and the next 1 byte is the valid read data.



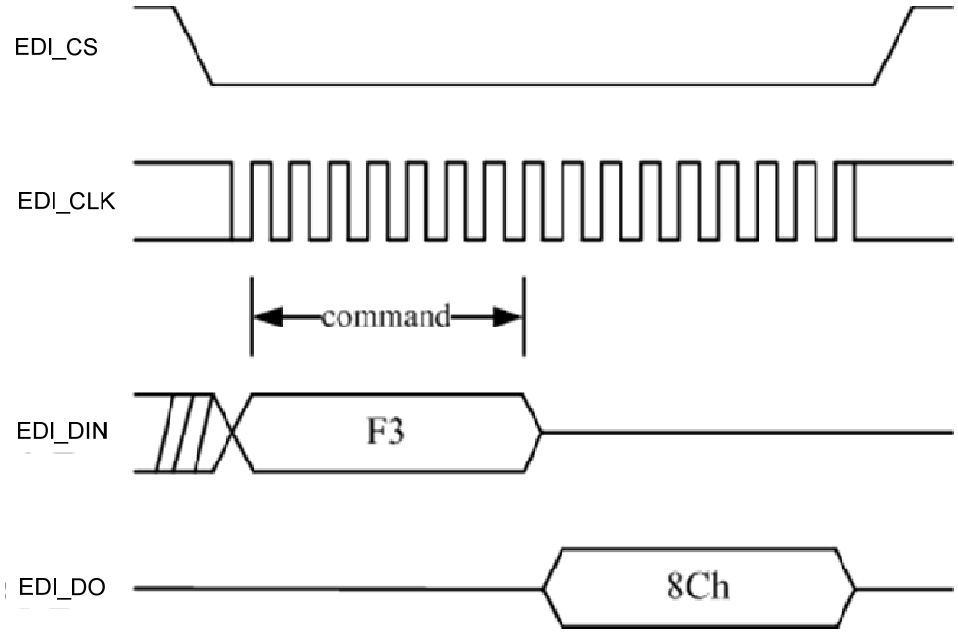
A.1.4 Write Command

Write command is by issuing command code **40h**, which is followed by 3-bytes of the target address. The 64k address boundary should be kept. Write command on EDI v2.0 (KB9010) only support EHB memory space (0h~FFFFh), in other words A[23:16] should be **0h**). For details memory space mapping, please refer Part B.



A.1.5 Disable EDI Command

Disable EDI command is by issuing command code **F3h**. On EDI_DO pin, **8Ch** would be shown to indicate the EDI is disabled.



5. Electrical Characteristics

5.1 Absolute Maximum Rating

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

5.2 DC Electrical Characteristics

BQC16IV

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.06		V	
Input High Threshold	V_{t+}		1.85		V	
Hysteresis	V_{TH}		0.79		V	
Output Low Voltage	V_{OL}			0.4	V	16mA Sink
Output High Voltage	V_{OH}	2.8			V	16mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		48.5K		Ω	V_I=0V
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

BQC04IV

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.06		V	
Input High Threshold	V_{t+}		1.85		V	
Hysteresis	V_{TH}		0.79		V	
Output Low Voltage	V_{OL}			0.4	V	4mA Sink
Output High Voltage	V_{OH}	2.8			V	4mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		48.5K		Ω	V_I=0V
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

BQC04IV_26K

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.06		V	
Input High Threshold	V_{t+}		1.85		V	
Hysteresis	V_{TH}		0.79		V	
Output Low Voltage	V_{OL}			0.4	V	4mA Sink
Output High Voltage	V_{OH}	2.8			V	4mA Source
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		29K		Ω	V_I=0V
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

IQADCI (ADC cell)

(Input only, No Pull-Up resistance function)

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.03		V	
Input High Threshold	V_{t+}		1.86		V	
Hysteresis	V_{TH}		0.83		V	
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		--		Ω	
Input Capacitance	C_{PU}		5.5		pF	

IQTI (ADC cell)

(Input only, No Pull-Up resistance function)

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold	V_{t-}		1.03		V	
Input High Threshold	V_{t+}		1.86		V	
Hysteresis	V_{TH}		0.83		V	
Input Leakage Current	I_{IL}		0.02		μA	No pull-up
Input Pull-Up Resistance	R_{PU}		--		Ω	
Input Capacitance	C_{PU}		5.5		pF	

OCT04H (DAC cell)

(Output only)

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

5.3 A/D & D/A Characteristics

ADC characteristics

Parameter	Limits			Unit
	Min	Typ	Max	
Resolution				Bit
Integral Non-linearity Error (INL)				LSB
Differential Non-linearity Error (DNL)				LSB
Offset Error				LSB
Gain Error				LSB
A/D Input Voltage Range				V
A/D Input Leakage Current				uA
A/D Input Resistance				MΩ
A/D Input Capacitance				pF
A/D Clock Frequency				MHz
Voltage Conversion Time				uS

DAC characteristics

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

5.4 Recommend Operation Condition

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
Vcc	Power Source Voltage	2.2	3.3	3.6	V
GND	Ground Voltage	-0.3	0	0.3	V
V_{CCA}	Analog Reference Voltage (for A/D and D/A)	3.0	3.3	3.6	V
AGND	Analog Ground Voltage	-0.3	0	0.3	V
T_{op}	Operating Temperature	0	25	70	℃

5.5 Operating Current

Symbol	Parameter	Limits	Unit
		Typ	
Icc	Typical current consumption in operating state under Windows environment. All clock domains are running, and no keyboard/mouse activities.	8	mA
I_{STOP}	Typical current consumption in STOP mode when PLL in low power state, WDT disable, functional modules OFF	50	uA

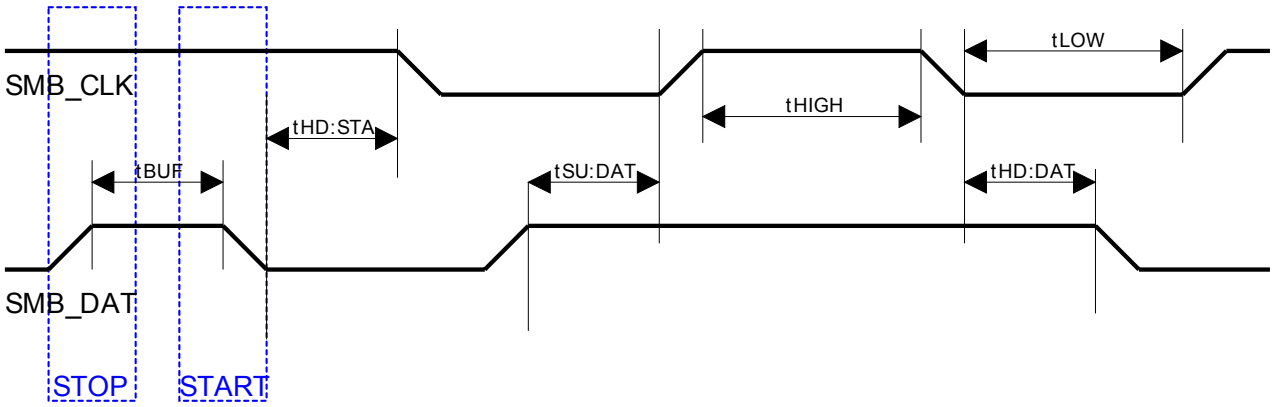
5.6 Package Thermal Information

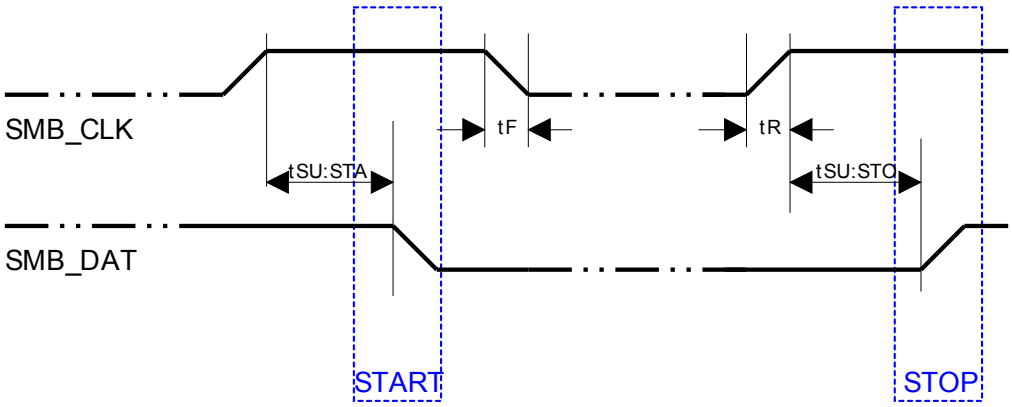
Thermal resistance (degrees C/W). $Theta_{JA}$ 、$Theta_{JC}$ values for IO3731

	$Theta_{JA}$ @ 0 m/s	**$Theta_{JC}$**
64-Pin LQFP	57.3	22.1

5.7 AC Electrical Characteristics

5.7.1 SMBus interface Timing





Timing Parameters

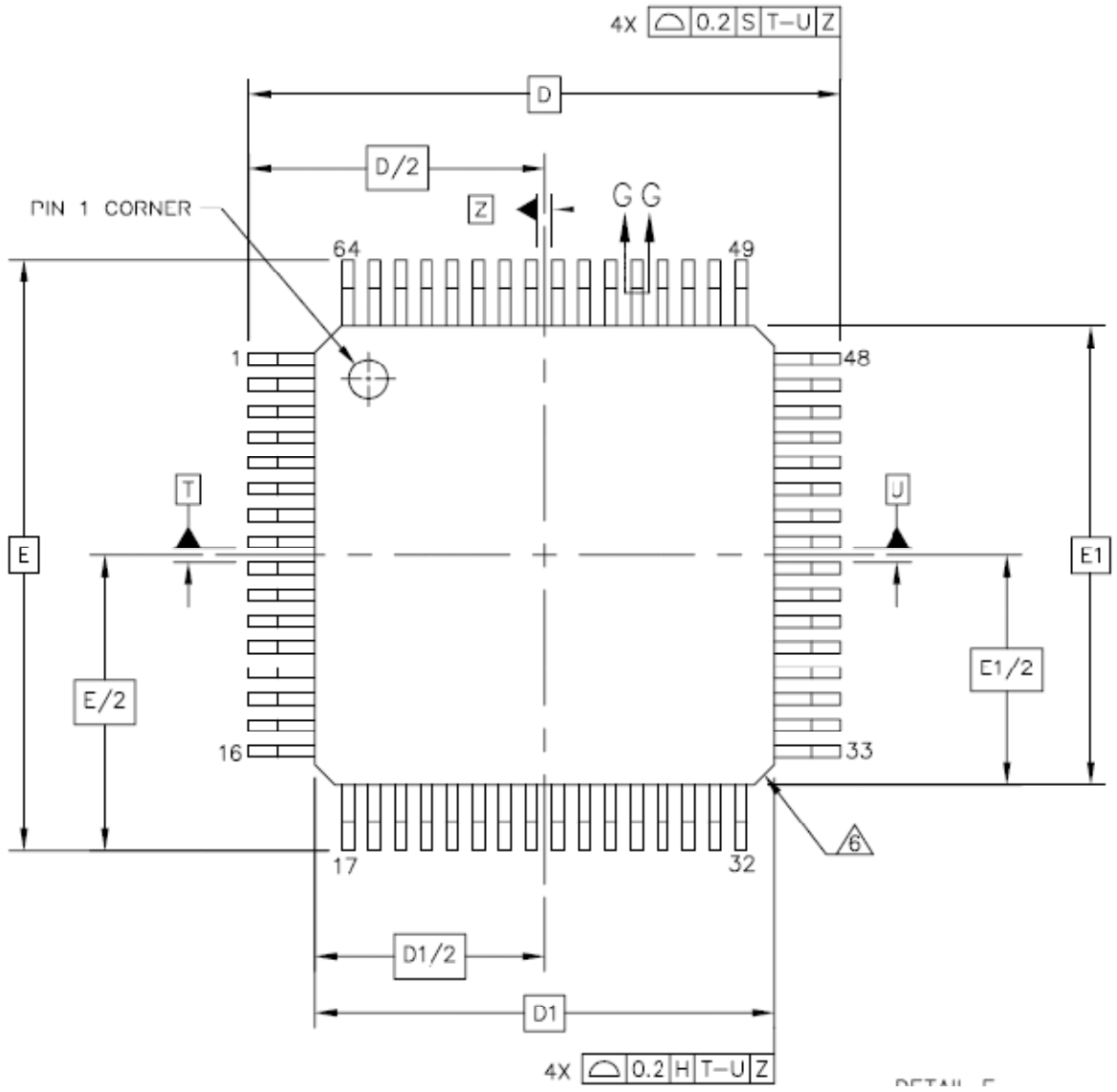
Symbol	Parameter	Min	Typ.	Max	Units	Notes
T_{buf}	Bus free time between Stop and Start Condition	4.7			µs	
$T_{hd:sta}$	Hold time after (repeated) star condition. After this period, the first clock is generated.	4.0			µs	
$T_{su:sta}$	Repeated start condition setup time	4.7			µs	
$T_{su:sto}$	Stop confition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

1. For characteristic only
2. SMBUS frequencry dependant

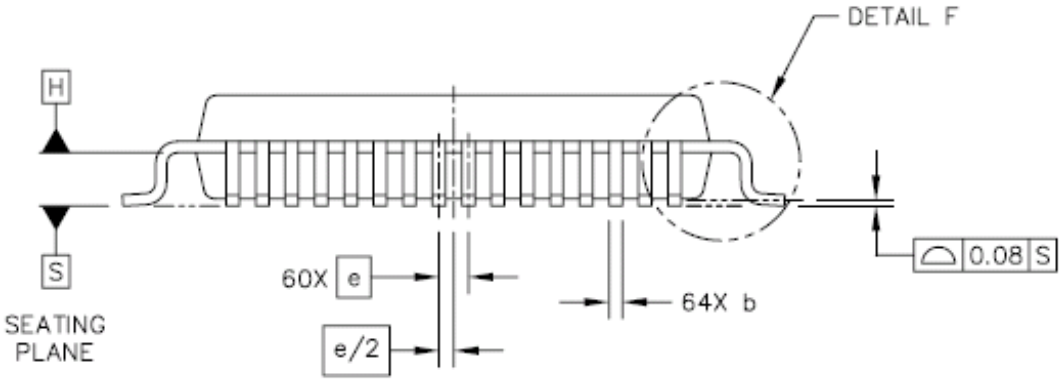
6. Package Information

6.1 LQFP 64-Pin Outline Diagram

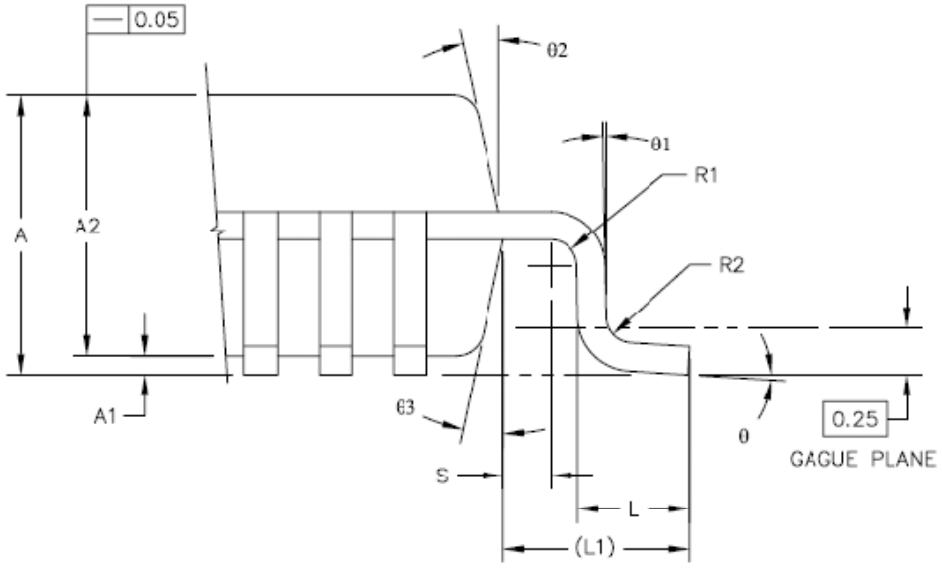
6.1.1 Top View



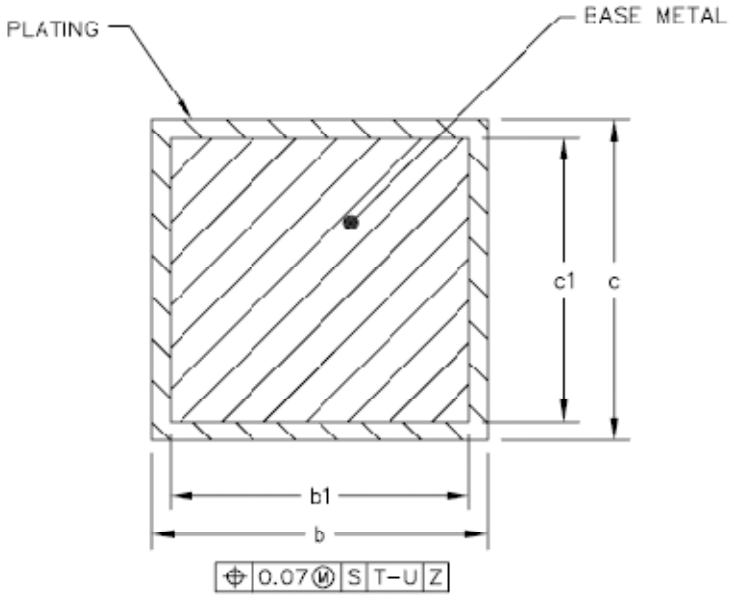
6.1.2 Side View



6.1.3 Lead View



DETAIL F



SECTION C–C

6.1.4 LQFP Outline Dimensions

DIM	Min.	Typ.	Max.	DIM	Min.	Typ.	Max.
A			1.6	E		9 BSC	
A1	0.05		0.15	E1		7 BSC	
A2	1.35	1.4	1.45	L	0.45	0.6	0.75
b	0.13	0.18	0.23	L1		1 REF	
b1	0.13	0.16	0.19	R1	0.08		
c	0.09		0.2	R2	0.08		0.2
c1	0.09		0.16	S	0.2		
D		9 BSC		θ	0°	3.5°	7°
D1		7 BSC		$\theta 1$	0°		
e		0.4 BSC		$\theta 2$	11°	12°	13°
				$\theta 3$	11°	12°	13°
Unit	mm						
Package	7 x 7 x 1.4 , 0.4 PITCH POD						

7. Part Number Description

Part Number	Package Size	Description
IO3731Q A1	7 x 7 x 1.4 LQFP-64	Lead Free