CMOS Single-chip 8-bit MCU with 10-bit ADC and Operational Amplifier

User's Manual Version 1.30

Introduction

This user's manual contains complete information for application developers who use A96L414/A96L416 for their specific needs.

A96L414/A96L416 is an advanced 8-bit CMOS MCU with 8/16Kbytes of Flash. Offering the convenience of Flash multi-programming features, this device can provide fire alarm and single type smoke detectors systems with a simple, robust, and cost effective solution as a complete set of semiconductor product to implement the smart alarm systems.

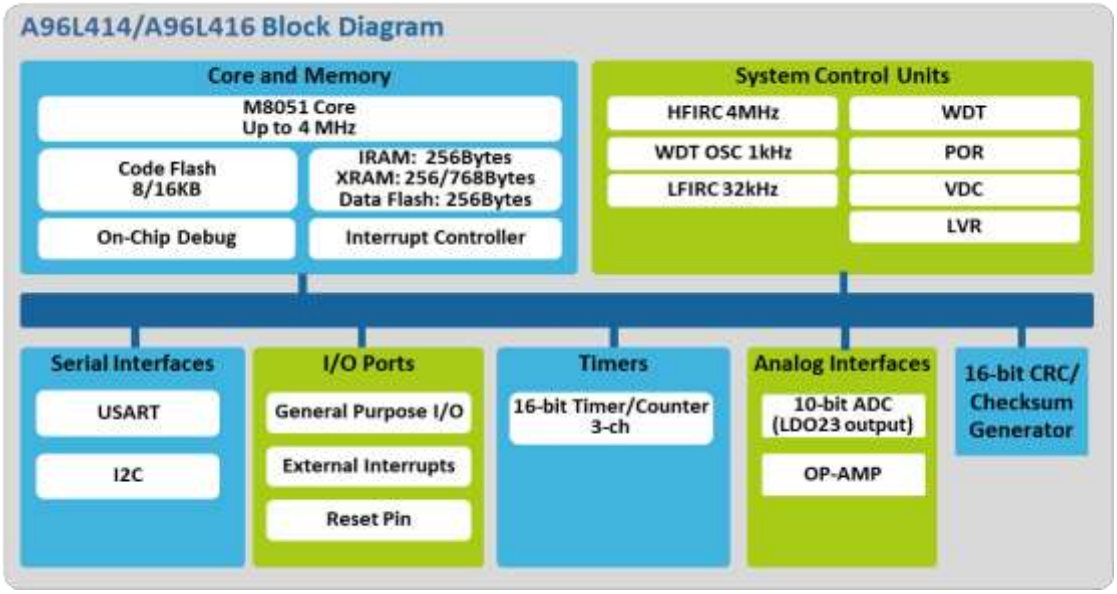


Figure 1. A96L414/A96L416 Block Diagram

Reference documents

- A96L414/A96L416 Datasheet: It includes information on mechanical characteristics, development methods, and ordering information. It is available at ABOV website, https://www.abovsemi.com.
- SDK 51 User's guide (System Design Kit): It was released by Intel in 1982. It contains all of components of a single board computer based on Intel's 8051 single chip microcomputer.
- Information on Mentor Graphics 8051 microcontroller: This technical document is provided at Mentor® website, https://www.mentor.com/products/ip/peripheral/microcontroller/

Contents

List of figures

List of tables

1 Description

A96L414/A96L416 is an advanced CMOS 8-bit microcontroller with 8/16Kbytes of FLASH. This is a powerful microcontroller which provides low power consumption and cost effective solution to smoke detector applications. A96L414/A96L416 supports power down modes to reduce power consumption.

Table 1 introduces features of A96L414/A96L416 and peripheral counts.

1.1 Device overview

Table 1. A96L414/A96L416 Device Features and Peripheral Counts

Peripheral		**A96L414/A96L416**
CPU		8-bit CISC core (M8051, 2 clocks per cycle)
Flash		• 8/16Kbytes with self r/w capability • On chip debug and ISP • Endurance: 10,000 cycles
IRAM		256 bytes
XRAM		256/768 bytes
Data Flash		• 256 bytes • Endurance: 100,000 cycles
GPIO		• Normal I/Os • 18 ports: P0[7:0], P1[5:0], P2[3:0]
Timer/ counter		• BIT 8-bit x 1-ch • WDT 8-bit x 1-ch: 1KHz internal RC oscillator for WDT or LF internal RC oscillator for WDT • 16-bit x 3-ch (T0/T1/T2)
Programmable pulse generation		Pulse generation (by T0/T1/T2)
I2C		8-bit x 1-ch
ADC		• 10-bit ADC, 7 input channels • V_{BGR}: 0.92V ±3%(T_A = -40℃ to +85℃) • LDO: 2.32V ±3.45%(T_A = 25℃)
Operational amplifier		• 2-ch • Rail-to-rail output
CRC and checksum generator		• 16-bit • Auto and user CRC/ checksum mode
Reset	Power on reset	Reset release level (1.4V)
	Low voltage reset	4 level detect (1.60V/ 2.20V/ 2.40V/ 2.70V)

Table 1. A96L414/A96L416 Device Features and Peripheral Counts (continued)

Peripheral	A96L414/A96L416
Constant sink current generator	• 2-ch • 16-steps selectable • Max. 274mA sink current
USART	UART + SPI • 8-bit UART x 1-ch • 8-bit SPI x 1-ch
Interrupt sources	• External interrupts: EINT0/1/2/3/10/11/12, 7 • Timer0/1/2, 3 • WDT1 • BIT1 • I2C1 • ADC 1 • USART Rx/ Tx 2
Internal RC oscillator	4MHz ± 3.0% (T_A = -40℃ to +85℃)
Power down mode	STOP, IDLE
Operating voltage and frequency	• 2.0V to 3.6V @ 0.5 to 4.0MHz with HFIRC • 2.0V to 3.6V @ 32KHz with LFIRC • Voltage dropout converter included for core
Minimum instruction execution time	0.25us @ 4MHz HFIRC
Operating temperature	-40℃ to +85℃
Package type	• 20 TSSOP • 16 SOPN • Pb-free package

1.2 Block diagram

Figure 2 describes A96L414/A96L416 in a block diagram.

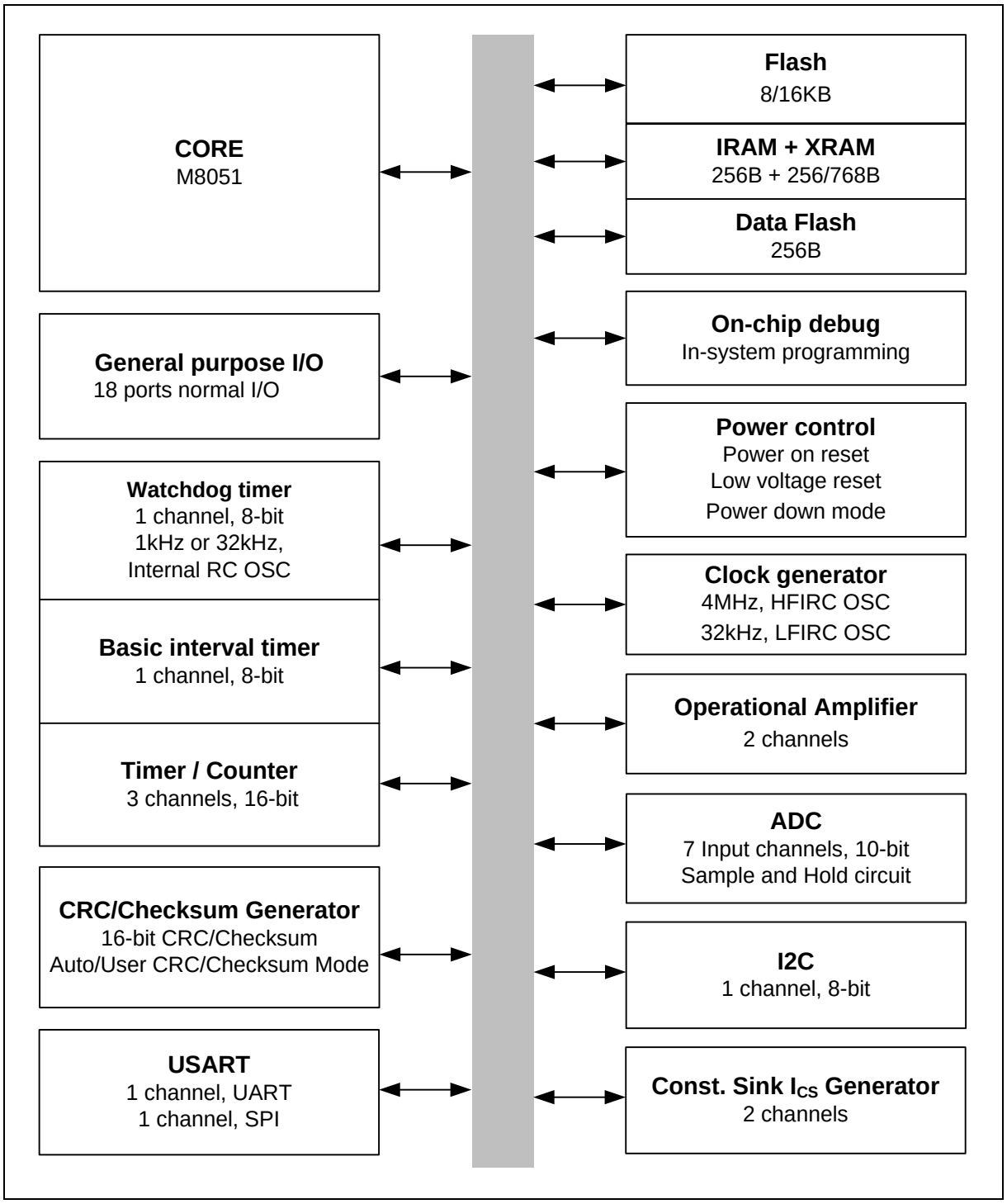
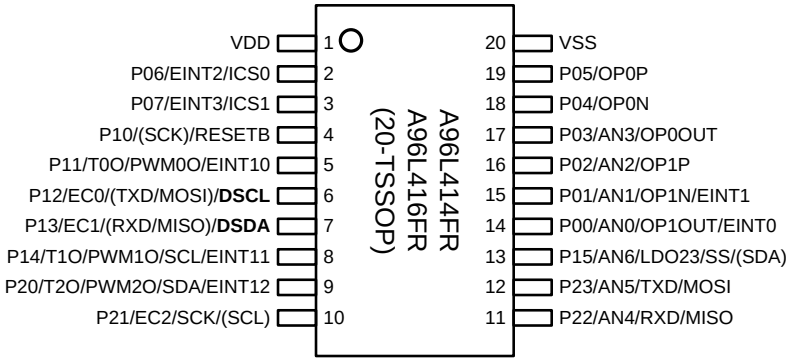


Figure 2. A96L414/A96L416 Block Diagram

2 Pinouts and pin descriptions

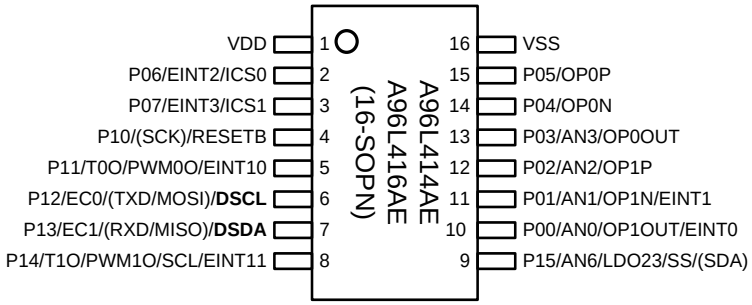
In this chapter, A96L414/A96L416 pinouts and pin descriptions are introduced.

2.1 Pinouts



NOTE: For the On-Chip Debugging, ISP assigns P1[2:3] pins for DSCL and DSDA respectively.

Figure 3. A96L414FR/A96L416FR 20 TSSOP Pinouts



NOTES:

1. For the On-Chip Debugging, ISP assigns P1[2:3] pins for DSCL and DSDA respectively.
2. The P20-P23 pins should be selected as a push-pull output or an input with pull-up resistor by software when the 16-pin package is used.

Figure 4. A96L414AE/A96L416AE 16 SOPN Pinouts

2.2 Pin description

Table 2. 20 TSSOP Pin Description

Pin name	I/O	Function	@reset	Shared with
P00	I/O	The port 0 is a bit-programmable I/O port which can be configured as an input (P00/P01/P06/P07: Schmitt trigger input), a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	AN0/OP1OUT/EINT0
P01				AN1/OP1N/EINT1
P02				AN2/OP1P
P03				AN3/OP0OUT
P04				AN4/OP0N
P05				AN5/OP0P
P06				EINT2/ICS0
P07				EINT3/ICS1
P10	I/O	The port 1 is a bit-programmable I/O port which can be configured as a Schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	(SCK)/RESETB
P11				T0O/PWM0O/EINT10
P12				EC0/(TXD/MOSI)/DSCL
P13				EC1/(RXD/MISO)/DSDA
P14				T1O/PWM1O/SCL/EINT11
P15				AN6/LDO23/SS/(SDA)
P20	I/O	The port 2 is a bit-programmable I/O port which can be configured as a Schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	T2O/PWM2O/SDA/EINT12
P21				EC2/SCK/(SCL)
P22				AN4/RXD/MISO
P23				AN5/TXD/MOSI
EINT0	I/O	External interrupt inputs	Input	P00/AN0/OP1OUT
EINT1				P01/AN1/OP1N
EINT2				P06/ICS0
EINT3				P07/ICS1
EINT10	I/O	External interrupt input and Timer 0 capture input	Input	P11/T0O/PWM0O
EINT11	I/O	External interrupt input and Timer 1 capture input		P14/T1O/PWM1O/SCL
EINT12	I/O	External interrupt input and Timer 2 capture input		P20/T2O/PWM2O/SDA
T0O	I/O	Timer 0 interval output	input	P11/PWM0O/EINT10
T1O	I/O	Timer 1 interval output		P14/PWM1O/SCL/EINT11

Table 2. 20 TSSOP Pin Description (continued)

Pin name	I/O	Function	@reset	Shared with
T2O	I/O	Timer 2 interval output		P20/PWM2O/SDA/EINT12
PWM0O	I/O	Timer 0 pulse output		P11/T0O/EINT10
PWM1O	I/O	Timer 1 pulse output		P14/T1O/SCL/EINT11
PWM2O	I/O	Timer 2 pulse output		P20/T2O/SDA/EINT12
EC0	I/O	Timer 0 event count input		P12/(TXD/MOSI)/DSCL
EC1	I/O	Timer 1 event count input		P13/(RXD/MISO)/DSDA
EC2	I/O	Timer 2 event count input		P21/SCK/(SCL)
AN0	I/O	A/D converter analog input channels	Input	P00/OP1OUT/EINT0
AN1				P01/OP1N/EINT1
AN2				P02/OP1P
AN3				P03/OP0OUT
AN4				P22/RXD/MISO
AN5				P23/TXD/MOSI
AN6				P15/LDO23/SS/(SDA)
LDO23	I/O	LDO voltage output	Input	P15/AN6/SS/(SDA)
OP0P	I/O	OP-AMP 0 positive input	Input	P05
OP0N	I/O	OP-AMP 0 negative input	Input	P04
OP0OUT	I/O	OP-AMP 0 output	Input	P03/AN3
OP1P	I/O	OP-AMP 1 positive input	Input	P02/AN2
OP1N	I/O	OP-AMP 1 negative input	Input	P01/AN1/EINT1
OP1OUT	I/O	OP-AMP 1 output	Input	P00/AN0/EINT0
TXD	I/O	UART data output	Input	P23/AN5/MOSI (P12/EC0/MOSI/DSCL)
RXD	I/O	UART data input	Input	P22/AN4/MISO (P13/EC1/MISO/DSDA)
MOSI	I/O	SPI master output, slave input	Input	P23/AN4/RXD (P12/EC0/TXD/DSCL)
MISO	I/O	SPI master input, slave output	Input	P22/AN4/RXD (P13/EC1/RXD/DSDA)
SCK	I/O	SPI clock input/output	Input	P21/EC2(SCL) (P10/RESETB)
SS	I/O	SPI slave select input	Input	P15/AN6/LDO23/(SDA)
ICS0	I/O	Constant sink current pins	Input	P06/EINT2
ICS1				P07/EINT3

Table 2. 20 TSSOP Pin Description (continued)

Pin name	I/O	Function	@reset	Shared with
SCL	I/O	I2C clock input/output	Input	P14/T1O/PWM1O/SCL (P21/EC2/SCK)
SDA	I/O	I2C data input/output	Input	P20/T2O/PWM2O/SDA (P15/AN6/LDO23/SS)
RESETB	I/O	System reset pin with a pull-up resistor when it is selected as the RESETB by CONFIGURE OPTION.	Input	P10/SCK
DSCL	I/O	On chip debugger clock input	Input	P12/EC0/(TXD/MOSI)
DSDA	I/O	On chip debugger data input/output	Input	P13/EC1/(RXD/MISO)
VDD, VSS	_	Power input pins	–	–

NOTES:

1. The P10/RESETB pin is configured as one of the P10/SCK and the RESETB pin by the "CONFIGURE OPTION".
2. If the P13/DSDA and P12/DSCL pins are connected to an emulator during reset or power-on reset, the pins are automatically configured as the debugger pins.
3. The P13/DSDA and P12/DSCL pins are configured as inputs with an internal pull-up resistor only during the reset or power-on reset.

3 Port structures

3.1 GPIO port structure

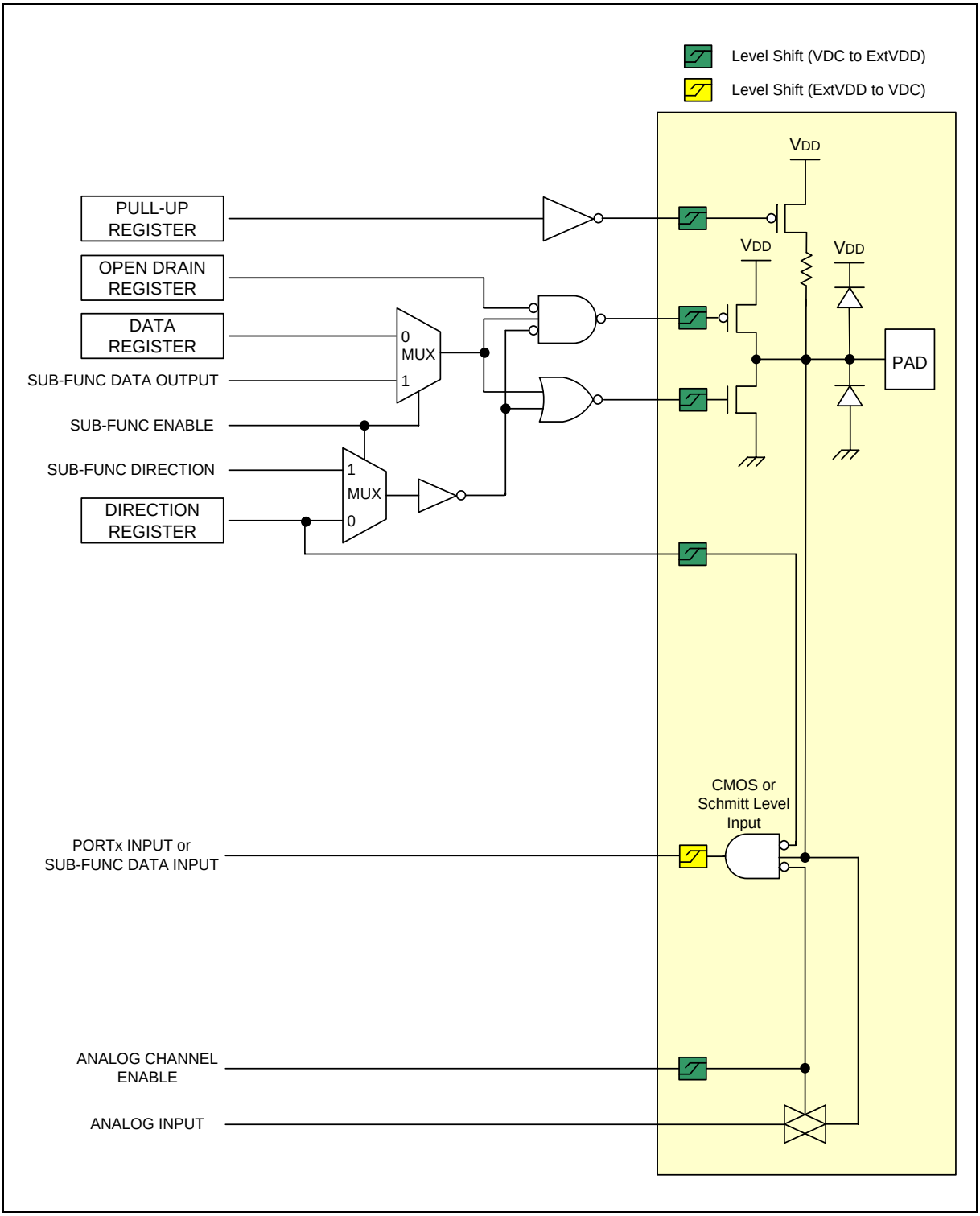


Figure 5. General Purpose I/O Port Structure

3.2 External interrupt I/O port structure

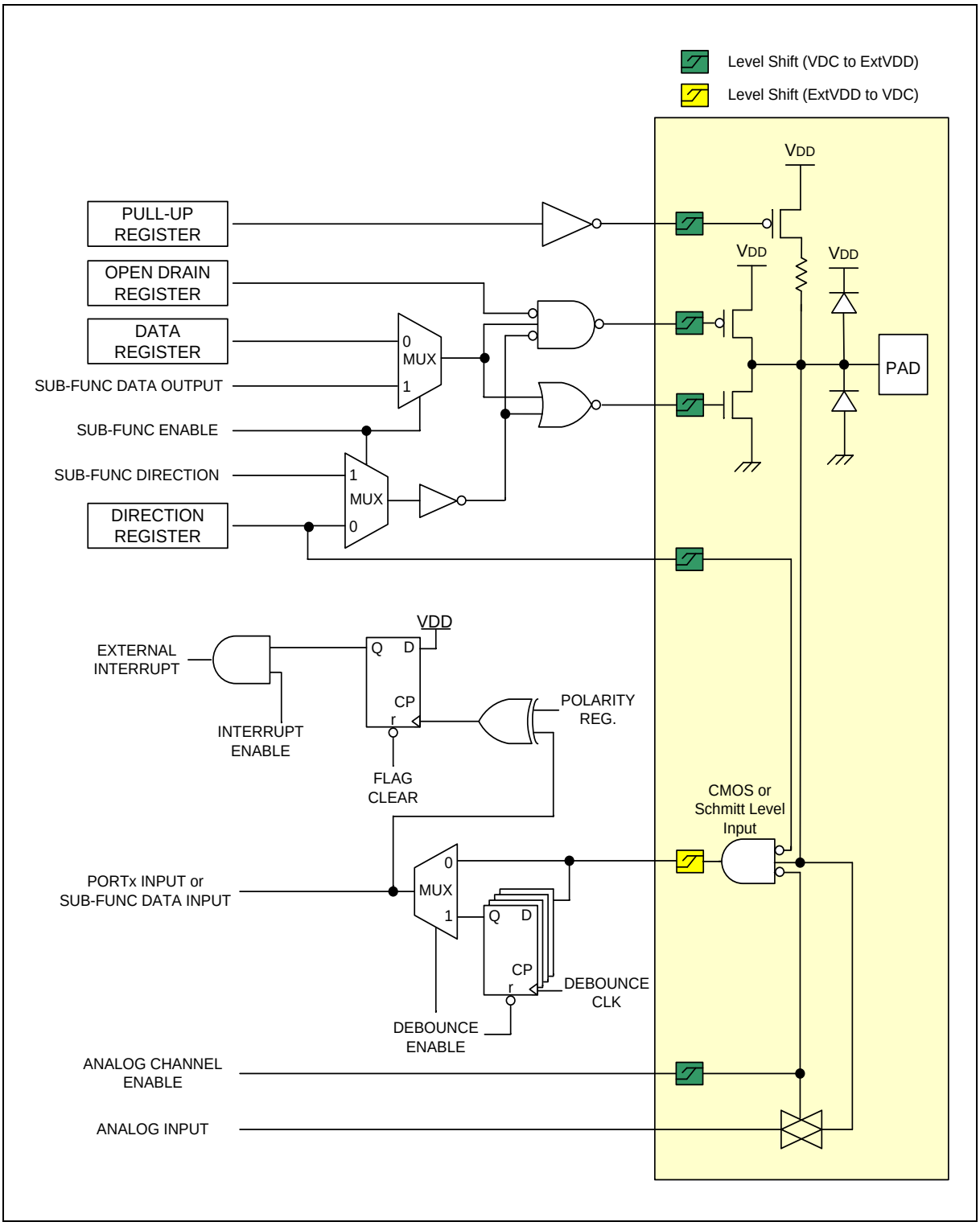


Figure 6. External Interrupt I/O Port Structure

4 Memory organization

A96L414/A96L416 addresses three separate memory spaces:

- Program memory
- Data memory
- XRAM memory

By means of this logical separation of the memory, 8-bit CPU address can access the data memory more rapidly. 16-bit data memory address is generated through the DPTR register.

A96L414/A96L416 provides on-chip 8/16 Kbytes of the ISP type Flash program memory, which is readable and writable. Internal data memory (iRAM) is 256 bytes and it includes the stack area. External data memory (XRAM) is 256/768 bytes.

4.1 Program memory

A 16-bit program counter is capable of addressing up to 64 Kbytes, but A96L414/A96L416 has only 8/16 Kbytes program memory space. After reset, CPU begins execution from location 0000H. Each interrupt is assigned to a fixed location of the program memory. The interrupt causes the CPU to jump to that location, where it commences an execution of a service routine.

For example, an external interrupt 1 is assigned to location 000BH. If the external interrupt 1 is going to be used, its service routine must begin at location 000BH. If the interrupt is not going to be used, its service location is available as general purpose program memory. If an interrupt service routine is short enough (frequent cases with a control application), the service routine can reside entirely within an 8 byte interval.

A longer service routine can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use. Figure 7 shows a map of the lower part of the program memory.

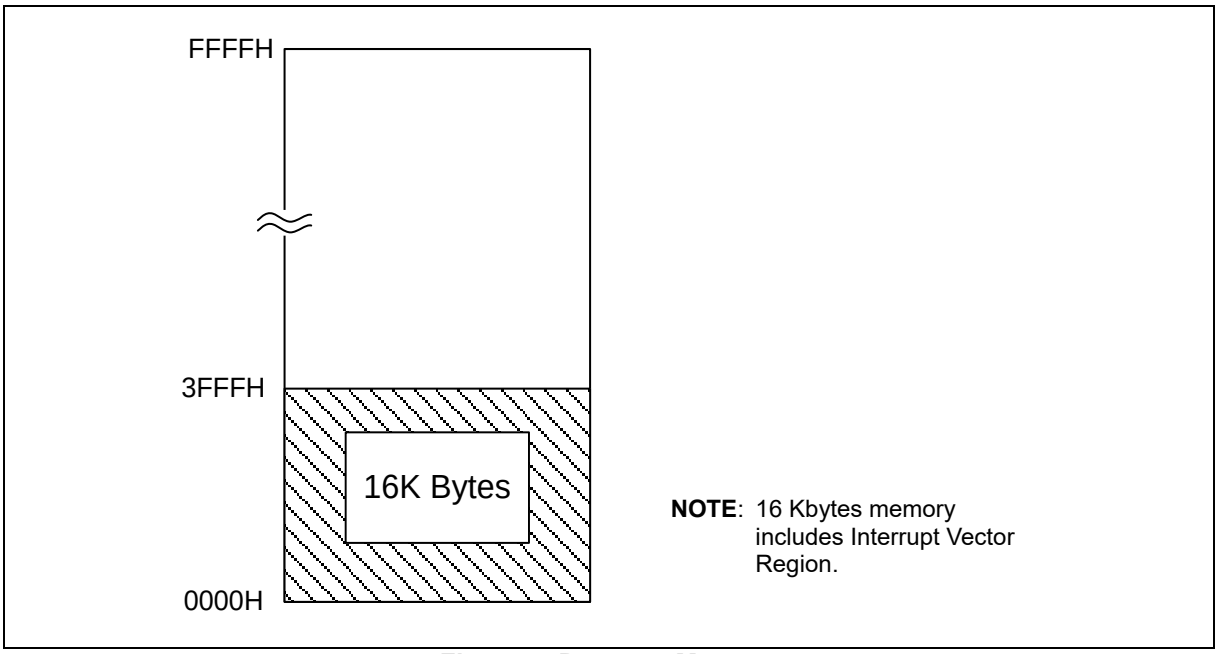


Figure 7. Program Memory

More detailed description of program memory is introduced in chapter 19. Flash memory later part in this document.

4.2 Internal data memory

Internal data memory is divided into three spaces as shown in Figure 8. Those three spaces are generally called as,

- Lower 128 bytes
- Upper 128 bytes
- Special Function Registers (SFR space)

Internal data memory addresses are always one byte wide, which implies an address space of 256 bytes.

In fact, the addressing modes of the internal data memory can accommodate up to 384 bytes by using a simple trick. Direct addresses higher than 7FH access one memory space and indirect addresses higher than 7FH access a different memory space. By means of this method, the upper 128 bytes and SFR space can occupy the same block of addresses, 80H through FFH, although they are physically separate entities as shown in Figure 8.

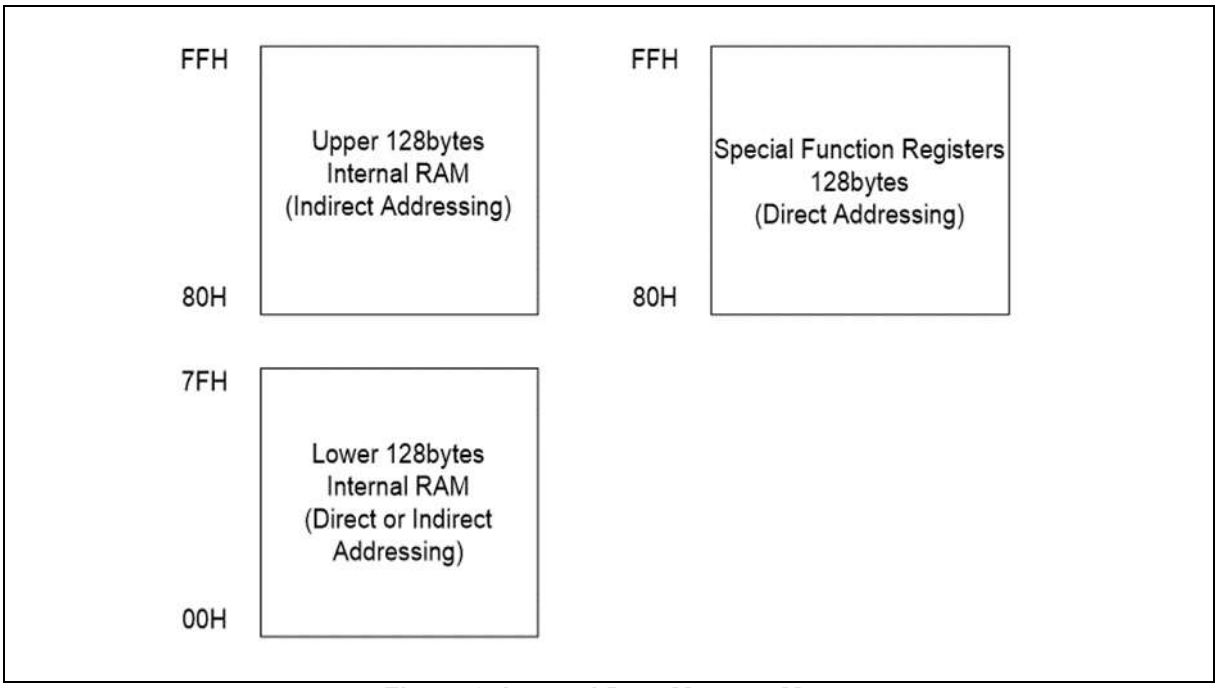


Figure 8. Internal Data Memory Map

The lower 128 bytes of RAM are present in all 8051 devices as mapped in Figure 9. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The 8051 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

Entire bytes in the lower 128 bytes can be accessed by either direct or indirect addressing, while the upper 128 bytes RAM can only be accessed by indirect addressing. These spaces are used for data RAM and stack.

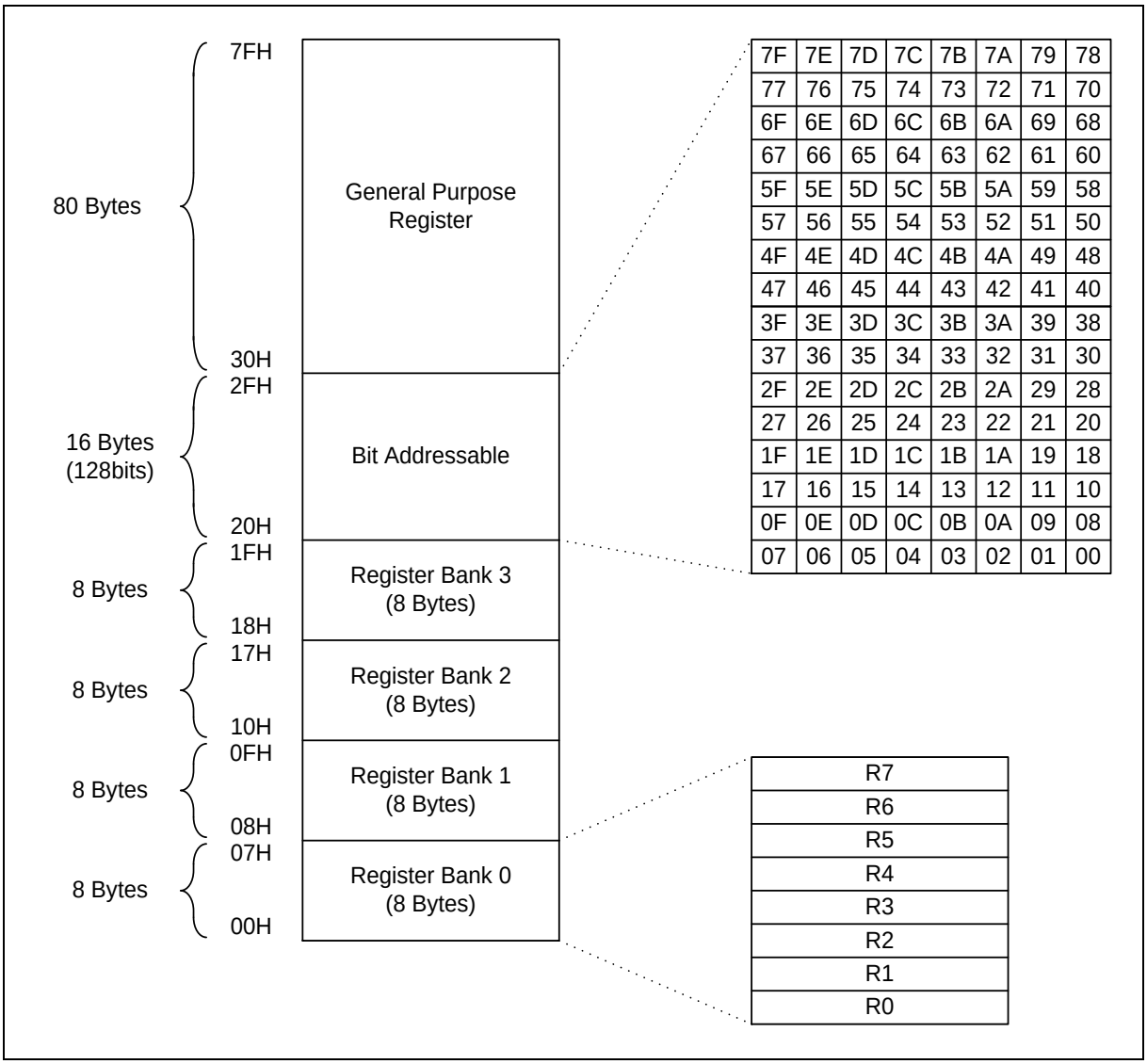


Figure 9. Lower 128 bytes Internal RAM

4.3 Extended SFR and data memory area

A96L414/A96L416 has 256/768 bytes XRAM and XSFR registers. Extended SFR area has no relation with RAM nor FLASH. This area can be read or written to by using SFR in 8-bit unit.

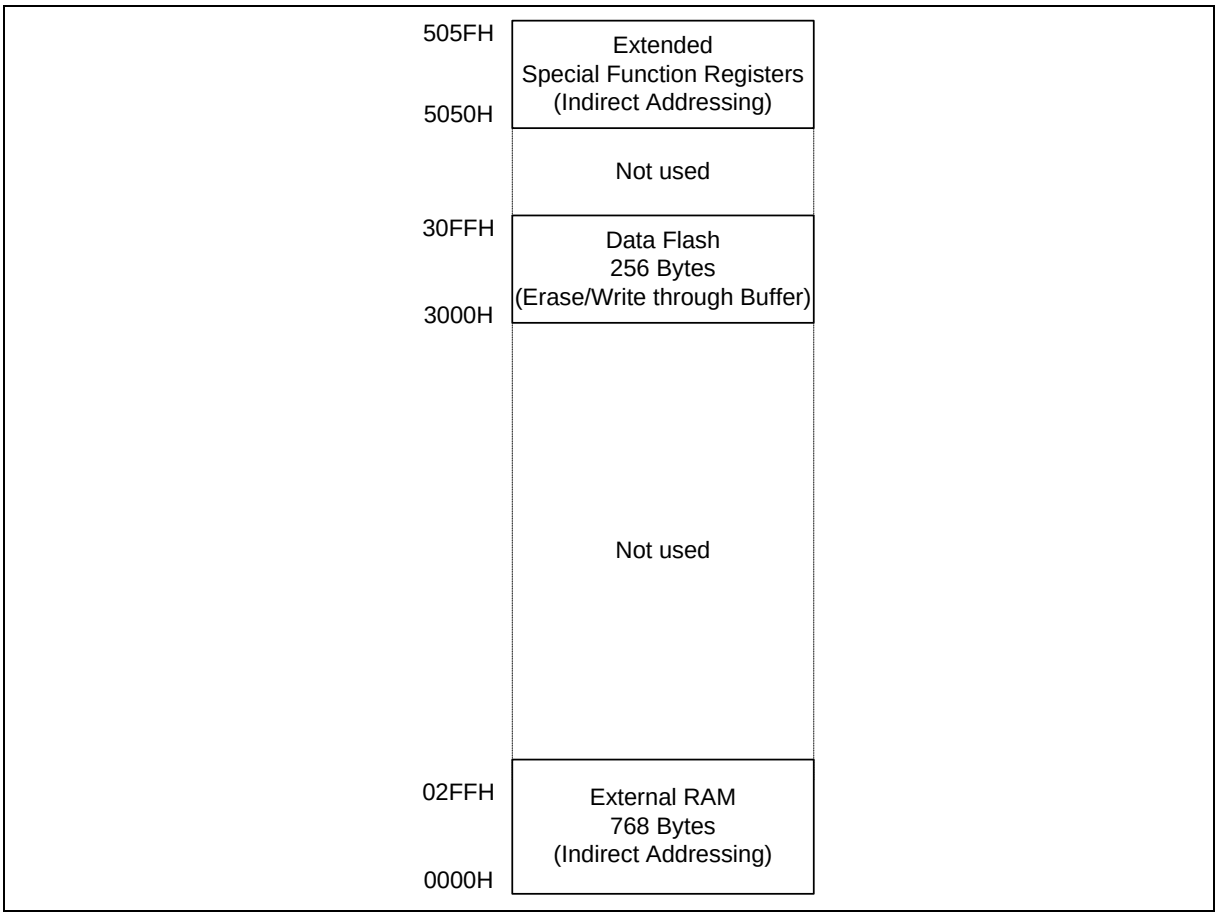


Figure 10. Extended SFR (XSFR) Area

4.4 Data Flash area

Data Flash area has no relation with RAM nor FLASH. This area can be read by using DPTR. Data Flash area can be erased or written to by using a buffer.

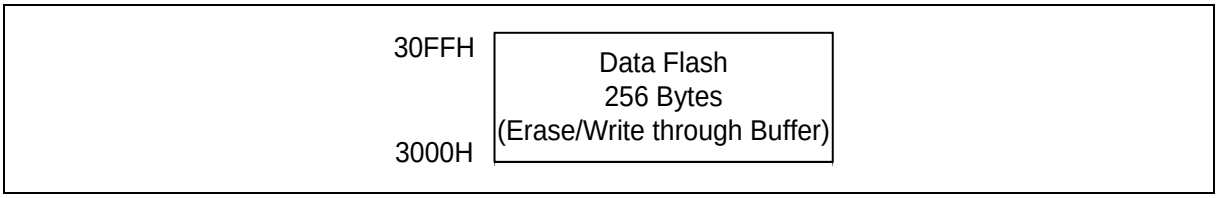


Figure 11. Data Flash Area

Detailed information about the Data Flash, please refer to Chapter 20 Data Flash memory.

4.5 SFR map

In this section, information of SFR map and map summaries are introduced through Table 3, Table 4, Table 5, and Table 6.

4.5.1 SFR map summary

Table 3. SFR Map Summary

– Reserved M8051 compatible

	00H/8H[NOTE]	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
0F8H	IP1	–	FSADRH	FSADRM	FSADRL	FIDR	FMCR	–
0F0H	B	I2CSAR1	DFSADRL	DFSADRH	DFIDR	DFMCR	–	–
0E8H	RSTFR	I2CCR	I2CSR	I2CSAR0	I2CDR	I2CSDHR	I2CSCLR	I2CSCHR
0E0H	ACC	–	ICSCR	ICSDR0	ICSDR1	–	–	–
0D8H	LVRCR	–	USTCR1	USTCR2	USTCR3	USTST	USTBD	USTDR
0D0H	PSW	–	P2OD	P2PU	P2FSR	–	–	FCDIN
0C8H	OSCCR	–	ADCCRL	ADCCRH	ADCDRL	ADCDRH	LDOCR	–
0C0H	–	–	T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
0B8H	IP	–	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
0B0H	–	–	T0CRL	T0CRH	T0ADRL	T0ADRH	T0BDRL	T0BDRH
0A8H	IE	IE1	IE2	IE3	–	CHPCR	AMPCR0	AMPCR1
0A0H	EIFLAG	P2IO	EO	–	EIPOL0	EIPOL1	–	–
98H	–	P1IO	P1OD	P1PU	P1FSRL	P1FSRH	P12DB	IRCIDR
90H	P2	P0IO	P0OD	P0PU	P0FSRL	P0FSRH	P0DB	IRCTRM
88H	P1	–	SCCR	BITCR	BITCNT	WDTCR	WDTDR/ WDTCNT	IRCTCR
80H	P0	SP	DPL	DPH	DPL1	DPH1	–	PCON

NOTE: Registers 00H/8H are bit-addressable.

4.5.2 Extended SFR map summary

Table 4. XSFR Map Summary

– Reserved

	00H/8H[NOTE]	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
5058H	FCDRL	–	–	–	–	–	–	LVRIDR
5050H	FCSARH	FCEARH	FCSARM	FCEARM	FCSARL	FCEARL	FCCR	FCDRH

NOTE: Registers 00H/8H are bit-addressable.

4.5.3 SFR map

Table 5. SFR Map

Address	Function	Symbol	R/W	@ Reset							
				7	6	5	4	3	2	1	0
80H	P0 Data Register	P0	R/W	0	0	0	0	0	0	0	0
81H	Stack Pointer	SP	R/W	0	0	0	0	0	1	1	1
82H	Data Pointer Register Low	DPL	R/W	0	0	0	0	0	0	0	0
83H	Data Pointer Register High	DPH	R/W	0	0	0	0	0	0	0	0
84H	Data Pointer Register Low 1	DPL1	R/W	0	0	0	0	0	0	0	0
85H	Data Pointer Register High 1	DPH1	R/W	0	0	0	0	0	0	0	0
86H	Reserved	–	–	–							
87H	Power Control Register	PCON	R/W	–	–	–	–	–	–	0	0
88H	P1 Data Register	P1	R/W	–	–	0	0	0	0	0	0
89H	Reserved	–	–	–							
8AH	System and Clock Control Register	SCCR	R	–	–	–	–	–	–	0	0
8BH	Basic Interval Timer Control Register	BITCR	R/W	0	0	0	–	0	0	0	1
8CH	Basic Interval Timer Counter Register	BITCNT	R	0	0	0	0	0	0	0	0
8DH	Watchdog Timer Control Register	WDTCR	R/W	0	0	0	–	–	0	0	0
8EH	Watchdog Timer Data Register	WDTDR	W	1	1	1	1	1	1	1	1
8EH	Watchdog Timer Counter Register	WDTCNT	R	0	0	0	0	0	0	0	0
8FH	Internal RC Trim Control Register	IRCTCR	R/W	0	0	0	0	0	0	0	0
90H	P2 Data Register	P2	R/W	–	–	–	–	0	0	0	0
91H	P0 Direction Register	P0IO	R/W	0	0	0	0	0	0	0	0
92H	P0 Open-drain Selection Register	P0OD	R/W	0	0	0	0	0	0	0	0
93H	P0 Pull-up Resistor Selection Register	P0PU	R/W	0	0	0	0	0	0	0	0
94H	Port 0 Function Selection Low Register	P0FSRL	R/W	0	0	0	0	0	0	0	0
95H	Port 0 Function Selection High Register	P0FSRH	R/W	0	0	0	0	0	0	0	0
96H	P0 Debounce Enable Register	P0DB	R/W	0	0	0	0	0	0	0	0

Table 5. SFR Map (continued)

Address	Function	Symbol	R/W	@ Reset							
				7	6	5	4	3	2	1	0
97H	Internal RC Trim Register	IRCTRM	R/W	x	x	x	x	x	x	x	x
98H	Reserved	–	–	–							
99H	P1 Direction Register	P1IO	R/W	–	–	0	0	0	0	0	0
9AH	P1 Open-drain Selection Register	P1OD	R/W	–	–	0	0	0	0	0	0
9BH	P1 Pull-up Resistor Selection Register	P1PU	R/W	–	–	0	0	0	0	0	0
9CH	Port 1 Function Selection Low Register	P1FSRL	R/W	–	0	–	0	–	0	–	0
9DH	Port 1 Function Selection High Register	P1FSRH	R/W	–	–	–	–	0	0	0	0
9EH	P1/P2 Debounce Enable Register	P12DB	R/W	–	–	0	0	–	0	0	0
9FH	Internal RC Trim Identification Register	IRCIDR	R/W	0	0	0	0	0	0	0	0
A0H	External Interrupt Flag Register	EIFLAG	R/W	0	0	0	0	0	0	0	0
A1H	P2 Direction Register	P2IO	R/W	–	–	–	–	0	0	0	0
A2H	Extended Operation Register	EO	R/W	–	–	–	0	–	0	0	0
A3H	Reserved	–	–	–							
A4H	External Interrupt Polarity 0 Register	EIPOL0	R/W	0	0	0	0	0	0	0	0
A5H	External Interrupt Polarity 1 Register	EIPOL1	R/W	–	–	0	0	0	0	0	0
A6H	Reserved	–	–	–							
A7H	Reserved	–	–	–							
A8H	Interrupt Enable Register	IE	R/W	0	–	0	0	0	0	0	0
A9H	Interrupt Enable Register 1	IE1	R/W	–	–	0	–	–	–	0	0
AAH	Interrupt Enable Register 2	IE2	R/W	–	–	0	0	–	0	0	0
ABH	Interrupt Enable Register 3	IE3	R/W	–	–	–	0	0	–	–	–
ACH	Reserved	–	–	–							
ADH	Chopper Control Register	CHPCR	R/W	–	–	–	–	–	–	0	0
AEH	OP-AMP Control Register 0	AMPCR0	R/W	–	0	–	–	–	–	0	0
AFH	OP-AMP Control Register 1	AMPCR1	R/W	0	0	0	0	0	–	0	0
B0H	Reserved	–	–	–							
B1H	Reserved	–	–	–							
B2H	Timer 0 Control Low Register	T0CRL	R/W	0	0	0	0	0	0	0	0

Table 5. SFR Map (continued)

Address	Function	Symbol	R/W	@ Reset							
				7	6	5	4	3	2	1	0
B3H	Timer 0 Control High Register	T0CRH	R/W	0	–	0	0	–	–	–	0
B4H	Timer 0 A Data Low Register	T0ADRL	R/W	1	1	1	1	1	1	1	1
B5H	Timer 0 A Data High Register	T0ADRH	R/W	1	1	1	1	1	1	1	1
B6H	Timer 0 B Data Low Register	T0BDRL	R/W	1	1	1	1	1	1	1	1
B7H	Timer 0 B Data High Register	T0BDRH	R/W	1	1	1	1	1	1	1	1
B8H	Interrupt Priority Register	IP	R/W	–	–	0	0	0	0	0	0
B9H	Reserved	–	–	–							
BAH	Timer 1 Control Low Register	T1CRL	R/W	0	0	0	0	0	0	0	0
BBH	Timer 1 Control High Register	T1CRH	R/W	0	–	0	0	–	–	–	0
BCH	Timer 1 A Data Low Register	T1ADRL	R/W	1	1	1	1	1	1	1	1
BDH	Timer 1 A Data High Register	T1ADRH	R/W	1	1	1	1	1	1	1	1
BEH	Timer 1 B Data Low Register	T1BDRL	R/W	1	1	1	1	1	1	1	1
BFH	Timer 1 B Data High Register	T1BDRH	R/W	1	1	1	1	1	1	1	1
C0H	Reserved	–	–	–							
C1H	Reserved	–	–	–							
C2H	Timer 2 Control Low Register	T2CRL	R/W	0	0	0	0	0	0	0	0
C3H	Timer 2 Control High Register	T2CRH	R/W	0	–	0	0	–	–	–	0
C4H	Timer 2 A Data Low Register	T2ADRL	R/W	1	1	1	1	1	1	1	1
C5H	Timer 2 A Data High Register	T2ADRH	R/W	1	1	1	1	1	1	1	1
C6H	Timer 2 B Data Low Register	T2BDRL	R/W	1	1	1	1	1	1	1	1
C7H	Timer 2 B Data High Register	T2BDRH	R/W	1	1	1	1	1	1	1	1
C8H	Oscillator Control Register	OSCCR	R/W	0	–	–	0	1	0	–	–
C9H	Reserved	–	–	–							
CAH	A/D Converter Control Low Register	ADCCRL	R/W	0	0	0	0	0	0	0	0
CBH	A/D Converter Control High Register	ADCCRH	R/W	0	–	–	–	0	0	0	0
CCH	A/D Converter Data Low Register	ADCDRL	R	x	x	x	x	x	x	x	x
CDH	A/D Converter Data High Register	ADCDRH	R	x	x	x	x	x	x	x	x
CEH	LDO Control Register	LDOCR	R/W	–	–	–	–	–	–	–	0
CFH	Reserved	–	–	–							
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0
D1H	Reserved	–	–	–							
D2H	P2 Open-drain Selection Register	P2OD	R/W	–	–	–	–	0	0	0	0

Table 5. SFR Map (continued)

Address	Function	Symbol	R/W	@ Reset							
				7	6	5	4	3	2	1	0
D3H	P2 Pull-up Resistor Selection Register	P2PU	R/W	–	–	–	–	0	0	0	0
D4H	Port 2 Function Selection Low Register	P2FSR	R/W	0	0	0	0	0	0	0	0
D5H	Reserved	–	–	–							
D6H	Reserved	–	–	–							
D7H	Flash CRC Data In Register	FCDIN	R/W	0	0	0	0	0	0	0	0
D8H	Low Voltage Reset Control Register	LVRCR	R/W	0	–	–	–	–	0	0	0
D9H	Reserved	–	–	–							
DAH	USART Control Register 1	USTCR1	R/W	0	0	0	0	0	0	0	0
DBH	USART Control Register 2	USTCR2	R/W	0	0	0	0	0	0	0	0
DCH	USART Control Register 3	USTCR3	R/W	0	0	0	0	0	0	0	0
DDH	USART Status Register	USTST	R/W	1	0	0	0	0	0	0	0
DEH	USART Baud Rate Generation Register	USTBD	R/W	1	1	1	1	1	1	1	1
DFH	USART Data Register	USTDR	R/W	0	0	0	0	0	0	0	0
E0H	Accumulator Register	ACC	R/W	0	0	0	0	0	0	0	0
E1H	Reserved	–	–	–							
E2H	Constant Sink Current Control Register	ICSCR	R/W	–	–	–	–	0	0	0	0
E3H	Constant Sink Current Data Register 0	ICSDR0	R/W	–	–	–	–	0	0	0	0
E4H	Constant Sink Current Data Register 1	ICSDR1	R/W	–	–	–	–	0	0	0	0
E5H	Reserved	–	–	–							
E6H	Reserved	–	–	–							
E7H	Reserved	–	–	–							
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	–	–	–
E9H	I2C Control Register	I2CCR	R/W	0	0	0	0	0	0	0	0
EAH	I2C Status Register	I2CSR	R/W	0	0	0	0	0	0	0	0
EBH	I2C Slave Address 0 Register	I2CSAR0	R/W	0	0	0	0	0	0	0	0
ECH	I2C Data Register	I2CDR	R/W	0	0	0	0	0	0	0	0
EDH	I2C SDA Hold Time Register	I2CSDHR	R/W	0	0	0	0	0	0	0	1

Table 5. SFR Map (continued)

Address	Function	Symbol	R/W	@ Reset							
				7	6	5	4	3	2	1	0
EEH	I2C SCL Low Period Register	I2CSCLR	R/W	0	0	1	1	1	1	1	1
EFH	I2C SCL High Period Register	I2CSCHR	R/W	0	0	1	1	1	1	1	1
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0
F1H	I2C Slave Address 1 Register	I2CSAR1	R/W	0	0	0	0	0	0	0	0
F2H	Data Flash Sector Address Low Register	DFSADRL	R/W	0	0	0	–	–	–	–	–
F3H	Data Flash Sector Address High Register	DFSADRH	R/W	0	0	0	0	0	0	0	0
F4H	Data Flash Identification Register	DFIDR	R/W	0	0	0	0	0	0	0	0
F5H	Data Flash Mode Control Register	DFMCR	R/W	0	–	–	–	–	0	0	0
F6H	Reserved	–	–	–							
F7H	Reserved	–	–	–							
F8H	Interrupt Priority Register 1	IP1	R/W	–	–	0	0	0	0	0	0
F9H	Reserved	–	–	–							
FAH	Flash Sector Address High Register	FSADRH	R/W	–	–	–	–	0	0	0	0
FBH	Flash Sector Address Middle Register	FSADRM	R/W	0	0	0	0	0	0	0	0
FCH	Flash Sector Address Low Register	FSADRL	R/W	0	0	0	0	0	0	0	0
FDH	Flash Identification Register	FIDR	R/W	0	0	0	0	0	0	0	0
FEH	Flash Mode Control Register	FMCR	R/W	0	–	–	–	–	0	0	0
FFH	Reserved	–	–	–							

4.5.4 Extended SFR map

Table 6. XSFR Map

Address	Function	Symbol	R/W	@ Reset							
				7	6	5	4	3	2	1	0
5050H	Flash CRC Start Address High Register	FCSARH	R/W	–	–	–	–	–	–	–	0
5051H	Flash CRC End Address High Register	FCEARH	R/W	–	–	–	–	–	–	–	0
5052H	Flash CRC Start Address Middle Register	FCSARM	R/W	0	0	0	0	0	0	0	0
5053H	Flash CRC End Address Middle Register	FCEARM	R/W	0	0	0	0	0	0	0	0
5054H	Flash CRC Start Address Low Register	FCSARL	R/W	0	0	0	0	–	–	–	–
5055H	Flash CRC End Address Low Register	FCEARL	R/W	0	0	0	0	1	1	1	1
5056H	Flash CRC Control Register	FCCR	R/W	0	0	0	–	0	0	0	0
5057H	Flash CRC Data High Register	FCDRH	R	1	1	1	1	1	1	1	1
5058H	Flash CRC Data Low Register	FCDRL	R	1	1	1	1	1	1	1	1
- -											
505FH	LVR Write Identification Register	LVRIDR	R/W	0	0	0	0	0	0	0	0

4.5.5 SFR map

ACC (Accumulator Register): E0H

7	6	5	4	3	2	1	0
ACC							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

ACC	Accumulator

B (B Register): F0H

7	6	5	4	3	2	1	0
B							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

B	B Register

SP (Stack Pointer): 81H

7	6	5	4	3	2	1	0
SP							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 07H

SP	Stack Pointer

DPL (Data Pointer Register Low): 82H

7	6	5	4	3	2	1	0
DPL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DPL	Data Pointer Low

DPH (Data Pointer Register High): 83H

7	6	5	4	3	2	1	0
DPH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DPH	Data Pointer High

DPL1 (Data Pointer Register Low 1): 84H

7	6	5	4	3	2	1	0
DPL1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DPL1	Data Pointer Low 1

DPH1 (Data Pointer Register High 1): 85H

7	6	5	4	3	2	1	0
DPH1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DPH1	Data Pointer High 1

PSW (Program Status Word Register): D0H

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

CY	Carry Flag
AC	Auxiliary Carry Flag
F0	General Purpose User-Definable Flag
RS1	Register Bank Select bit 1
RS0	Register Bank Select bit 0
OV	Overflow Flag
F1	User-Definable Flag
P	Parity Flag. Set/Cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator

EO (Extended Operation Register): A2H

7	6	5	4	3	2	1	0
–	–	–	TRAP_EN	–	DPSEL2	DPSEL1	DPSEL0
–	–	–	R/W	–	R/W	R/W	R/W

Initial value: 00H

TRAP_EN Select the Instruction **(Keep always '0')**.

0 Select MOVC @(DPTR++), A

1 Select Software TRAP Instruction

DPSEL[2:0] Select Banked Data Pointer Register

DPSEL2	DPSEL1	SPSEL0	Description
0	0	0	DPTR0
0	0	1	DPTR1
Reserved			

5 Ports

5.1 I/O ports

A96L414/A96L416 has three groups of I/O ports, P0, P1 and P2. Each port can be easily configured as an input pin, an output, or an internal pull up and open-drain pin by software. The port configuration pursues to meet various system configurations and design requirements. P0, P1 and P2 have a function generating interrupts in accordance with a change of state of the pin.

5.2 Port registers

5.2.1 Data register (Px)

Data register (Px) is related to a bidirectional I/O port. If a port is configured as an output port, data can be written to the corresponding bit of the Px. If a port is configured as an input, data can be read from the corresponding bit of the Px.

5.2.2 Direction register (PxIO)

Each I/O pin can be used as an input or an output independently by setting a PxIO register. If a bit is cleared in this read/write register, the corresponding pin of Px will be an input. While setting bits in this register will configure the corresponding pins to output.

Most bits are cleared by a system reset, but some bits are set by the system reset.

5.2.3 Pull-up register selection register (PxPU)

On-chip pull-up resistors can be connected to I/O ports individually by configuring a pull-up resistor selection register (PxPU). Setting a PxPU register can enable or disable a pull-up resister of each port. If a certain bit in PxPU register is 1, a pull-up resister of the corresponding pin is enabled. While the bit is 0, the pull-up resister is disabled. All bits are cleared by a system reset.

5.2.4 Open-drain selection register (PxOD)

There are internal open-drain selection registers (PxOD) for Px. Setting a PxOD register can enable or disable an open-drain of each port.

Most ports become push-pull by a system reset, but some ports become open-drain by the system reset.

5.2.5 Debounce enable register (P0DB, P12DB)

P00, P01, P06, P07, P11, P14, P20 and P21 support a debounce function. Debounce clocks of the ports are fx/1, fx/4, fx/16, and fx/64 respectively.

5.2.6 Port function selection register (P0FSRH, P0FSRL, P1FSRH, P1FSRL, P2FSR)

Port function selection registers define alternative functions of ports. Please remember that these registers must be set properly for alternative port functions. A reset clears the P0FSRH, P0FSRL, P1FSRH, P1FSRL and P2FSR register to '00H', which makes all pins to normal I/O ports.

5.2.7 Register map

Table 7. Port Register Map

Name	Address	Direction	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	91H	R/W	00H	P0 Direction Register
P0OD	92H	R/W	00H	P0 Open-drain Selection Register
P0PU	93H	R/W	00H	P0 Pull-up Resistor Selection Register
P0DB	96H	R/W	00H	P0 Debounce Enable Register
P0FSRH	95H	R/W	00H	P0 Function Selection High Register
P0FSRL	94H	R/W	00H	P0 Function Selection Low Register
P1	88H	R/W	00H	P1 Data Register
P1IO	99H	R/W	00H	P1 Direction Register
P1OD	9AH	R/W	00H	P1 Open-drain Selection Register
P1PU	9BH	R/W	00H	P1 Pull-up Resistor Selection Register
P12DB	9EH	R/W	00H	P1/P2 Debounce Enable Register
P1FSRH	9DH	R/W	00H	Port 1 Function Selection High Register
P1FSRL	9CH	R/W	00H	Port 1 Function Selection Low Register
P2	90H	R/W	00H	P2 Data Register
P2IO	A1H	R/W	00H	P2 Direction Register
P2OD	D2H	R/W	00H	P2 Open-drain Selection Register
P2PU	D3H	R/W	00H	P2 Pull-up Resistor Selection Register
P2FSR	D4H	R/W	00H	Port 2 Function Selection High Register

5.3 Port P0

5.3.1 Port description of P0

As an 8-bit I/O port, P0 controls the following registers:

- P0 data register (P0)
- P0 direction register (P0IO)
- P0 debounce enable register (P0DB)
- P0 pull-up resistor selection register (P0PU)
- P0 open-drain selection register (P0OD)
- P0 Function selection registers (P0FSRH/P0FSRL)

5.3.2 Register description of P0

P0 (P0 Data Register): 80H

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0[7:0] I/O Data

P0IO (P0 Direction Register): 91H

7	6	5	4	3	2	1	0
P07IO	P06IO	P05IO	P04IO	P03IO	P02IO	P01IO	P00IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0IO[7:0] P0 Data I/O Direction.
0 Input
1 Output
NOTE: EINT0/EINT1/EINT2/EINT3 function possible when input

P0PU (P0 Pull-up Resistor Selection Register): 93H

7	6	5	4	3	2	1	0
P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0PU[7:0] Configure Pull-up Resistor of P0 Port
0 Disable
1 Enable

P0OD (P0 Open-drain Selection Register): 92H

7	6	5	4	3	2	1	0
P07OD	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0OD[7:0] Configure Open-drain of P0 Port
- 0 Push-pull output
- 1 Open-drain output

P0DB (P0 Debounce Enable Register): 96H

7	6	5	4	3	2	1	0
DBCLK1	DBCLK0	–	–	P07DB	P06DB	P01DB	P00DB
R/W	R/W	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

DBCLK[1:0] Configure Debounce Clock of Port

DBCLK1	DBCLK0	Description
0	0	fx/(SCLK)
0	1	fx/4
1	0	fx/16
1	1	fx/64

P07DB Configure Debounce of P07 Port
- 0 Disable
- 1 Enable

P06DB Configure Debounce of P06 Port
- 0 Disable
- 1 Enable

P01DB Configure Debounce of P01 Port
- 0 Disable
- 1 Enable

P00DB Configure Debounce of P00 Port
- 0 Disable
- 1 Enable

NOTES:

1. If the same level is not detected on enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.
2. A pulse level should be input for the duration of 3 clocks or more to be actually detected as a valid edge.
3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.

P0FSRH (Port 0 Function Selection High Register): 95H

7	6	5	4	3	2	1	0
–	P0FSRH6	–	P0FSRH4	–	P0FSRH2	–	P0FSRH0
–	R/W	–	R/W	–	R/W	–	R/W

Initial value: 00H

P0FSRH6 P07 Function select

P0FSRH6	Description
0	I/O Port (EINT3 function possible when input)
1	ICS1 Function

P0FSRH4 P06 Function select

P0FSRH4	Description
0	I/O Port (EINT2 function possible when input)
1	ICS0 Function

P0FSRH2 P05 Function Select

P0FSRH2	Description
0	I/O Port
1	OP0P Function

P0FSRH0 P04 Function Select

P0FSRH0	Description
0	I/O Port
1	OP0N Function

NOTE: If OP-AMP0 is used, the P04 and P05 pins must be set to OP0N and OP0P functions regardless of using internal or external gain resistors.

P0FSRL (Port 0 Function Selection Low Register): 94H

7	6	5	4	3	2	1	0
P0FSRL7	P0FSRL6	P0FSRL5	P0FSRL4	P0FSRL3	P0FSRL2	P0FSRL1	P0FSRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0FSRL[7:6] P03 Function select

P0FSRL7	P0FSRL6	Description
0	0	I/O Port
0	1	OP0OUT Function
1	0	AN3 Function
1	1	Not used

P0FSRL[5:4] P02 Function Select

P0FSRL5	P0FSRL4	Description
0	0	I/O Port
0	1	OP1P Function
1	0	AN2 Function
1	1	Not used

P0FSRL[3:2] P01 Function select

P0FSRL3	P0FSRL2	Description
0	0	I/O Port (EINT1 function possible when input)
0	1	OP1N Function
1	0	AN1 Function
1	1	Not used

P0FSRL[1:0] P00 Function select

P0FSRL1	P0FSRL0	Description
0	0	I/O Port (EINT0 function possible when input)
0	1	OP1OUT Function
1	0	AN0 Function
1	1	Not used

NOTES:

1. If OP-AMP0 is used, the P03 pin must be set to OP0OUT function regardless of using internal or external gain resistors.
2. If OP-AMP1 is used, the P00, P01, and P02 pins must be set to OP1OUT, OP1N, and OP1P functions regardless of using internal or external gain resistors.

5.4 Port P1

5.4.1 Port description of P1

As a 6-bit I/O port, P1 controls the following registers:

- P1 data register (P1)
- P1 direction register (P1IO)
- P1 pull-up resistor selection register (P1PU)
- P1/P2 debounce enable register (P12DB)
- P1 open-drain selection register (P1OD)
- P1 Function selection registers (P1FSRH/P1FSRL)

5.4.2 Register description of P1

P1 (P1 Data Register): 88H

7	6	5	4	3	2	1	0
–	–	P15	P14	P13	P12	P11	P10
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P1[5:0] I/O Data

P1IO (P1 Direction Register): 99H

7	6	5	4	3	2	1	0
–	–	P15IO	P14IO	P13IO	P12IO	P11IO	P10IO
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P1IO[5:0] P1 Data I/O Direction.
0 Input
1 Output

NOTE: EINT10/EINT11/EC0/EC1/SS function possible when input

P1PU (P1 Pull-up Resistor Selection Register): 9BH

7	6	5	4	3	2	1	0
–	–	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P1PU[5:0] Configure Pull-up Resistor of P1 Port
0 Disable
1 Enable

P1OD (P1 Open-drain Selection Register): 9AH

7	6	5	4	3	2	1	0
–	–	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P1OD[5:0]	Configure Open-drain of P1 Port	
	0	Push-pull output
	1	Open-drain output

P12DB (P1/P2 Debounce Enable Register): 9EH

7	6	5	4	3	2	1	0
–	–	P21DB	P20DB	–	P14DB	P11DB	P10DB
–	–	R/W	R/W	–	R/W	R/W	R/W

Initial value: 00H

P21DB	Configure Debounce of P21 Port	
	0	Disable
	1	Enable
P20DB	Configure Debounce of P20 Port	
	0	Disable
	1	Enable
P14DB	Configure Debounce of P14 Port	
	0	Disable
	1	Enable
P11DB	Configure Debounce of P11 Port	
	0	Disable
	1	Enable
P10DB	Configure Debounce of P10 Port	
	0	Disable
	1	Enable

NOTES:

1. If the same level is not detected on enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.
2. A pulse level should be input for the duration of 3 clocks or more to be actually detected as a valid edge.
3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.
4. Refer to the port 0 debounce enable register (P0DB) for the debounce clock of port 1 and 2.

P1FSRH (Port 1 Function Selection High Register): 9DH

7	6	5	4	3	2	1	0
–	–	–	–	P1FSRH3	P1FSRH2	P1FSRH1	P1FSRH0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P1FSRH[3:2] P15 Function select

P1FSRH3	P1FSRH2	Description
0	0	I/O Port(SS function possible when input)
0	1	LDO23 Function
1	0	AN6 Function
1	1	(SDA) Function

P1FSRH[1:0] P14 Function select

P1FSRH1	P1FSRH0	Description
0	0	I/O Port (EINT11 function possible when input)
0	1	T1O/PWM1O Function
1	0	SCL Function
1	1	Not used

P1FSRL (Port 1 Function Selection Low Register): 9CH

7	6	5	4	3	2	1	0
–	P1FSRL6	–	P1FSRL4	–	P1FSRL2	–	P1FSRL0
–	R/W	–	R/W	–	R/W	–	R/W

Initial value: 00H

P1FSRL6 P13 Function select

0	I/O Port (EC1 function possible when input)
1	(RXD/MISO) Function

P1FSRL4 P12 Function select

0	I/O Port (EC0 function possible when input)
1	(TXD/MOSI) Function

P1FSRL2 P11 Function select

0	I/O Port (EINT10 function possible when input)
1	T0O/PWM0O Function

P1FSRL0 P10 Function select

0	I/O Port
1	(SCK) Function

NOTE: For more information settings of P10/RESETB, please refer to **Appendix: Configure option**.

5.5 Port P2

5.5.1 Port description of P2

As a 4-bit I/O port, P2 controls the following registers:

- P2 data register (P2)
- P2 direction register (P2IO)
- P2 pull-up resistor selection register (P2PU)
- P2 open-drain selection register (P2OD)
- P2 function selection register (P2FSR)

5.5.2 Register description of P2

P2 (P2 Data Register): 90H

7	6	5	4	3	2	1	0
–	–	–	–	P23	P22	P21	P20
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P2[3:0] I/O Data

P2IO (P2 Direction Register): A1H

7	6	5	4	3	2	1	0
–	–	–	–	P23IO	P22IO	P21IO	P20IO
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P2IO[3:0] P2 Data I/O Direction.
0 Input
1 Output

NOTE: EINT12 /EC2 function possible when input

P2PU (P2 Pull-up Resistor Selection Register): D3H

7	6	5	4	3	2	1	0
–	–	–	–	P23PU	P22PU	P21PU	P20PU
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P2PU[3:0] Configure Pull-up Resistor of P2 Port
0 Disable
1 Enable

P2OD (P2 Open-drain Selection Register): D2H

7	6	5	4	3	2	1	0
–	–	–	–	P23OD	P22OD	P21OD	P20OD
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

P2OD[3:0] Configure Open-drain of P2 Port
0 Push-pull output
1 Open-drain output

P2FSR (Port 2 Function Selection Register): D4H

7	6	5	4	3	2	1	0
P2FSR7	P2FSR6	P2FSR5	P2FSR4	P2FSR3	P2FSR2	P2FSR1	P2FSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P2FSR[7:6] P23 Function select

P2FSR7	P2FSR6	Description
0	0	I/O Port
0	1	TXD/MOSI Function
1	0	AN5 Function
1	1	Not used

P2FSR[5:4] P22 Function select

P2FSR5	P2FSR4	Description
0	0	I/O Port
0	1	RXD/MISO Function
1	0	AN4 Function
1	1	Not used

P2FSR[3:2] P21 Function select

P2FSR3	P2FSR2	Description
0	0	I/O Port(EC2 function possible when input)
0	1	SCK Function
1	0	(SCL) Function
1	1	Not used

P2FSR[1:0] P20 Function select

P2FSR1	P2FSR0	Description
0	0	I/O Port (EINT12 function possible when input)
0	1	T2O/PWM2O Function
1	0	SDA Function
1	1	Not used

6 Interrupt controller

Up to 16 interrupt sources are available in the A96L414/A96L416. Allowing software control, each interrupt source can be enabled by defining separate enable register bit associated with it. It can also have four levels of priority assigned. The non-maskable interrupt source is always enabled with a higher priority than any other interrupt sources, and is not controllable by software.

The interrupt controller features the followings:

- Receives requests from 16 interrupt sources
- 6 group priority
- 4 priority levels
- Multi interrupt possibility
- If requests of different priority levels are received simultaneously, a request with higher priority level is served first.
- Each interrupt source can be controlled by an EA bit and an IEx bit
- Interrupt latency varies ranging from 3 to 9 machine cycles in a single interrupt system.

Non-maskable interrupt is always enabled, while maskable interrupts can be enabled through four pairs of interrupt enable registers (IE, IE1, IE2, and IE3). Each bit of the four registers can individually enable or disable a particular interrupt source. Especially bit 7 (EA) in the register IE provides overall control. It must be set to '1' to enable interrupts as introduced in the followings:

- When EA is set to '0' ➔ All interrupts are disabled.
- When EA is set to '1' ➔ A particular interrupt can be individually enabled or disabled by the associate bit of the interrupt enable registers.

EA is always cleared to '0' jumping to an interrupt service vector and set to '1' executing the [RETI] instruction. A96L414/A96L416 supports a four-level priority scheme. Each maskable interrupt is individually assigned to one of the four levels according to IP and IP1.

Interrupt Group	Highest → Lowest			
0 (Bit0)	Interrupt 0	Interrupt 6	Interrupt 12	Interrupt 18
1 (Bit1)	Interrupt 1	Interrupt 7	Interrupt 13	Interrupt 19
2 (Bit2)	Interrupt 2	Interrupt 8	Interrupt 14	Interrupt 20
3 (Bit3)	Interrupt 3	Interrupt 9	Interrupt 15	Interrupt 21
4 (Bit4)	Interrupt 4	Interrupt 10	Interrupt 16	Interrupt 22
5 (Bit5)	Interrupt 5	Interrupt 11	Interrupt 17	Interrupt 23

Highest

Lowest

Figure 12. Interrupt Group Priority Level

Figure 12 introduces interrupt groups and their priority levels that is available for sharing interrupt priority. Priority of a group is set by 2 bits of Interrupt Priority (IP) registers: 1 bit from IP and another 1 bit from IP1.

Interrupt Service Routine serves an interrupt having higher priority first. If two requests of different priority levels are received simultaneously, the request with higher priority level is served prior to the lower one.

6.1 External interrupt

External interrupts on pins of INT0 to INT5 receive various interrupt requests in accordance with the external interrupt polarity 0 register (EIPOL0) and external interrupt polarity 1 register (EIPOL1) as shown in Figure 13. Each external interrupt source has enable/disable bits. An external interrupt flag register (EIFLAG) provides status of the external interrupts.

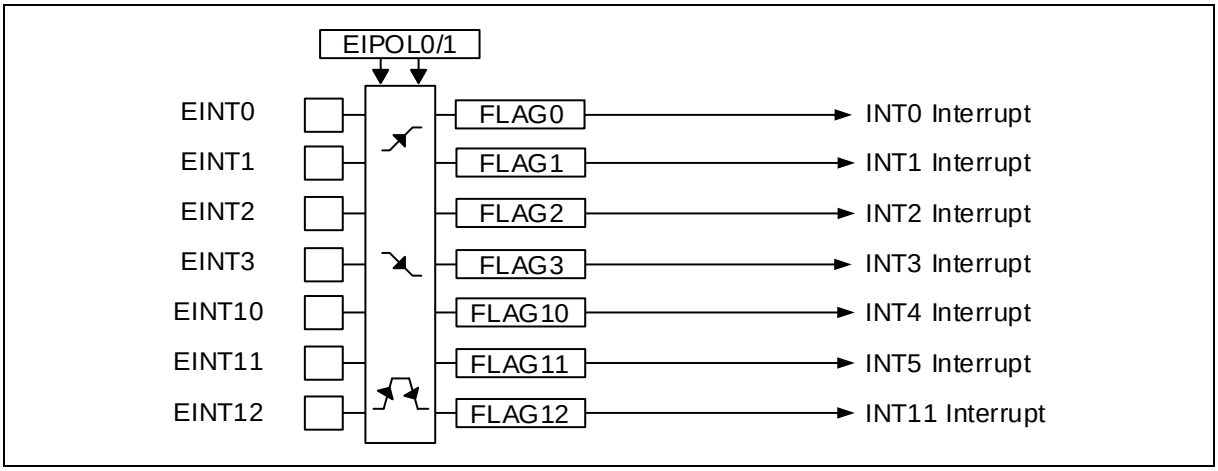


Figure 13. External Interrupt Description

6.2 Interrupt controller block diagram

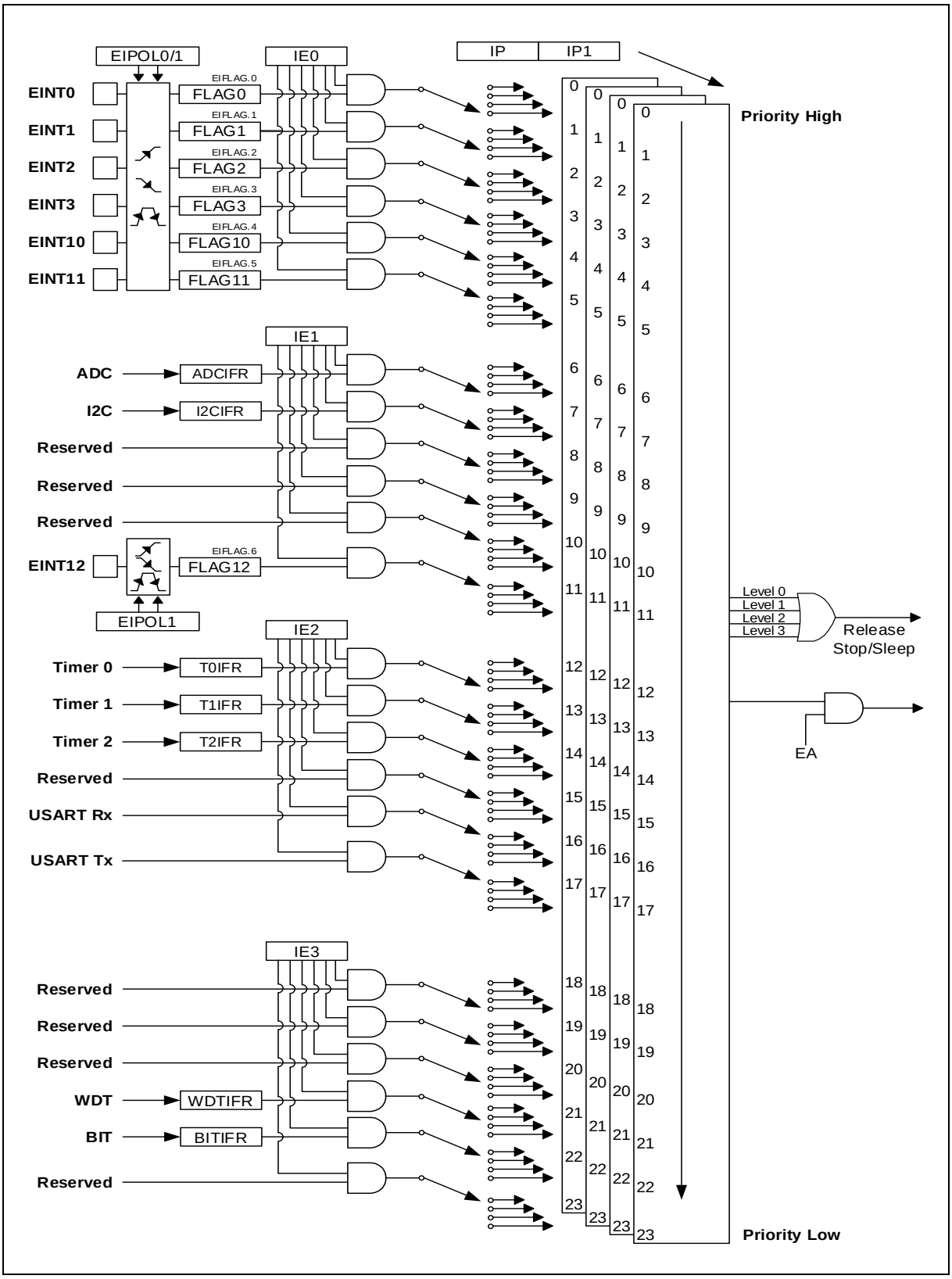


Figure 14. Interrupt Controller Block Diagram

In Figure 14, release signal for STOP and IDLE mode can be generated by all interrupt sources which are enabled without reference to priority level. An interrupt request will be delayed while data is written to one of the registers IE, IE1, IE2, IE3, IP, IP1, and PCON.

6.3 Interrupt vector table

When a certain interrupt occurs, a LCALL (Long Call) instruction pushes the contents of the PC (Program Counter) onto the stack, and loads the appropriate vector address. CPU pauses from its current task for some time and processes the interrupt at the vector address.

Interrupt controller supports 24 interrupt sources and each interrupt source has a determined priority order as shown in Table 8.

Table 8. Interrupt Vector Address Table

Interrupt source	Symbol	Interrupt Enable Bit	Priority	Mask	Vector address
Hardware RESET	RESETB	-	0	Non-Maskable	0000H
External Interrupt 0	INT0	IE.0	1	Maskable	0003H
External Interrupt 1	INT1	IE.1	2	Maskable	000BH
External Interrupt 2	INT2	IE.2	3	Maskable	0013H
External Interrupt 3	INT3	IE.3	4	Maskable	001BH
External Interrupt 10	INT4	IE.4	5	Maskable	0023H
External Interrupt 11	INT5	IE.5	6	Maskable	002BH
ADC Interrupt	INT6	IE1.0	7	Maskable	0033H
I2C Interrupt	INT7	IE1.1	8	Maskable	003BH
_	INT8	IE1.2	9	Maskable	0043H
_	INT9	IE1.3	10	Maskable	004BH
_	INT10	IE1.4	11	Maskable	0053H
External Interrupt 12	INT11	IE1.5	12	Maskable	005BH
T0 Interrupt	INT12	IE2.0	13	Maskable	0063H
T1 Interrupt	INT13	IE2.1	14	Maskable	006BH
T2 Interrupt	INT14	IE2.2	15	Maskable	0073H
_	INT15	IE2.3	16	Maskable	007BH
USART Rx Interrupt	INT16	IE2.4	17	Maskable	0083H
USART Tx Interrupt	INT17	IE2.5	18	Maskable	008BH
_	INT18	IE3.0	19	Maskable	0093H
_	INT19	IE3.1	20	Maskable	009BH
_	INT20	IE3.2	21	Maskable	00A3H
WDT Interrupt	INT21	IE3.3	22	Maskable	00ABH
BIT Interrupt	INT22	IE3.4	23	Maskable	00B3H
_	INT23	IE3.5	24	Maskable	00BBH

To execute the maskable interrupts, both EA bit and a corresponding bit of IEx associated with a specific interrupt source must be set to '1'. When an interrupt request is received, a particular interrupt request flag is set to '1' and maintains its status until CPU accepts the interrupt. After the interrupt acceptance, the interrupt request flag will be cleared automatically.

6.4 Interrupt sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to '0' by a reset or an instruction. The interrupt acceptance always happens at the last cycle of the instruction process. So rather than fetching the current instruction, CPU executes internally LCALL instruction and saves a PC onto the stack.

To begin an ISR (Interrupt Service Routine), the interrupt controller uses a branch instruction LJMP (Long Jump). The interrupt controller gives address of LJMP instruction to CPU. Since the end of the execution of current instruction, it needs 3~9 machine cycles to go to the interrupt service routine. The interrupt service task is terminated by the interrupt return instruction [RETI]. Once an interrupt request is generated, the following process is performed.

Table 9 introduces LJMP example code.

Table 9. LJMP Description and Example Code

<table>
<tr><td>Instruction</td><td>LJMP</td></tr>
<tr><td>Example code</td><td>LJMP 4000H

<table>
<tr><th>Address</th><th>Data</th><th>Instruction</th></tr>
<tr><td>1280H</td><td>02</td><td rowspan="3">LJMP 4000H</td></tr>
<tr><td>1281H</td><td>40</td></tr>
<tr><td>1282H</td><td>00</td></tr>
<tr><td>1283H</td><td>E4</td><td>CLR A</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>4000H</td><td>00</td><td>NOP</td></tr>
<tr><td>4001H</td><td>23</td><td>RL A</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
NOTE:
After finishing LJMP, NOP located at the address 400H will be executed as the next instruction.</td></tr>
</table>

Figure 15 shows a flow diagram of an ISR process.

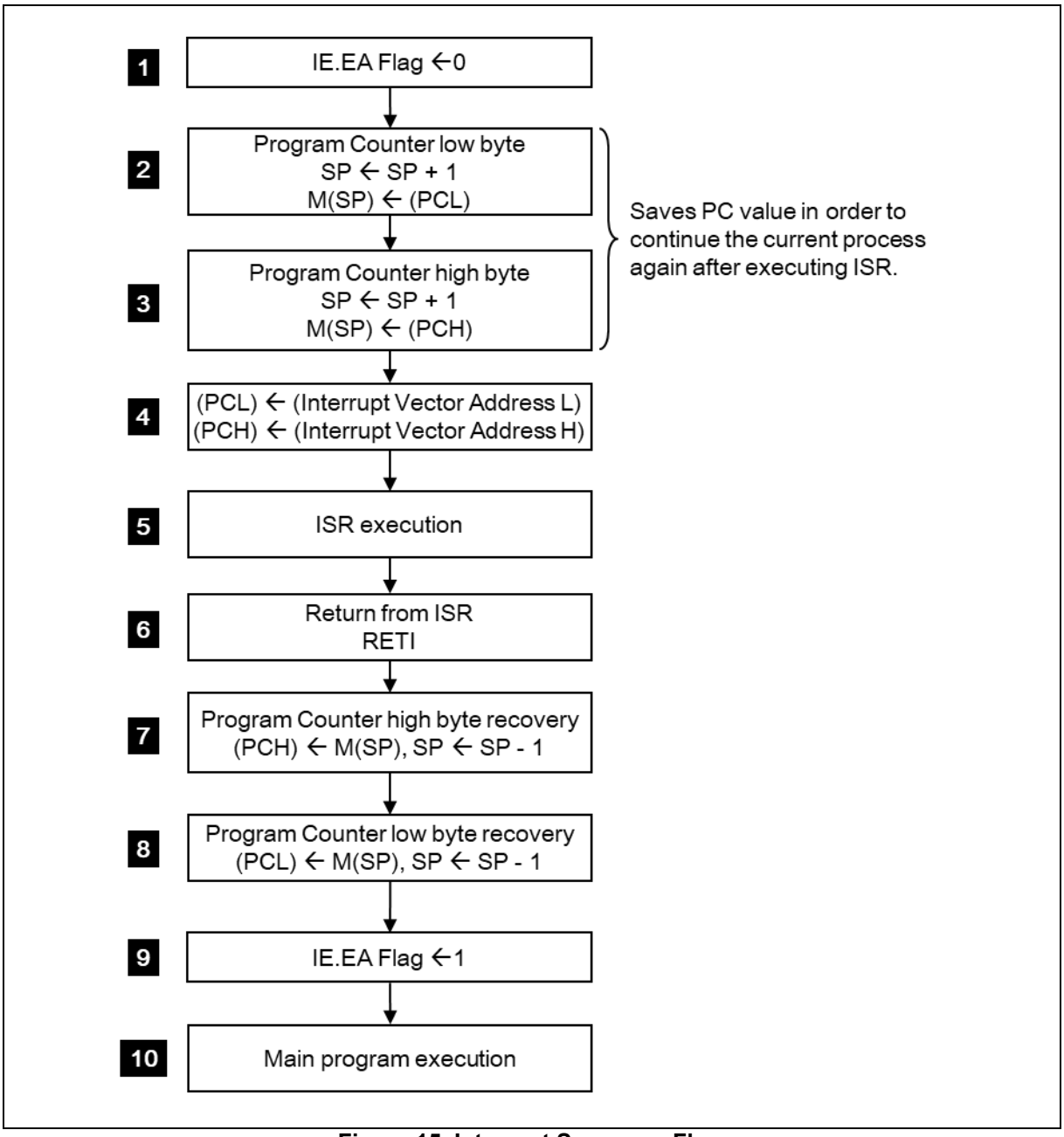


Figure 15. Interrupt Sequence Flow

6.5 Effective timing after controlling interrupt bit

Case A in Figure 16 shows an effective time of Control Interrupt Enable Register (IE, IE1, IE2, and IE3).

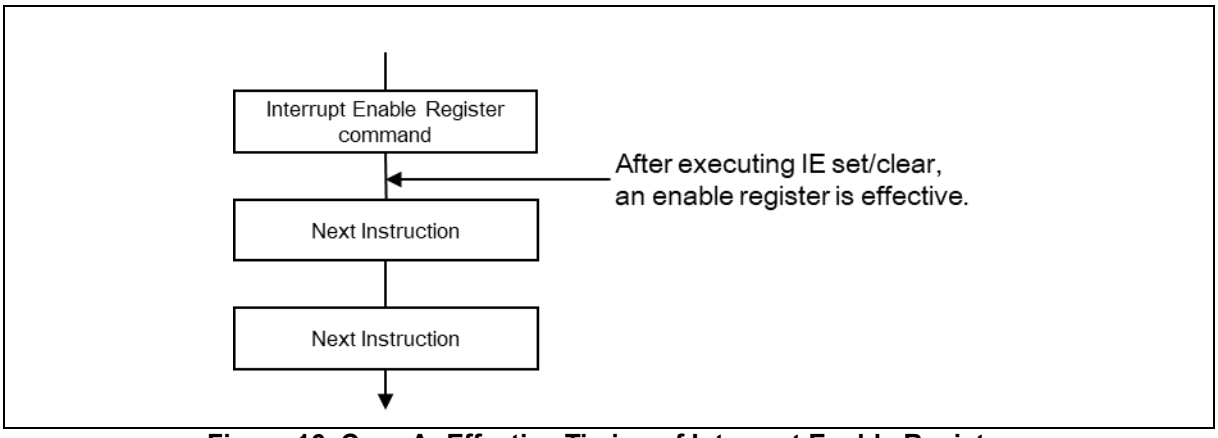


Figure 16. Case A: Effective Timing of Interrupt Enable Register

Case B in Figure 17 shows an effective time of Interrupt Flag Register.

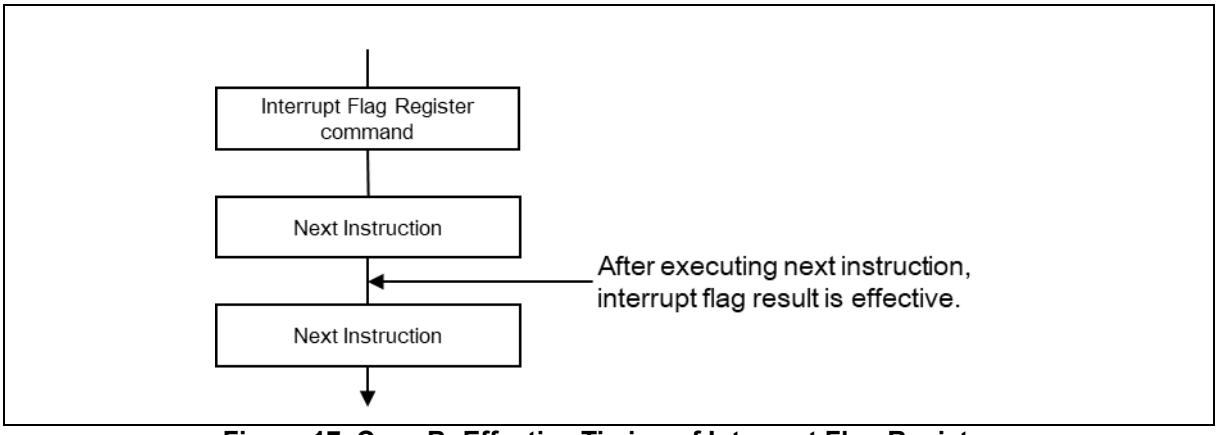


Figure 17. Case B: Effective Timing of Interrupt Flag Register

6.6 Multi interrupt

If two requests of different priority levels are received simultaneously, the request with higher priority level is served first. If more than one interrupt request are received, the interrupt polling sequence determines which request is served first by hardware. However, for special features, multi-interrupt processing can be executed by software.

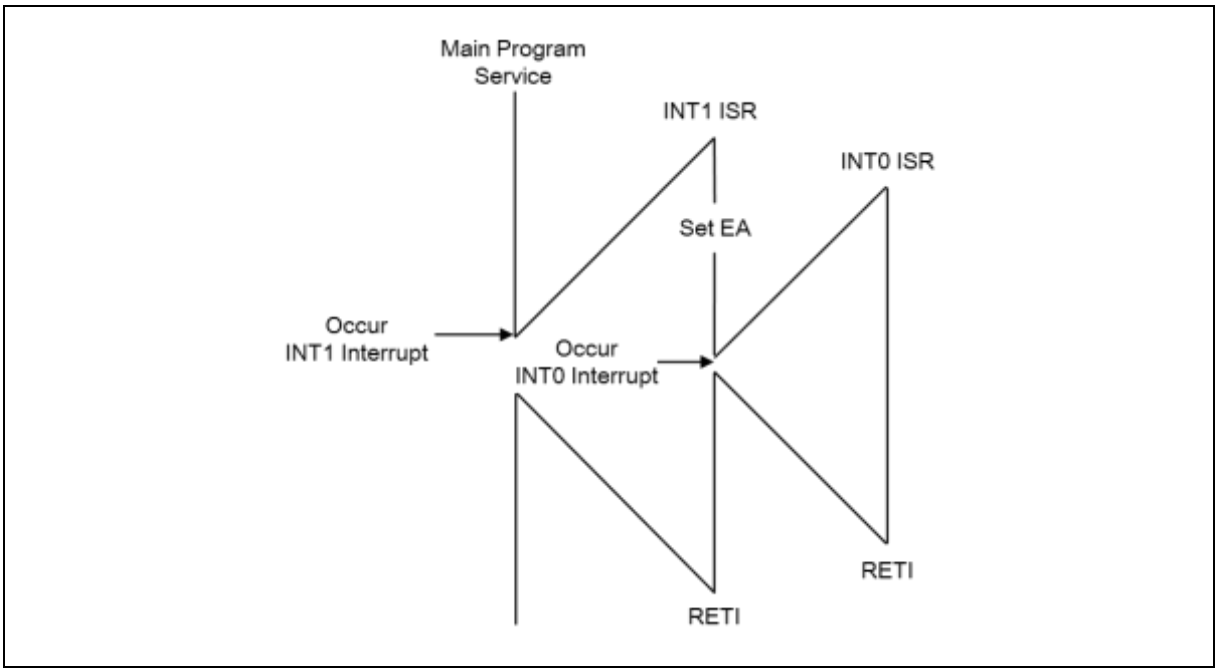


Figure 18. Effective Timing of Multi Interrupt

Figure 18 shows an example of multi-interrupt processing. While INT1 is served, INT0 which has higher priority than INT1 is occurred. Then INT0 is served immediately, then remain part of INT1 service routine is executed. If the priority level of INT0 is same or lower than INT1, INT0 will be served after the INT1 service has completed.

An interrupt service routine can be interrupted only by an interrupt with higher priority, and if two interrupts of different priority occur at the same time, the interrupt with higher priority level will be served first. An interrupt cannot be interrupted by another interrupt with the same or a lower priority level. If two interrupts having the same priority level occur simultaneously, the service order for those interrupts will be determined by the scan order.

6.7 Interrupt enable accept timing

Figure 19 implies that some period of time is required to response to the latched interrupt signal. In this figure, 4 machine cycles will be taken for the processes of LCALL and LJMP.

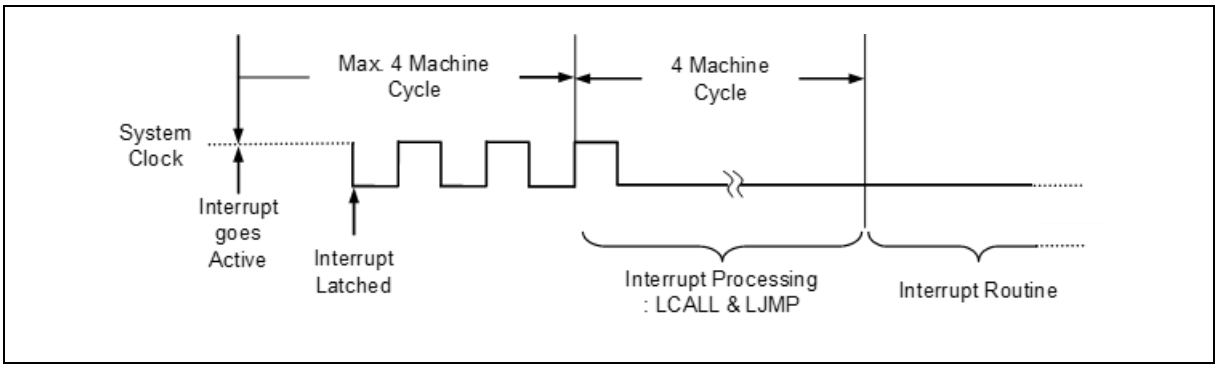


Figure 19. Interrupt Response Timing Diagram

6.8 Interrupt Service Routine address

As shown in Figure 20, ISR can be placed at any other location in program memory, and program memory must provide an unconditional jump to the starting address of ISR from the corresponding vector address.

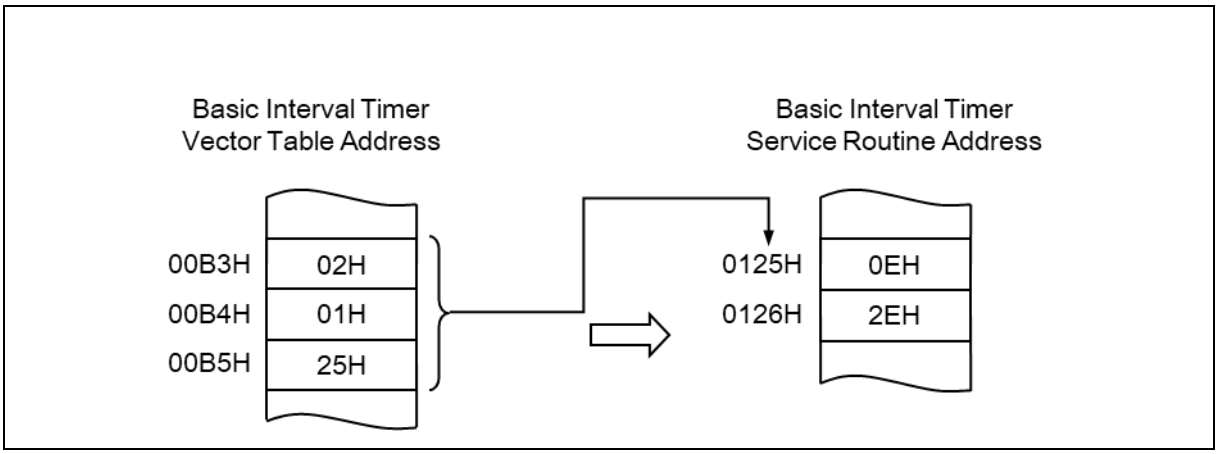


Figure 20. Correspondence between Vector Table Address and ISR Entry Address

6.9 Saving/ restore general-purpose registers

Let's assume there occurs an urgent condition. CPU needs to pause from its current task (Main Task in Figure 21) for some time to execute something else (Interrupt Service Task in Figure 21). After finishing the something else, CPU will return to the current task (Main Task).

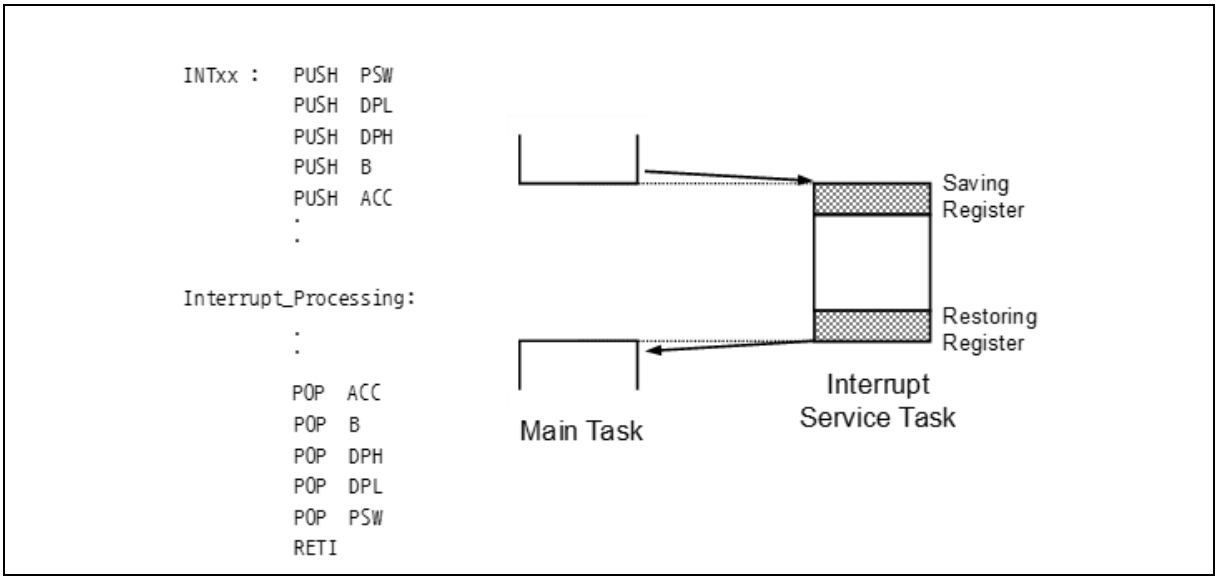


Figure 21. Saving and Restore Process Diagram and Example Code

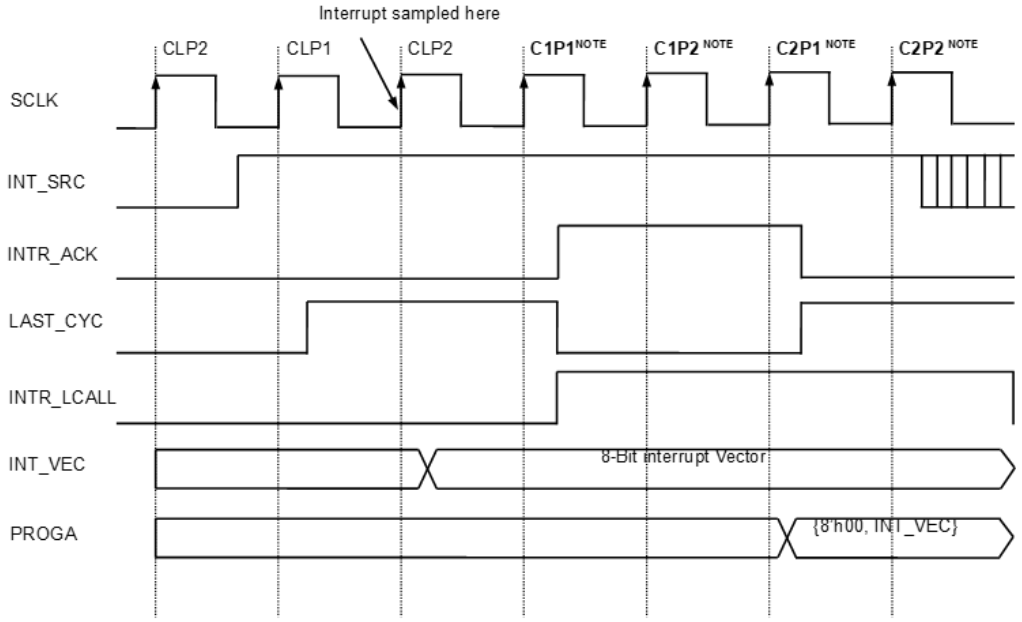
Example code in Figure 21 performs the followings:

1. Interrupt INTXX occurs.
2. PUSH PSW: the SP is incremented by one, and the value of the specified byte operand is stored at the internal RAM address indirectly referenced by the SP.
3. PUSH DPL, PUSH DPH: PSW in memory stack by help of PUSH instruction.
4. CPU stores low
5. CPU pops the value of flag register and stores it in register H by help of POP Instruction.

6.10 Interrupt timing

As seen in Figure 22 below, an interrupt source is sampled at the last cycle of a command. Upon the sampling, low 8-bit of interrupt vector is decided.

M8051W core makes the interrupt acknowledge at the first cycle of a command, executes LCALL instruction to jump interrupt routine at the address referenced by INT_VEC.



NOTE: Command cycle CLPx imply the followings:

- L ➔ Last cycle
- 1 ➔ 1st cycle or 1st phase
- 2 ➔ 2nd cycle or 2nd phase

Figure 22. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

6.11 Interrupt register

Interrupt registers are memory space used to control interrupt functions. As shown in Table 10, the interrupt registers consist of Interrupt Enable Registers, Interrupt Priority Registers, External Interrupt Flag Registers, and External Interrupt Polarity Register.

6.11.1 Interrupt Enable registers (IE, IE1, IE2, IE3)

Interrupt enable register consists of global interrupt control bit (EA) and peripheral interrupt control bits. Total 24 peripherals are able to control interrupts.

6.11.2 Interrupt Priority registers (IP, IP1)

24 interrupts are divided into 6 groups where each group has 4 interrupt sources respectively. A group can be assigned 4 levels of interrupt priority by using an interrupt priority register. Level 3 is the highest priority, while level 0 is the lowest priority. After a reset, IP and IP1 are cleared to '00H'. If interrupts have the same priority level, lower number interrupt is served first.

6.11.3 External Interrupt Flag register (EIFLAG)

External interrupt flag (EIFLAG) is set to '1' when the external interrupt generating condition is satisfied. The flag is cleared when the interrupt service routine is executed. Alternatively, the flag can be cleared by writing '0' to it.

6.11.4 External Interrupt Polarity registers (EIPOL0, EIPOL1)

External interrupt polarity 0 register (EIPOL0) and external interrupt polarity 1 register (EIPOL1) determine one from rising edge, falling edge, and both edges for interrupt. No interrupt is at any edge by default.

6.11.5 Register map

Table 10. Interrupt Register Map

Name	Address	Direction	Default	Description
IE	A8H	R/W	00H	Interrupt Enable Register
IE1	A9H	R/W	00H	Interrupt Enable Register 1
IE2	AAH	R/W	00H	Interrupt Enable Register 2
IE3	ABH	R/W	00H	Interrupt Enable Register 3
IP	B8H	R/W	00H	Interrupt Priority Register
IP1	F8H	R/W	00H	Interrupt Priority Register 1
EIFLAG	A0H	R/W	00H	External Interrupt Flag Register
EIPOL0	A4H	R/W	00H	External Interrupt Polarity 0 Register
EIPOL1	A5H	R/W	00H	External Interrupt Polarity 1 Register

6.11.6 Interrupt register description

IE (Interrupt Enable Register): A8H

7	6	5	4	3	2	1	0
EA	–	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EA Enable or Disable All Interrupt bits
- 0 All Interrupt disable
- 1 All Interrupt enable

INT5E Enable or Disable External Interrupt 11 (EINT11)
- 0 Disable
- 1 Enable

INT4E Enable or Disable External Interrupt 10 (EINT10)
- 0 Disable
- 1 Enable

INT3E Enable or Disable External Interrupt 3 (EINT3)
- 0 Disable
- 1 Enable

INT2E Enable or Disable External Interrupt 2 (EINT2)
- 0 Disable
- 1 Enable

INT1E Enable or Disable External Interrupt 1 (EINT1)
- 0 Disable
- 1 Enable

INT0E Enable or Disable External Interrupt 0 (EINT0)
- 0 Disable
- 1 Enable

IE1 (Interrupt Enable Register 1): A9H

7	6	5	4	3	2	1	0
–-	–	INT11E	–	–-	–	INT7E	INT6E
–	–	R/W	–	–	–	R/W	R/W

Initial value: 00H

INT11E Enable or Disable External Interrupt 12 (EINT12)
- 0 Disable
- 1 Enable

INT7E Enable or Disable I2C Interrupt
- 0 Disable
- 1 Enable

INT6E Enable or Disable ADC Interrupt
- 0 Disable
- 1 Enable

IE2 (Interrupt Enable Register 2): AAH

7	6	5	4	3	2	1	0
–	–	INT17E	INT16E	–	INT14E	INT13E	INT12E
–	–	R/W	R/W	–	R/W	R/W	R/W

Initial value: 00H

INT17E Enable or Disable USART Tx Interrupt
- 0 Disable
- 1 Enable

INT16E Enable or Disable USART Rx Interrupt
- 0 Disable
- 1 Enable

INT14E Enable or Disable Timer 2 Interrupt
- 0 Disable
- 1 Enable

INT13E Enable or Disable Timer 1 Interrupt
- 0 Disable
- 1 Enable

INT12E Enable or Disable Timer 0 Interrupt
- 0 Disable
- 1 Enable

IE3 (Interrupt Enable Register 3): ABH

7	6	5	4	3	2	1	0
–	–	–	INT22E	INT21E	–	–	–
–	–	–	R/W	R/W	–	–	–

Initial value: 00H

INT22E Enable or Disable BIT Interrupt
- 0 Disable
- 1 Enable

INT21E Enable or Disable WDT Interrupt
- 0 Disable
- 1 Enable

IP (Interrupt Priority Register): B8H

7	6	5	4	3	2	1	0
–	–	IP5	IP4	IP3	IP2	IP1	IP0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

IP1 (Interrupt Priority Register 1): F8H

7	6	5	4	3	2	1	0
–	–	IP15	IP14	IP13	IP12	IP11	IP10
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

IP[5:0], IP1[5:0] Select Interrupt Group Priority

IP1x	IPx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

EIFLAG (External Interrupt Flag Register): A0H

7	6	5	4	3	2	1	0
IICIFR	FLAG12	FLAG11	FLAG10	FLAG3	FLAG2	FLAG1	FLAG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

IICIFR This is an I2C interrupt flag bit. When the interrupt occurs, this bit becomes '1'. This bit is cleared when write any values in the I2CSR register.

0 I2C Interrupt not occurred

1 I2C Interrupt occurred

EIFLAG[6:0] When an external interrupt (EINT0/1/2/3/10/11/12) is occurred, the flag becomes '1'. The flag is cleared by writing a '0' to the bit or automatically cleared by INT_ACK signal. Writing "1" has no effect.

0 External Interrupt 0/1/2/3/10/11/12 not occurred

1 External Interrupt 0/1/2/3/10/11/12 occurred

EIPOL0 (External Interrupt Polarity 0 Register): A4H

7	6	5	4	3	2	1	0
POL3		POL2		POL1		POL0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL0[7:0] External interrupt (EINT0, EINT1, EINT2 and EINT3) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n = 0, 1, 2 and 3

EIPOL1 (External Interrupt Polarity 1 Register): A5H

7	6	5	4	3	2	1	0
–-	–	POL12		POL11		POL10	
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL1[5:0] External interrupt (EINT10, EINT11 and EINT12) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n = 10, 11 and 12

7 Clock generator

As shown in Figure 23, a clock generator produces basic clock pulses which provide a CPU and peripherals with a system clock. A default system clock is a 1MHz HF INT-RC oscillator and default division rate is four. To stabilize system internally, it is used 1MHz HF INT-RC oscillator on POR.

A96L414/A96L416 incorporates three types of oscillators:

- Calibrated HF Internal RC Oscillator (4MHz)
  - HF INT-RC OSC/8 (0.5MHz)
  - HF INT-RC OSC/4 (1MHz, default system clock)
  - HF INT-RC OSC/2 (2MHz)
  - HF INT-RC OSC/1 (4MHz)
- Internal WDTRC Oscillator (1KHz)
- LF INT-RC Oscillator(32KHz)

7.1 Block diagram

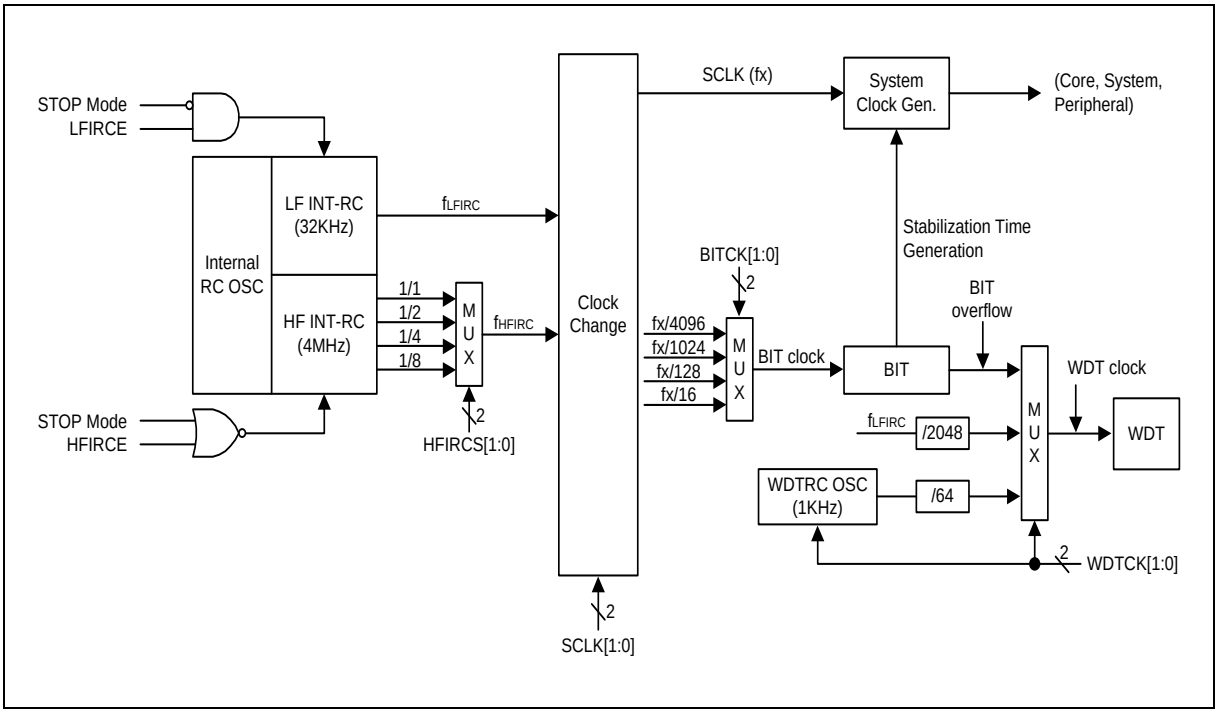


Figure 23. Clock Generator in Block Diagram

7.2 Register map

Table 11. Clock Generator Register Map

Name	Address	Direction	Default	Description
SCCR	8AH	R/W	00H	System and Clock Control Register
OSCCR	C8H	R/W	08H	Oscillator Control Register
IRCTCR	8FH	R/W	00H	Internal RC Trim Control Register
IRCTRM	97H	R/W	xxH	Internal RC Trim Register
IRCIDR	9FH	R/W	00H	Internal RC Identification Register

7.3 Register description

SCCR (System and Clock Control Register): 8AH

7	6	5	4	3	2	1	0
–	–	–	–	–	–	SCLK1	SCLK0
–	–	–	–	–	–	R/W	R/W

Initial value: 00H

SCLK[1:0] System Clock Selection Bit

SCLK1	SCLK0	Description
0	0	HFIRC(f_{HFIRC}) for system clock
1	1	LFIRC(f_{LFIRC}) for system clock
Other Values		Not available(No effect)

NOTE: If a target system clock is disabled by the OSCCR register, the SCCR register won't be changed in case of selecting the corresponding clock as a system clock.

OSCCR (Oscillator Control Register): C8H

7	6	5	4	3	2	1	0
LFIRCE	–	–	IRCS1	IRCS0	HFIRCE	–	–
R/W	–	–	R/W	R/W	R/W	–	–

Initial value: 08H

LFIRCE Control the operation of the low frequency internal RC oscillator

0	Disable operation of LFIRC OSC
1	Enable operation of LFIRC OSC

HFIRCS[1:0] Internal RC Oscillator Post-divider Selection

HFIRCS1	HFIRCS0	Description
0	0	f_{HFIRC}/8 (0.5MHz)
0	1	f_{HFIRC}/4 (1MHz)
1	0	f_{HFIRC}/2 (2MHz)
1	1	f_{HFIRC}/1 (4MHz)

HFIRCE Control the operation of the high frequency internal RC oscillator

0	Enable operation of HFIRC OSC
1	Disable operation of HFIRC OSC

NOTE: The system clock is not disabled by the corresponding bit of the OSCCR register.
Ex) The high frequency internal RC oscillator won't be disabled by the HFIRCE bit during the f_{HFIRC} is selected as the system clock.

IRCIDR (Internal RC Trim Identification Register): 9FH

7	6	5	4	3	2	1	0
IRCID7	IRCID6	IRCID5	IRCID4	IRCID3	IRCID2	IRCID1	IRCID0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

IRCID[7:0] Internal RC Trim Identification.
Others No identification value
01000110b Identification value for IRC Trim
(These bits are automatically cleared to logic '00H' immediately after one time operation.)

IRCTRM (Internal RC Trim Register): 97H

7	6	5	4	3	2	1	0
ITRM7	ITRM6	ITRM5	ITRM4	ITRM3	ITRM2	ITRM1	ITRM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: xxH

ITRM[7:0] Internal RC Trim bits.
These bits are read from "Configure Area" when a system reset occurs. These bits provide a user programmable trimming value on operation. The range is -128 to +127. The ITRM7 is sign bit. The IRC frequency is faster by minus value and slower by plus. The frequency is changed by about 11[kHz] step-by-step. This register can be written with valid ID value and IRCTCR=0xB3.

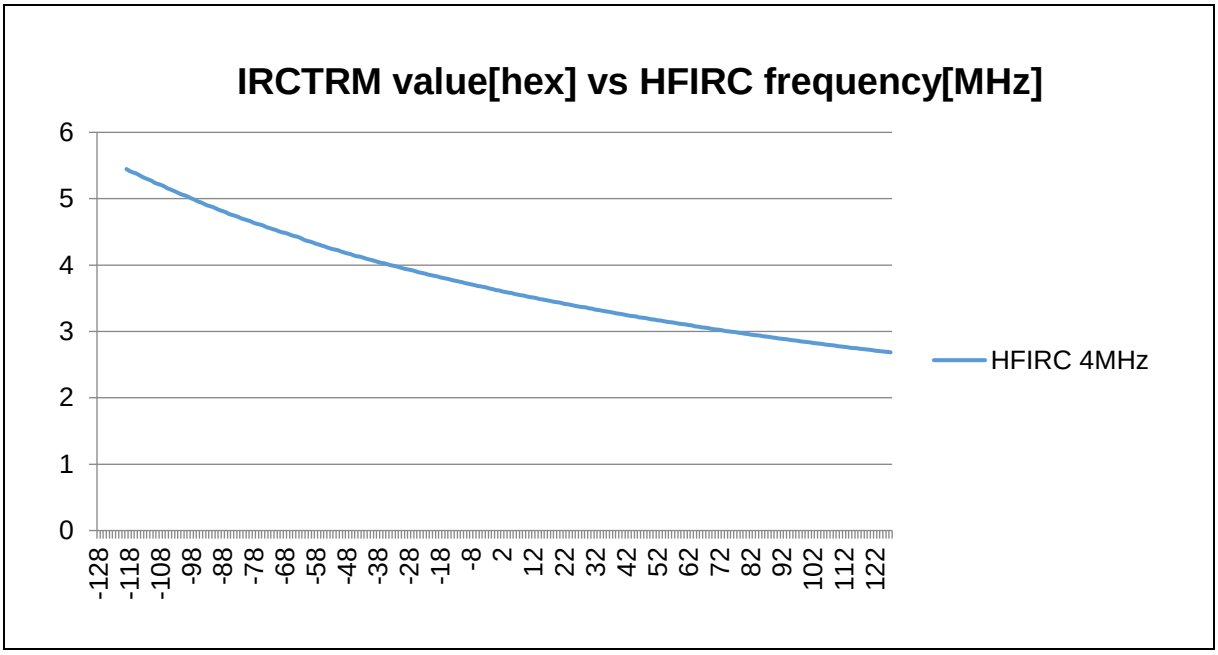


Figure 24. IRCTRM Value vs. IRC Frequency Graph

NOTE: Due to the HFIRC 4MHz filter, more than 5.5MHz frequency range can't measure.

IRCTCR (Internal RC Trim Control Register): 8FH

7	6	5	4	3	2	1	0
ITCR7	ITCR6	ITCR5	ITCR4	ITCR3	ITCR2	ITCR1	ITCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

ITCR[7:0]	Internal RC Trim Control Register.	
	Others	No effect
	10110011b	IRCTRM register is used for IRC frequency. This register can be written with valid ID value.

8 Basic Interval Timer (BIT)

A96L414/A96L416 has a free running 8-bit Basic Interval Timer (BIT). This BIT generates the time base for Watchdog Timer counting, and provides a basic interval timer interrupt (BITIFR).

The BIT of A96L414/A96L416 features the followings:

- During Power On, BIT gives a stable clock generation time
- On exiting Stop mode, BIT gives a stable clock generation time
- As a timer, BIT generates a timer interrupt.

8.1 Block diagram

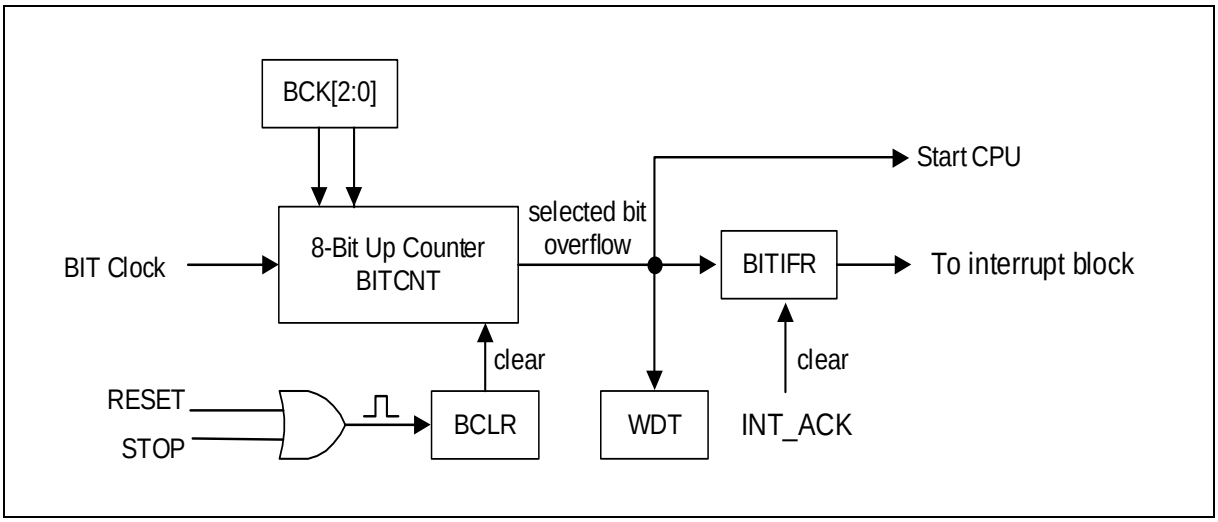


Figure 25. Basic Interval Timer in Block Diagram

8.2 Register map

Table 12. Basic Interval Timer Register Map

Name	Address	Direction	Default	Description
BITCNT	8CH	R	00H	Basic Interval Timer Counter Register
BITCR	8BH	R/W	01H	Basic Interval Timer Control Register

8.3 Register description

BITCNT (Basic Interval Timer Counter Register): 8CH

7	6	5	4	3	2	1	0
BITCNT7	BITCNT6	BITCNT5	BITCNT4	BITCNT3	BITCNT2	BITCNT1	BITCNT0
R	R	R	R	R	R	R	R

Initial value: 00H

BITCNT[7:0] BIT Counter

BITCR (Basic Interval Timer Control Register): 8BH

7	6	5	4	3	2	1	0
BITIFR	BITCK1	BITCK0	–	BCLR	BCK2	BCK1	BCK0
R/W	R/W	R/W	–	R/W	R/W	R/W	R/W

Initial value: 01H

BITIFR When BIT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.
0 BIT interrupt no generation
1 BIT interrupt generation

BITCK[1:0] Select BIT clock source

BITCK1	BITCK0	Description
0	0	fx/4096
0	1	fx/1024
1	0	fx/128
1	1	fx/16

BCLR If this bit is written to '1', BIT Counter is cleared to '0'
0 Free Running
1 Clear Counter

BCK[2:0] Select BIT overflow period

BCK2	BCK1	BCK0	Description
0	0	0	Bit 0 overflow (BIT Clock * 2)
0	0	1	Bit 1 overflow (BIT Clock * 4) (default)
0	1	0	Bit 2 overflow (BIT Clock * 8)
0	1	1	Bit 3 overflow (BIT Clock * 16)
1	0	0	Bit 4 overflow (BIT Clock * 32)
1	0	1	Bit 5 overflow (BIT Clock * 64)
1	1	0	Bit 6 overflow (BIT Clock * 128)
1	1	1	Bit 7 overflow (BIT Clock * 256)

9 Watchdog Timer (WDT)

Watchdog Timer (WDT) is used to rapidly detect the CPU malfunctions such as endless looping caused by noise. In addition, it is used to resume the CPU in a normal state. A watchdog timer signal for malfunction detection can be used as either a CPU reset or an interrupt request. When the Watchdog Timer is not being used for the detection of the CPU malfunctions, it can be used as a timer generating an interrupt at fixed intervals.

The Watchdog Timer can be used in a free running 8-bit timer mode or in a Watchdog Timer mode by setting WDTRSON bit, which is WDTCR[5]. If '1' is written to WDTCR[5], WDT counter value is cleared and counts up. After 1 machine cycle, this bit is cleared to '0' automatically.

The Watchdog Timer consists of an 8-bit binary counter and the Watchdog Timer Data Register. When value of an 8-bit binary counter is equal to the 8 bits of WDTCNT, an interrupt request flag is generated. This can be used as a watchdog timer interrupt or a reset of CPU in accordance with a bit WDTRSON.

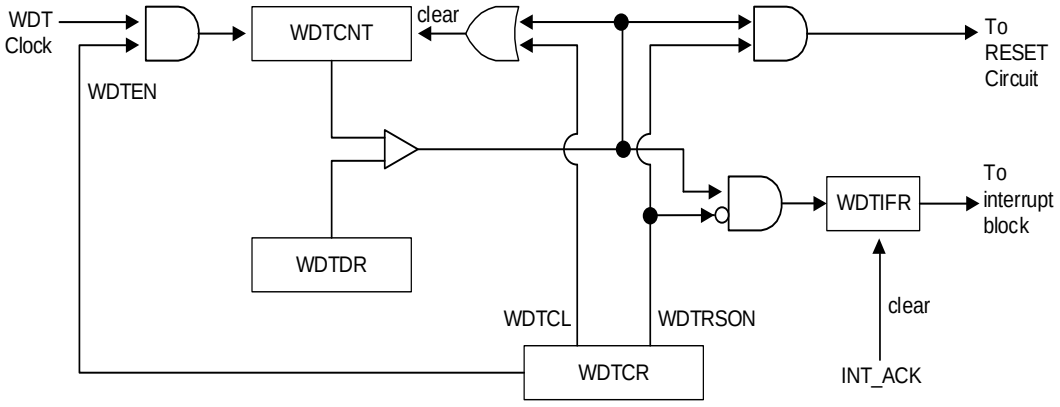
The input clock source of Watchdog Timer is BIT overflow, LFIRC, WDTRC. Interval of the watchdog timer interrupt is decided by the BIT overflow period and WDTDR set value. Equation of the WDT interrupt interval is described in the followings:

WDT Interrupt Interval = (BIT Interrupt Interval) X (WDTDR Value+1)

WDT Interrupt Interval = $2048/f_{LFIRC}$ x (WDTDR Value+1) when LFIRC

WDT Interrupt Interval = $64/f_{WDTRC}$ x (WDTDR Value+1) when WDTRC

9.1 Block diagram



NOTE: Watchdog timer clock depends on WDTCK bit in WDTCR register. For detailed information of the WDTCR register, please refer to Watchdog timer register description later part in this chapter.

Figure 26. Watchdog Timer in Block Diagram

9.2 WDT interrupt timing waveform

Figure 27 shows a timing diagram when a Watchdog Timer generates system reset signal and an interrupt signal.

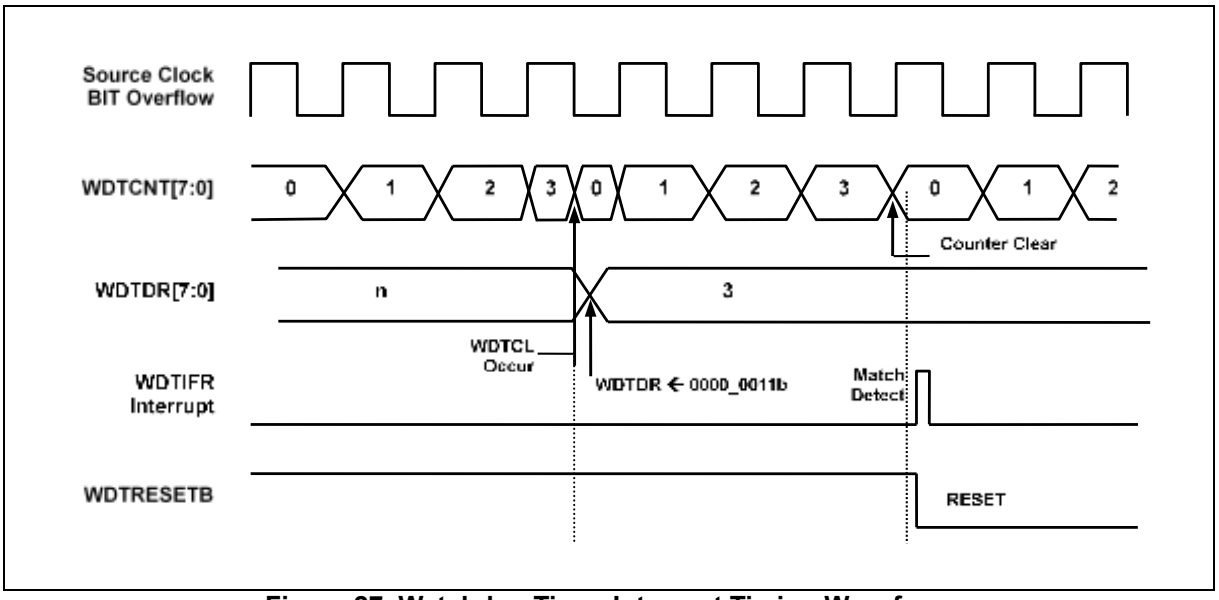


Figure 27. Watchdog Timer Interrupt Timing Waveform

9.3 Register map

Name	Address	Direction	Default	Description
WDTCNT	8EH	R	00H	Watchdog Timer Counter Register
WDTDR	8EH	W	FFH	Watchdog Timer Data Register
WDTCR	8DH	R/W	00H	Watchdog Timer Control Register

9.4 Register description

WDTCNT (Watchdog Timer Counter Register: Read Case): 8EH

7	6	5	4	3	2	1	0
WDTCNT7	WDTCNT6	WDTCNT5	WDTCNT4	WDTCNT3	WDTCNT2	WDTCNT1	WDTCNT0
R	R	R	R	R	R	R	R

Initial value: 00H

WDTCNT[7:0] WDT Counter

WDTDR (Watchdog Timer Data Register: Write Case): 8EH

7	6	5	4	3	2	1	0
WDTDR7	WDTDR6	WDTDR5	WDTDR4	WDTDR3	WDTDR2	WDTDR1	WDTDR0
W	W	W	W	W	W	W	W

Initial value: FFH

WDTDR[7:0] Set a period

WDT Interrupt Interval is as follows:

- (BIT Interrupt Interval) x(WDTDR Value+1)
- $64/f_{WDTRC}$ x (WDTDR Value+1) when WDTRC
- $2048/f_{LFIRC}$ x (WDTDR Value+1) when LFIRC

NOTE: Do not write "0" in the WDTDR register.

WDTCR (Watchdog Timer Control Register): 8DH

7	6	5	4	3	2	1	0
WDTEN	WDTRSON	WDTCL	–	–	WDTCK1	WDTCK0	WDTIFR
R/W	R/W	R/W	–	–	R/W	R/W	R/W

Initial value: 00H

WDTEN Control WDT Operation
0 Disable
1 Enable

WDTRSON Control WDT RESET Operation
0 Free Running 8-bit timer
1 Watchdog Timer RESET ON

WDTCL Clear WDT Counter
0 Free Run
1 Clear WDT Counter (auto clear after 1 Cycle)

WDTCK[1:0] Control WDT Clock Selection Bit

WDTCK1	WDTCK0	description
0	0	BIT overflow for WDT clock (WDTRC disable)
0	1	LFIRC for WDT clock(WDTRC disable)
1	0	WDTRC for WDT clock (WDTRC enable)
1	1	Not available(No effect)

WDTIFR When WDT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.
0 WDT Interrupt no generation
1 WDT Interrupt generation

10 TIMER 0/1/2

A 16-bit timer 0/1/2 incorporates a multiplexer and six registers such as timer 0/1/2A data register high/low, timer 0/1/2B data register high/low, and timer 0/1/2 control register high/low (TnADRH, TnADRL, TnBDRH, TnBDRL, TnCRH, TnCRL).

TIMER 0/1/2 operates in one of four operating modes:

- 16-bit capture mode
- 16-bit timer/ counter mode
- 16-bit PPG output mode (one-shot mode)
- 16-bit PPG output mode (repeat mode)

Specifically in capture mode, data is captured into input capture data register (TnBDRH/TnBDRL) by EINT10/EINT11/EINT12. Timer 0/1/2 outputs the comparison result between counter and data register through TnO port in timer/counter mode. Timer 0/1/2 outputs PWM wave form through PWMnO port in the PPG mode).

A timer/counter 0/1/2 uses an internal clock or an external clock (ECn) as an input clock source. The clock sources are introduced below, and one is selected by clock selection logic which is controlled by clock selection bits (TnCK[2:0]).

- Timer 0/1/2 clock sources: f_X/1, 2, 4, 8, 64, 512, 2048 and ECn

Table 13. TIMER 0/1/2 Operating Modes

TnEN	P1FSRL[2](T0)/ P1FSRH[1:0](T1)/ P2FSR[1:0](T2)	TnMS[1:0]	TnCK[2:0]	Timer n
1	1/01/01	00	XXX	16 Bit Timer/Counter Mode
1	0/00/00	01	XXX	16 Bit Capture Mode
1	1/01/01	10	XXX	16 Bit PPG Mode(one-shot mode)
1	1/01/01	11	XXX	16 Bit PPG Mode(repeat mode)

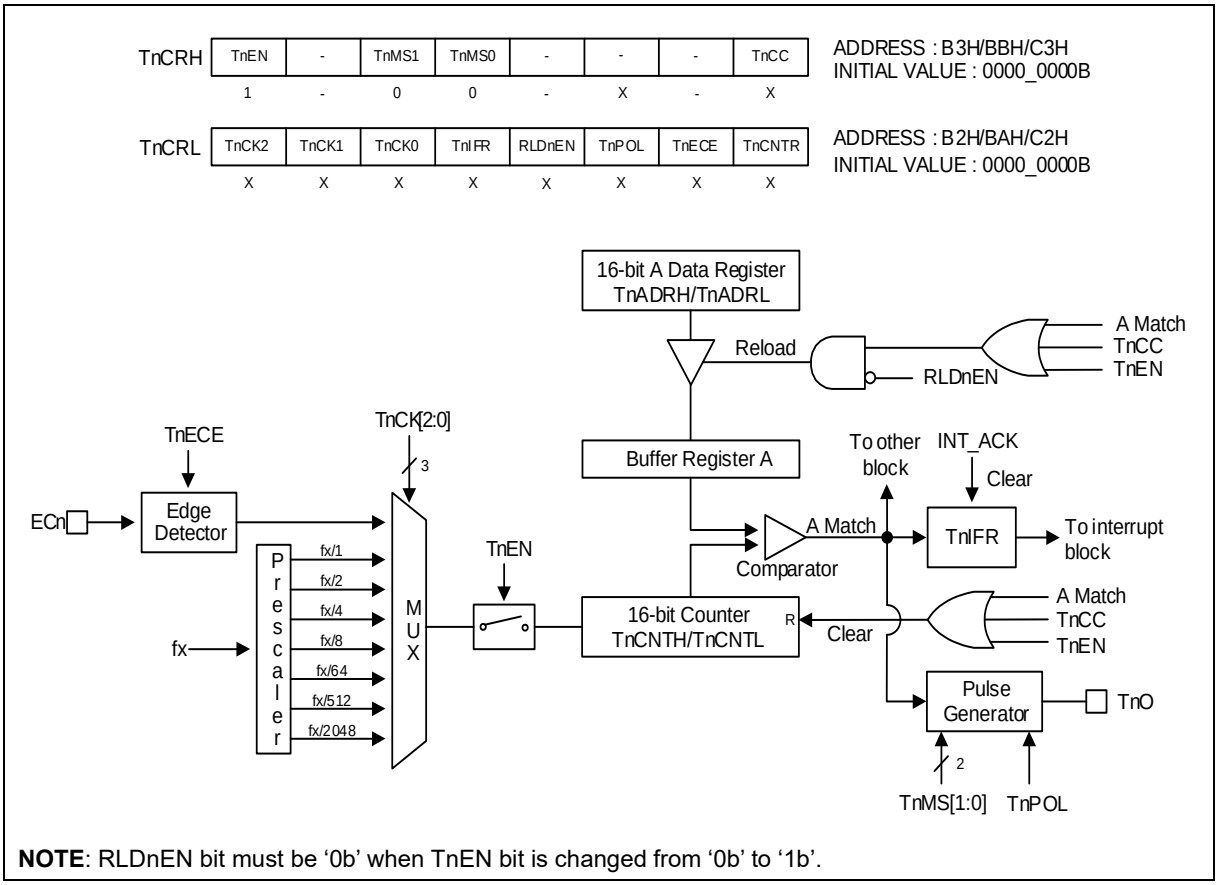
10.1 16-bit timer/ counter mode

16-bit timer/counter mode is selected by control register as shown in Figure 28. This figure shows that a 16-bit timer has a counter and data registers. Counter registers have increasing values by internal or external clock input. TIMER 0/1/2 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (TnCK[2:0]).

When the value of TnCNTH, TnCNTL and the value of TnADRH, TnADRL are identical each other in Timer 0/1/2, a match signal is generated and the interrupt of Timer 0/1/2 occurs.

The TnCNTH, TnCNTL value is automatically cleared by match signal. It can be cleared by software (TnCC) too.

The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], ECn port should be set to the input port by P11IO/P14IO/P20IO bit.



NOTE: RLDnEN bit must be '0b' when TnEN bit is changed from '0b' to '1b'.

Figure 28. 16-bit Timer/ Counter Mode of TIMER 0/1/2

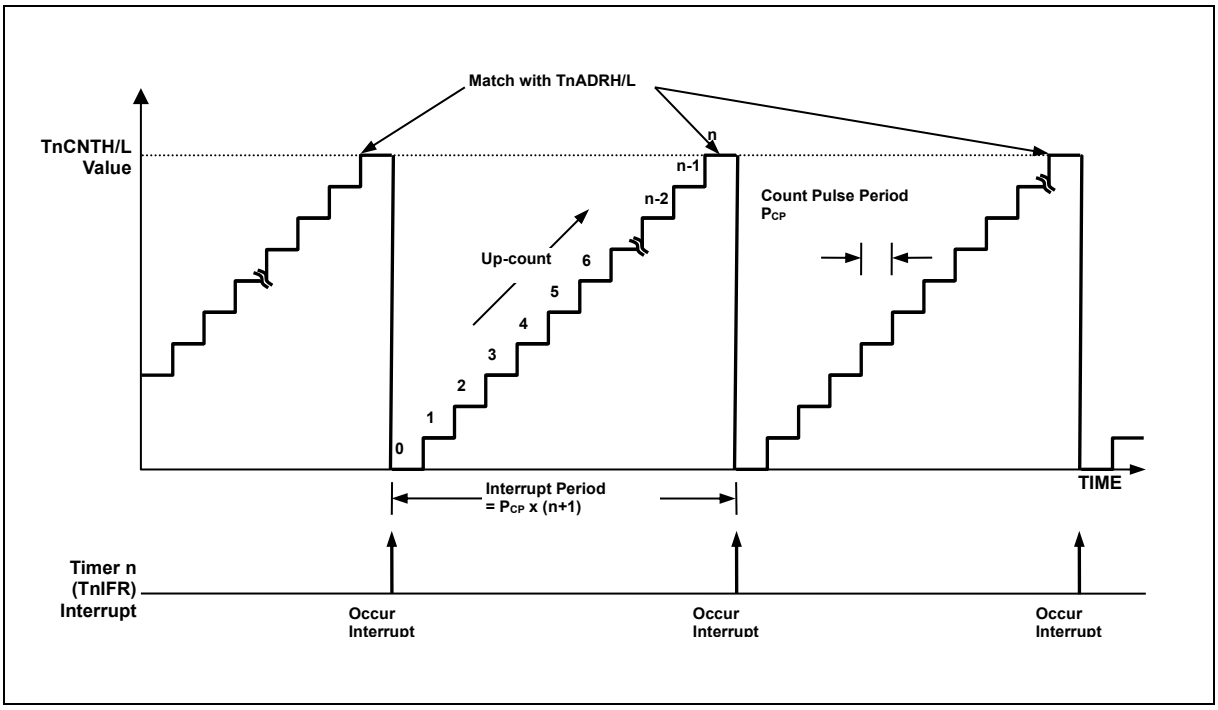
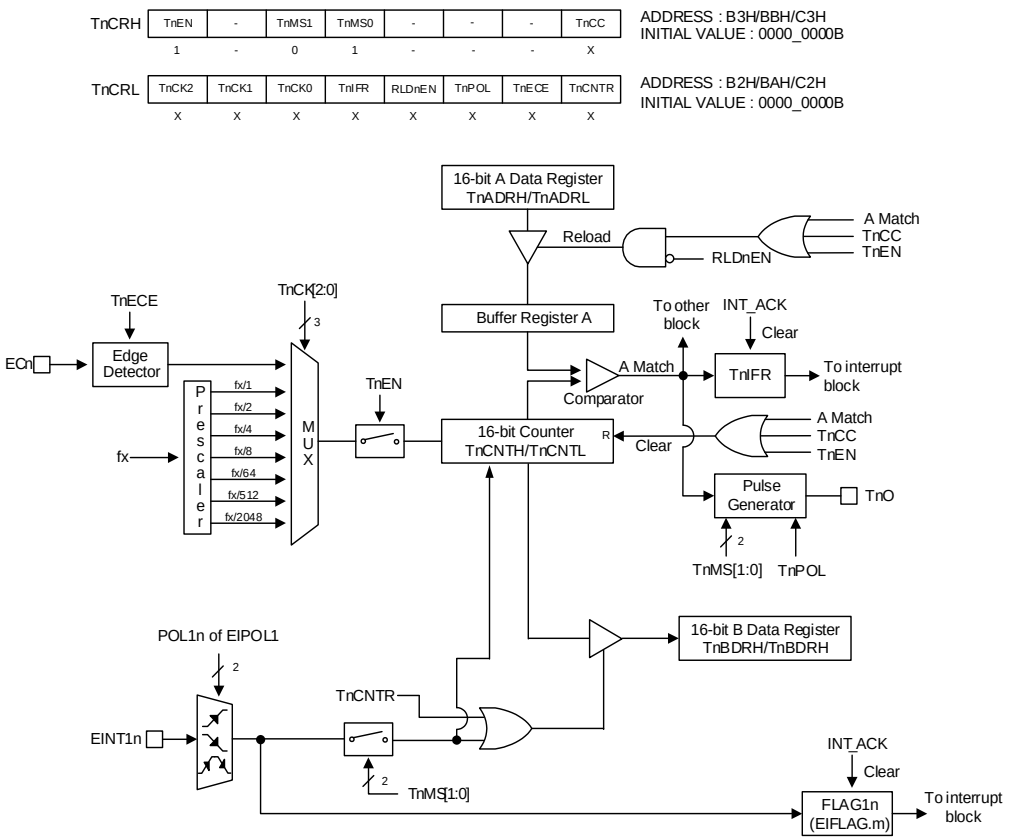


Figure 29. 16-bit Timer/ Counter 0/1/2 Interrupt Example

10.2 16-bit capture mode

16-bit timer 0/1/2 capture mode is set by configuring TnMS[1:0] as '01'. It uses an internal/external clock as a clock source. Basically, the 16-bit timer 0/1/2 capture mode has the same function as the 16-bit timer/counter mode, and the interrupt occurs when TnCNTH/TnCNTL is equal to TnADRH/TnADRL. The TnCNTH, TnCNTL values are automatically cleared by match signal. It can be cleared by software (TnCC) too.

A timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. Capture result is loaded into TnBDRH/TnBDRL. According to EIPOL1 registers settings, the external interrupt EINT10/EINT11/EINT12 function is selected. EINT10/EINT11/EINT12 pin must be set as an input port.



NOTE: RLDnEN bit must be '0b' when TnEN bit is changed from '0b' to '1b'.

Figure 30. 16-bit Capture Mode of TIMER 0/1/2 (Where n=0, 1, and 2, m=4, 5, and 6)

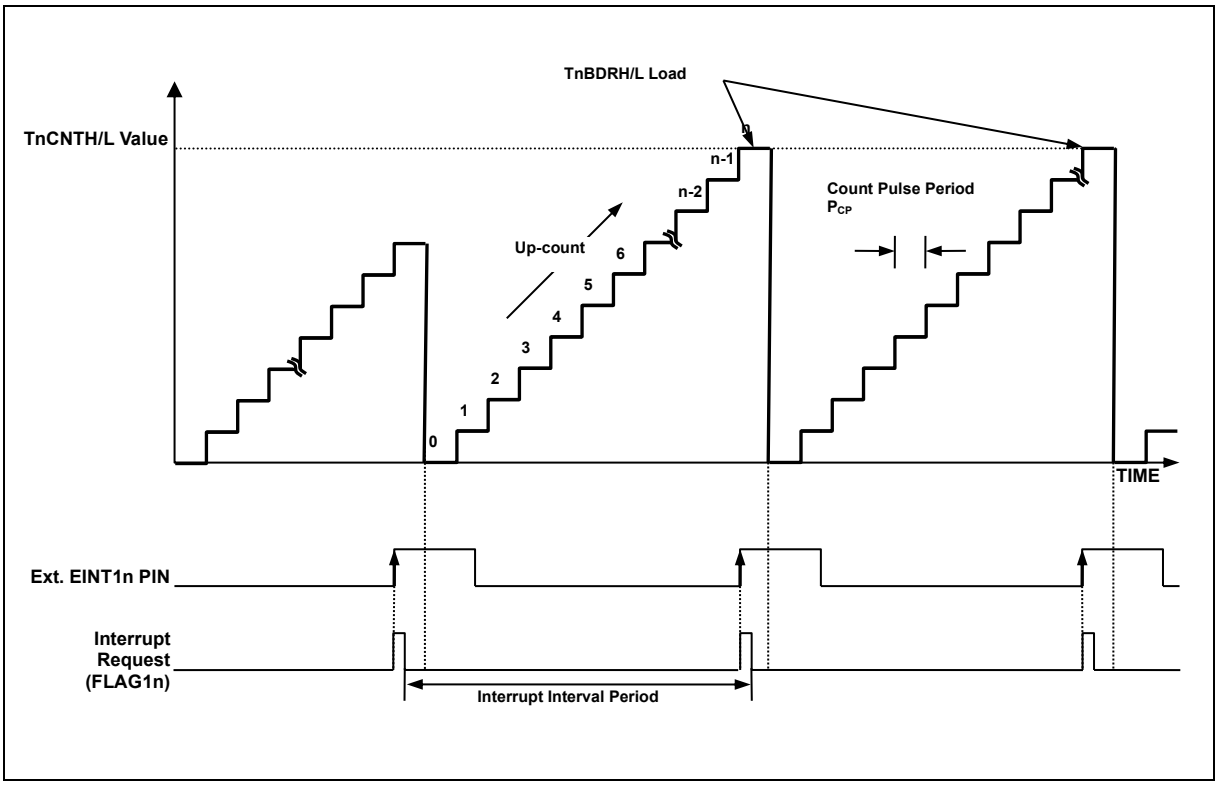


Figure 31. Input Capture Mode Operation of TIMER 0/1/2

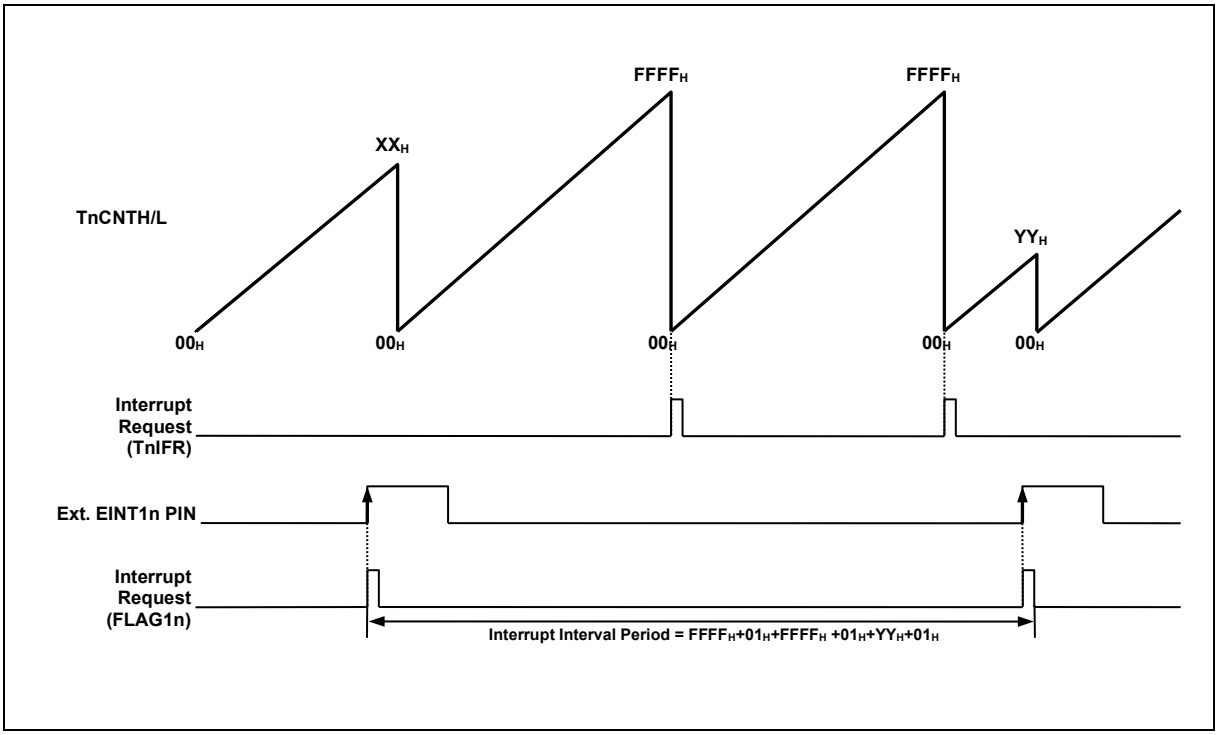
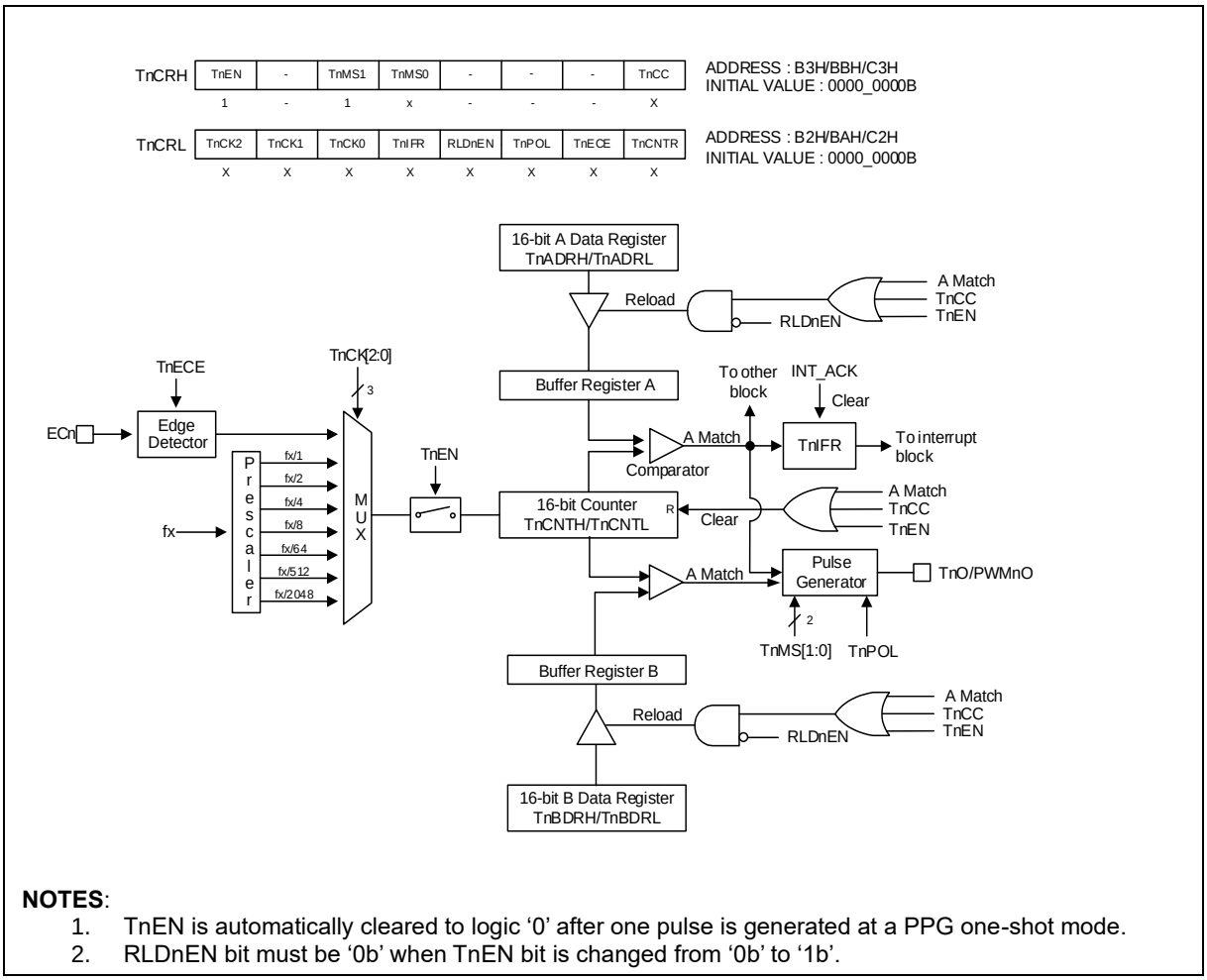


Figure 32. Express Timer Overflow in Capture Mode

10.3 16-bit PPG mode

TIMER 0/1/2 has a PPG (Programmable Pulse Generation) function. In PPG mode, TnO/PWMnO pin outputs up to 16-bit resolution PWM output.

For this function, TnO/PWMnO pin must be configured as a PWM output by setting P1FSRL[2] to '1'(T0), P1FSRH[0] to '1'(T1), P2FSR [0] to '1'(T2). Period of the PWM output is determined by TnADRH/TnADRL, and duty of the PWM output is determined by TnBDRH/TnBDRL.



NOTES:

1. TnEN is automatically cleared to logic '0' after one pulse is generated at a PPG one-shot mode.
2. RLDnEN bit must be '0b' when TnEN bit is changed from '0b' to '1b'.

Figure 33. 16-bit PPG Mode of TIMER 0/1/2

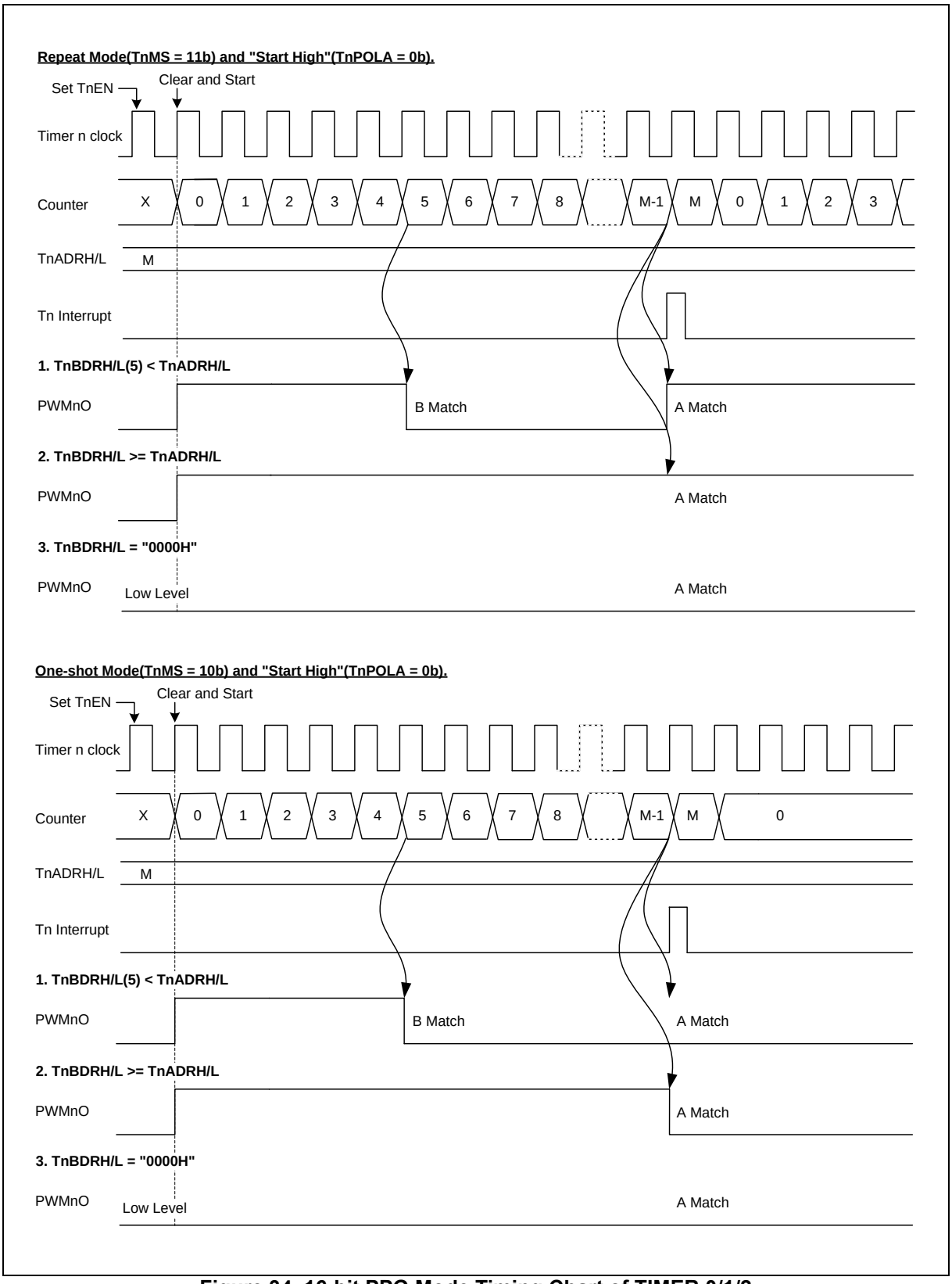
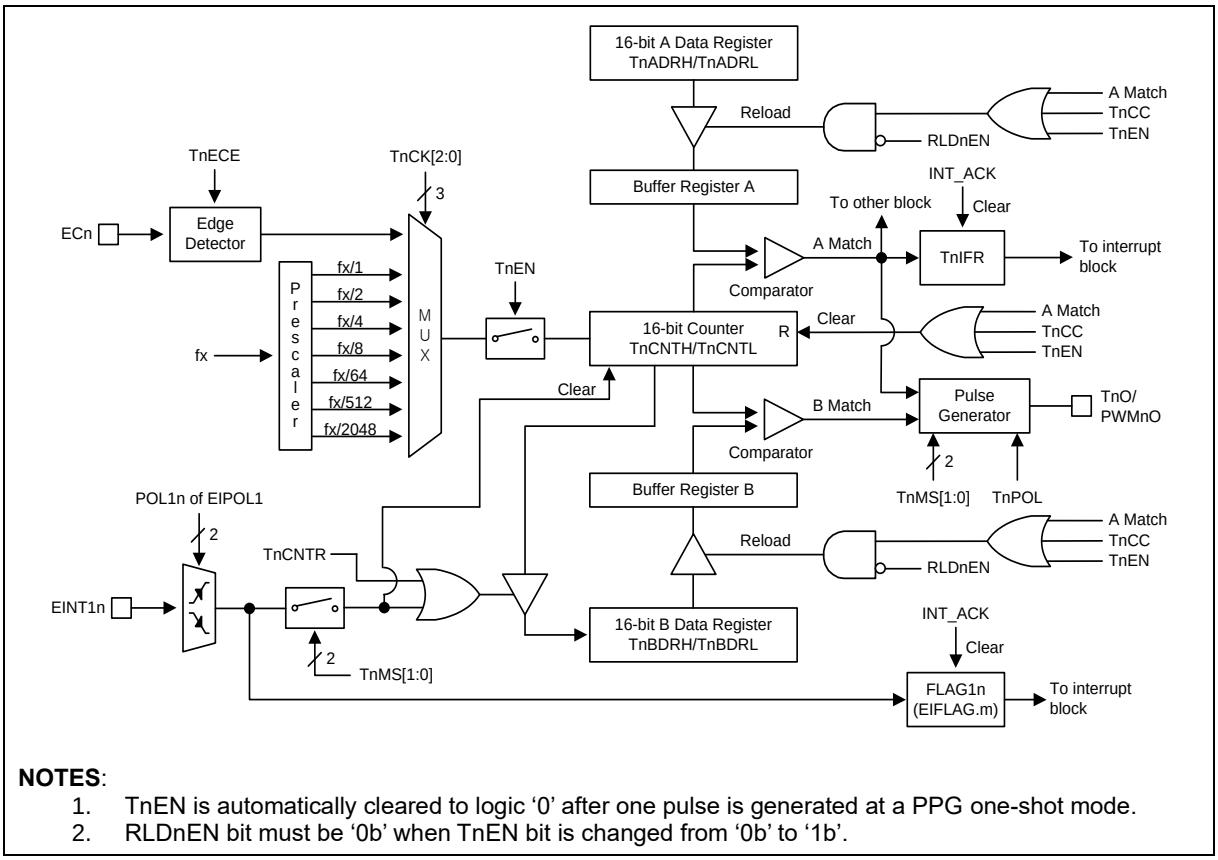


Figure 34. 16-bit PPG Mode Timing Chart of TIMER 0/1/2

10.4 Block diagram



NOTES:

1. TnEN is automatically cleared to logic '0' after one pulse is generated at a PPG one-shot mode.
2. RLDnEN bit must be '0b' when TnEN bit is changed from '0b' to '1b'.

Figure 35. 16-bit Timer n in Block Diagram (Where n = 0, 1, and 2, m=4, 5, and 6)

10.5 Register map

Table 14. TIMER n Register Map (Where n = 0, 1, and 2)

Name	Address	Direction	Default	Description
TnCRH	B3H/BBH/C3H	R/W	00H	Timer n Control High Register
TnCRL	B2H/BAH/C2H	R/W	00H	Timer n Control Low Register
TnADRH	B5H/BDH/C5H	R/W	FFH	Timer n A Data High Register
TnADRL	B4H/BCH/C4H	R/W	FFH	Timer n A Data Low Register
TnBDRH	B7H/BFH/C7H	R/W	FFH	Timer n B Data High Register
TnBDRL	B6H/BEH/C6H	R/W	FFH	Timer n B Data Low Register

10.6 Timer/counter 0/1/2 register description

TnADRH (Timer n A data High Register): B5H/BDH/C5H, Where n = 0, 1, and 2

7	6	5	4	3	2	1	0
TnADRH7	TnADRH6	TnADRH5	TnADRH4	TnADRH3	TnADRH2	TnADRH1	TnADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

TnADRH[7:0] Tn A Data High Byte

TnADRL (Timer n A Data Low Register): B4H/BCH/C4H, Where n = 0, 1, and 2

7	6	5	4	3	2	1	0
TnADRL7	TnADRL6	TnADRL5	TnADRL4	TnADRL3	TnADRL2	TnADRL1	TnADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

TnADRL[7:0] Tn A Data Low Byte

NOTE: Do not write "0000H" in the TnADRH/TnADRL register when PPG mode

TnBDRH (Timer n B Data High Register): B7H/BFH/C7H, Where n = 0, 1, and 2

7	6	5	4	3	2	1	0
TnBDRH7	TnBDRH6	TnBDRH5	TnBDRH4	TnBDRH3	TnBDRH2	TnBDRH1	TnBDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

TnBDRH[7:0] Tn B Data High Byte

TnBDRL (Timer n B Data Low Register): B6H/BEH/C6H, Where n = 0, 1, and 2

7	6	5	4	3	2	1	0
TnBDRL7	TnBDRL6	TnBDRL5	TnBDRL4	TnBDRL3	TnBDRL2	TnBDRL1	TnBDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

TnBDRL[7:0] Tn B Data Low Byte

TnCRH (Timer n Control High Register): B3H/BBH/C3H, Where n = 0, 1, and 2

7	6	5	4	3	2	1	0
TnEN	–	TnMS1	TnMS0	–	–	–	TnCC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value: 00H

TnEN Control Timer n
- 0 Timer n disable
- 1 Timer n enable (Counter clear and start)

TnMS[1:0] Control Timer n Operation Mode

TnMS1	TnMS0	Description
0	0	Timer/counter mode (TnO: toggle at A match)
0	1	Capture mode (The A match interrupt can occur)
1	0	PPG one-shot mode (PWMnO)
1	1	PPG repeat mode (PWMnO)

TnCC Clear Timer n Counter
- 0 No effect
- 1 Clear the Timer n counter (When write, automatically cleared "0" after being cleared counter)

TnCRL (Timer n Control Low Register): B2H/BAH/C2H, Where n = 0, 1, and 2

7	6	5	4	3	2	1	0
TnCK2	TnCK1	TnCK0	TnIFR	RLDnEN	TnPOL	TnECE	TnCNTR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

TnCK[2:0] Select Timer n clock source. fx is main system clock frequency

TnCK2	TnCK1	TnCK0	Description
0	0	0	fx/2048
0	0	1	fx/512
0	1	0	fx/64
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	External clock (ECn)

TnIFR When Tn Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.
- 0 Tn Interrupt no generation
- 1 Tn Interrupt generation

RLDnEN Control Timer n Reload Signal
- 0 Enable Timer n reload signal
- 1 Disable Timer n reload signal

TnPOL TnO/PWMnO Polarity Selection
- 0 Start High (TnO/PWMnO is low level at disable)
- 1 Start Low (TnO/PWMnO is high level at disable)

TnECE Timer n External Clock Edge Selection
- 0 External clock falling edge
- 1 External clock rising edge

TnCNTR Timer n Counter Read Control
- 0 No effect
- 1 Load the counter value to the B data register (When write, automatically cleared "0" after being loaded)

11 10-bit A/D Converter

Analog-to-digital (A/D) converter allows conversion of an analog input signal to a corresponding 10-bit digital output. The A/D module has 9 analog inputs, and the output of the multiplexer becomes the input into the converter, which generates a result via successive approximation.

The A/D module incorporates four registers as listed in the followings. Each register can be selected for the corresponding channel by setting ADSEL[3:0]. When conversion is completed, two registers ADCDRH and ADCDRL contain the results of the conversion, the conversion status bit AFLAG is set to '1', and A/D interrupt is set. During the A/D conversion, AFLAG bit is read as '0'.

- A/D converter control high register (ADCCRH)
- A/D converter control low register (ADCCRL)
- A/D converter data high register (ADCDRH)
- A/D converter data low register (ADCDRL)
- LDO control register (LDOCR)

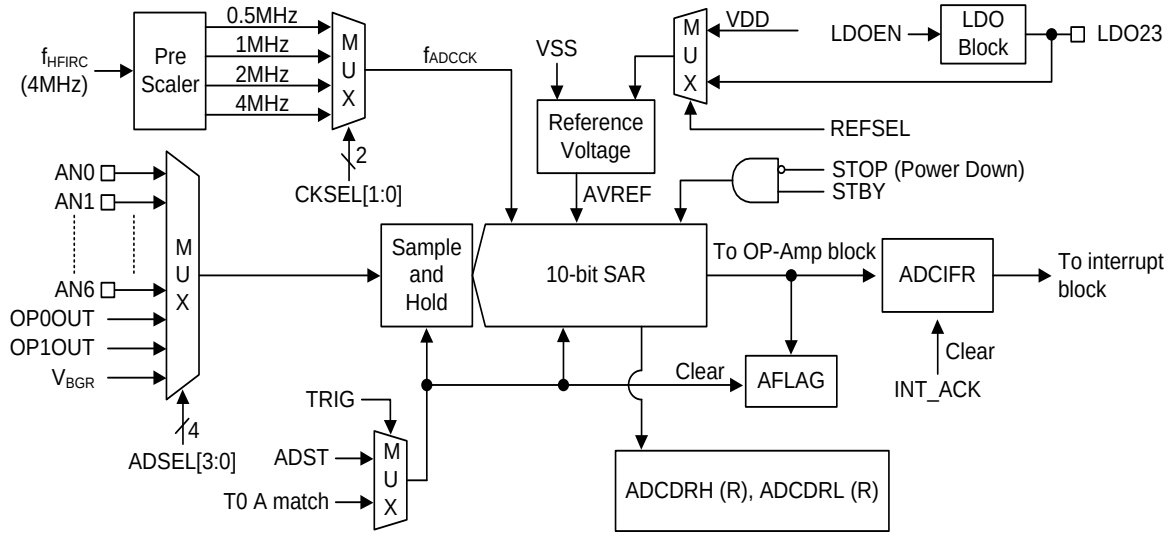
11.1 Conversion timing

A/D conversion process requires 6 clocks to sample and hold, 2 steps (2 clock edges) to convert each bit, and 2 clocks to set up A/D conversion. Therefore, total of 28 clocks are required to complete a 10-bit conversion.

For example, in a case that conversion clock operates with a 2MHz ADC clock frequency, one clock cycle is 0.5 us. Each bit conversion requires 2 clocks, the conversion rate is calculated as follows:

'6 clocks for S&H' + '2 clocks/bit × 10 bits' + set-up time = 28 clocks
28 clock × 0.5 us = 14 us at 2 MHz

11.2 Block diagram



NOTES:

1. The conversion clock is 14-clock including ADC sample and hold.
2. The time of ADC sample/hold and conversion is at least 8usec.
3. When the LDO is enabled, a 0.1uF capacitor should be connected between LDO23 pin and VSS.
4. The LDO should only be used as the reference voltage of A/D converter and OP-amp.

Figure 36. 10-bit ADC Block Diagram

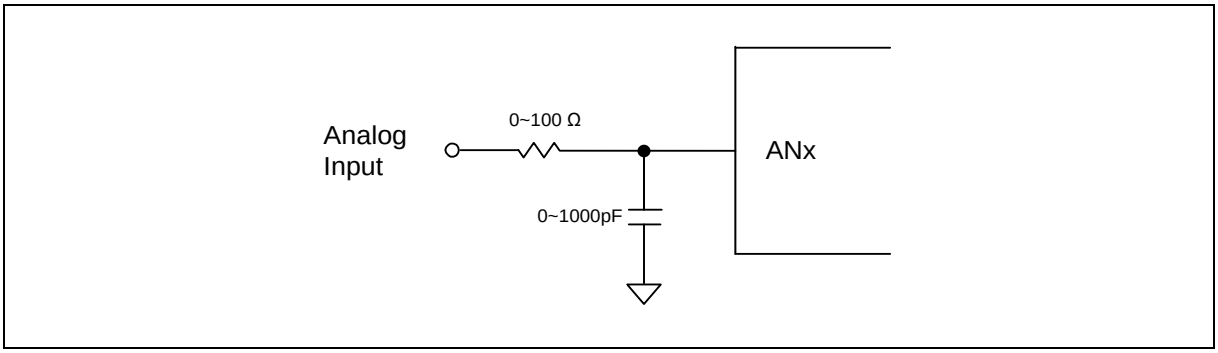


Figure 37. AD Analog Input Pin with Capacitor

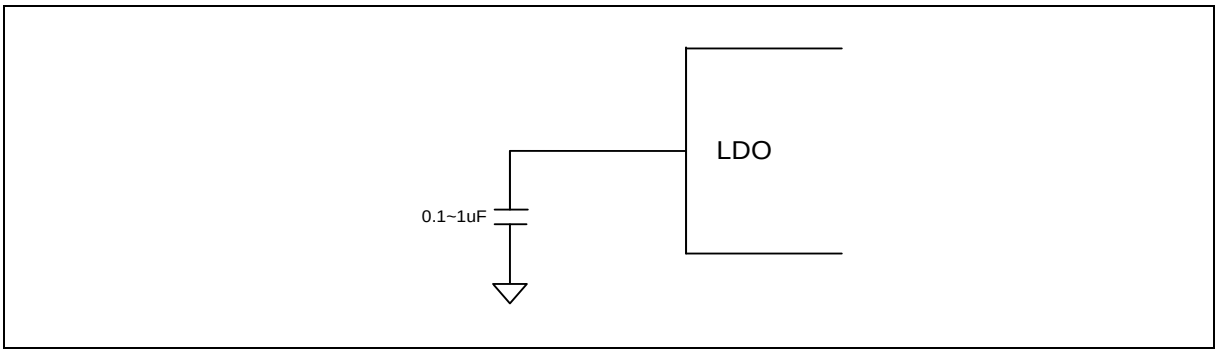


Figure 38. LDO23 Pin with Capacitor

11.3 ADC operation

In this section, ADC operation is described through figures 39 to 41. As shown in Figure 39, ADC conversion starts after configuring ADC Control High/ Low registers. By checking AFLAG, it is defined whether the conversion is completed or not. If AFLG is '1', the conversion is completed and ADC Data High/ Low registers are read to finish ADC operation.

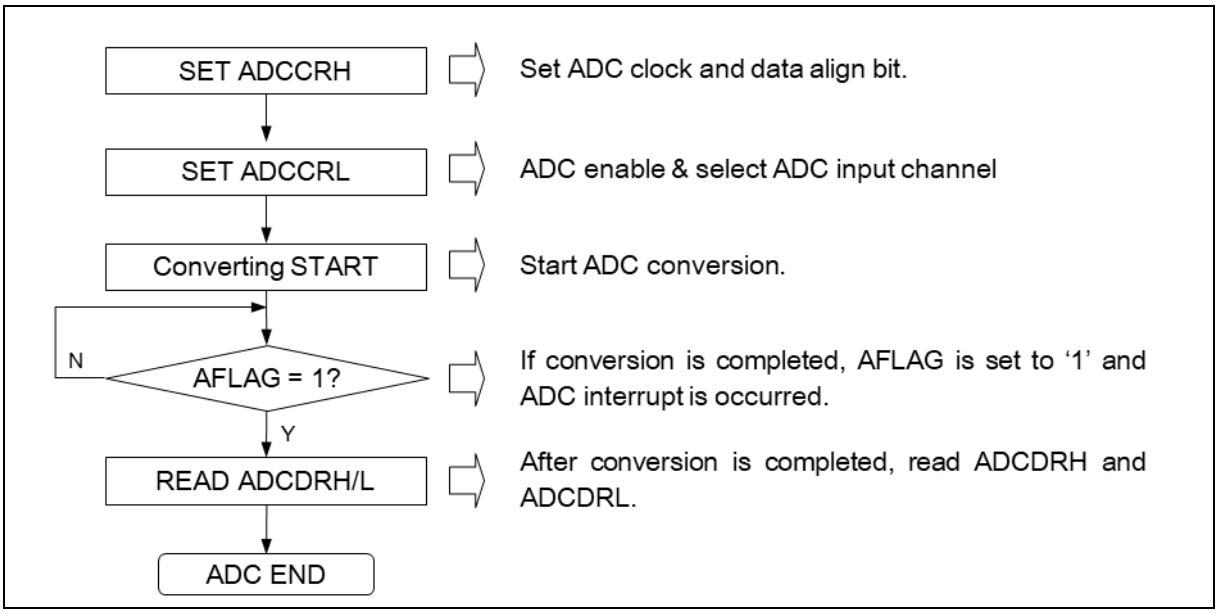


Figure 39. ADC Operation Flow

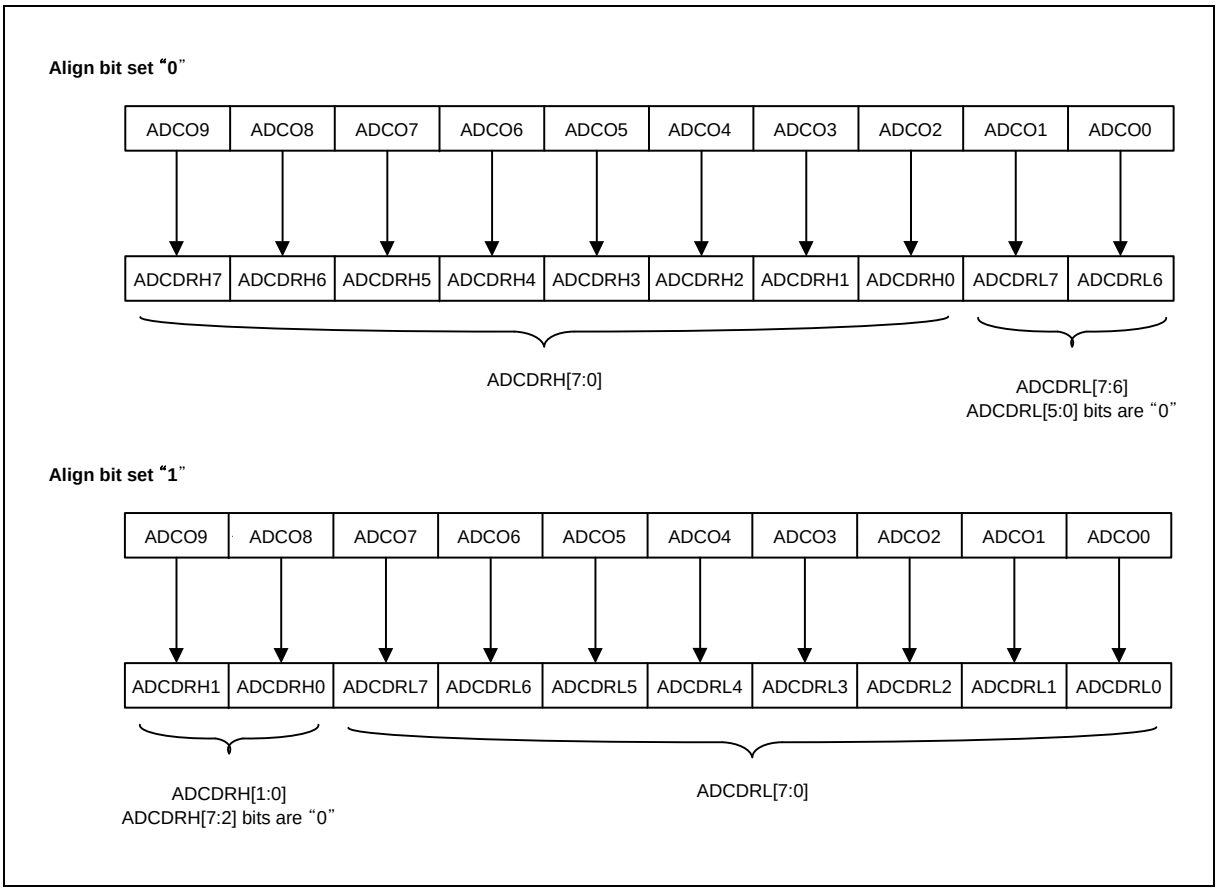


Figure 40. ADC Operation for Align Bit

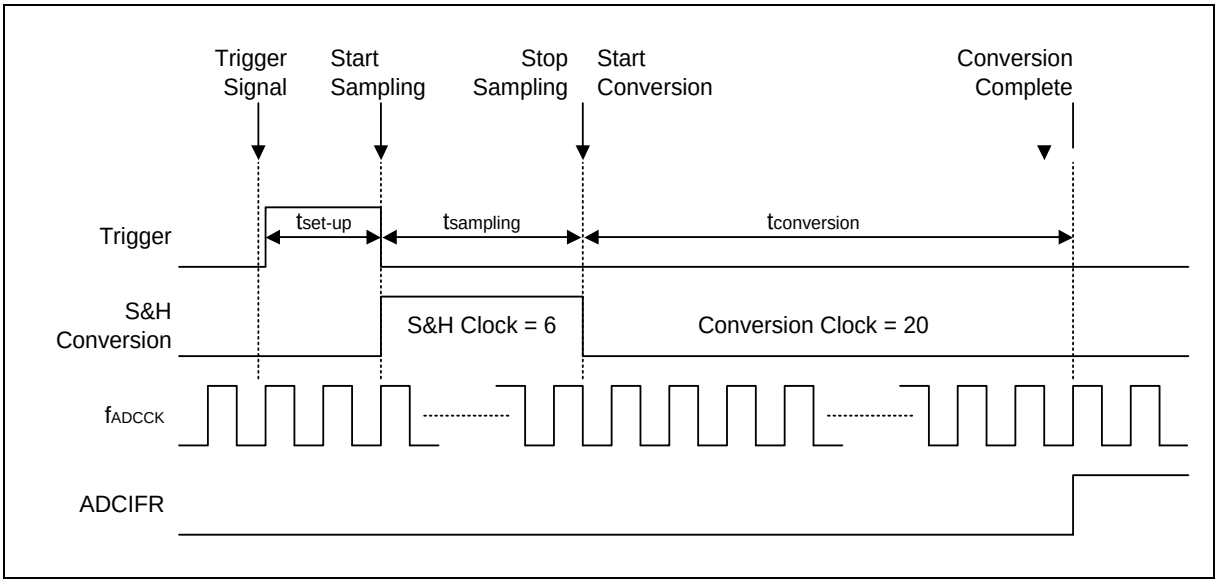


Figure 41. ADC Timing Chart

11.4 Register map

Table 15. 10-bit ADC Register Map

Name	Address	Direction	Default	Description
ADCCRH	CBH	R/W	00H	A/D Converter Control High Register
ADCCRL	CAH	R/W	00H	A/D Converter Control Low Register
ADCDRH	CDH	R	xxH	A/D Converter Data High Register
ADCDRL	CCH	R	xxH	A/D Converter Data Low Register
LDOCR	CEH	R/W	00H	LDO Control Register

11.5 Register description

ADCDRH (A/D Converter Data High Register): CDH

7	6	5	4	3	2	1	0
ADDM9	ADDM8	ADDM7	ADDM6	ADDM5	ADDM4	ADDM3 ADDL9	ADDM2 ADDL8
R	R	R	R	R	R	R	R

Initial value: xxH

ADDM[9:2] MSB align, A/D Converter High Result (8-bit)
ADDL[9:8] LSB align, A/D Converter High Result (2-bit)

ADCDRL (A/D Converter Data Low Register): CCH

7	6	5	4	3	2	1	0
ADDM1 ADDL7	ADDM0 ADDL6	ADDL5	ADDL4	ADDL3	ADDL2	ADDL1	ADDL0
R	R	R	R	R	R	R	R

Initial value: xxH

ADDM[1:0] MSB align, A/D Converter Low Result (2-bit)
ADDL[7:0] LSB align, A/D Converter Low Result (8-bit)

ADCCRH (A/D Converter Control High Register): CBH

7	6	5	4	3	2	1	0
ADCIFR	–	–	–	TRIG	ALIGN	CKSEL1	CKSEL0
R/W	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

ADCIFR — When ADC Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.
- 0 ADC Interrupt no generation
- 1 ADC Interrupt generation

TRIG — A/D Trigger Signal Selection
- 0 ADST
- 1 Timer 0 A match signal

ALIGN — A/D Converter data align selection.
- 0 MSB align (ADCDRH[7:0], ADCDRL[7:6])
- 1 LSB align (ADCDRH[1:0], ADCDRL[7:0])

CKSEL[1:0] — A/D Converter Clock selection

CKSEL1	CKSEL0	Description
0	0	4MHz
0	1	2MHz
1	0	1MHz
1	1	0.5MHz

ADCCRL (A/D Converter Control Low Register): CAH

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	ADSEL3	ADSEL2	ADSEL1	ADSEL0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Initial value: 00H

STBY Control Operation of A/D
(The ADC module is automatically disabled at stop mode)
0 ADC module disable
1 ADC module enable

ADST Control Trigger Signal for Conversion Start.
0 No effect
1 Trigger signal generation for conversion start

REFSEL A/D Converter Reference Selection
0 Internal Reference (VDD)
1 Internal LDO out (LDO23)

AFLAG A/D Converter Operation State (This bit is cleared to '0' when the STBY bit is set to '0' or when the CPU is at STOP mode)
0 During A/D Conversion
1 A/D Conversion finished

ADSEL[3:0] A/D Converter input selection

ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
1	0	0	1	Output of OP-AMP 0
1	0	1	0	Output of OP-AMP 1
1	1	1	1	V_{BGR}(0.92V)
Other values				Not available

LDOCR (LDO Control Register): CEH

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	LDOEN
–	–	–	–	–	–	–	R/W

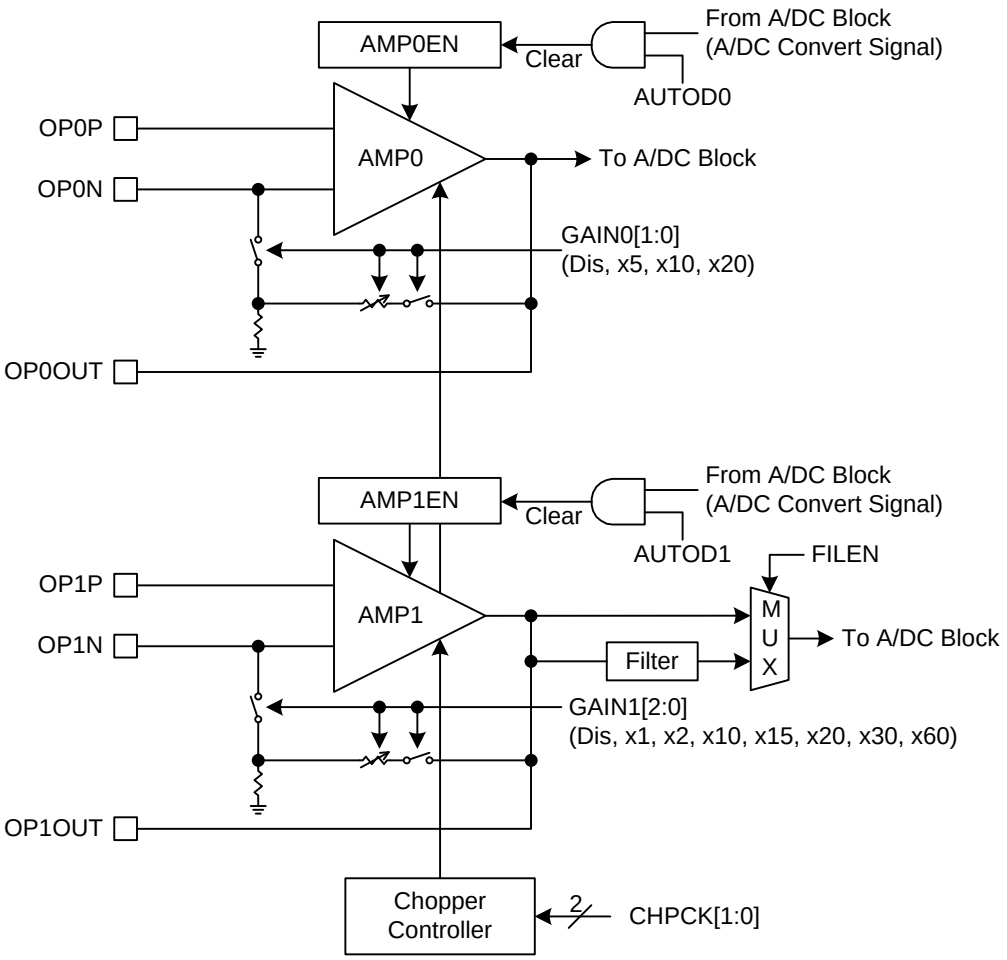
Initial value: 00H

LDOEN Control Operation of LDO
0 Disable LDO block
1 Enable LDO block

12 Operational amplifier

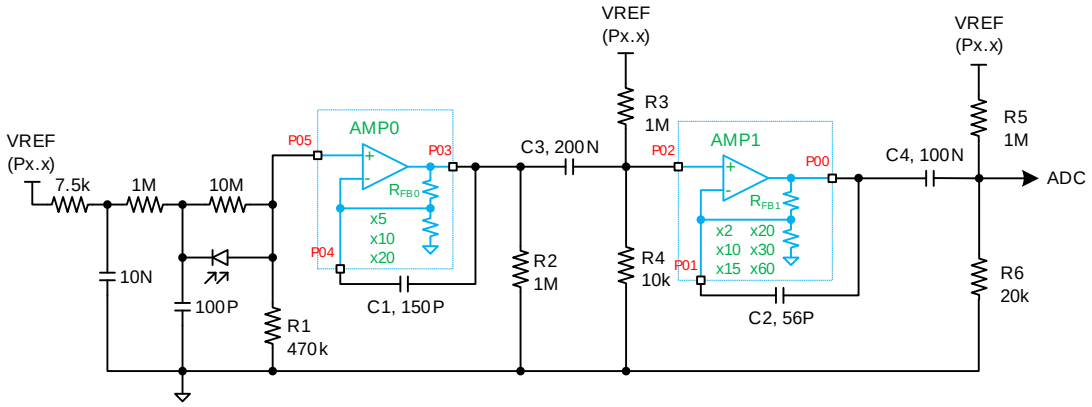
A96L414/A96L416 offers two channels of an operational amplifier (OP-Amp). OP-Amp consists of three registers such as OP-AMP control register 0 (AMPCR0), OP-AMP control register 1 (AMPCR1), and Chopper control register (CHPCR).

12.1 Block diagram



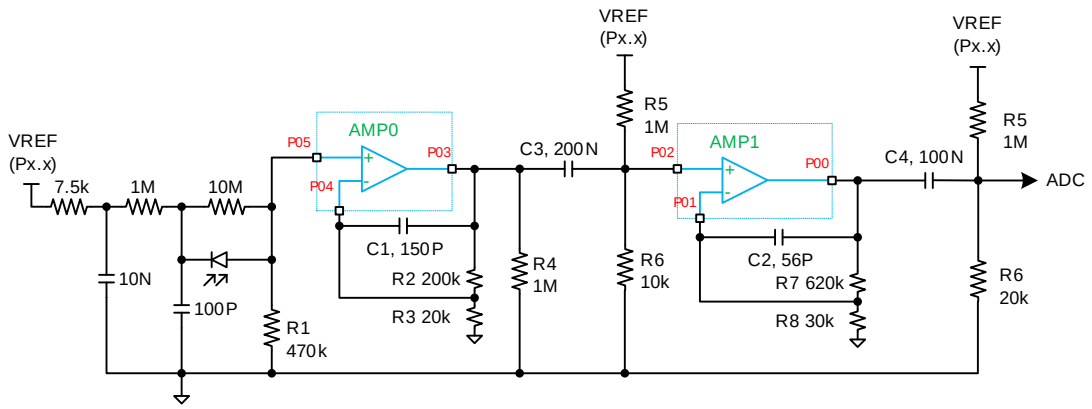
NOTE: Feedback resistors for gain on each Amp are disconnected during one or more Amp and gain are disabled.

Figure 42. OP Amp Block Diagram



NOTE: It is recommended to set the OP-AMP enabled and then VREF on.

Figure 43. Recommend Circuit for Internal Gain.



NOTE: It is recommended to set the OP-AMP enabled and then VREF on.

Figure 44. Recommend Circuit for External Gain.

12.2 Register map

Table 16. OP Amp Register Map

Name	Address	Direction	Default	Description
CHPCR	ADH	R/W	00H	Chopper Control Register
AMPCR0	AEH	R/W	00H	OP-AMP Control Register 0
AMPCR1	AFH	R/W	00H	OP-AMP Control Register 1

12.3 Register description

AMPCR0 (Operational Amplifier Control Register 0): AEH

7	6	5	4	3	2	1	0
–	FILEN	–	–	–	–	AUTOD1	AUTOD0
–	R/W	–	–	–	–	R/W	R/W

Initial value: 00H

FILEN Filter Control bit
- 0 Disable filter and select no filtered OP1OUT for ADC
- 1 Enable filter and select the filtered OP1OUT for ADC

AUTOD1 Control disable of OP-AMP1 Block
- 0 Not automatically disable
- 1 Automatically disable by A/DC convert signal

AUTOD0 Control disable of OP-AMP0 Block.
- 0 Not automatically disable
- 1 Automatically disable by A/DC convert signal

NOTE: The FILEN bits should always be '0'.

AMPCR1 (Operational Amplifier Control Register 1): AFH

7	6	5	4	3	2	1	0
AMP1EN	GAIN12	GAIN11	GAIN10	AMP0EN	–	GAIN01	GAIN00
R/W	R/W	R/W	R/W	R/W	–	R/W	R/W

Initial value: 00H

AMP1EN Control operation of OP-AMP1 Block, This bit is automatically cleared by A/DC convert signal when the AUTOD1 bit is "1".

0	OP-AMP1 block disable
1	OP-AMP1 block enable

GAIN1[2:0] Select Gain of OP-AMP1

GAIN12	GAIN11	GAIN10	Description
0	0	0	Disable gain
0	0	1	x1
0	1	0	x2
0	1	1	x10
1	0	0	x15
1	0	1	x20
1	1	0	x30
1	1	1	x60

AMP0EN Control operation of OP-AMP0 Block, This bit is automatically cleared by A/DC convert signal when the AUTOD0 bit is "1".

0	OP-AMP0 block disable
1	OP-AMP0 block enable

GAIN0[1:0] Select Gain of OP-AMP0

GAIN01	GAIN00	Description
0	0	Disable gain
0	1	x5
1	0	x10
1	1	x20

CHPCR (Chopper Control Register): ADH

7	6	5	4	3	2	1	0
–	–	–	–	–	–	CHPCK1	CHPCK0
–	–	–	–	–	–	R/W	R/W

Initial value: 00H

CHPCK[1:0] Chopper Clock Selection bits

CHPCK1	CHPCK0	Description
0	0	125 kHz
0	1	167 kHz
1	0	Not available
1	1	Not available

NOTE: It is recommended to always keep this register at 0x00 to optimize the OP-Amp noise characteristics.

13 USART

USART (Universal Synchronous/Asynchronous Receiver/Transmitter) is a microchip that facilitates communication through a computer's serial port using RS-232C protocol. A96L414/A96L416 incorporates a USART function block inside. The USART function block consists of USART control register1/2/3/4, USART status register, USART baud-rate generation register and USART data register.

Operation mode is selected by the operation mode of USART selection bits (USTMS[1:0]).

It has three operating modes as listed in the followings:

- Asynchronous mode (UART)
- Synchronous mode (USART)
- SPI mode

13.1 USART UART mode

Universal Asynchronous serial Receiver and Transmitter (UART) is a highly flexible serial communication device. Its main features are listed below:

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data Overrun Detection
- Framing Error Detection
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete

The UART has a baud rate generator, a transmitter and a receiver. A baud rate generator is used for asynchronous operation. A transmitter consists of a single write buffer, a serial shift register, parity generator and control logic, and is used for handling different serial frame formats. A write buffer allows continuous transfer of data without any delay between frames. A receiver is the most complex part of the UART module because of its clock and data recovery units. A recovery unit is used for asynchronous data reception. In addition to the recovery unit, the receiver includes a parity checker, a shift register, a two-level receive FIFO (USTDR) and control logic. The receiver supports identical frame formats to the transmitter's and can detect frame error, data overrun and parity errors.

13.2 UART block diagram

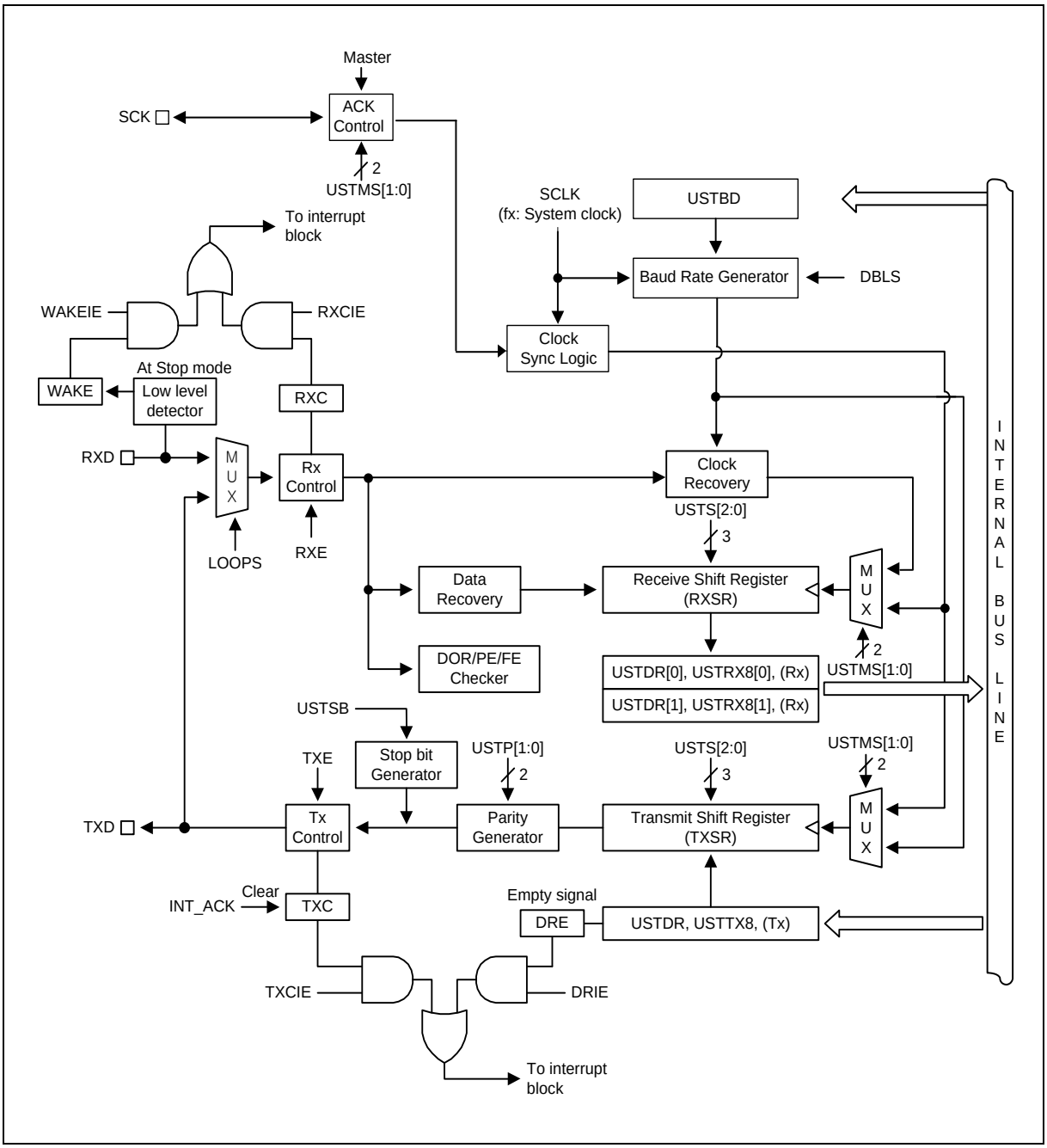


Figure 45. UART Block Diagram

13.3 Clock generator

A clock generation logic generates a base clock for the transmitter and the receiver. The USART supports four modes of clock operation such as normal asynchronous mode, double speed asynchronous mode, master synchronous mode and slave synchronous mode.

A clock generation scheme for master SPI and slave SPI mode is the same as master synchronous and slave synchronous operation mode. By configuring USTMS[1:0] bits in USTCR1 register, asynchronous operation or synchronous operation can be selected. Asynchronous double speed mode is controlled by the DBLS bit in the USTCR2 register.

MASTER bit in USTCR3 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCK pin is active only when the USART operates in synchronous or SPI mode.

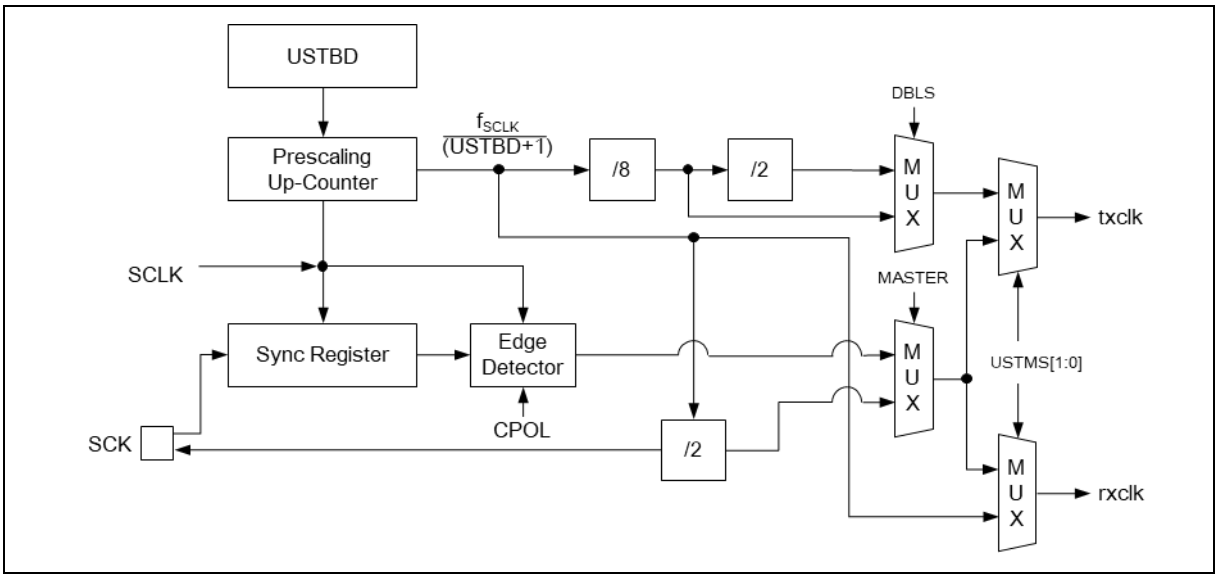


Figure 46. Clock Generator Block Diagram

Table 17 introduces equations for calculating baud rate (in bps).

Table 17. Equations for Baud Rate Register Settings

Operating mode	Equation for calculating baud rate
Normal Mode(DBLS=0)	Baud Rate= fx/(16(USTBD+1))
Double Speed Mode(DBLS=1)	Baud Rate= fx/(8(USTBD+1))
Synchronous or SPI Master Mode	Baud Rate= fx/(2(USI0BD+1))

13.4 External clock (SCK)

External clocking is used in the synchronous mode of operation.

External clock input from the SCK pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and the receiver. This process brings two CPU clock period delays. The maximum frequency of the external SCK pin is limited up to 1MHz.

13.5 Synchronous mode operation

When synchronous or SPI mode is used, the SCK pin will be used as either a clock input (slave) or a clock output (master). Data is sampled and transmitter is issued on different edges of SCK clock respectively.

For example, if data input on RXD (MISO in SPI mode) pin is sampled on the rising edge of SCK clock, data output on TXD (MOSI in SPI mode) pin is altered on the falling edge.

By configuring CPOL bit in USTCR1 register, an edge of SCK clock for data sampling and for data change can be selected. As shown in the Figure 47 below, when CPOL is zero, the data will be changed at rising SCK edge and sampled at falling SCK0 edge.

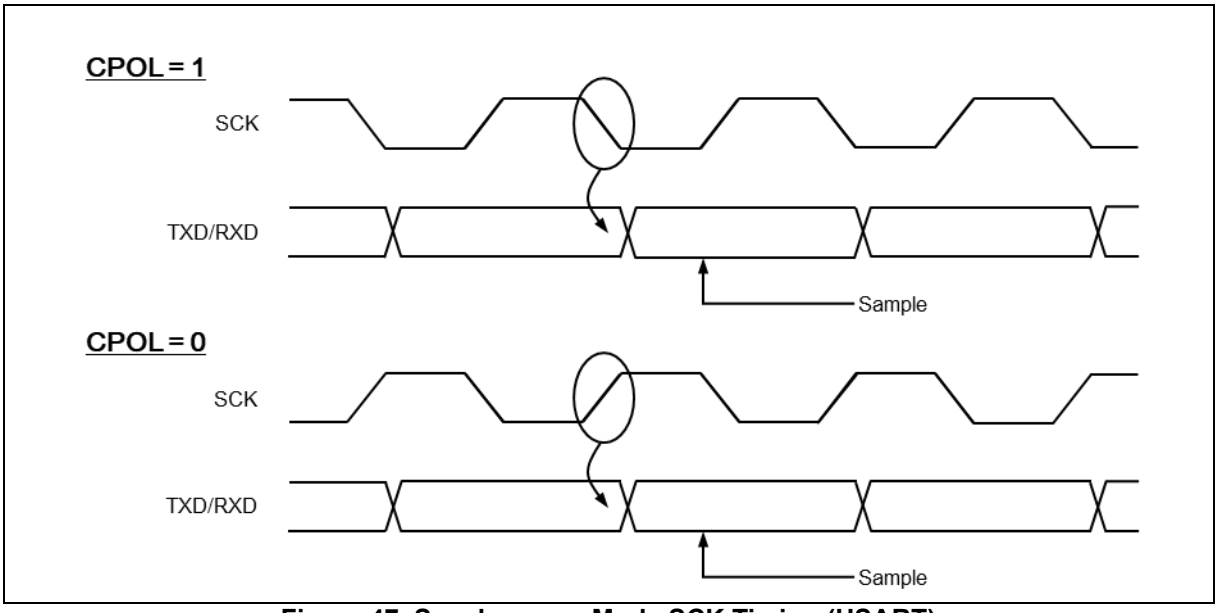


Figure 47. Synchronous Mode SCK Timing (USART)

13.6 Data format

A serial frame is defined as a single character of data bits that consist of synchronization bits (start and stop bits), and optionally a parity bit for error detection.

The UART supports 30 combinations of the followings as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with a start bit, followed by the least significant data bit (LSB). Then the next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If parity function is enabled, the parity bit is inserted between the last data bit and the stop bit. A high-to-low signal transition on a data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line enters to an idle state. The idle means high state of data pin.

The following Figure 48 shows possible combinations of the frame format. Bits inside brackets are optional.

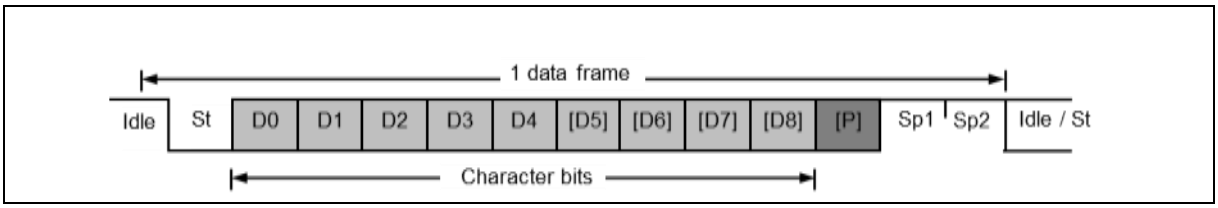


Figure 48. Frame Format Diagram

Every single frame will have the following bits:

- Idle: no communication on communication line (TxD/RxD)
- St: starting the frame (Start bit: Low)
- Dn: data bits (0 to 8)
- Parity bit: even parity, odd parity, no parity
- Stop bit(s): end of the frame (1 bit or 2 bits)

Frame format of the UART is defined by configuring USTS [2:0], USTP [1:0] and USTSB in registers USTCR1 and USTCR3. The transmitter and the receiver use the same settings.

13.7 Parity bit

The parity bit is calculated by doing XOR of all data bits. If odd parity is used, result of the XOR is inverted. The parity bit is located between the MSB and the first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$

$P_{odd} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$

P_{even}: Parity bit using even parity

P_{odd}: Parity bit using odd parity

D_n: Data bit n of the character

13.8 UART transmitter

UART transmitter is enabled by setting TXE bit in USTCR2 register. When the Transmitter is enabled, TXD pin must be set to TXD function for the serial output pin of UART. This can be done by configuring P2FSR[7:6]. Baud-rate, operation mode and frame format must be set once before starting any transmission.

13.8.1 Sending Tx data

Data transmission is initiated by loading a transmit buffer (USTDR register I/O location) with data to be transmitted.

The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used, the ninth bit must be written to the TX8 bit in USTCR3 register before it is loaded to the transmit buffer (USTDR register).

13.8.2 Transmitter flag and interrupt

The UART transmitter has 2 flags indicating its status. One is a UART data register empty flag (UDRE) and the other is transmit complete flag (TXC). Both flags can be interrupt sources. UDRE flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty, and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register. This flag can be cleared by writing '0' to this bit position too. Writing '1' to this bit position is prevented.

When the data register empty interrupt enable (UDRIE) bit in USTCR2 register is set and the global interrupt is enabled, UART data register empty interrupt is generated while UDRE flag is set.

The transmit complete (TXC) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer. The TXC flag is automatically cleared when the transmit complete interrupt service routine is executed, or it can be cleared by writing '0' to TXC bit in USTST register.

When the transmit complete interrupt enable (TXCIE) bit in USTCR2 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXC flag is set.

13.8.3 Parity generator

A parity generator calculates a parity bit for the serial frame data to be sent. When parity bit is enabled (USTP[1] = 1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

13.8.4 Disabling transmitter

Disabling the transmitter by clearing TXE bit will not be effective until ongoing transmission is completed. When the Transmitter is disabled, the TXD pin can be used as a normal general purpose I/O (GPIO).

13.9 UART receiver

UART receiver is enabled by setting RXE bit in USTCR2 register. When the receiver is enabled, the RXD pin must be set to RXD function for the serial input pin of UART. This can be done by configuring by P2FSR [5:4]. Baud-rate, mode of operation and frame format must be set before starting serial reception.

13.9.1 Receiving Rx data

The receiver starts data reception when it detects a valid start bit (LOW) on RXD pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2nd stop bit in the frame, the 2nd stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the USTDR register.

If 9-bit characters are used (USTS[2:0] = "111"), the ninth bit is stored in the RX8 bit position in the USTCR3 register. The 9th bit must be read from the RX8 bit before reading the low 8 bits from the USTDR register. Likewise, the error flags FE, DOR, PE must be read before reading the data from USTDR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

13.9.2 Receiver flag and interrupt

The UART receiver has one flag that indicates the receiver's status. Receive complete (RXC) flag indicates whether there are unread data in the receive buffer. This flag is set when there are unread data in the receive buffer, and cleared when the receive buffer is empty. If the receiver is disabled (RXE=0), the receiver buffer is flushed and the RXC flag is cleared.

When receive complete interrupt enable (RXCIE) bit in register USTCR2 is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXC flag is set.

The UART receiver has three error flags which are frame error (FE), data overrun (DOR) and parity error (PE). These error flags can be read from the USTST register. As received data are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from USTDR register, read the USTST register first which contains error flags.

The frame error (FE) flag indicates the state of the first stop bit. The FE flag is '0' when the stop bit was correctly detected as '1', and the FE flag is '1' when the stop bit was incorrect, i.e. detected as '0'. This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DOR) flag indicates data loss due to a receive buffer full condition. DOR occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DOR flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The parity error (PE) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USTP[1] = 0), the PE bit is always read '0'.

13.9.3 Parity checker

Parity checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame, if parity bit is enabled (USTP[1]=1).

13.9.4 Disabling receiver

Unlike the transmitter, disabling the Receiver by clearing RXE bit makes the receiver inactive immediately. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXD pin can be used as a normal general purpose I/O (GPIO).

13.9.5 Asynchronous data reception

To receive asynchronous data frame, the UART has a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXD pin. The data recovery logic samples and low pass filters the incoming bits, to remove the noise of RXD pin.

Figure 49 illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud-rate for normal mode (DBLS=0) and 8 times the baud-rate for double speed mode (DBLS=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the double speed mode.

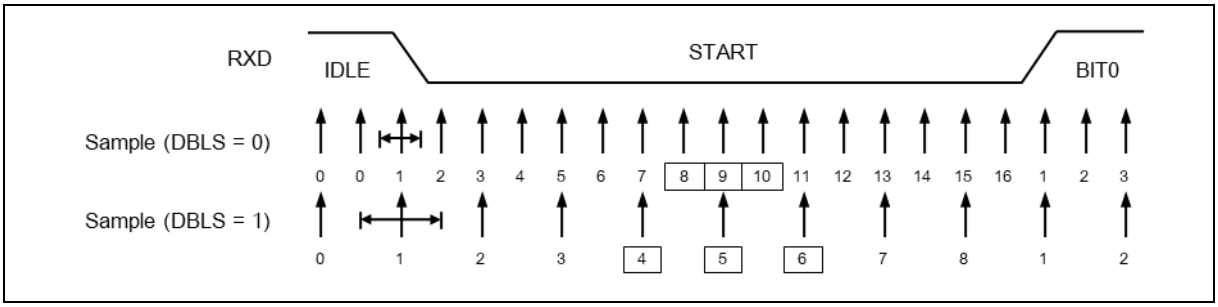


Figure 49. Start Bit Sampling

When the receiver is enabled (RXE=1), the clock recovery logic tries to find a high-to-low transition on the RXD line, the start bit condition.

After detecting high to low transition on RXD line, the clock recovery logic uses samples 8, 9, and 10 for normal mode, and samples 4, 5, and 6 for double speed mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described in Figure 50, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for normal mode and 8 times for double speed mode. And uses sample 8, 9, and 10 to decide data value for normal mode, and samples 4, 5, and 6 for double speed mode. If more than 2 samples have low levels, the received bit is considered to a logic '0' and if more than 2 samples have high levels, the received bit is considered to a logic '1'. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order.

Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

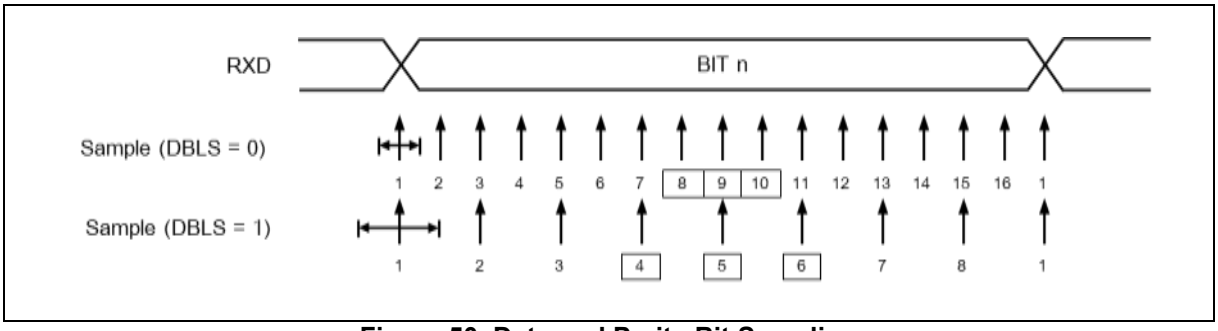


Figure 50. Data and Parity Bit Sampling

The process for detecting stop bit is like clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (FE) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXD line to check a valid high to low transition is detected (start bit detection).

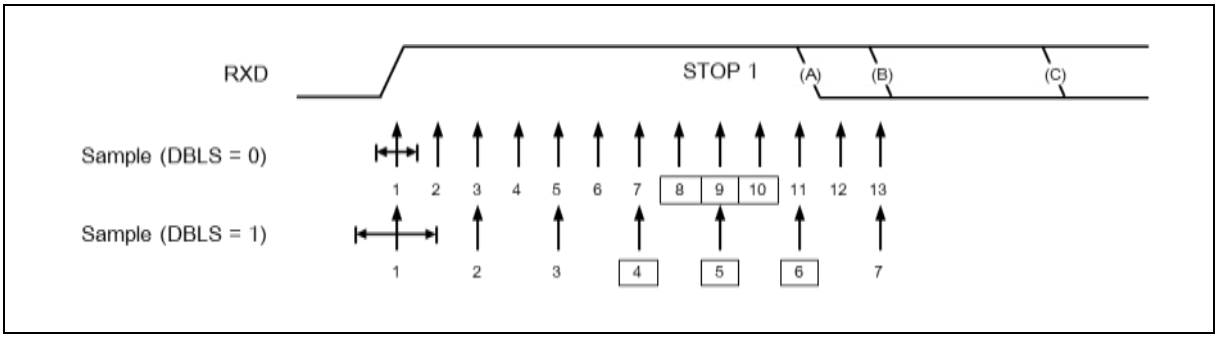


Figure 51. Stop Bit Sampling and Next Stop Bit Sampling

13.10 USART SPI mode

USART can be configured to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full Duplex, Three-wire synchronous data transfer
- Mater and Slave Operation
- Supports all four SPI modes of operation (mode 0, 1, 2, and 3)
- Selectable LSB first or MSB first data transfer
- Double buffered transmit and receive
- Programmable transmit bit rate

When SPI mode is enabled (USTMS[1:0]="11"), the slave select (SS) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USTSSEN bit is set to '0'.

Note that during SPI mode of operation, the pin RXD is renamed as MISO and TXD is renamed as MOSI for compatibility to other SPI devices.

13.11 SPI block diagram

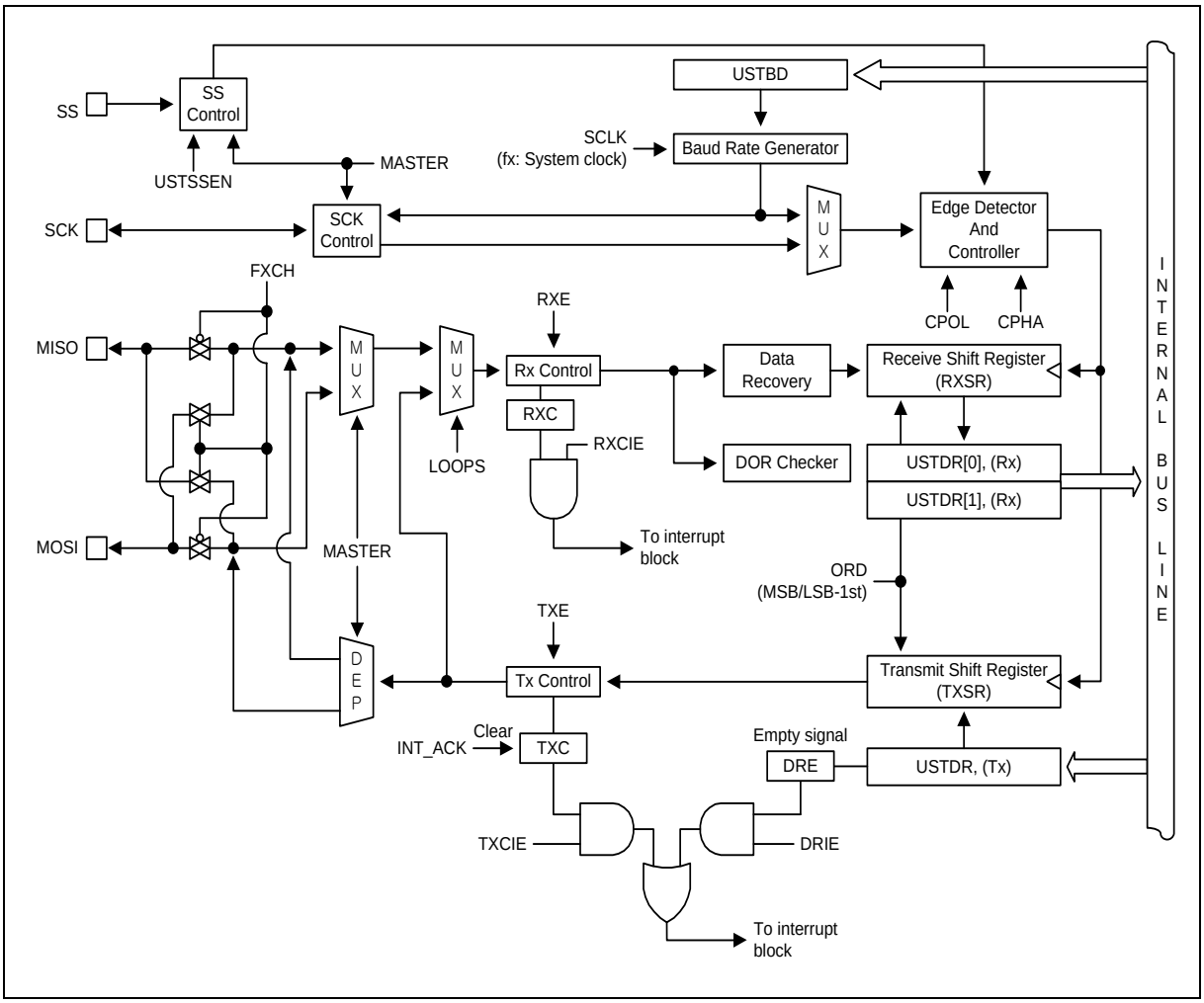


Figure 52. SPI Block Diagram

13.12 SPI clock formats and timing

To accommodate a wide variety if synchronous serial peripherals from different manufacturers, the USART has a clock polarity bit (CPOL) and a clock phase control bit (CPHA) to select one of four clock formats for data transfers. CPOL selectively insert an inverter in series with the clock. CPHA chooses between two different clock phase relationships between the clock and data. Note that CPHA and CPOL bits in USTCR1 register have different meanings according to the USTMS[1:0] bits which decides the operating mode of USART.

Table 18 introduces four combinations of CPOL and CPHA for SPI mode 0, 1, 2 and 3.

Table 18. CPOL Functionality

SPI Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

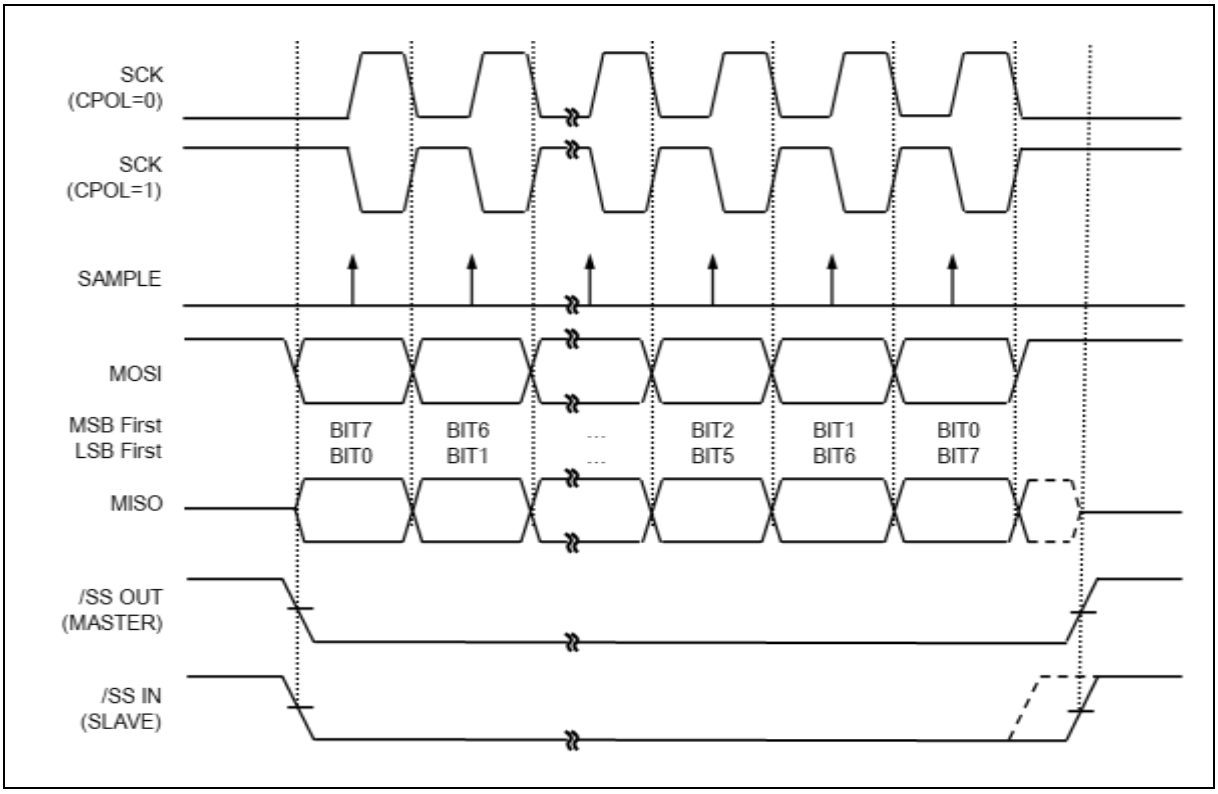


Figure 53. SPI Clock Formats when CPHA=0

When CPHA=0, the slave begins to drive its MISO output with the first data bit value when SS goes to active low. The first SCK edge causes both the master and the slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the second SCK edge, the USART shifts the second data bit value out to the MOSI and MISO outputs of the master and slave, respectively. Unlike the case of CPHA=1, when CPHA=0, the slave's SS input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SS input.

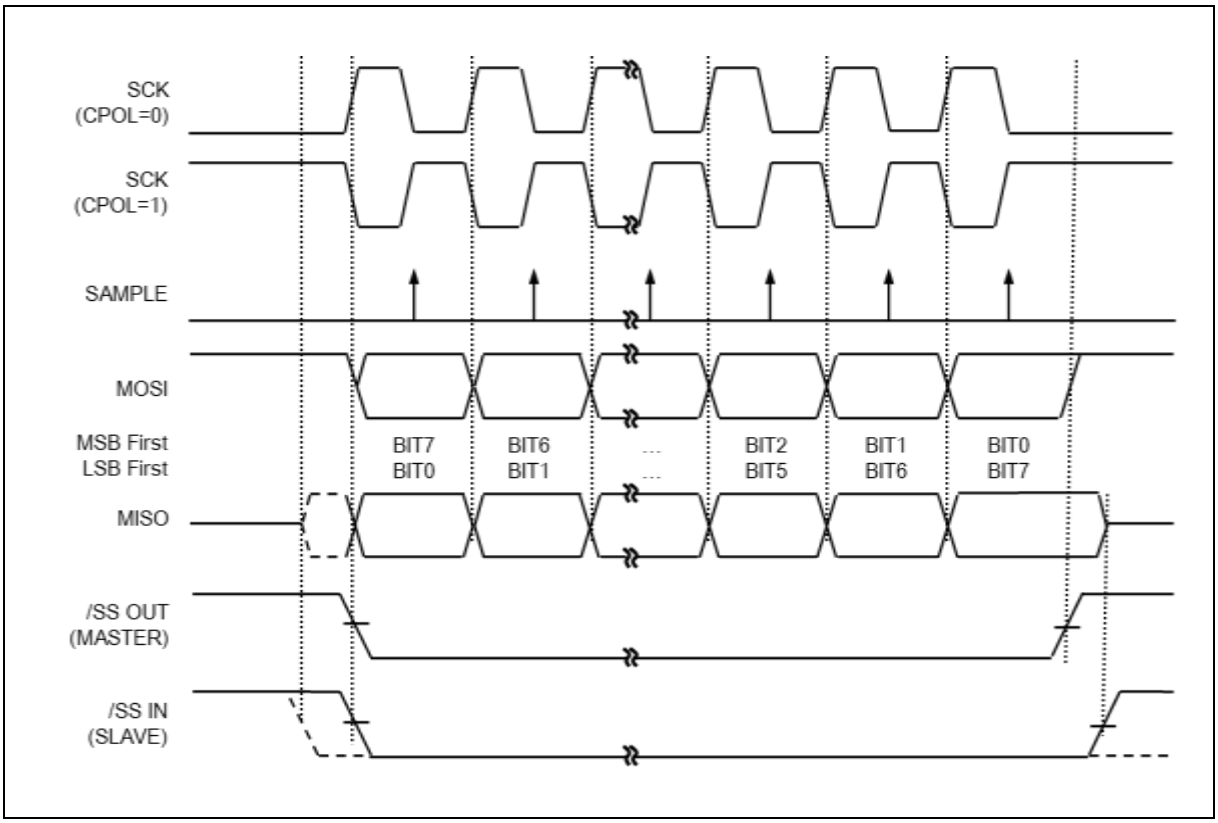


Figure 54. SPI Clock Formats when CPHA=1

When CPHA=1, the slave begins to drive its MISO output when SS goes active low, but the data is not defined until the first SCK edge. The first SCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next SCK edge causes both the master and slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the third SCK edge, the USART shifts the second data bit value out to the MOSI and MISO output of the master and slave respectively. When CPHA=1, the slave's SS input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USART resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USART Data Register Empty flag (DRE=1) and then writing a byte of data to the USTDR Register. In master mode of operation, even if transmission is not enabled (TXE=0), writing data to the USTDR register is necessary because the clock SCK is generated from transmitter block.

13.13 Register map

Table 19. USART Register Map

Name	Address	Direction	Default	Description
USTCR1	DAH	R/W	00H	USART Control Register 1
USTCR2	DBH	R/W	00H	USART Control Register 2
USTCR3	DCH	R/W	00H	USART Control Register 3
USTST	DDH	R/W	80H	USART Status Register
USTBD	DEH	R/W	FFH	USART Baud Rate Generation Register
USTDR	DFH	R/W	00H	USART Data Register

13.14 Register description

USTBD (USART Baud-Rate Generation Register): DEH

7	6	5	4	3	2	1	0
USTBD7	USTBD6	USTBD5	USTBD4	USTBD3	USTBD2	USTBD1	USTBD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

USTBD[7:0]	The value in this register is used to generate internal baud rate in UART mode or to generate SCK clock in SPI mode. To prevent malfunction, do not write '0' in UART mode and do not write '0' or '1' in synchronous or SPI mode.

USTDR (USART Data Register): DFH

7	6	5	4	3	2	1	0
USTDR7	USTDR6	USTDR5	USTDR4	USTDR3	USTDR2	USTDR1	USTDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

USTDR[7:0]	The USART Transmit buffer and Receive buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the USTDR register. Reading the USTDR register returns the contents of the Receive Buffer. Write to this register only when the DRE flag is set. In SPI master mode, the SCK clock is generated when data are written to this register.

USTCR1 (USART Control Register 1): DAH

7	6	5	4	3	2	1	0
USTMS1	USTMS0	USTP1	USTP0	USTS2	USTS1 ORD	USTS0 CPHA	CPOL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

USTMS[1:0] Selects Operation Mode of USART

USTMS1	USTMS0	Operation mode
0	0	Asynchronous Mode (UART)
0	1	Synchronous Mode
1	0	Reserved
1	1	SPI mode

USTP[1:0] Selects Parity Generation and Check method (only UART mode)

USTP1	USTP0	Parity
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

USTS[2:0] When in Asynchronous or Synchronous mode of operation, selects the length of data bits in a frame.

USTS2	USTS1	USTS0	Data Length
0	0	0	5 bit
0	0	1	6 bit
0	1	0	7 bit
0	1	1	8 bit
1	1	1	9 bit
Other values			Reserved

ORD This bit is in the same bit position with USTS1. The MSB of the data byte is transmitted first when set to '1' and the LSB when set to '0' (only SPI mode)

0	LSB-first
1	MSB-first

CPOL This bit determines the clock polarity of ACK in synchronous or SPI mode

0	TXD Change @Rising Edge, RXD Change @Falling Edge
1	TXD Change @Falling Edge, RXD Change @Rising Edge

CPHA This bit is in the same bit position with USTS0. This bit determines if data are sampled on the leading or trailing edge of SCK (only SPI mode)

CPOL	CPHA	Leading edge	Trailing edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

USTCR2 (USART Control Register 2): DBH

7	6	5	4	3	2	1	0
DRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USTEN	DBLS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DRIE Interrupt enable bit for Data Register
- 0 Interrupt from DRE is inhibited (use polling)
- 1 When DRE is set, request an interrupt

TXCIE Interrupt enable bit for Transmit Complete
- 0 Interrupt from TXC is inhibited (use polling)
- 1 When TXC is set, request an interrupt

RXCIE Interrupt enable bit for Receive Complete
- 0 Interrupt from RXC is inhibited (use polling)
- 1 When RXC is set, request an interrupt

WAKEIE Interrupt enable bit for Asynchronous Wake in STOP mode. When device is in stop mode, if RXD goes to Low level, an interrupt can be requested to wake-up system (only UART mode)
- 0 Interrupt from Wake is inhibited
- 1 When WAKE is set, request an interrupt

TXE Enables the Transmitter unit
- 0 Transmitter is disabled
- 1 Transmitter is enabled

RXE Enables the Receiver unit
- 0 Receiver is disabled
- 1 Receiver is enabled

USTEN Activate USART Function Block by supplying.
- 0 USART is disabled
- 1 USART is enabled

DBLS This bit selects receiver sampling rate (only UART mode)
- 0 Normal asynchronous operation
- 1 Double speed asynchronous operation

USTCR3 (USART Control Register 3): DCH

7	6	5	4	3	2	1	0
MASTER	LOOPS	DISSCK	USTSSEN	FXCH	USTSB	USTTX8	USTRX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

MASTER	Selects master or slave in SPI or Synchronous mode operation and controls the direction of SCK pin.	
	0	Slave operation (External clock for SCK)
	1	Master operation (Internal clock for SCK)
LOOPS	Control the Loop Back mode of USART for test mode	
	0	Normal operation
	1	Loop Back mode
DISSCK	In synchronous mode operation, selects the waveform of SCK output.	
	0	SCK is free-running while UART is enabled in synchronous master mode
	1	SCK is active while any frame is on transferring
USTSSEN	This bit controls the SS pin operation (only SPI mode)	
	0	Disable
	1	Enable (The SS pin should be a normal input)
FXCH	SPI port function exchange control bit (only SPI mode)	
	0	No effect
	1	Exchange MOSI and MISO function
USTSB	Selects the length of stop bit in Asynchronous or Synchronous mode of operation.	
	0	1 Stop bit
	1	2 Stop bit
USTTX8	The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Write this bit first before loading the USTDR register.	
	0	MSB (9th bit) to be transmitter is '0'
	1	MSB (9th bit) to be transmitter is '1'
USTRX8	The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Read this bit first before reading the receive buffer (only UART mode)	
	0	MSB (9th bit) to be received is '0'
	1	MSB (9th bit) to be received is '1'

USTST (USART Status Register): DDH

7	6	5	4	3	2	1	0
DRE	TXC	RXC	WAKE	USTRST	DOR	FE	PE
R/W	R/W	R	R/W	R/W	R	R/W	R/W

Initial value: 80H

DRE — The DRE flag indicates if the transmit buffer (USTDR) is ready to receive new data. If DRE is '1', the buffer is empty and ready to be written. The flag can generate a DRE interrupt.

0 Transmit buffer is not empty

1 Transmit buffer is empty

TXC — This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXC interrupt is executed. This flag can generate a TXC interrupt.

0 Transmission is ongoing

1 Transmit buffer is empty and the data in transmit shift register are shifted out completely

RXC — This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read. The RXC flag can be used to generate a RXC interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKE — This flag is set when the RXD pin is detected low while the CPU is in STOP mode. This flag can be used to generate a WAKE interrupt (only UART mode)

0 No WAKE interrupt is generated

1 WAKE interrupt is generated

USTRST — This is an internal reset and only has effect on USART. Writing '1' to this bit initializes the internal logic of USART and is automatically cleared to '0'.

0 No effect

1 Reset USART

DOR — This bit is set if data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.

0 No Data OverRun

1 Data OverRun detected

FE — This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read (only UART mode)

0 No Frame Error

1 Frame Error detected

PE — This bit is set if the next character in the receive buffer has a Parity Error while Parity Checking is enabled. This bit is valid until the receive buffer is read (only UART mode)

0 No Parity Error

1 Parity Error detected

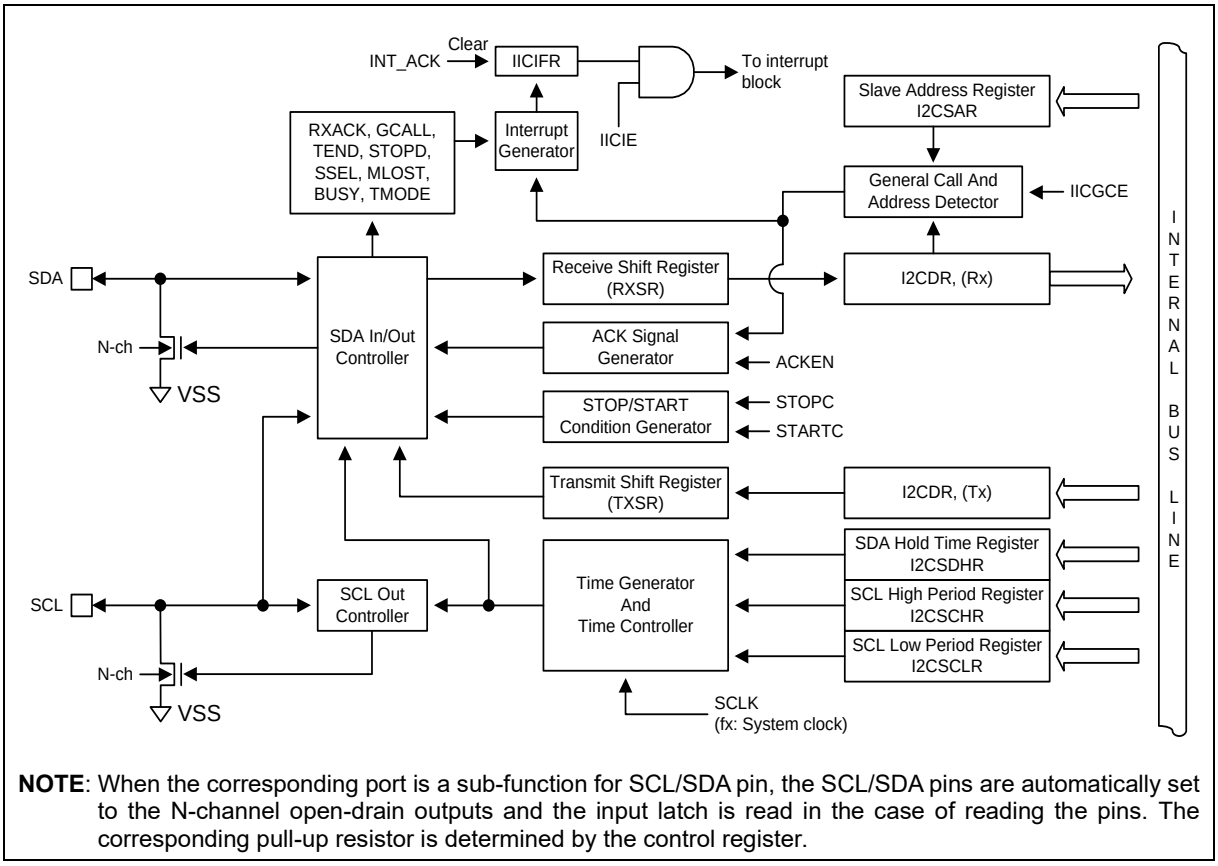
14 Inter Integrated Circuit (I2C)

Inter Integrated Circuit (I2C) is one of industrial standard serial communication protocols. It uses two bus lines such as Serial Data Line (SDA) and Serial Clock Line (SCL) to exchange data. Because both SDA and SCL lines are open-drain output, each line needs pull-up resistor.

Features of the I2C are shown below:

- Compatible with I2C bus standard
- Multi-master operation
- Up to 400kHz data transfer read speed
- 7 bit address
- Support two slave address
- Both master and slave operation
- Bus busy detection

14.1 Block diagram



NOTE: When the corresponding port is a sub-function for SCL/SDA pin, the SCL/SDA pins are automatically set to the N-channel open-drain outputs and the input latch is read in the case of reading the pins. The corresponding pull-up resistor is determined by the control register.

Figure 55. I2C Block Diagram

14.2 I2C bit transfer

The data on the SDA line must be stable during HIGH period of the clock, SCL. The HIGH or LOW state of the data line can only change when the clock signal in the SCL line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

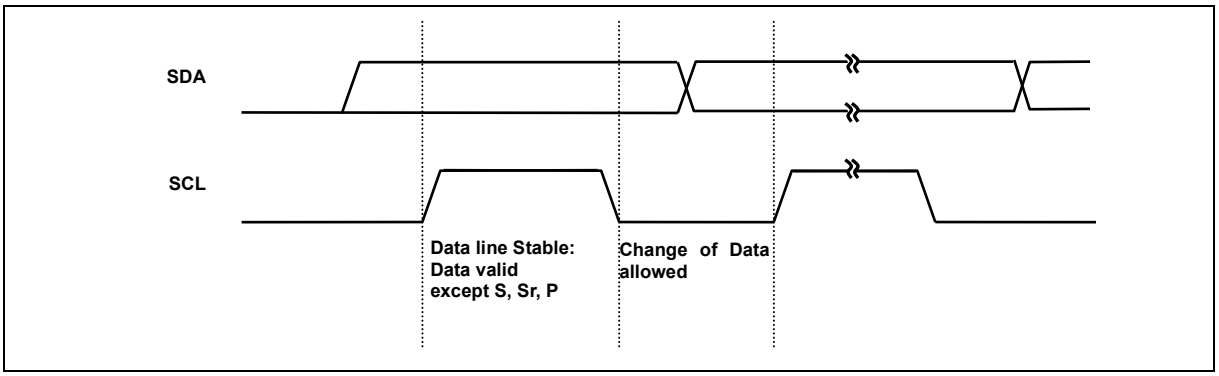


Figure 56. Bit Transfer in the I2C-Bus

14.3 Start/ Repeated Start/ Stop

One master can issue a START (S) condition to recognize other devices connected to the SCL, SDA lines that it will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

A high to low transition on the SDA line while SCL is high defines a START (S) condition.

A low to high transition on the SDA line while SCL is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, i.e., the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays busy. So, the START and repeated START conditions are functionally identical.

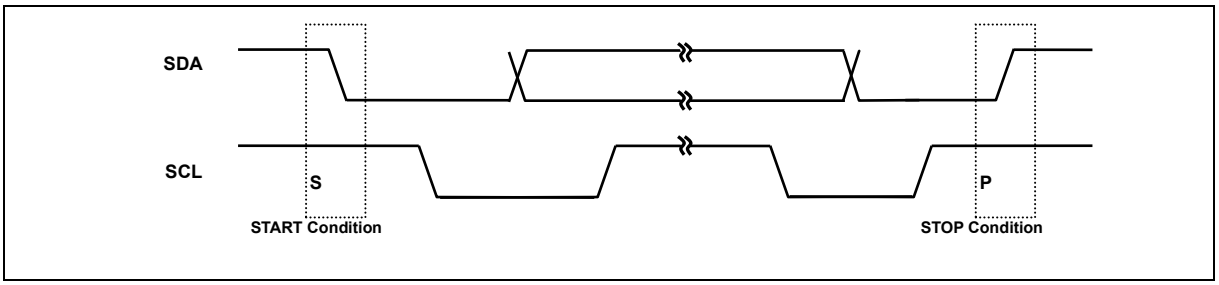


Figure 57. START and STOP Condition

14.4 Data transfer

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCL LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line.

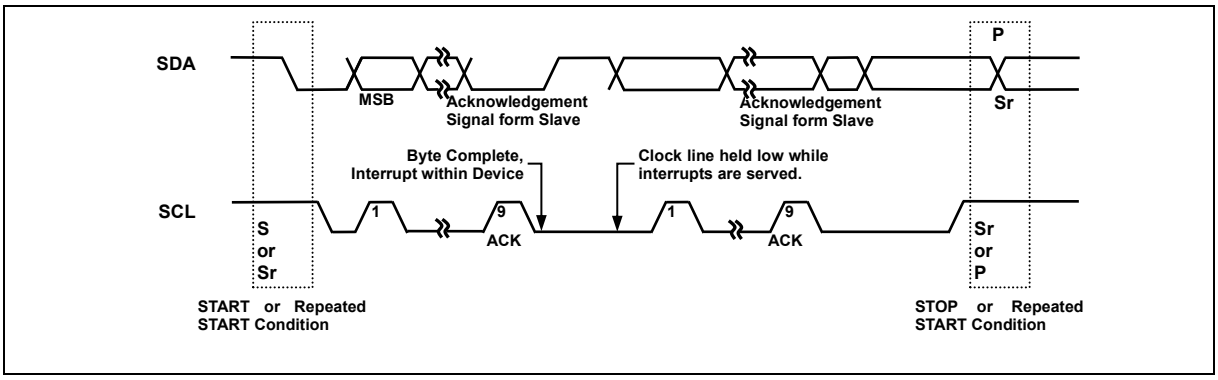


Figure 58. Data Transfer on the I2C-Bus

14.5 I2C acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it's performing some real time function, the data line must be left HIGH by the slave. And also, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDA line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

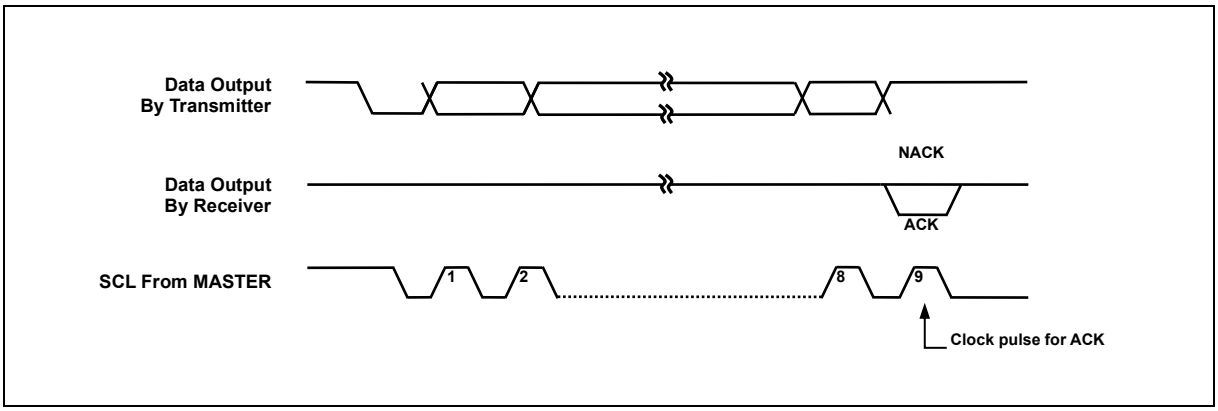


Figure 59. Acknowledge on the I2C-Bus

14.6 Synchronization/ arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and it will hold the SCL line in that state until the clock HIGH state is reached. However the LOW to HIGH transition of this clock may not change the state of the SCL line if another clock is still within its LOW period. In this way, a synchronized SCL clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition. Arbitration takes place on the SDA line, while the SCL line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output state because the level on the bus doesn't correspond to its own level. Arbitration continues for many bits until a winning master gets the ownership of I2C bus. Its first stage is comparison of the address bits.

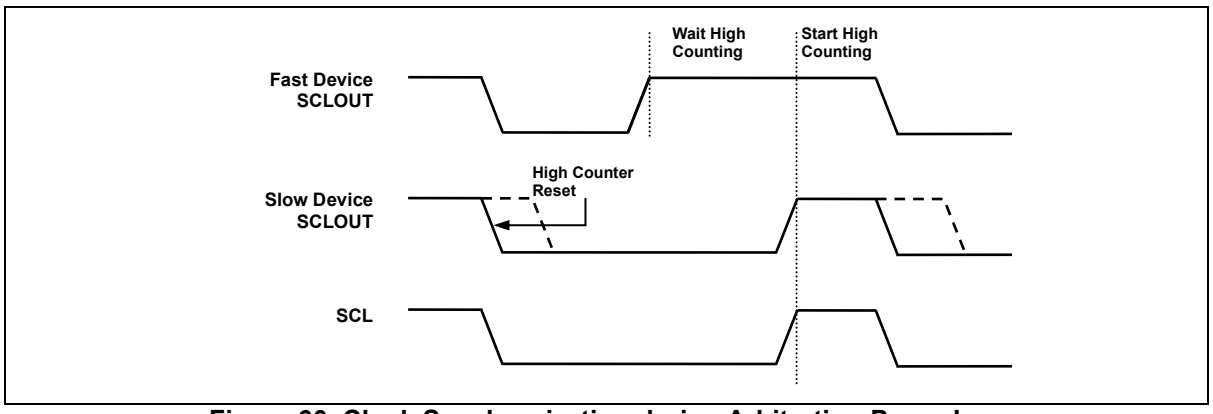


Figure 60. Clock Synchronization during Arbitration Procedure

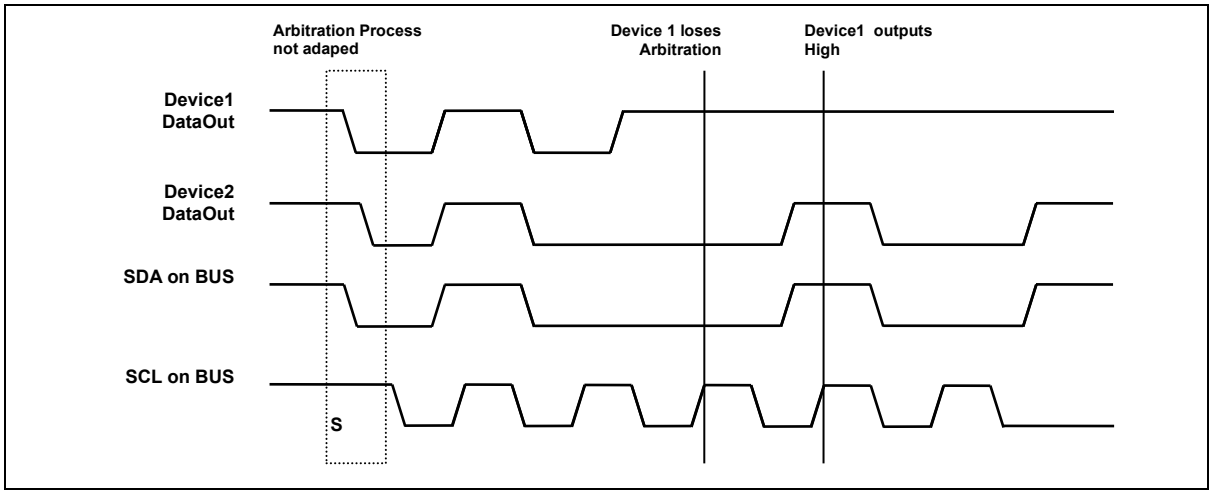


Figure 61. Arbitration Procedure of Two Masters.

14.7 Operation

The I2C operates in byte units and is based on interrupts. The interrupts are issued after all bus events except for a transmission of a START condition. Because the I2C is interrupt based, the application software is free to carry on other operations during an I2C byte transfer.

Please remember that when the I2C interrupt is generated, IICIFR flag in IIFLAG register is set and it is cleared when all interrupt source bits in the I2CSR register are cleared to "0b". When the I2C interrupt occurs, the SCL line is hold LOW until clearing "0b" all interrupt source bits in I2CSR register. When the IICIFR flag is set, the I2CSR contains a value indicating the current state of the I2C bus. According to the value in I2CSR, software can decide what to do next.

The I2C can operate in 4 modes by configuring master/slave, transmitter/receiver. The operating mode is configured by a winning master. A more detailed explanation follows below.

14.8 Master transmitter

To operate I2C in master transmitter, follow the recommended steps below.

1. Enable I2C by setting IICEN bit in I2CCR. This provides main clock to the peripheral.
2. Load SLA+W into the I2CDR where SLA is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is '0'. Note that I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLR and I2CSCHR for the Low and High period of SCL line.
4. Configure the I2CSDHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLR to the I2CSDHR.
5. Set the STARTC bit in I2CCR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTC bit is set, 8-bit data in I2CDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCL. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave.

To operate as a slave when the MLOST bit in I2CSR is set, the ACKEN bit in I2CCR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

I2C (Master) can choose one of the following cases regardless of the reception of ACK signal from slave.

1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2CDR.
2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC bit in I2CCR.
3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2CDR and set STARTC bit in I2CCR.

After doing one of the actions above, clear to "0b" all interrupt source bits in I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR and if transfer direction bit is '1' go to master receiver section.

7. 1-Byte of data is being transmitted. During data transfer, bus arbitration continues.

8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCL LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOST bit in I2CSR is set. If then, I2C waits in idle state. When the data in I2CDR is transmitted completely, I2C generates TEND interrupt.

   I2C can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2CDR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC bit in I2CCR.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2CDR and set the STARTC bit in I2CCR.

After doing one of the actions above, clear to "0b" all interrupt source bits in I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR, and if transfer direction bit is '1' go to master receiver section.

9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write "0" to I2CSR. After this, I2C enters idle state.

14.9 Master receiver

To operate I2C in master receiver, follow the recommended steps below.

1. Enable I2C by setting IICEN bit in I2CCR. This provides main clock to the peripheral.
2. Load SLA+R into the I2CDR where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. Note that I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLR and I2CSCHR for the Low and High period of SCL line.
4. Configure the I2CSDHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLR to the I2CSDHR.
5. Set the STARTC bit in I2CCR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTC bit is set, 8-bit data in I2CDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCL. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOST bit in I2CSR is set, the ACKEN bit in I2CCR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

1) Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKEN bit in I2CCR to decide whether I2C Acknowledges the next data to be received or not.

2) Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOPC bit in I2CCR.

3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA+R/W into the I2CDR and set STARTC bit in I2CCR.

After doing one of the actions above, clear to "0b" all interrupt source bits in I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR and if transfer direction bit is '0' go to master transmitter section.

7. 1-Byte of data is being received.

8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCL LOW. When 1-Byte of data is received completely, I2C generates TEND interrupt.

   I2C can choose one of the following cases according to the RXACK flag in I2CSR.

   1) Master continues receiving data from slave. To do this, set ACKEN bit in I2CCR to Acknowledge the next data to be received.

   2) Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKEN bit in I2CCR.

   3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPC bit in I2CCR.

   4) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA+R/W into the I2CDR and set the STARTC bit in I2CCR.

   After doing one of the actions above, clear to "0b" all interrupt source bits in I2CSR to release SCL line. In case of 1) and 2), move to step 7. In case of 3), move to step 9 to handle STOP interrupt. In case of 4), move to step 6 after transmitting the data in I2CDR, and if transfer direction bit is '0' go to master transmitter section.

9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write "0" value to I2CSR. After this, I2C enters idle state.

14.10 Slave transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCL, load value 0x00 into I2CSDHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICIE bit and IICEN bit in I2CCR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state i.e., waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and the SCL line is held LOW. Note that even if the address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs, load transmit data to I2CDR and clear to "0b" all interrupt source bits in I2CSR to release SCL line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TEND interrupt and holds the SCL line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

   1) No ACK signal is detected and I2C waits STOP or repeated START condition.
   2) ACK signal from master is detected. Load data to transmit into I2CDR.

   After doing one of the actions above, clear to "0b" all interrupt source bits in I2CSR to release SCL line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOPC bit indicates that data transfer between master and slave is over. To clear I2CSR, write "0" to I2CSR. After this, I2C enters idle state.

14.11 Slave receiver

To operate I2C in slave receiver, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCL, load value 0x00 into I2CSDHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICIE bit in I2CCR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state i.e., waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and the SCL line is held LOW. Note that even if the address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs and I2C is ready to receive data, clear to "0b" all interrupt source bits in I2CSR to release SCL line.
5. 1-Byte of data is being received.
6. In this step, I2C generates TEND interrupt and holds the SCL line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

   1) No ACK signal is detected (ACKEN=0) and I2C waits STOP or repeated START condition.

   2) ACK signal is detected (ACKEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, clear to "0b" all interrupt source bits in I2CSR to release SCL line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOPC bit indicates that data transfer between master and slave is over. To clear I2CSR, write "0" to I2CSR. After this, I2C enters idle state.

14.12 Register map

Table 20. I2C Register Map

Name	Address	Direction	Default	Description
I2CCR	E9H	R/W	00H	I2C Control Register
I2CSR	EAH	R/W	00H	I2C Status Register
I2CSAR0	EBH	R/W	00H	I2C Slave Address 0 Register
I2CSAR1	F1H	R/W	00H	I2C Slave Address 1 Register
I2CDR	ECH	R/W	00H	I2C Data Register
I2CSDHR	EDH	R/W	01H	I2C SDA Hold Time Register
I2CSCLR	EEH	R/W	3FH	I2C SCL Low Period Register
I2CSCHR	EFH	R/W	3FH	I2C SCL High Period Register

14.13 Register description

I2CDR (I2C Data Register): ECH

7	6	5	4	3	2	1	0
I2CDR7	I2CDR6	I2CDR5	I2CDR4	I2CDR3	I2CDR2	I2CDR1	I2CDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

I2CDR[7:0]	The I2CDR transmit buffer and receive buffer share the same I/O address with this DATA register. The transmit data buffer is the destination for data written to the I2CDR register. Reading the I2CDR register returns the contents of the receive buffer.

I2CSDHR (I2C SDA Hold Time Register): EDH

7	6	5	4	3	2	1	0
I2CSDHR7	I2CSDHR6	I2CSDHR5	I2CSDHR4	I2CSDHR3	I2CSDHR2	I2CSDHR1	I2CSDHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 01H

I2CSDHR[7:0]	The register is used to control SDA output timing from the falling edge of SCL. Note that SDA is changed after t_{SCLK} X (I2CSDHR+2), in master mode, load half the value of I2CSCLR to this register to make SDA change in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after t_{SCLK} X (I2CSDHR+2) in master mode. So, to insure operation in slave mode, the value. t_{SCLK} X (I2CSDHR +2) must be smaller than the period of SCL.

I2CSCHR (I2C SCL High Period Register): EFH

7	6	5	4	3	2	1	0
I2CSCHR7	I2CSCHR6	I2CSCHR5	I2CSCHR4	I2CSCHR3	I2CSCHR2	I2CSCHR1	I2CSCHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 3FH

I2CDR[7:0]	This register defines the high period of SCL in master mode. The transmit data buffer is the destination for data written to the I2CDR register. The base clock is SCLK, the system clock, and the period is calculated by the formula: t_{SCLK} X (4 X I2CSCHR + 2) where t_{SCLK} is the period of SCLK.

So, the operating frequency of I2C master mode is calculated by the following equation.

$$f_{I2C} = \frac{1}{t_{SCLK} \text{ X } (4 \text{ X } (I2CSCLR + I2CSCHR) + 4)}$$

I2CSCLR (I2C SCL Low Period Register): EEH

7	6	5	4	3	2	1	0
I2CSCLR7	I2CSCLR6	I2CSCLR5	I2CSCLR4	I2CSCLR3	I2CSCLR2	I2CSCLR1	I2CSCLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 3FH

I2CDR[7:0]	This register defines the low period of SCL in master mode. The transmit data buffer is the destination for data written to the I2CDR register. The base clock is SCLK, the system clock, and the period is calculated by the formula: t_{SCLK} X (4 X I2CSCLR + 2) where t_{SCLK} is the period of SCLK.

I2CSAR0 (I2C Slave Address 0 Register): EBH

7	6	5	4	3	2	1	0
I2CSLA06	I2CSLA05	I2CSLA04	I2CSLA03	I2CSLA02	I2CSLA01	I2CSLA00	GCALL0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

I2CSLA0[6:0]	These bits configure the slave address 0 in slave mode.
GCALL0EN	This bit decides whether I2C allows general call address or not in I2C slave mode.
	0 Ignore general call address
	1 Allow general call address

I2CSAR1 (I2C Slave Address 1 Register): F1H

7	6	5	4	3	2	1	0
I2CSLA16	I2CSLA15	I2CSLA14	I2CSLA13	I2CSLA12	I2CSLA11	I2CSLA10	GCALL1EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

I2CSLA1[6:0]	These bits configure the slave address 1 in slave mode.
GCALL1EN	This bit decides whether I2C allows general call address or not in I2C slave mode.
	0 Ignore general call address
	1 Allow general call address

I2CCR (I2C Control Register): E9H

7	6	5	4	3	2	1	0
IICRST	IICEN	TXDLYENB	IICIE	ACKEN	IMASTER	STOPC	STARTC
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Initial value: 00H

IICRST	Initialize Internal Registers of I2C.	
	0	No effect
	1	Initialize I2C, auto cleared
IICEN	Activate I2C Function Block by Supplying	
	0	I2C is disabled
	1	I2C is enabled
TXDLYENB	I2CSDHR register control bit	
	0	Enable I2CSDHR Register
	1	Disable I2CSDHR Register
IICIE	Interrupt Enable bit.	
	0	Interrupt of I2C is inhibited (use polling)
	1	Enable interrupt for I2C
ACKEN	Controls ACK signal generation at ninth SCL period.	
	0	No ACK signal is generated (SDA = 1)
	1	ACK signal is generated (SDA = 0)
	NOTES: ACK signal is output (SDA = 0) for the following 3 cases. When received address packet equals to I2CSLA bits in I2CSAR. When received address packet equals to value 0x00 with GCALLn enabled. When I2C operates as a receiver (master or slave)	
IMASTER	Represent operating mode of I2C	
	0	I2C is in slave mode
	1	I2C is in master mode
STOPC	When I2C is master, STOP condition generation	
	0	No effect
	1	STOP condition is to be generated
STARTC	When I2C is master, START condition generation	
	0	No effect
	1	START condition is to be generated

NOTE: Refer to the external interrupt flag register (EIFLAG) for the I2C interrupt flag.

I2CSR (I2C Status Register): EAH

7	6	5	4	3	2	1	0
GCALLn	TEND	STOPD	SSEL	MLOST	BUSY	TMODE	RXACK
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

GCALL[note)] This bit has different meaning depending on whether I2C is master or slave. When I2C is a master, this bit represents whether it received AACK (address ACK) from slave
- 0 No AACK is received (Master mode)
- 1 AACK is received (Master mode)

When I2C is a slave, this bit is used to indicate general call
- 0 General call address is not detected (Slave mode)
- 1 General call address is detected (Slave mode)

TEND [note)] This bit is set when 1-byte of data is transferred completely
- 0 1 byte of data is not completely transferred
- 1 1 byte of data is completely transferred

STOPD [note)] This bit is set when a STOP condition is detected
- 0 No STOP condition is detected
- 1 A STOP condition is detected

SSEL [note)] This bit is set when I2C is addressed by other master
- 0 I2C is not selected as a slave
- 1 I2C is addressed by other master and acts as a slave

MLOST [note)] This bit represents the result of bus arbitration in master mode
- 0 I2C maintains bus mastership
- 1 I2Chas lost bus mastership during arbitration process

BUSY This bit reflects bus status
- 0 I2C bus is idle, so a master can issue a START condition
- 1 I2C bus is busy

TMODE This bit is used to indicate whether I2C is transmitter or receiver
- 0 I2C is a receiver
- 1 I2C is a transmitter

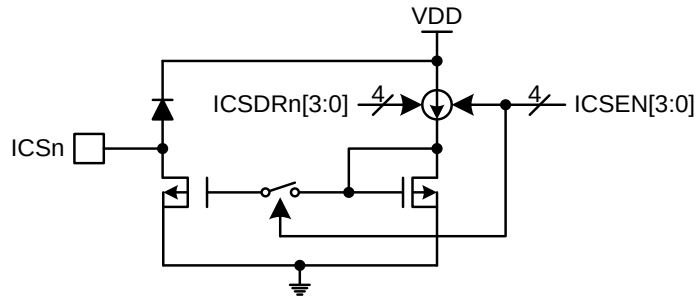
RXACK This bit shows the state of ACK signal
- 0 No ACK is received
- 1 ACK is received at ninth SCL period

NOTES:
1. These bits can be source of interrupt.
2. When an I2C interrupt occurs except for STOP mode, the SCL line is hold LOW. To release SCL, write arbitrary value to I2CSR. When I2CSR is written, The TEND, STOPD, SSEL, MLOST, and RXACK bits are cleared.

15 Constant sink current generator

Constant sink current generator supplies constant current regardless of variable I_{CS} voltage ranging from 2.0V to 3.6V. The constant current value is controlled by registers ICSDR0 and ICSDR1, and the sink current ranges from 49mA to 274mA.

15.1 Block diagram



NOTE: If a sink current generator of an ICSn pin is disabled by ICSCR[3:0], the corresponding ICSn pin is high and the current flowing is zero.

Figure 62. Constant Sink Current Generator Block Diagram (n=0 and 1)

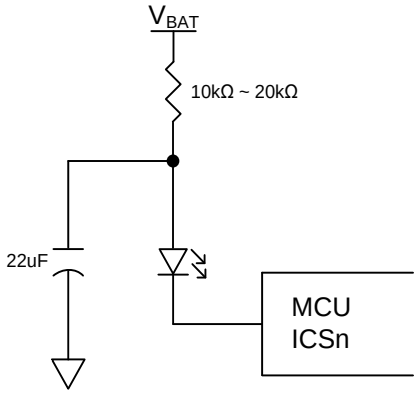


Figure 63. Constant Sink Current Generator Pin with Capacitor

15.2 Register map

Table 21. Constant Sink Current Generator Register Map

Name	Address	Direction	Default	Description
ICSCR	E2H	R/W	00H	Constant Sink Current Control Register
ICSDR0	E3H	R/W	00H	Constant Sink Current Data Register 0
ICSDR1	E4H	R/W	00H	Constant Sink Current Data Register 1

15.3 Register description

ICSCR (Constant Sink Current Control Register): E2H

7	6	5	4	3	2	1	0
–	–	–	–	ICSEN3	ICSEN2	ICSEN1	ICSEN0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

ICSEN[3:0]	Constant Sink Current Enable bits	
	0101b	Enable for the ICS0 pin and disable for the ICS1 pin
	1010b	Enable for the ICS1 pin and disable for the ICS0 pin
	Others	Disable sink current generator for the ICS0 and ICS1 pins

ICSDR0 (Constant Sink Current Data Register 0): E3H

7	6	5	4	3	2	1	0
–	–	–	–	ICSD03	ICSD02	ICSD01	ICSD00
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

ICSD0[3:0]	ICS0 pin Constant Sink Current Data bits
	ICS0 pin current [mA] ≒ 50 + 15 x ICSD0[3:0]

ICSDR1 (Constant Sink Current Data Register 1): E4H

7	6	5	4	3	2	1	0
–	–	–	–	ICSD13	ICSD12	ICSD11	ICSD10
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

ICSD1[3:0]	ICS1 pin Constant Sink Current Data bits
	ICS1 pin current [mA] ≒ 50 + 15 x ICSD1[3:0]

16 Flash CRC and Checksum generator

Flash CRC (Cyclic Redundancy Check) generator of A96L414/A96L416 generates 16-bit CRC code bits from Flash and a generator polynomial. The CRC code for each input data frame is appended to the frame.

Specifically CRC-based technique is used to verify data transmission or storage integrity. In the scope of the functional safety standards, this technique offers a means of verifying the Flash memory integrity. The Flash CRC generator helps compute a signature of software during runtime, to be compared with a reference signature.

The CRC generator has following features:

- Auto CRC and User CRC Mode
- CRC Clock : f_{HFIRC}, $f_{HFIRC}/2$, $f_{HFIRC}/4$, $f_{HFIRC}/8$ and fx (System clock)
- CRC-16 polynomial: 0x8C81 ($X^{16} + X^{15} + X^{11} + X^{10} + X^7 + 1$)

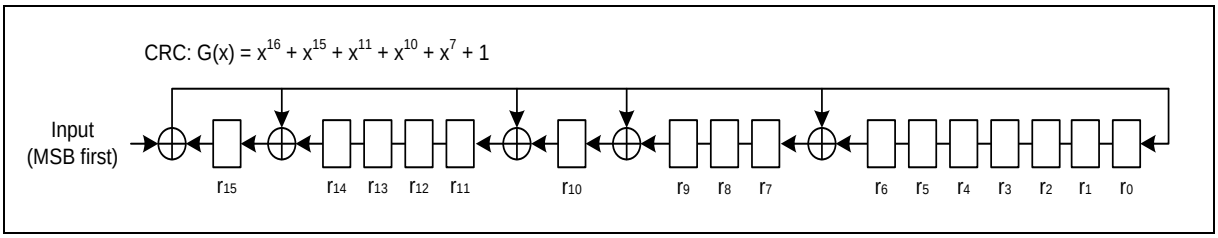


Figure 64. CRC-16 Polynomial Structure

16.1 Block diagram

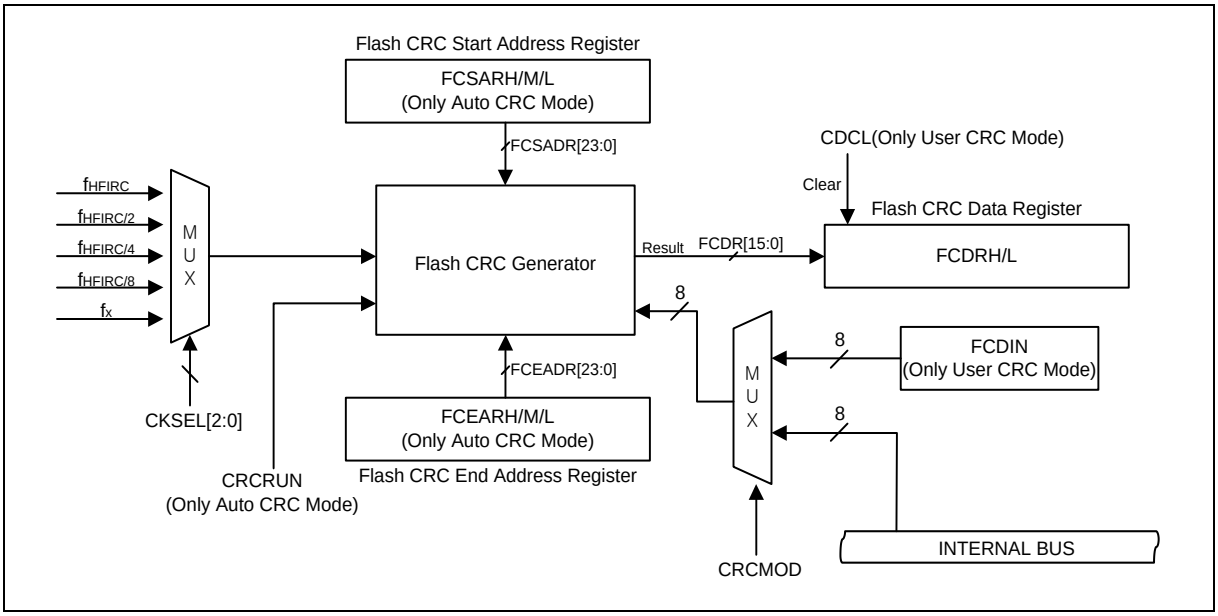


Figure 65. Flash CRC/ Checksum Generator Block Diagram

16.2 Operation procedure and example code of CRC and Checksum

The CRC operation procedure in Auto CRC/Checksum mode is introduced in the following list, and Figure 66 shows example program tip:

1. Global interrupt Disable (EA = 0)
2. Select Auto CRC/Checksum Mode and CRC
3. Select CRC Clock
4. Set CRC start address register (FCSARH/FCSARM/FCSARL)
5. Set CRC end address register (FCEARH/FCEARM/FCEARL)
6. CRC operation starts (CRCRUN = 1)
7. Read the CRC result
8. Global interrupt Enable (EA = 1)

```
//**** Global interrupt Disable
EA = 0;

//**** Flash CRC Auto CRC/Checksum Mode and CRC
FCCR &= _0101_1111;

OSCCR &= _1111_1011;    // IRC Enable
FCCR &= _1111_0001;                 // CRC clk = fIRC/1

//**** CRC start address set
FCSARH = 0x00;
FCSARM = 0x00;
FCSARL = 0x00;

//**** CRC end address set
FCEARH = 0x00;
FCEARM = 0x3F;
FCEARL = 0xFF;

//**** CRC start
FCCR |= _0000_0001;
_nop_();            //Dummy instruction, This instruction must be needed.
_nop_();            //Dummy instruction, This instruction must be needed.
_nop_();            //Dummy instruction, This instruction must be needed.

//**** Read CRC result
Temp0 = FCDRH;
Temp1 = FCDRL;

//**** Global interrupt Enable
EA = 1;
```

NOTES:

1. Three or more NOP instructions must immediately follow the CRC start operation in auto CRC/Checksum mode.
2. During a CRC operation (when CRCRUN bit is Running state) in auto CRC/Checksum mode, the CPU is hold and the global interrupt is on disable state regardless of the IE.7 (EA) bit. But should be set the global interrupt is disabled (EA = 0) before the CRC operation is started in use auto CRC/Checksum mode, recommend.

Figure 66. Program Tip for CRC Operation in Auto CRC/ Checksum Mode

The CRC operation procedure in User CRC/Checksum mode is introduced in the following list, and Figure 67 shows example program tip:

1. Select User CRC/Checksum Mode and CRC
2. Clear Flash CRC data register (FCDRH/FCDRL)
3. Read data from the Flash
4. Write the data to FCDIN Register
5. Read the CRC result

```
unsigned char code *rom_addr=0x0000;
unsigned int i=0;

    FCCR |= _1000_0000;      // Flash CRC User CRC/Checksum Mode
    FCCR &= _1101_1111;      // Flash CRC CRC Mode
    FCCR |= _0100_0000;      // Flash CRC data register clear

    for(i=0x0000; i <= 0x3FFF; i++)       // 0000H~3FFFH
    {
            FCDIN = rom_addr[i];
            WDTCR |= _0010_0000;   // Clear WDT counter
    }

//**** Read CRC result
        Temp0 = FCDRH;
        Temp1 = FCDRL;
```

Figure 67. Program Tip for CRC Operation in User CRC/ Checksum Mode

The Checksum operation procedure in Auto CRC/Checksum mode is introduced in the following list, and Figure 68 shows example program tip:

1. Global interrupt Disable (EA = 0)
2. Select Auto CRC/Checksum Mode and Checksum
3. Select CRC Clock
4. Set CRC start address register (FCSARH/FCSARM/FCSARL)
5. Set CRC end address register (FCEARH/FCEARM/FCEARL)
6. CRC operation starts (CRCRUN = 1)
7. Read the Checksum result
8. Global interrupt Enable (EA = 1)

```
//**** Global interrupt Disable
EA = 0;

//**** Flash CRC Auto CRC/Checksum Mode and Checksum
FCCR &= _0111_1111;
FCCR |= _0010_0000;                    // Checksum mode

OSCCR &= _1111_1011;                   // IRC Enable
FCCR &= _1111_0001;                    // CRC clk = fIRC/1

//**** Checksum start address set
FCSARH = 0x00;
FCSARM = 0x00;
FCSARL = 0x00;

//**** Checksum end address set
FCEARH = 0x00;
FCEARM = 0x3F;
FCEARL = 0xFF;

//**** Checksum start
FCCR |= _0000_0001;
_nop_();              //Dummy instruction, This instruction must be needed.
_nop_();              //Dummy instruction, This instruction must be needed.
_nop_();              //Dummy instruction, This instruction must be needed.

//**** Read Checksum result
Temp0 = FCDRH;
Temp1 = FCDRL;

//**** Global interrupt Enable
EA = 1;
```

NOTES:

1. Three or more NOP instructions must immediately follow the Checksum start operation in auto CRC/Checksum mode.
2. During a checksum operation (when CRCRUN bit is Running state) in auto CRC/Checksum mode, the CPU is hold and the global interrupt is on disable state regardless of the IE.7 (EA) bit. But should be set the global interrupt is disabled (EA = 0) before the Checksum operation is started in use auto CRC/Checksum mode, recommend.

Figure 68. Program Tip for Checksum Operation in Auto CRC/ Checksum Mode

The Checksum operation procedure in User CRC/Checksum mode is introduced in the following list, and Figure 69 shows example program tip:

1. Select User CRC/Checksum Mode and Checksum
2. Clear Flash CRC data register (FCDRH/FCDRL)
3. Read data from the Flash
4. Write the data to FCDIN Register
5. Read the Checksum result

```
unsigned char code *rom_addr=0x0000;
unsigned int i=0;

FCCR |= _1000_0000;                    // Flash CRC User CRC/Checksum Mode
FCCR &= _0010_0000;                    // Flash CRC Checksum
FCCR |= _0100_0000;                    // Flash CRC data register clear

for(i=0x0000; i <= 0x3FFF; i++)        // 0000H~3FFFH
{
    FCDIN = rom_addr[i];
    WDTCR |= _0010_0000;               // Clear WDT counter
}

//**** Read Checksum result
        Temp0 = FCDRH;
        Temp1 = FCDRL;
```

Figure 69. Program Tip for Checksum Operation in User CRC/ Checksum Mode

16.3 Register map

Table 22. Flash CRC/Checksum Generator Register Map

Name	Address	Direction	Default	Description
FCSARH	5050H (XSFR)	R/W	00H	Flash CRC Start Address High Register
FCEARH	5051H (XSFR)	R/W	00H	Flash CRC End Address High Register
FCSARM	5052H (XSFR)	R/W	00H	Flash CRC Start Address Middle Register
FCEARM	5053H (XSFR)	R/W	00H	Flash CRC End Address Middle Register
FCSARL	5054H (XSFR)	R/W	00H	Flash CRC Start Address Low Register
FCEARL	5055H (XSFR)	R/W	0FH	Flash CRC End Address Low Register
FCCR	5056H (XSFR)	R/W	00H	Flash CRC Control Register
FCDRH	5057H (XSFR)	R	FFH	Flash CRC Data High Register
FCDRL	5058H (XSFR)	R	FFH	Flash CRC Data Low Register
FCDIN	D7H	R/W	00H	Flash CRC Data In Register

16.4 Register description

FCSARH (Flash CRC Start Address High Register): 5050H

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	FCSARH0
–	–	–	–	–	–	–	R/W

Initial value: 00H

FCSARH0	Flash CRC Start Address High **NOTE:** Used only to Auto CRC Mode.

FCSARM (Flash CRC Start Address Middle Register): 5052H

7	6	5	4	3	2	1	0
FCSARM7	FCSARM6	FCSARM5	FCSARM4	FCSARM3	FCSARM2	FCSARM1	FCSARM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

FCSARM[7:0]	Flash CRC Start Address Middle **NOTE:** Used only to Auto CRC Mode.

FCSARL (Flash CRC Start Address Low Register): 5054H

7	6	5	4	3	2	1	0
FCSARL7	FCSARL6	FCSARL5	FCSARL4	FCSARL3	FCSARL2	FCSARL1	FCSARL0
R/W	R/W	R/W	R/W	–	–	–	–

Initial value: 00H

FCSARL[7:4]	Flash CRC Start Address Low **NOTE**: Used only to Auto CRC Mode.
FCSARL[3:0]	These bits are always "0000b".

FCEARH (Flash CRC End Address High Register): 5051H

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	FCEARH0
–	–	–	–	–	–	–	R/W

Initial value: 00H

FCEARH0	Flash CRC End Address High **NOTE:** Used only to Auto CRC Mode.

FCEARM (Flash CRC End Address Middle Register): 5053H

7	6	5	4	3	2	1	0
FCEARM7	FCEARM6	FCEARM5	FCEARM4	FCEARM3	FCEARM2	FCEARM1	FCEARM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

FCEARM[7:0]	Flash CRC End Address Middle **NOTE:** Used only to Auto CRC Mode.

FCEARL (Flash CRC End Address Low Register): 5055H

7	6	5	4	3	2	1	0
FCEARL7	FCEARL6	FCEARL5	FCEARL4	FCEARL3	FCEARL2	FCEARL1	FCEARL0
R/W	R/W	R/W	R/W	–	–	–	–

Initial value: 0FH

FCEARL[7:4] Flash CRC End Address Low
NOTE: Used only to Auto CRC Mode.
FCEARL[3:0] These bits are always "1111b".

FCDRH (Flash CRC Data High Register): 5057H

7	6	5	4	3	2	1	0
FCDRH7	FCDRH6	FCDRH5	FCDRH4	FCDRH3	FCDRH2	FCDRH1	FCDRH0
R	R	R	R	R	R	R	R

Initial value: FFH

FCDRH[7:0] Flash CRC Data High

FCDRL (Flash CRC Data Low Register): 5058H

7	6	5	4	3	2	1	0
FCDRL7	FCDRL6	FCDRL5	FCDRL4	FCDRL3	FCDRL2	FCDRL1	FCDRL0
R	R	R	R	R	R	R	R

Initial value: FFH

FCDRL[7:0] Flash CRC Data Low

FCDIN (Flash CRC Data IN Register): D7H

7	6	5	4	3	2	1	0
FCDIN7	FCDIN6	FCDIN5	FCDIN4	FCDIN3	FCDIN2	FCDIN1	FCDIN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

FCDIN[7:0] Flash CRC Data In
NOTE: Used only to User CRC Mode.

FCCR (Flash CRC Control Register): 5056H

7	6	5	4	3	2	1	0
CRCMOD	CDCL	MDSEL	–	CKSEL2	CKSEL1	CKSEL0	CRCRUN
R/W	R/W	R/W	–	R/W	R/W	R/W	R/W

Initial value: 00H

CRCMOD Select CRC/Checksum Mode
- 0 Auto CRC/Checksum Mode
- 1 User CRC/Checksum Mode

CDCL Flash CRC Data Register Clear
- 0 No effect
- 1 Clear Flash CRC Data register

NOTE: This bit is cleared to '0' automatically, after Flash CRC Data register is cleared. The FCDRH/L is set to "FFH" if the MDSEL is set to "0b" and "00H" if the MDSEL is set to "1b". Used only to User CRC/Checksum Mode.

MDSEL CRC/Checksum Selection
- 0 Select CRC
- 1 Select Checksum

CKSEL[2:0] Select Flash CRC/Checksum Clock

CKSEL2	CKSEL1	CKSEL0	Description
0	0	0	f_{HFIRC}
0	0	1	$f_{HFIRC}/2$
0	1	0	$f_{HFIRC}/4$
0	1	1	$f_{HFIRC}/8$
1	0	0	f_X(system clock)
Other values			Not used

CRCRUN CRC/Checksum Start Signal & Busy Flag, Used only to Auto CRC/Checksum mode.
- 0 Indicates that CRC/Checksum operation is not running or has finished. When written "0", CRC/Checksum operation is finished by force even if CRC/Checksum operation is running. It has no effect to write "0" if CRC/Checksum is not running currently.
- 1 When written "1", CRC/Checksum operation starts and this bit remains "1" as long as CRC/Checksum operation is on-going. This bit is cleared to "0" automatically after CRC/Checksum operation finishes.

17 Power down operation

A96L414/A96L416 offers two power-down modes to minimize power consumption of itself. Programs under the two power saving modes IDLE and STOP, are stopped and power consumption is reduced considerably.

17.1 Peripheral operation in IDLE/STOP mode

Peripheral's operations during IDLE/STOP mode is introduced in Table 23.

Table 23. Peripheral Operation during Power-down Mode

Peripheral	IDLE mode	STOP mode
CPU	ALL CPU Operation are Disable	ALL CPU Operation are Disable
RAM	Retain	Retain
Basic Interval Timer	Operates Continuously	Stop
Watchdog Timer	Operates Continuously	Stop (Can be operated with WDTRC OSC, LFIRC OSC)
Timer0~1	Operates Continuously	Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally)
ADC	Operates Continuously	Stop
USART	Operates Continuously	Stop
Siren	Operates Continuously	Stop
Line Interface	Operates Continuously	Stop
Internal OSC	Oscillation	Stop
WDTRC OSC (1KHz)	Can be operated with setting value	Can be operated with setting value
Constant Sink Current	Retain	Retain
I/O Port	Retain	Retain
Control Register	Retain	Retain
Address Data Bus	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt (EC0, EC1, EC2), External Interrupt, WDT, USART

17.2 IDLE mode

The power control register is set to '01h' to enter the IDLE Mode. In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally but CPU stops. It is released by reset or interrupt. To be released by interrupt, interrupt should be enabled before IDLE mode. If using reset, because the device becomes initialized state, the registers have reset value.

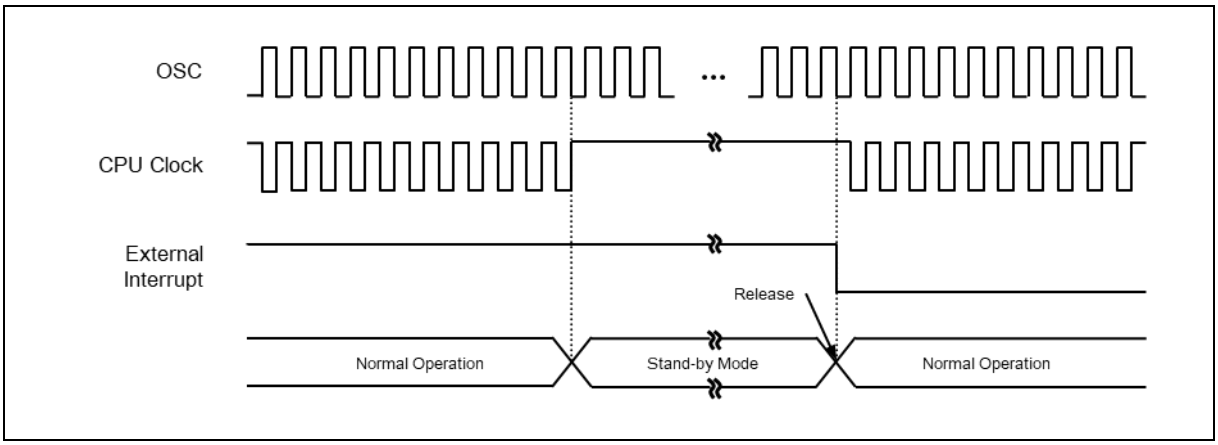


Figure 70. IDLE Mode Release Timing by External Interrupt

17.3 STOP mode

The power control register is set to '03H' to enter the STOP Mode. In the stop mode, the selected oscillator, system clock and peripheral clock is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and control registers are held.

The source for exit from STOP mode is hardware reset and interrupts. The reset re-defines all the control registers.

When exit from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 71 shows the timing diagram. When released from STOP mode, the Basic interval timer is activated on wake-up. Therefore, before STOP instruction, user must be set its relevant prescale divide ratio to have long enough time. This guarantees that oscillator has started and stabilized.

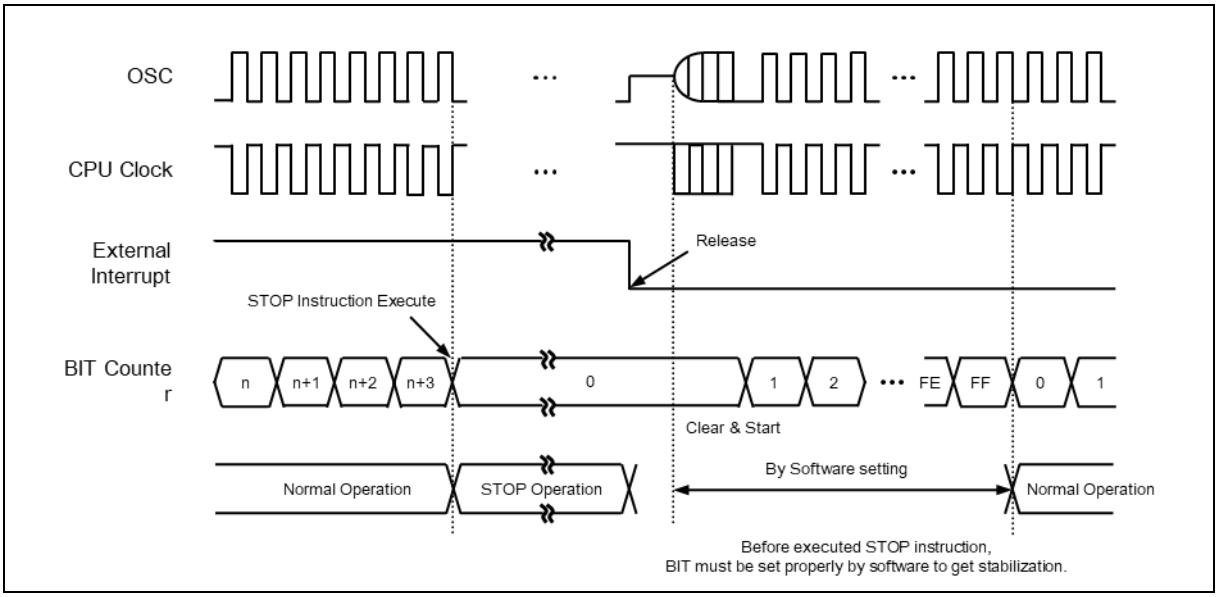


Figure 71. STOP Mode Release Timing by External Interrupt

17.4 Release operation of STOP mode

After STOP mode is released, the operation begins according to content of related interrupt register just before STOP mode start (Refer to Figure 72).

If the global interrupt Enable Flag (IE.EA) is set to `1`, the STOP mode is released by the interrupt which each interrupt enable flag = `1` and the CPU jumps to the relevant interrupt service routine. Even if the IE.EA bit is cleared to '0', the STOP mode is released by the interrupt of which the interrupt enable flag is set to '1'.

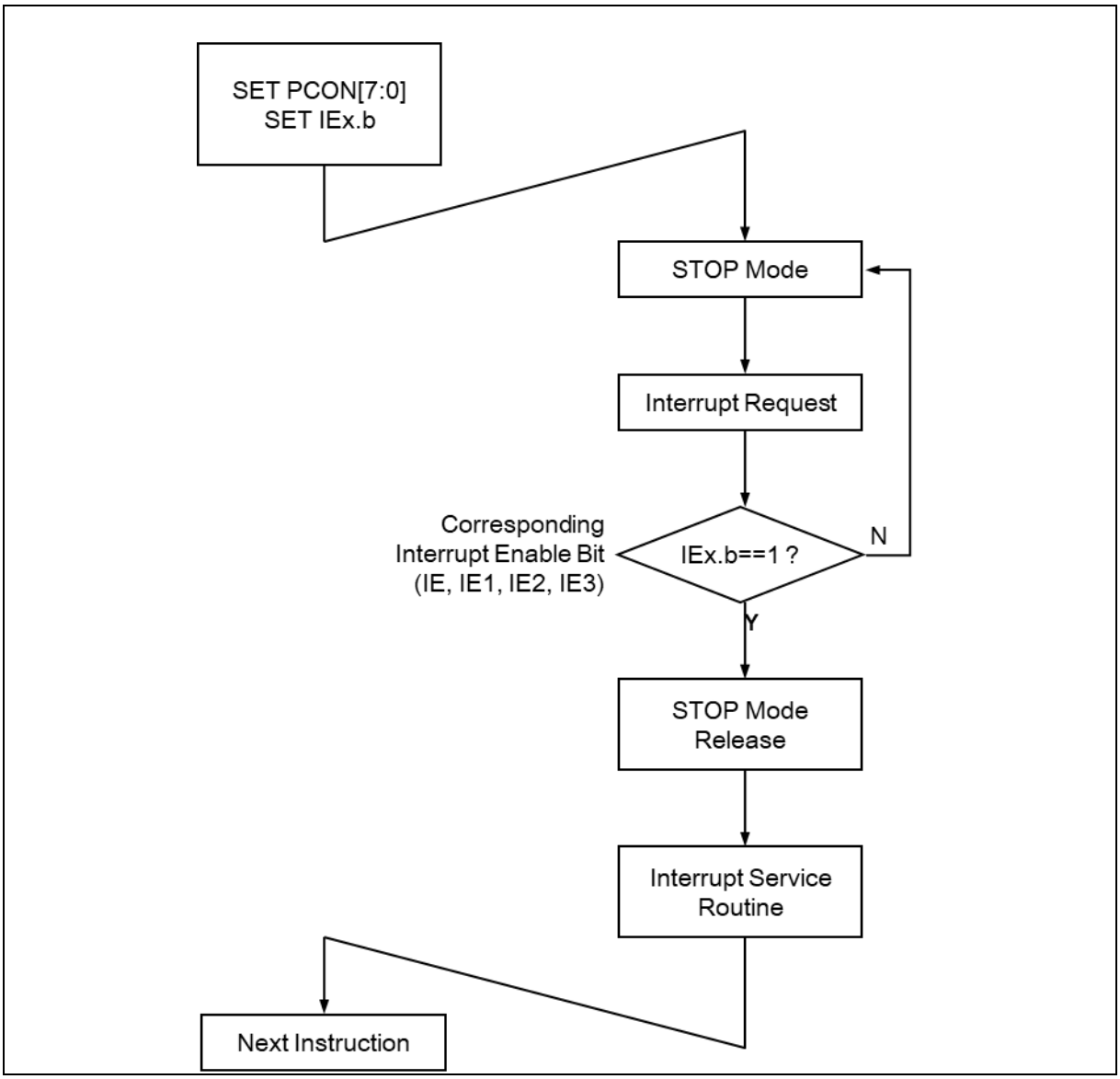


Figure 72. STOP Mode Release Flow

17.5 Register map

Table 24. Power-down Operation Register Map

Name	Address	Direction	Default	Description
PCON	87H	R/W	00H	Power control register

17.6 Register description

PCON (Power Control Register): 87H

7	6	5	4	3	2	1	0
–	–	–	–	–	–	PCON1	PCON0
–	–	–	–	–	–	R/W	R/W

Initial value: 00H

PCON[1:0]	Power Control
01H	IDLE mode enable
03H	STOP mode enable
Other Values	Normal operation

Before configuring a register PCON, please be aware of the followings:

1. To enter IDLE mode, PCON must be set to '01H'.
2. To enter STOP mode, PCON must be set to '03H'.
3. The PCON register is automatically cleared by a release signal in STOP/IDLE mode.
4. Three or more NOP instructions must follow immediately after the instruction that makes the device enter in STOP/IDLE mode. Refer to the following example code in Table 25.

Table 25. Example Code with 3 or more NOP Instructions

Example code 1	Example code 2
MOV PCON, #01H ; IDLE mode NOP NOP NOP • • •	MOV PCON, #03H ; STOP mode NOP NOP NOP • • •

18 Reset

When a reset event occurs, an internal register is selected to be initialized in accordance with a reset value. Each reset value introduced in Table 26 indicates a corresponding On Chip Hardware that is to be initialized.

Table 26. Reset Value and the Relevant On Chip Hardware

On Chip Hardware	Reset Value
Program Counter (PC)	0000H
Accumulator	00H
Stack Pointer (SP)	07H
Peripheral clock	On
Control register	Refer to peripheral registers.

A96L414/A96L416 has 5 types of reset sources as listed in the followings:

- External RESETB
- Power On RESET (POR)
- WDT overflow reset (in a case of WDTEN='1')
- Low voltage reset (in a case of LVREN='0')
- OCD RESET

18.1 Reset block diagram

Figure 73 shows a reset block of A96L414/A96L416.

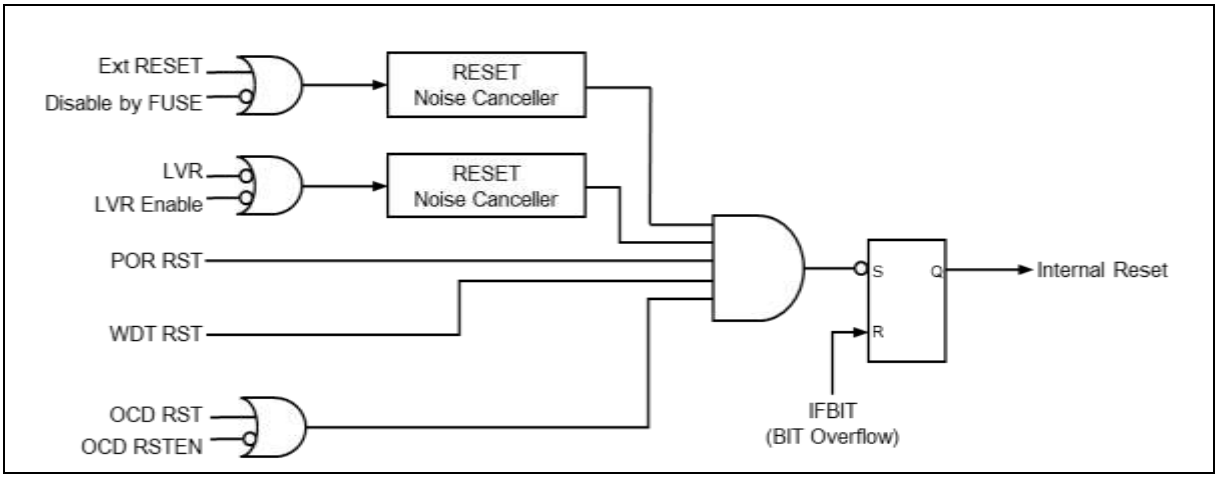


Figure 73. Reset Block Diagram

18.2 Reset noise canceller

Figure 74 is a noise canceller timing diagram for noise cancellation of RESET. It has the noise cancellation value of about 2us (@VDD=5V) to the low input of system reset.

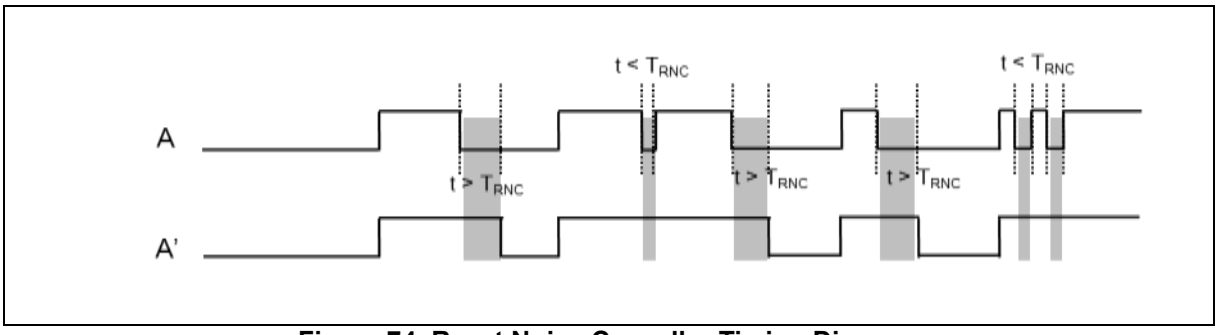


Figure 74. Reset Noise Canceller Timing Diagram

18.3 Power on Reset

When device power is increasing, POR (Power on Reset) executes a function to reset the device. If POR is used to reset the device, it executes the device reset function instead of RESET IC or RESET Circuit.

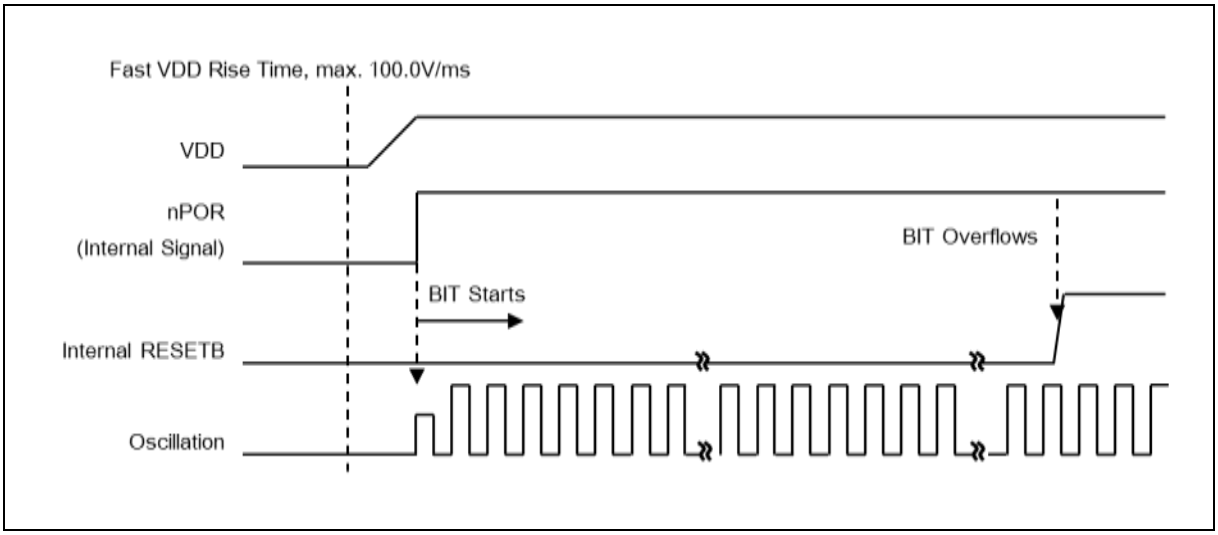


Figure 75. Fast VDD Rising Time

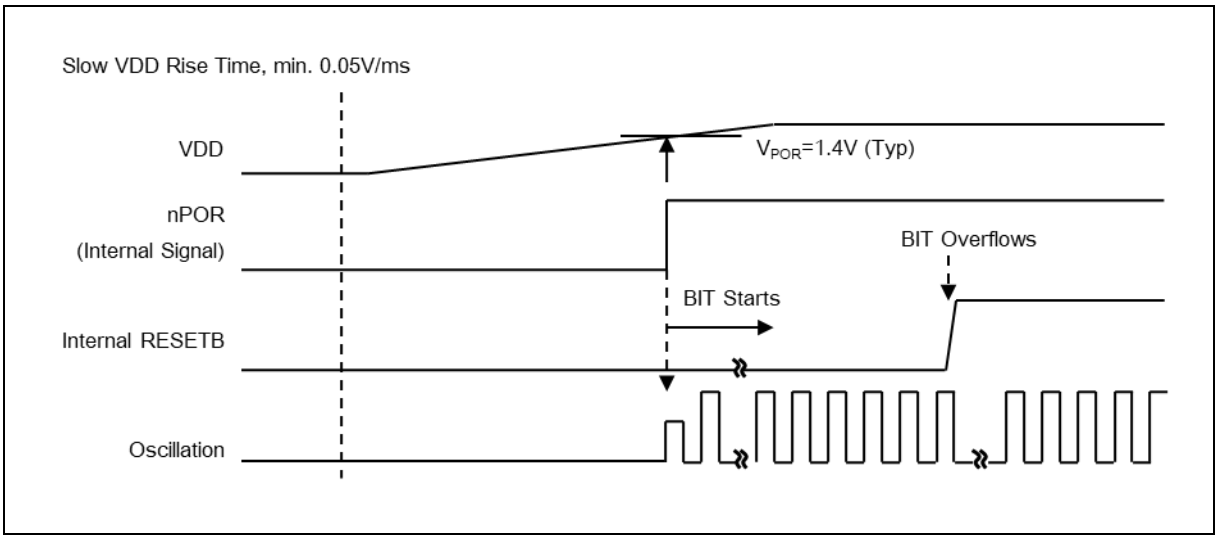


Figure 76. Internal Reset Release Timing on Power-Up

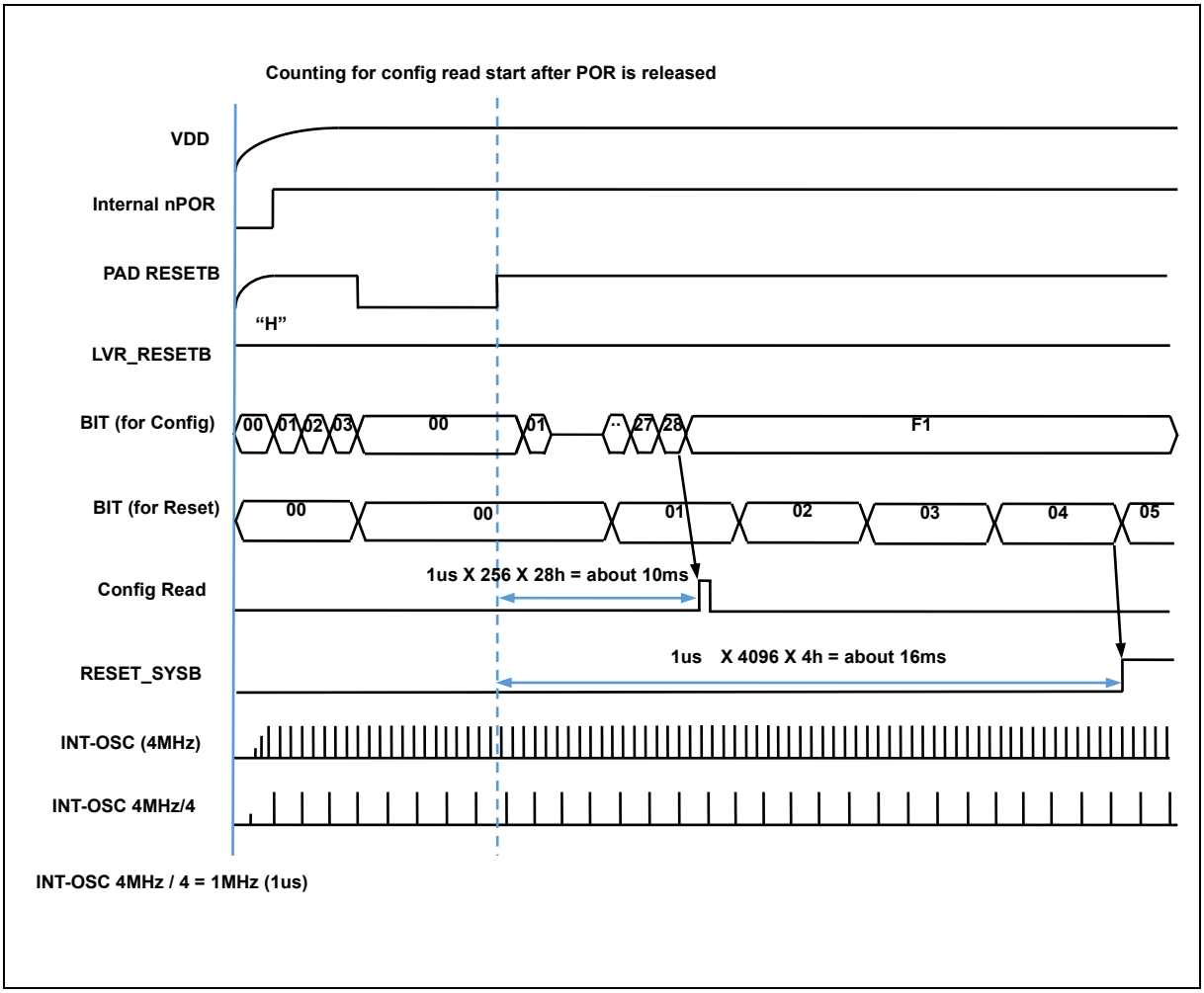


Figure 77. Configuration Timing when Power-On

Relationship between VDD input and internal oscillator is described in Figure 78 and Table 27.

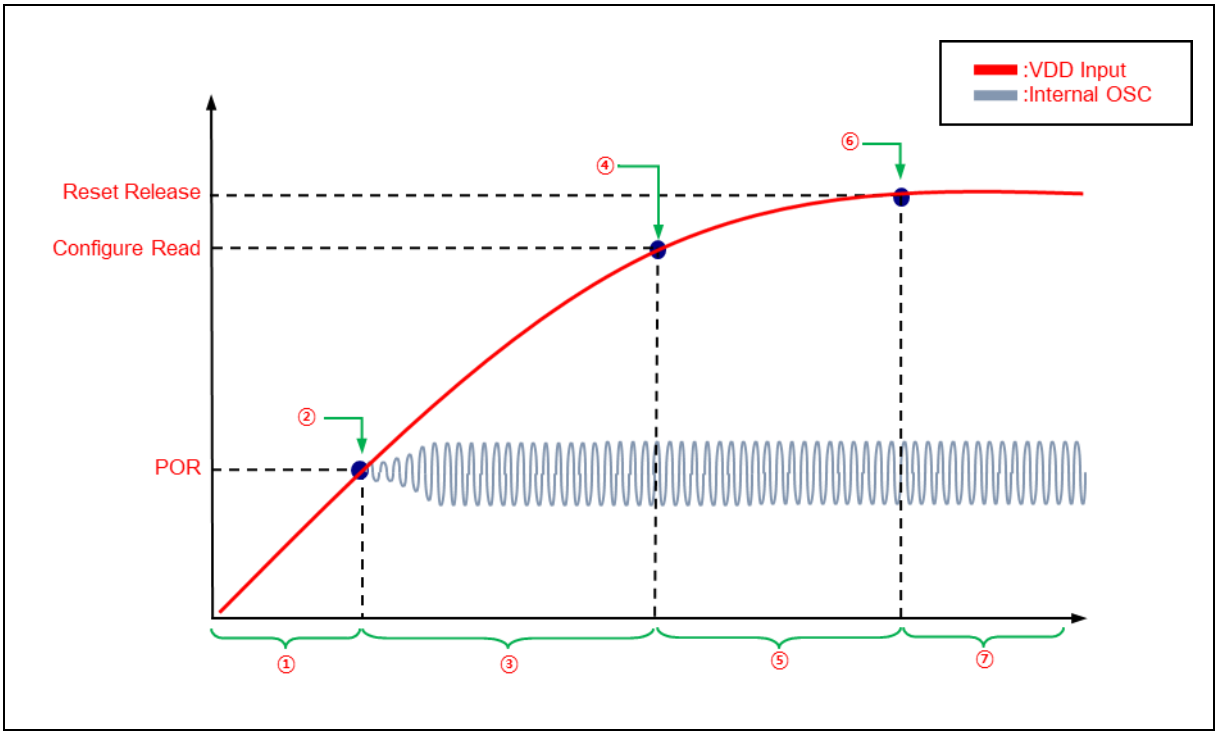


Figure 78. Boot Process Waveform

Table 27. Boot Process Description

Process	Description	Remark
①	No operation	
②	1st POR level detection	About 1.4V
③	• (INT-OSC 4MHz/4) x 256 x 28h Delay section (=10ms) • VDD input voltage must rise over than Flash operating voltage for Config read.	Slew rate >= 0.05V/ms
④	Config read point	• About 1.5V ~ 1.6V • Config Value is determined by Writing Option.
⑤	Rising section to reset release level	16ms point after POR or Ext_reset release
⑥	Reset release section (BIT overflow) • After16ms, after external reset release (external reset) • 16ms point after POR (POR only)	BIT is used for peripheral stability.
⑦	Normal operation	

18.4 External RESETB input

External RESETB is the input to a Schmitt trigger. If RESETB pin is held with low for at least 10us over within the operating voltage range and stable oscillation, it is applied and the internal state is initialized. When reset state becomes '1', it needs stabilization time with 16ms and after the stable state, the internal RESET becomes '1'. The Reset process step needs 5 oscillator clocks. And the program execution starts at the vector address stored at address 0000H.

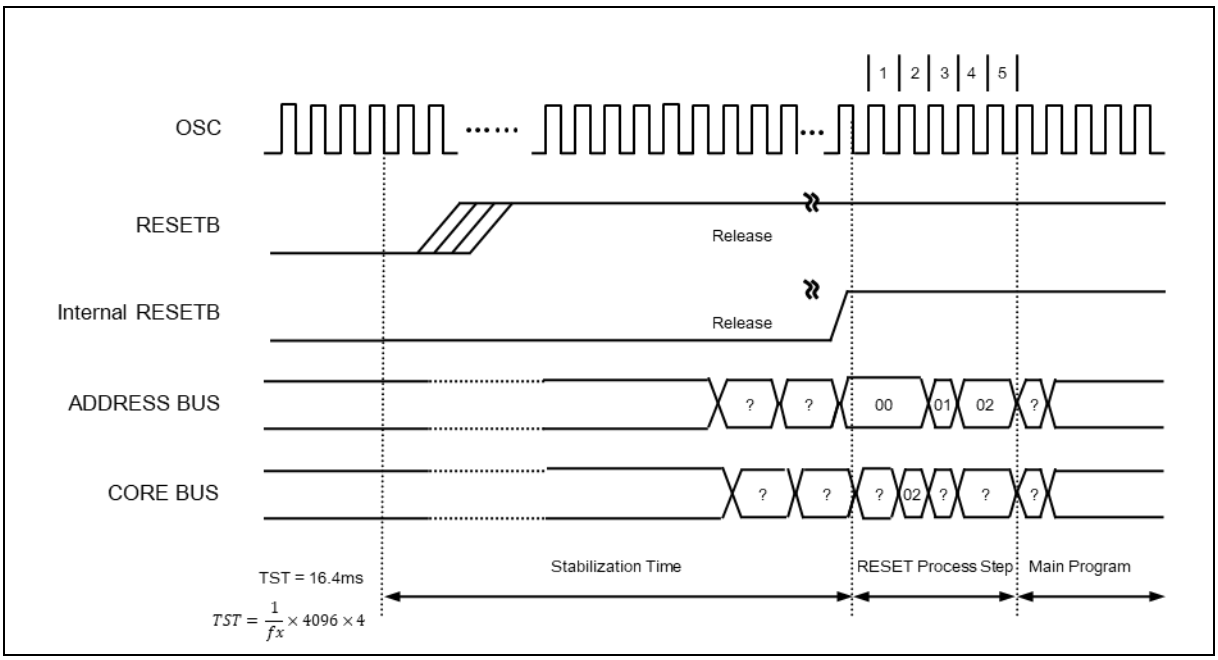
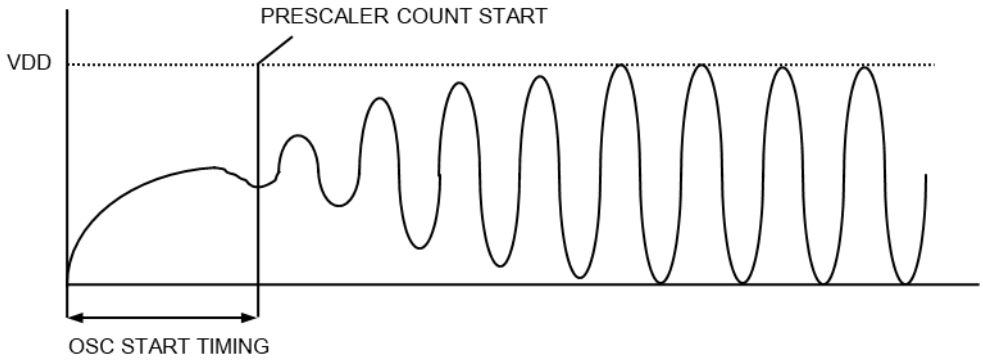


Figure 79. Timing Diagram after RESET



NOTES:

1. Stable generating time is not included in the start-up time.
2. RESETB pin has a pull-up resistor by H/W.

Figure 80. Oscillator Generating Waveform Example

18.5 Brown out detector processor

A96L414/A96L416 has an On-chip brown-out detection circuit (BOD) to monitor VDD level during its operation. It compares VDD level to a fixed trigger level which can be selected to be one of 1.60V, 2.20V, 2.40V, and 2.70V by LVRVS[1:0] bits. In STOP mode, since the BOD will contribute significantly to the total current consumption, the LVREN bit is set to off by software to minimize the current consumption.

18.5.1 Block diagram

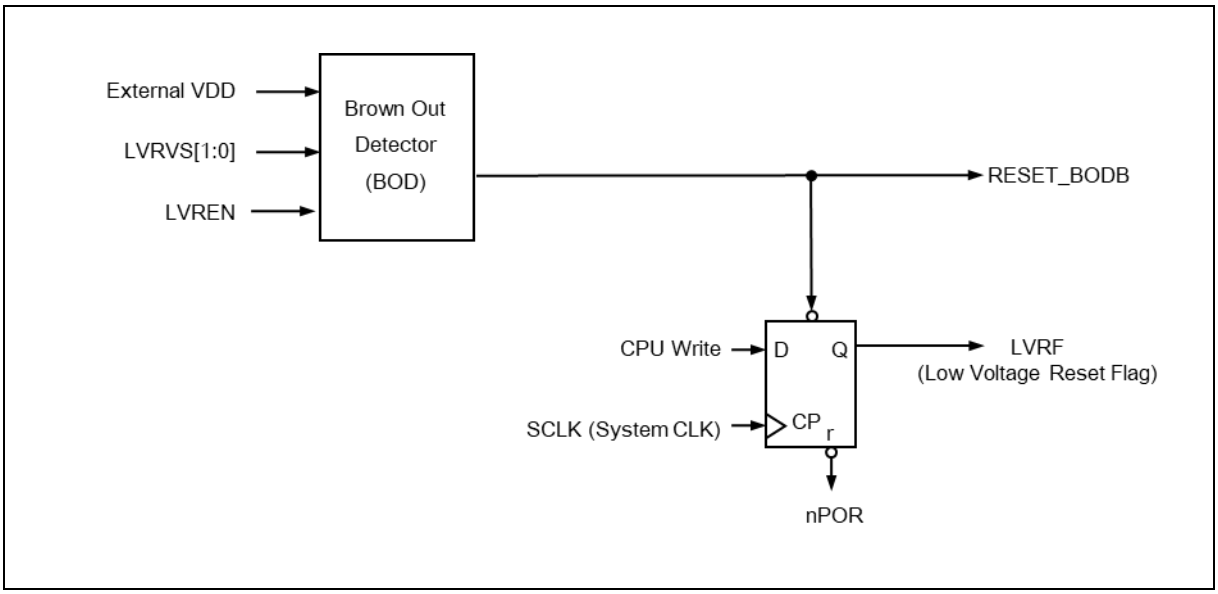


Figure 81. BOD Block Diagram

18.5.2 Internal reset and BOD reset in timing diagram

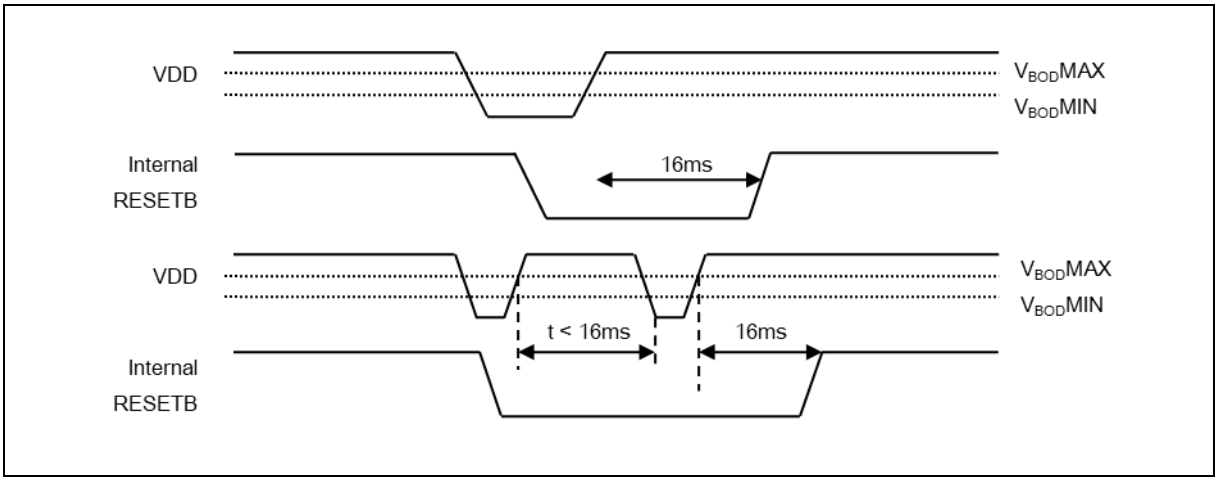


Figure 82. Internal Reset at Power Fail Situation

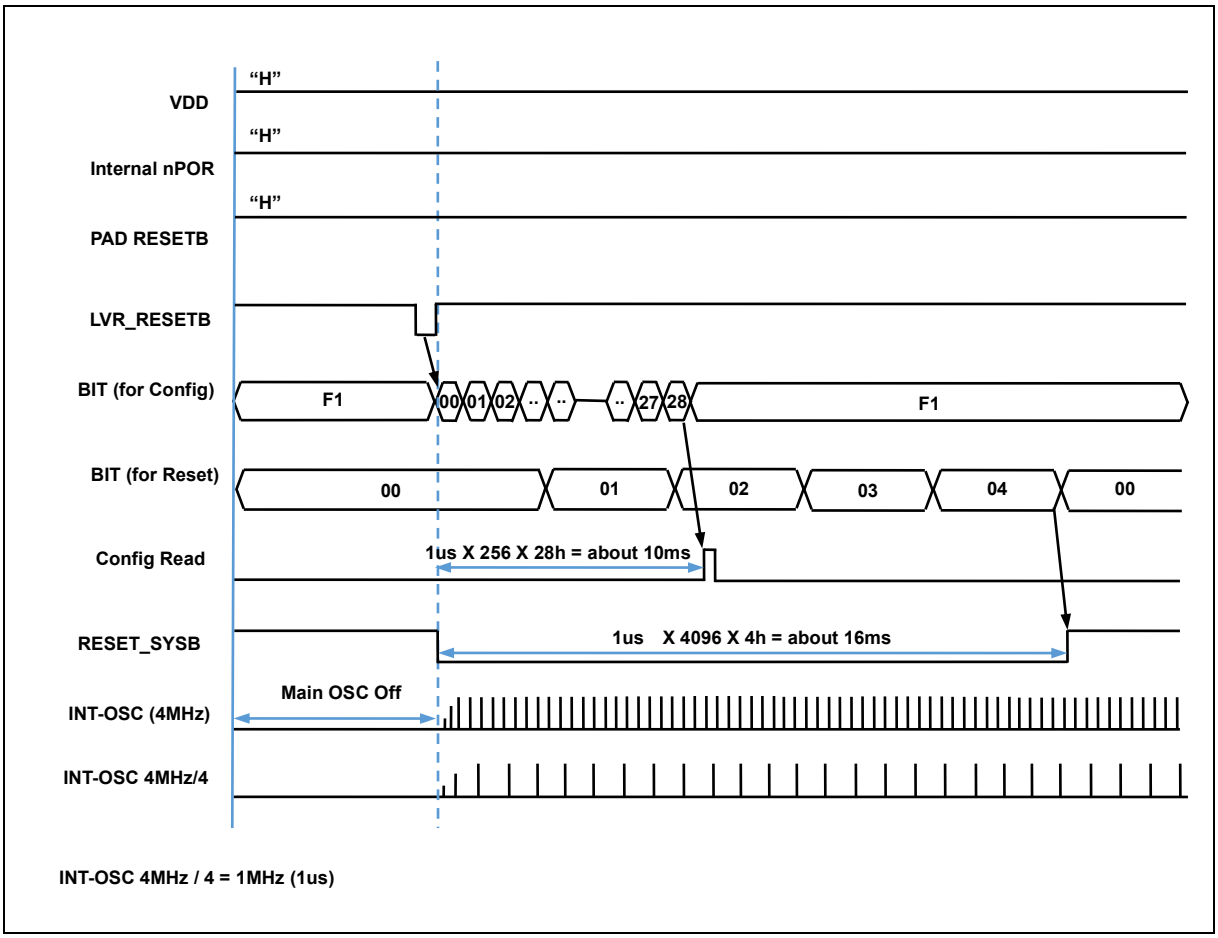


Figure 83. Configuration Timing when BOD Reset

18.6 Register map

Table 28. Reset Operation Register Map

Name	Address	Direction	Default	Description
RSTFR	E8H	R/W	80H	Reset Flag Register
LVRCR	D8H	R/W	00H	Low Voltage Reset Control Register
LVRIDR	505FH (XSFR)	R/W	00H	LVR Write Identification Register

18.7 Register description

RSTFR (Reset Flag Register): E8H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	LVRF	–	–	–
R/W	R/W	R/W	R/W	R/W	–	–	–

Initial value: 80H

PORF Power-On Reset flag bit. The bit is reset by writing '0' to this bit.
0 No detection
1 Detection

EXTRF External Reset (RESETB) flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.
0 No detection
1 Detection

WDTRF Watchdog Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.
0 No detection
1 Detection

OCDRF On-Chip Debug Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.
0 No detection
1 Detection

LVRF Low Voltage Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.
0 No detection
1 Detection

NOTES

1. When the Power-On Reset occurs, the PORF bit is only set to "1", the WDTRF/OCDRF bits are cleared to "0".
2. When the Power-On Reset occurs, the EXTRF bit is unknown, at that time, the EXTRF bit can be set to "1" when External Reset (RESETB) occurs.
3. When the Power-On Reset occurs, the LVRF bit is unknown, at that time, the LVRF bit can be set to "1" when LVR Reset occurs.
4. When a reset except the POR occurs, the corresponding flag bit is only set to "1", the other flag bits are kept in the previous values.

LVRCR (Low Voltage Reset Control Register): D8H

7	6	5	4	3	2	1	0
LVRST	–	–	–	–	LVRVS1	LVRVS0	LVREN
R/W	–	–	–	–	R/W	R/W	R/W

Initial value: 00H

LVRST LVR Enable when Stop Release
- 0 Not effect at stop release
- 1 LVR enable at stop release

NOTES:
When this bit is '1', the LVREN bit is cleared to '0' by stop mode release. (LVR enable)
When this bit is '0', the LVREN bit is not effect by stop mode release.

LVRVS[1:0] LVR Voltage Select

LVRVS1	LVRVS0	Description
0	0	1.60V
0	1	2.20V
1	0	2.40V
1	1	2.70V

LVREN LVR Operation
- 0 LVR Enable
- 1 LVR Disable

NOTES:
1. The LVRST and LVRVS[1:0] bits are cleared by a power-on reset but are retained by other reset signals.
2. The LVRVS[1:0] bits should be set to '00b' while LVREN bit is "1".
3. This register can be written with valid ID value (LVRIDR == 0x59).

LVRIDR (LVR Write Identification Register): 505FH (XSFR)

7	6	5	4	3	2	1	0
LVRID7	LVRID6	LVRID5	LVRID4	LVRID3	LVRID2	LVRID1	LVRID0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

LVRID[7:0] LVR Write Identification
- Others No identification value
- 01011001b Identification value for LVR register write

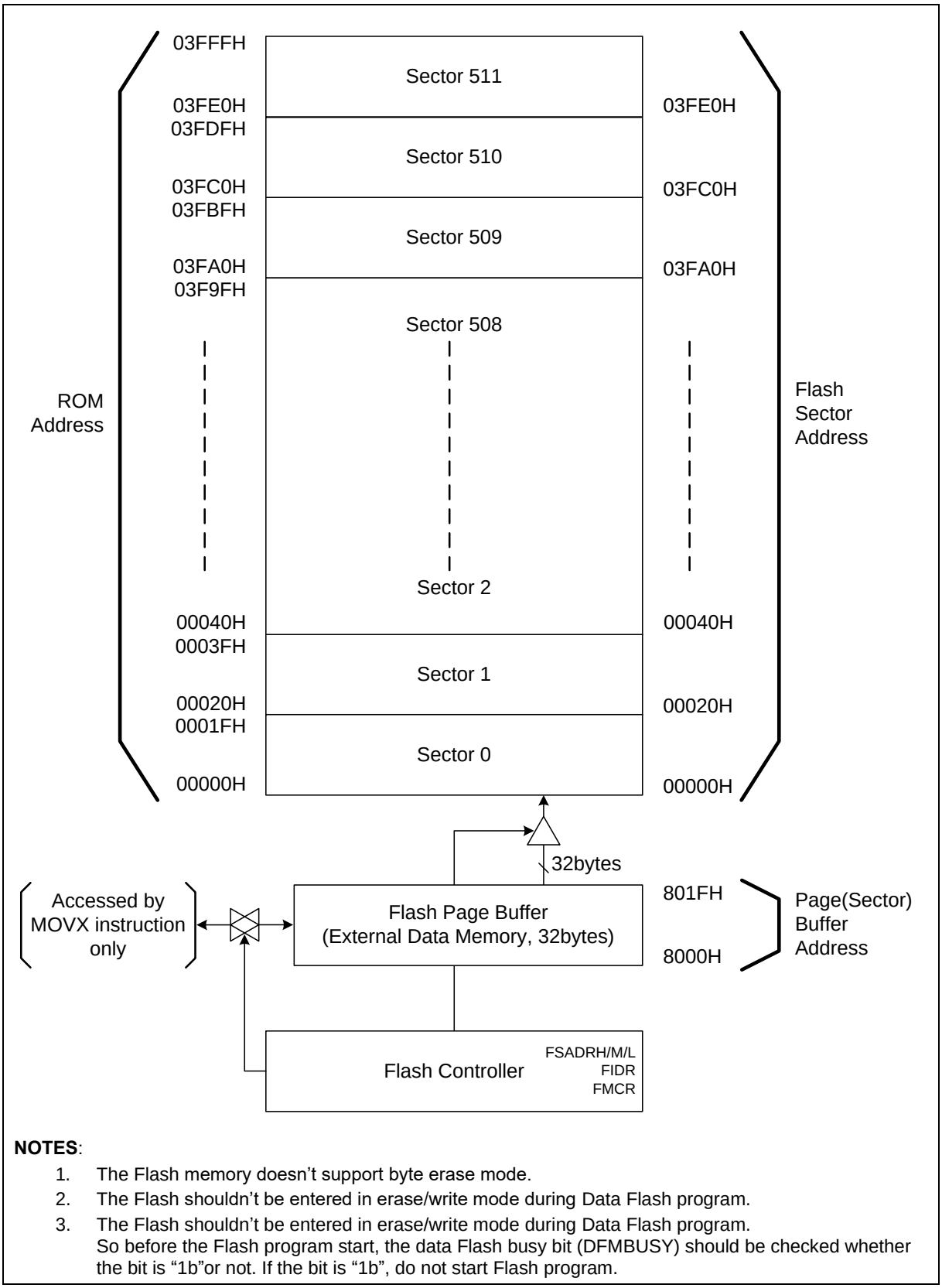
(These bits are automatically cleared to logic '00H' immediately after one time operation)

19 Flash memory

A96L414/A96L416 incorporates Flash memory inside. Program can be written, erased, and overwritten on the Flash memory while it is mounted on a board. The Flash memory can be read by 'MOVC' instruction and programmed in OCD, serial ISP mode or user program mode. Followings are features summary of Flash memory.

- Flash size: 8/16Kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 10,000 program/erase cycles at typical voltage and temperature for Flash memory

19.1 Flash program ROM structure



NOTES:

1. The Flash memory doesn't support byte erase mode.
2. The Flash shouldn't be entered in erase/write mode during Data Flash program.
3. The Flash shouldn't be entered in erase/write mode during Data Flash program.
   So before the Flash program start, the data Flash busy bit (DFMBUSY) should be checked whether the bit is "1b"or not. If the bit is "1b", do not start Flash program.

Figure 84. Flash Program ROM Structure

19.2 Register map

Table 29. Flash Memory Register Map

Name	Address	Direction	Default	Description
FSADRH	FAH	R/W	00H	Flash Sector Address High Register
FSADRM	FBH	R/W	00H	Flash Sector Address Middle Register
FSADRL	FCH	R/W	00H	Flash Sector Address Low Register
FIDR	FDH	R/W	00H	Flash Identification Register
FMCR	FEH	R/W	00H	Flash Mode Control Register

19.3 Register description

FSADRH (Flash Sector Address High Register): FAH

7	6	5	4	3	2	1	0
–	–	–	–	FSADRH3	FSADRH 2	FSADRH1	FSADRH0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

FSADRH[3:0] Flash Sector Address High

FSADRM (Flash Sector Address Middle Register): FBH

7	6	5	4	3	2	1	0
FSADRM7	FSADRM6	FSADRM5	FSADRM4	FSADRM3	FSADRM2	FSADRM1	FSADRM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

FSADRM[7:0] Flash Sector Address Middle

FSADRL (Flash Sector Address Low Register): FCH

7	6	5	4	3	2	1	0
FSADRL7	FSADRL6	FSADRL5	FSADRL4	FSADRL3	FSADRL2	FSADRL1	FSADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

FSADRL[7:0] Flash Sector Address Low

FIDR (Flash Identification Register): FDH

7	6	5	4	3	2	1	0
FIDR7	FIDR6	FIDR5	FIDR4	FIDR3	FIDR2	FIDR1	FIDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

FIDR[7:0] Flash Identification

Others No identification value

10100101 Identification value for a Flash mode

(These bits are automatically cleared to logic '00H' immediately after one time operation except "Flash page buffer reset mode")

FMCR (Flash Mode Control Register): FEH

7	6	5	4	3	2	1	0
FMBUSY	–	–	–	–	FMCR2	FMCR1	FMCR0
R	–	–	–	–	R/W	R/W	R/W

Initial value: 00H

FMBUSY Flash Mode Busy Bit. This bit will be used for only debugger.
0 No effect when "1" is written
1 Busy

FMCR[2:0] Flash Mode Control Bits. During a Flash mode operation, the CPU is hold and the global interrupt is on disable state regardless of the IE.7 (EA) bit.

FMCR2	FMCR1	FMCR0	Description
0	0	1	Select Flash page buffer reset mode and start regardless of the FIDR value (Clear all 32bytes to '0')
0	1	0	Select Flash sector erase mode and start operation when the FIDR="10100101b'
0	1	1	Select Flash sector write mode and start operation when the FIDR="10100101b'
1	0	0	Select Flash hard lock and start operation when the FIDR="10100101b'

Others Values: No operation
(These bits are automatically cleared to logic '00H' immediately after one time operation)

19.4 Serial In-System Program (ISP) mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to Chapter 24. **Development tools** in details about debugger.

19.5 Protection area (User Program mode)

A user can program Flash memory (protection area) of A96L414/A96L416. The protection area cannot be erased or programmed if any protection area is enabled by the configure option 2. If the protection area is disabled (PAEN ='0'), this area can be erased or programmed.

The user can choose size of protection area can by using configure option 2. For more information about configure option 2, please refer to **Appendix A. Configure option**.

Table 30 and Table 31 introduce protection area size and relative information.

Table 30. Protection Area Size and its Relative Information on A96L414

Protection area size select			Size of protection area	Address of protection area
PASS2	PASS1	PASS0		
0	0	0	0.7Kbytes	0100H – 03FFH
0	0	1	1.7Kbytes	0100H – 07FFH
0	1	0	2.7Kbytes	0100H – 0BFFH
0	1	1	3.8Kbytes	0100H – 0FFFH
1	0	0	5.7Kbytes	0100H – 17FFH
1	0	1	6.7Kbytes	0100H – 1BFFH
1	1	0	7.2Kbytes	0100H – 1DFFH
1	1	1	7.5Kbytes	0100H – 1EFFH

NOTE: Please refer to **Appendix A. Configure option**.

Table 31. Protection Area Size and its Relative Information on A96L416

Protection area size select			Size of protection area	Address of protection area
PASS2	PASS1	PASS0		
0	0	0	0.7Kbytes	0100H – 03FFH
0	0	1	1.7Kbytes	0100H – 07FFH
0	1	0	2.7Kbytes	0100H – 0BFFH
0	1	1	3.8Kbytes	0100H – 0FFFH
1	0	0	13.7Kbytes	0100H – 37FFH
1	0	1	14.7Kbytes	0100H – 3BFFH
1	1	0	15.2Kbytes	0100H – 3DFFH
1	1	1	15.5Kbytes	0100H – 3EFFH

NOTE: Please refer to **Appendix A. Configure option**.

19.6 Erase mode

The sector erase program procedure in user program mode:

1. Page buffer clear (FMCR=0x01).
2. Write '0' to the page buffer.
3. Set Flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set Flash identification register (FIDR).
5. Check User ID to prevent invalid work[NOTE].
6. Set Flash mode control register (FMCR).
7. Erase verify

NOTE: Please refer to a subsection **19.8 Protection for invalid erase/write**.

Figure 85 shows example program tip regarding sector erase.

```
        MOV     FMCR,#0x01      ;page buffer clear
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.

        MOV     A,#0
        MOV     R0,#SectorSize  ;Sector size of Device
        MOV     DPH,#0x80       ;Page Buffer Address is 8000H
        MOV     DPL,#0

    Pgbuf_clr:
        MOVX    @DPTR,A
        INC     DPTR
        DJNZ    R0,Pgbuf_clr    ;Write '0' to all page buffer


        MOV     FSADRH,#SAH     ;Sector Address High Byte.
        MOV     FSADRM,#SAM     ;Sector Address Middle Byte
        MOV     FSADRL,#SAL     ;Sector Address Low Byte
        MOV     FIDR,#0xA5      ;Identification value

        MOV     A,#ID_DATA_1    ;Check the UserID(written by user)
        CJNE    A,UserID1,No_WriteErase;This routine for UserID must be needed.
        MOV     A,#ID_DATA_2
        CJNE    A,UserID2,No_WriteErase

        MOV     FMCR,#0x02      ;Start Flash erase mode
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.

        LJMP    Erase_verify
                ---
    No_WriteErase:
        MOV     FIDR,#00H
        MOV     UserID1,#00H
        MOV     UserID2,#00H
                ---
    Erase_verify:
                ---
    Verify_error:
                ---
```

Figure 85. Program Tip: Sector Erase

19.7 Write mode

The sector Write program procedure in user program mode:

1. Page buffer clear (FMCR=0x01)
2. Write data to page buffer
3. Set Flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set Flash identification register (FIDR).
5. Check the UserID for to prevent the invalid work [NOTE1].
6. Set Flash mode control register (FMCR).
7. Write verify

NOTES

1. Please refer to "19.8. Protection for Invalid Erase/Write".
2. All data of the sector should be "00H" before writing data to a sector.

Figure 86 shows example program tip regarding sector write.

```
        MOV     FMCR,#0x01      ;page buffer clear
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.

        MOV     A,#0
        MOV     R0,#SectorSize  ;Sector size of Device
        MOV     DPH,#0x80       ;Page Buffer Address is 8000H
        MOV     DPL,#0

Pgbuf_WR:   MOVX    @DPTR,A
        INC     A
        INC     DPTR
        DJNZ    R0,Pgbuf_WR     ;Write data to all page buffer

        MOV     FSADRH,#SAH     ;Sector Address High Byte.
        MOV     FSADRM,#SAM     ;Sector Address Middle Byte
        MOV     FSADRL,#SAL     ;Sector Address Low Byte
        MOV     FIDR,#0xA5      ;Identification value

        MOV     A,#ID_DATA_1    ;Check the UserID(written by user)
        CJNE    A,UserID1,No_WriteErase;This routine for UserID must be needed.
        MOV     A,#ID_DATA_2
        CJNE    A,UserID2,No_WriteErase

        MOV     FMCR,#0x03      ;Start Flash write mode
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.
        NOP             ;Dummy instruction, This instruction must be needed.

        LJMP    Write_verify
                ---
No_WriteErase:
        MOV     FIDR,#00H
        MOV     UserID1,#00H
        MOV     UserID2,#00H
                ---
Write_verify:
                ---
Verify_error:
                ---
```

Figure 86. Program Tip: Sector Write

The Byte Write program procedure in user program mode:

1. Page buffer clear (FMCR=0x01)
2. Write data to page buffer
3. Set Flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set Flash identification register (FIDR).
5. Check the UserID for to prevent the invalid work [NOTE1].
6. Set Flash mode control register (FMCR).
7. Write verify

NOTES:

1. Please refer to "19.8. Protection for invalid erase/write".
2. Data of the address should be "00H" before writing data to an address.

Figure 87 shows example program tip regarding byte write.

```
        MOV    FMCR,#0x01      ;page buffer clear
        NOP                    ;Dummy instruction, This instruction must be needed.
        NOP                    ;Dummy instruction, This instruction must be needed.
        NOP                    ;Dummy instruction, This instruction must be needed.

        MOV    A,#5
        MOV    DPH,#0x80
        MOV    DPL,#0
        MOVX   @DPTR,A                ;Write data to page buffer

        MOV    A,#6
        MOV    DPH,#0x80
        MOV    DPL,#0x05
        MOVX   @DPTR,A                ;Write data to page buffer

       MOV     FSADRH,#SAH    ;Sector Address High Byte.
       MOV     FSADRM,#SAM    ;Sector Address Middle Byte
       MOV     FSADRL,#SAL    ;Sector Address Low Byte
       MOV     FIDR,#0xA5     ;Identification value

       MOV     A,#ID_DATA_1   ;Check the UserID(written by user)
       CJNE    A,UserID1,No_WriteErase;This routine for UserID must be needed.
       MOV     A,#ID_DATA_2
       CJNE    A,UserID2,No_WriteErase

       MOV     FMCR,#0x03     ;Start Flash write mode
       NOP                    ;Dummy instruction, This instruction must be needed.
       NOP                    ;Dummy instruction, This instruction must be needed.
       NOP                    ;Dummy instruction, This instruction must be needed.

       LJMP    Write_verify
               ---
No_WriteErase:
       MOV     FIDR,#00H
       MOV     UserID1,#00H
       MOV     UserID2,#00H
               ---
Write_verify:
               ---
Verify_error:
               ---
```

Figure 87. Program Tip: Byte Write

19.8 Protection for invalid erase/ write

It needs to be careful when programming Flash erase/write operation in code. In addition, it needs preparations for invalid jump to the Flash erase/write code occurred by malfunction, noise, and power off.

NOTE: For more information about the invalid erase and write operation, please refer to Appendix: Flash protection for invalid erase/write.

Following procedure instructs to protect for invalid erase and write operation:

1. User ID check routine for the Flash erase/write code

```
ErWt_rtn:
    ---
    MOV    FIDR,#10100101B                ;ID Code
    MOV    A,#ID_DATA_1          ;Ex) ID_DATA_1: 93H, ID_DATA_2: 85H, ID_DATA_3: 5AH
    CJNE   A,UserID1,No_WriteErase
    MOV    A,#ID_DATA_2
    CJNE   A,UserID2,No_WriteErase
    MOV    A,#ID_DATA_3
    CJNE   A,UserID3,No_WriteErase
    MOV    FMCR,#0x??             ;0x03 if write, 0x02 if erase
    ---
    ---
    RET

No_WriteErase:
    MOV    FIDR,#00H
    MOV    UserID1,#00H
    MOV    UserID2,#00H
    MOV    UserID3,#00H
    MOV    Flash_flag,#00H
    RET
```

Figure 88. User ID Check Routine for Flash Erase/ Write Code

With codes in Figure 88 invalid Flash erase/write can be avoided.

2. It is important the location where the UserID1/2/3 will be written. The invalid Flash erase/write problem will remain if the UserID1/2/3 is written at the above line of the instruction "MOV FIDR,#10100101B". Therefore, it is recommended to write the UserID1/2/3 in different routine after returning.

   Figure 89 shows example code regarding the recommendation.

```
Decide_ErWt:
    ---
    MOV    Flash_flag1,#38H  ;Random value for example, in case of erase/write needs
    MOV    FSADRL,#20H               ;Here 20H is example,
    MOV    Flash_flag2,#75H
    RET
```

Figure 89. Example Code regarding the Recommendation

3. The Flash sector address (FSADRH/FSADRM/FSADRL) must always keep the address of the Flash which is used for data area. For example, The FSADRH/FSADRM is always 0x00/0x0f" if 0x0f00 to 0x0fff is used for data.

4. Overview of main

```
        ---
        CALL    Work1
        CALL    Decide_ErWt
        CALL    Work2
        CALL    ID_write
        CALL    Work3
        CALL    Flash_erase
        CALL    Flash_write
        ---
        ---
        ---

ID_wire:
        MOV     A,#38H
        CJNE    A,Flash_flag1,No_write_ID
        MOV     A,#75H
        CJNE    A,Flash_flag2,No_write_ID
        MOV     UserID1,#ID_DATA_1      ;Write Uiser ID1
        MOV     A,#38H
        CJNE    A,Flash_flag1,No_write_ID
        MOV     A,#75H
        CJNE    A,Flash_flag2,No_write_ID
        MOV     UserID2,#ID_DATA_2      ;Write Uiser ID2
        MOV     A,#38H
        CJNE    A,Flash_flag1,No_write_ID
        MOV     A,#75H
        CJNE    A,Flash_flag2,No_write_ID
        MOV     UserID3,#ID_DATA_3      ;Write Uiser ID3
        RET

No_write_ID:
        MOV     UserID1,#00H
        MOV     UserID2,#00H
        MOV     UserID3,#00H
        RET
```

Figure 90. Overview of Main

19.8.1 Protection flow of invalid erase/ write

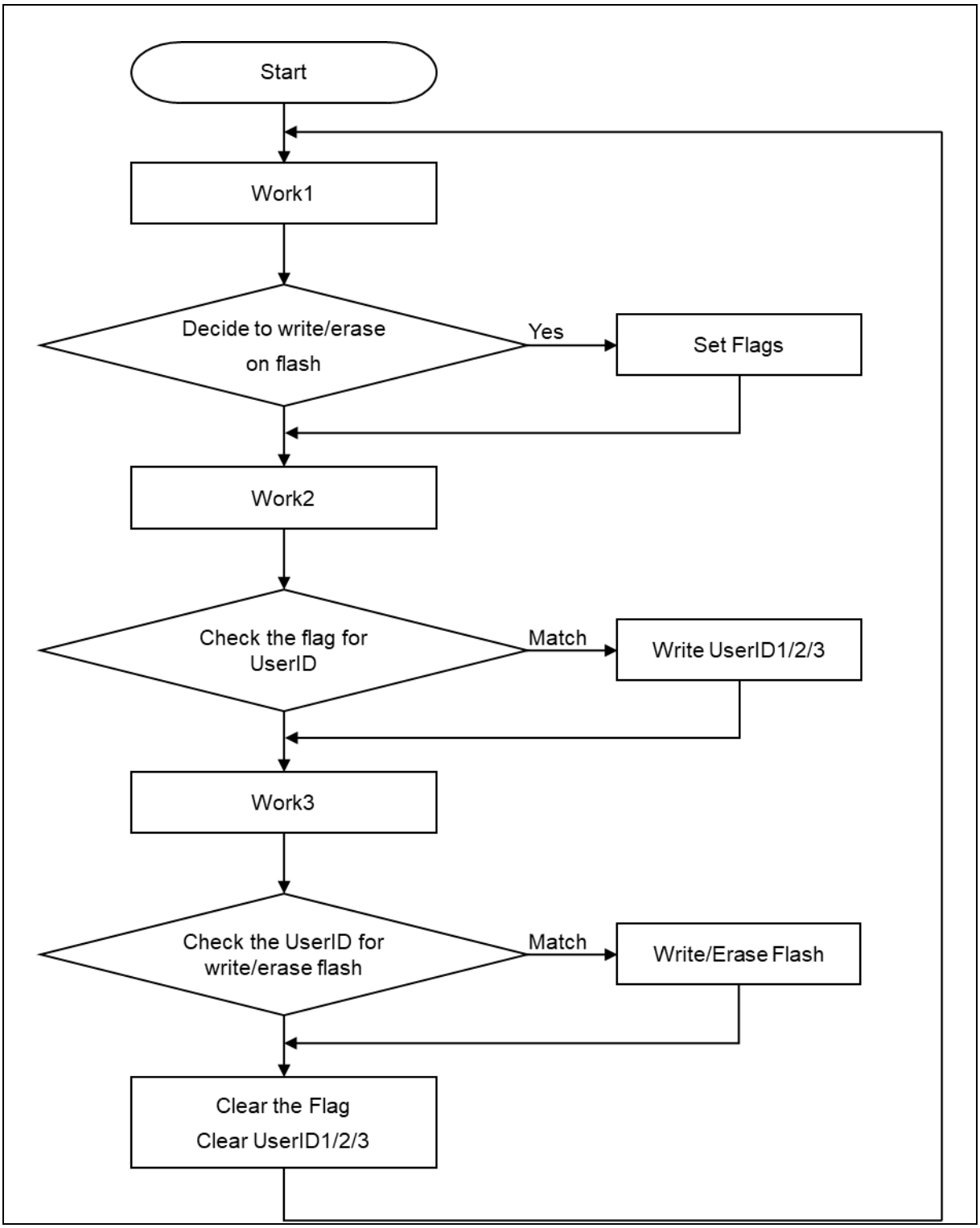


Figure 91. Protection Flow of Invalid Erase/ Write

19.9 Read mode

The Reading program procedure in user program mode is shown in the followings:

- Load received data from Flash memory on MOVC instruction by indirectly addressing mode.

```
MOV     A,#0
MOV     DPH,#0x0F
MOV     DPL,#0xA0           ;Flash memory address

MOVC    A,@A+DPTR           ;read data from Flash memory
```

Figure 92. Program Tip: Reading

19.10 Code write protection mode

The code write protection program procedure in user program mode:

1. Set Flash identification register (FIDR).
2. Check the UserID for to prevent the invalid work [NOTE].
3. Set Flash mode control register (FMCR).

NOTE: Please refer to 19.8 Protection for Invalid Erase/Write.

```
        MOV     FIDR,#0xA5      ;Identification value

        MOV     A,#ID_DATA_1    ;Check the UserID(written by user)
        CJNE    A,UserID1,No_WriteErase ;This routine for UserID must be needed.
        MOV     A,#ID_DATA_2
        CJNE    A,UserID2,No_WriteErase

        MOV     FMCR,#0x04      ;Start Flash Code Write Protection mode
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.

No_WriteErase:
        MOV     FIDR,#00H
        MOV     UserID1,#00H
        MOV     UserID2,#00H
        ---
```

Figure 93. Program Tip: Code Write Protection

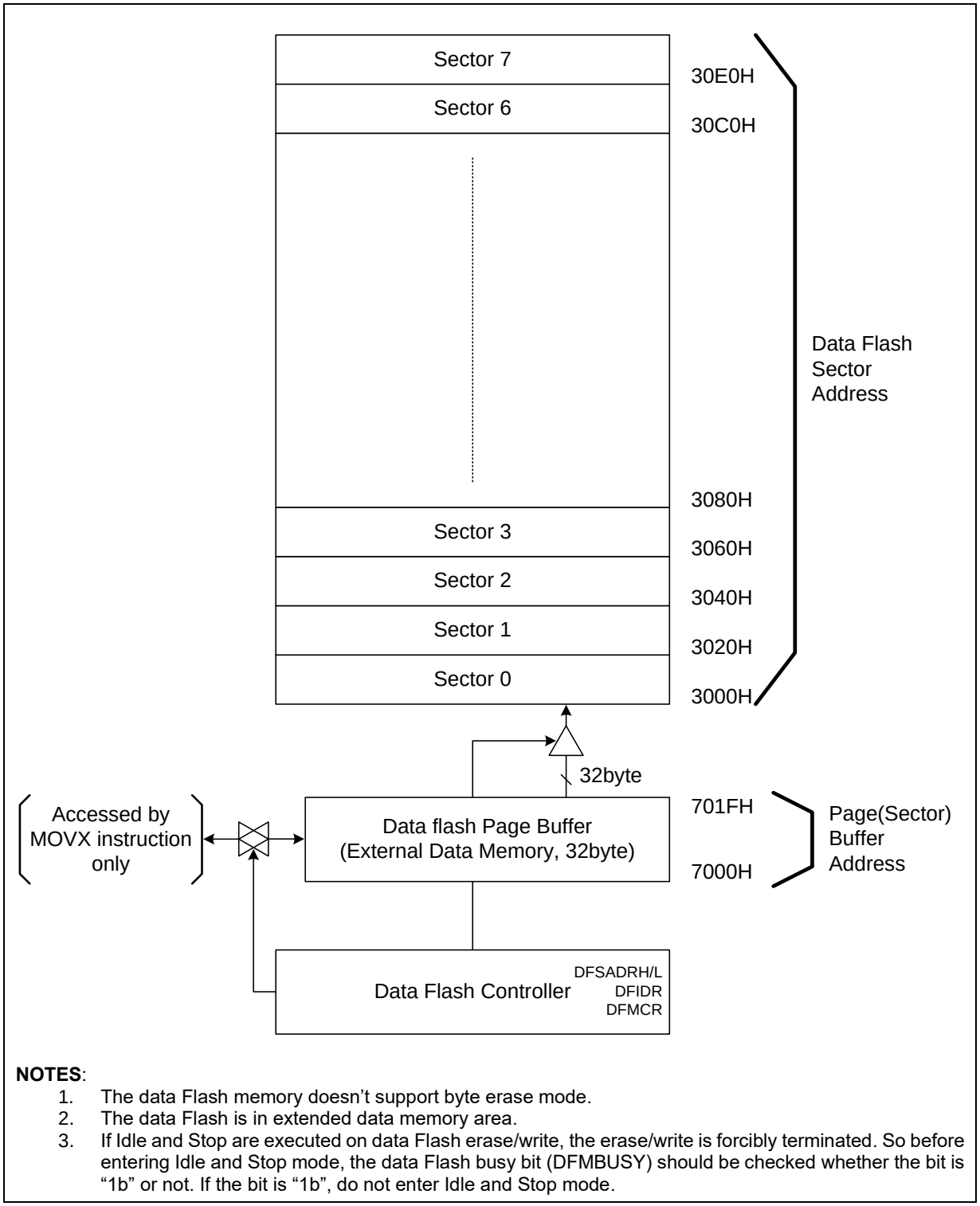
20 Data Flash memory

The A96L414/A96L416 includes Data Flash memory of 256bytes. It can be written, erased, and overwritten. The Data Flash memory can be read by 'MOVX' instruction.

- Data Flash Size: 256bytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 100,000 program/erase cycles at typical voltage and temperature for memory

The write/erase cycles of the internal Data Flash can be increased significantly if it is divided into smaller and used in turn. If 256bytes are divided into 8 areas with 32bytes and the each area from 1st to 8th is used up to 100,000 cycles, the total erase/write is for 800,000 cycles.

Figure 94 describes the relationship between Data Flash page buffer, Data Flash controller, and Data Flash sector addresses.



NOTES:

1. The data Flash memory doesn't support byte erase mode.
2. The data Flash is in extended data memory area.
3. If Idle and Stop are executed on data Flash erase/write, the erase/write is forcibly terminated. So before entering Idle and Stop mode, the data Flash busy bit (DFMBUSY) should be checked whether the bit is "1b" or not. If the bit is "1b", do not enter Idle and Stop mode.

Figure 94. Data Flash Structure

20.1 Register map

Table 32. Data Flash Register Map

Name	Address	Direction	Default	Description
DFSADRH	F3H	R/W	00H	Data Flash Sector Address High Register
DFSADRL	F2H	R/W	00H	Data Flash Sector Address Low Register
DFIDR	F4H	R/W	00H	Data Flash Identification Register
DFMCR	F5H	R/W	00H	Data Flash Mode Control Register

20.2 Register description: Data Flash control and status

DFSADRH (Data Flash Sector Address High Register): F3H

7	6	5	4	3	2	1	0
DFSADRH7	DFSADRH6	DFSADRH5	DFSADRH4	DFSADRH3	DFSADRH2	DFSADRH1	DFSADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DFSADRH[7:0] Data Flash Sector Address High

DFSADRL (Data Flash Sector Address Low Register): F2H

7	6	5	4	3	2	1	0
DFSADRL7	DFSADRL6	DFSADRL5	–	–	–	–	–
R/W	R/W	R/W	–	–	–	–	–

Initial value: 00H

DFSADRL[7:5] Data Flash Sector Address Low

DFIDR (Data Flash Identification Register): F4H

7	6	5	4	3	2	1	0
DFIDR7	DFIDR6	DFIDR5	DFIDR4	DFIDR3	DFIDR2	DFIDR1	DFIDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

DFIDR[7:0] Data Flash Identification
Others No identification value
01101001 Identification value for a Data Flash mode
(These bits are automatically cleared to logic '00H' immediately after one time operation except "Data Flash page buffer reset mode")

DFMCR (Data Flash Mode Control Register): F5H

7	6	5	4	3	2	1	0
DFMBUSY	–	–	–	–	DFMCR2	DFMCR1	DFMCR0
R	–	–	–	–	R/W	R/W	R/W

Initial value: 00H

DFMBUSY Data Flash busy bit.

0 No effect when "1" is written

1 Busy

DFMCR[2:0] Data Flash Mode Control Bits

DFMCR2	DFMCR1	DFMCR0	Description
0	0	1	Select Data Flash page buffer reset mode and start regardless of the DFIDR value.(Clear all 16bytes to '0')
0	1	0	Select Data Flash sector erase mode and start operation when the DFIDR="01101001b"
1	0	0	Select Data Flash sector write mode and start operation when the DFIDR=" 01101001b"
1	1	0	Select Data Flash bulk erase mode and start operation when the DFIDR=" 01101001b"

Others Values: No operation

(Automatically cleared to logic '00H' immediately after one time operation)

20.3 Erase mode

The sector erase program procedure in user program mode:

1. Page buffer clear (DFMCR=0x01)
2. Write '0' to page buffer
3. Set Data Flash sector address register (DFSADRH/DFSADRL).
4. Set Data Flash identification register (DFIDR).
5. Check the UserID for to prevent the invalid work[NOTE].
6. Set Data Flash mode control register (DFMCR).
7. Erase verify

NOTE: Please refer to 19.8 Protection for Invalid Erase/Write.

```
        ANL     EO,#0xF8        ;Set DPTR0
        MOV     DFMCR,#0x01     ;page buffer clear
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.

        MOV     A,#0
        MOV     R0,#DF_SectorSize       ;Sector size of Data Flash
        MOV     DPH,#0x70               ;Page Buffer Address is 7000H
        MOV     DPL,#0

    DF_Pgbuf_clr:
        MOVX    @DPTR,A
        INC     DPTR
        DJNZ    R0,DF_Pgbuf_clr ;Write '0' to all page buffer

        MOV     DFSADRH,#SAH            ;Sector Address High Byte.
        MOV     DFSADRL,#SAL            ;Sector Address Low Byte
        MOV     DFIDR,#0x69             ;Identification value

        MOV     A,#DF_ID_DATA_1         ;Check the UserID(written by user)
        CJNE    A,DF_UserID1,No_DFWriteErase;This routine for UserID must be needed.
        MOV     A,#DF_ID_DATA_2
        CJNE    A,DF_UserID2,No_DFWriteErase

        MOV     DFMCR,#0x02     ;Start Data Flash erase mode
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.

        LJMP    DF_Erase_verify
        ---
    No_DFWriteErase:
        MOV     DFIDR,#00H
        MOV     DF_UserID1,#00H
        MOV     DF_UserID2,#00H
        ---
    DF_Erase_verify:
        MOV     A,DFMCR
        JNB     ACC.7,DF_Erase_verify
        ---
    DF_Verify_error:
        ---
```

Figure 95. Program Tip: Sector Erase

20.4 Write mode

The sector Write program procedure in user program mode

1. Page buffer clear (DFMCR=0x01)
2. Write data to page buffer
3. Set Data Flash sector address register (DFSADRH/DFSADRL).
4. Set Data Flash identification register (DFIDR).
5. Check the UserID for to prevent the invalid work [NOTE].
6. Set Data Flash mode control register (DFMCR).
7. Write verify

NOTE: Data of the address must be "00H" before writing data to an address.

```
        ANL     EO,#0xF8        ;Set DPTR0
        MOV     DFMCR,#0x01     ;page buffer clear
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.

        MOV     A,#0
        MOV     R0,#DF_SectorSize       ;Sector size of DATA FLASH
        MOV     DPH,#0x70               ;Page Buffer Address is 7000H
        MOV     DPL,#0

    DF_Pgbuf_WR:
        MOVX    @DPTR,A
        INC     A
        INC     DPTR
        DJNZ    R0,DF_Pgbuf_WR          ;Write data to all page buffer

        MOV     DFSADRH,#SAH            ;Sector Address High Byte.
        MOV     DFSADRL,#SAL            ;Sector Address Low Byte
        MOV     DFIDR,#0x69             ;Identification value

        MOV     A,#DF_ID_DATA_1                 ;Check the UserID(written by user)
        CJNE    A,DF_UserID1,No_DFWriteErase;This routine for UserID must be needed.
        MOV     A,#DF_ID_DATA_2
        CJNE    A,DF_UserID2,No_DFWriteErase

        MOV     DFMCR,#0x04     ;Start DATA FLASH write mode
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.

        LJMP    DF_Write_verify
        ---
    No_DFWriteErase:
        MOV     DFIDR,#00H
        MOV     DF_UserID1,#00H
        MOV     DF_UserID2,#00H
        ---
    DF_Write_verify:
        MOV     A,DFMCR
        JNB     ACC.7,DF_Write_verify
        ---
    DF_Verify_error:
        ---
```

Figure 96. Program Tip: Sector Write

The Byte Write program procedure in user program mode

1. Page buffer clear (DFMCR=0x01)
2. Write data to page buffer
3. Set Data Flash sector address register (DFSADRH/DFSADRL).
4. Set Data Flash identification register (DFIDR).
5. Check the UserID for to prevent the invalid work [NOTE].
6. Set Data Flash mode control register (DFMCR).
7. Write verify

NOTE: Data of the address must be "00H" before writing data to an address.

```
        ANL     EO,#0xF8        ;Set DPTR0
        MOV     DFMCR,#0x01     ;page buffer clear
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.

        MOV     A,#5
        MOV     DPH,#0x70
        MOV     DPL,#0
        MOVX    @DPTR,A                 ;Write data to page buffer

        MOV     A,#6
        MOV     DPH,#0x70
        MOV     DPL,#0x05
        MOVX    @DPTR,A                 ;Write data to page buffer

        MOV     DFSADRH,#SAH ;Sector Address High Byte.
        MOV     DFSADRL,#SAL  ;Sector Address Low Byte
        MOV     DFIDR,#0x69     ;Identification value

        MOV     A,#DF_ID_DATA_1 ;Check the UserID(written by user)
        CJNE    A,DF_UserID1,No_DFWriteErase;This routine for UserID must be needed.
        MOV     A,#DF_ID_DATA_2
        CJNE    A,DF_UserID2,No_DFWriteErase

        MOV     DFMCR,#0x04     ;Start DATA FLASH write mode
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.
        NOP                     ;Dummy instruction, This instruction must be needed.

        LJMP    DF_Write_verify
        ---
    No_DFWriteErase:
        MOV     DFIDR,#00H
        MOV     DF_UserID1,#00H
        MOV     DF_UserID2,#00H
        ---
    DF_Write_verify:
        MOV     A,DFMCR
        JNB     ACC.7,DF_Write_verify
        ---
    DF_Verify_error:
        ---
```

Figure 97. Program Tip: Byte Write

20.5 Read mode

The Reading program procedure in user program mode

- Load the received data from Data Flash memory on MOVX instruction by indirectly addressing mode.

```
MOV     DPH,#0x30
MOV     DPL,#0x10           ;Data Flash memory address

MOVX    A,@DPTR             ;read data from Data Flash memory
```

Figure 98. Program Tip: Reading

21 Electrical characteristics

21.1 Absolute maximum ratings

Table 33. Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Remark
Supply voltage	VDD	-0.3 ~ +4.0	V	–
Normal voltage Pin	VI	-0.3 ~ VDD+0.3	V	Voltage on any pin with respect to VSS
	VO	-0.3 ~ VDD+0.3	V	
	IOH	-20	mA	Maximum current output sourced by (I_{OH} per I/O pin)
	∑IOH	-80	mA	Maximum current ($\sum I_{OH}$)
	IOL	60	mA	Maximum current sunk by (I_{OL} per I/O pin)
	∑IOL	120	mA	Maximum current ($\sum I_{OL}$)
Constant sink pin	IOL	390	mA	Maximum current sink by ICS0 and ICS1
Total power dissipation	PT	600	mW	–
Storage temperature	TSTG	-65 ~ +150	°C	–

Caution: Stresses beyond those listed under 'Absolute Maximum Ratings' may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

21.2 Operating conditions

The device must be used in operating conditions that comply with the parameters in Table 34.

Table 34. Recommended Operating Conditions

(TA=-40℃ to +85℃)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Operating voltage	VDD	fx=0.5 to 4.0MHz, HFIRC	2.0	—	3.6	V
		fx=32KHz, LFIRC	2.0	—	3.6	
Operating temperature	T_{OPR}	VDD=2.0 to 3.6V	-40	—	85	℃

21.3 ADC characteristics

Table 35. ADC Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V, VSS=0V)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Resolution	—	—		—	10	—	bit
Integral linear error	ILE	AVREF= 2.2V to 3.6V f_{ADCCK}=2MHz		—	—	±3	LSB
Differential linearity error	DLE			—	—	±1	
Top offset error	TOE			—	—	±5	
Zero offset error	ZOE			—	—	±5	
Conversion time	t_{CON}	AVREF= 2.2V to 3.6V		14	—	—	us
Analog input voltage	V_{AN}	—		VSS	—	VDD	V
Sample/ hold time	t_{SH}	—		3	—	—	us
Band gap reference voltage	V_{BGR}	T_A = +25℃		-3	0.92	+3	%
A/DC input leakage current	IAN	VDD=3V		—	—	2	uA
A/DC current	I_{ADC}	Enable	VDD=3V	—	200	350	uA
		Disable		—	—	0.1	uA

NOTES:

1. Zero offset error is the difference between 0000000000 and the converted output for zero input voltage (VSS).
2. Top offset error is the difference between 1111111111 and the converted output for top input voltage (AVREF).
3. If AVREF is less than 2.7V, the resolution degrades by 1-bit whenever AVREF drops 0.1V. (@ADCLK = 0.5MHz, under 2.7V resolution has no test.)

21.4 LDO characteristics

Table 36. LDO Characteristics

(T_A=-40℃ to +85℃, VDD=2.7V to 3.6V, VSS=0V)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Supply voltage	VDD	–		2.6	–	3.6	V
Output voltage	V_{LDO23}	T_A = +25℃		2.24	2.32	2.40	V
LDO output current	I_{LDO23}	VDD=3V, CL=0.1uF, T_A = +25℃		2	–	–	mA
Stabilization time	t_{STA}			–	–	100	us
LDO block current	I_{LDO}	Enable	VDD=3V, No load	–	30	50	uA
		Disable		–	–	0.1	

21.5 Power on Reset

Table 37. Power on Reset Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V, VSS=0V

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
RESET release level	V_{POR}	–	–	1.4	–	V
VDD voltage rising time	t_R	0.2V to 2.0V	0.05	–	100	V/ms
POR current	I_{POR}	–	–	0.2	–	uA

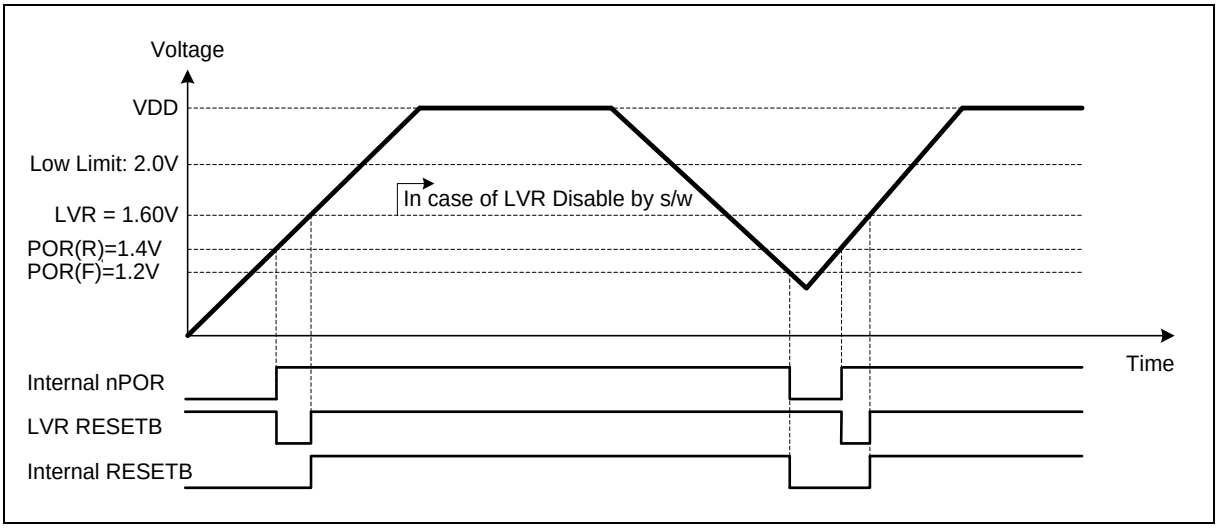


Figure 99. Power-On Reset Timing

21.6 Low voltage reset characteristics

Table 38. LVR Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V V to 3.6V, VSS=0V)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Detection level	V_{LVR}	The LVR can select all levels.		_	1.60	1.89	V
				2.05	2.20	2.35	
				2.20	2.40	2.60	
				2.45	2.70	2.95	
Hysteresis	△V	–		–	10	100	mV
Minimum pulse width	t_{LW}	–		100	–	–	us
LVR current	I_{LVR}	Enable	VDD= 3V	_	0.8	1.6	uA
		Disable		_	–	0.1	

21.7 Operational amplifier 0/1 characteristics

Table 39. Operational Amplifier 0/1 Characteristic

(T_A=-40℃ to +85℃, VDD=2.2V to 3.6V, VSS=0V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input offset voltage	V_{OF}	VDD=3.0V	—	±10	±100	uV
Input offset current	I_{OF}	VDD=3.0V, VCM=0V	—	15	50	pA
Common-mode rejection ratio	CMRR	VDD=3.0V, DC VCM=0V to VDD-1.2V	80	100	—	dB
Power supply rejection ration	PSRR	VDD=3.0V	80	100	—	
Open loop voltage gain	—	VDD=3.0V	100	120	—	dB
Gain error	ERR	VDD=3.0V, VIN ≥ 0.1V, x10 VIN<(Input x Gain)	—	1	—	%
Input Common-mode voltage range	V_{IN}	VDD=3.0V	0	—	VDD-1.2	V
Output voltage range	V_O	VDD=3.0V, RL=10kΩ	VSS+0.1	—	VDD-0.1	V
Output short circuit current	ISCH	VDD=3.0V, Absolute	—	9	—	mA
	ISCL		—	12	—	
Gain bandwidth	f_{GB}	VDD=3.0V	1	2	—	MHz
Voltage follower pulse response	T_{AR}	VDD=3.0V, Small Signal	—	5	10	us
OP-AMP 0/1 total current	I_{AMP}	Enable, VDD=3.0V, No Load	—	220	300	uA
		Disable, VDD=3.0V, No Load	—	—	0.1	
Enable time of AMP0/1	t_{ON}	VDD=3.0V, Gain=x20/x30, RL=10KΩ with 50pF	—	—	150	us
Input noise voltage density	e_{ni}	Input Referred f=1Hz	—	0.1	—	uV/rtHz
		Input Referred f=1KHz	—	50	—	nV/rtHz
Slew rate	S_R	VDD=3.0V, RL=10K, CL=50pF	—	0.7	—	V/us
Phase margin	P_M	VDD=3.0V, RL=10K, CL=50pF	—	60	—	Degrees
Chopping clock	f_{CHOP}	—	125	—	167	kHz

21.8 High frequency internal RC oscillator characteristics

Table 40. High Frequency Internal RC Oscillator Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V, VSS=0V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Frequency	f_{HFIRC}	VDD=3.0V	—	4	—	MHz
Tolerance	—	T_A = -10°C to +50°C, with user (S/W) trim.	—	—	±1.0	%
		T_A = -10°C to +50°C			±2.0	
		T_A = -40°C to +85°C			±3.0	
Clock duty ratio	T_{OD}	—	40	50	60	%
Stabilization time	T_{FS}	—	—	—	4	us

21.9 Low frequency internal RC oscillator characteristics

Table 41. Low Frequency Internal RC Oscillator Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V, VSS=0V)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Frequency	f_{LFIRC}	VDD=2.0V – 3.6V		—	32	—	kHz
Tolerance	—	T_A = -40°C to +85°C	With 0.1uF bypass capacitor	—	—	±10	%
Clock duty ratio	T_{OD}	—		40	50	60	%
Stabilization time	T_{FS}	—		—	—	120	us
LFIRC current	I_{LFIRC}	Enable	VDD=3V	—	1	—	uA
		Disable		—	—	0.1	

21.10 Internal Watchdog Timer RC oscillator characteristics

Table 42. Internal WDTRC Oscillator Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V, VSS=0V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Frequency	f_{WDTRC}	—	0.5	1	2	kHz
Stabilization time	t_{WDTS}	—	—	—	1	ms
WDTRC current	I_{WDTRC}	Enable	—	1	—	uA
		Disable	—	—	0.1	

21.11 DC characteristics

Table 43. DC Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V, VSS=0V, f_{HFIRC}=4MHz)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input high voltage	V_{IH1}	All input pins except VIH2, RESETB		0.8VDD	—	VDD	V
	V_{IH2}	P02–P05		0.7VDD	—	VDD	
Input low voltage	V_{IL1}	All input pins except VIL2, RESETB		—	—	0.2VDD	V
	V_{IL2}	P02–P05		—	—	0.3VDD	
Output high voltage	V_{OH}	VDD=3V, All output ports, IOH = – 4mA		VDD-1.0	—	—	V
Output low voltage	V_{OL}	VDD=3V, IOL = 6mA		—	—	1.0	V
Input high leakage current	I_{IH}	All output ports		—	—	1.0	uA
Input low leakage current	I_{IL}	All output ports		-1.0	—	—	uA
Pull-up resistor	R_{PU1}	VI=0V, T_A=25℃, All Input ports	VDD=3.0V	25	50	100	kΩ
	R_{PU2}	VI=0V, T_A=25℃, RESETB	VDD=3.0V	100	200	400	kΩ
Supply current	I_{DD1} (RUN)	f_{HFIRC} = 4MHz	VDD=3V±10%	—	400	600	uA
		f_{HFIRC} = 2MHz		—	240	360	
		f_{HFIRC} = 1MHz		—	170	250	
	I_{DD2} (IDLE)	f_{HFIRC} = 4MHz	VDD=3V±10%	—	100	150	uA
		f_{HFIRC} = 2MHz		—	90	135	
		f_{HFIRC} = 1MHz		—	75	110	
	I_{DD3} (RUN)	VDD=3V, f_{LFIRC}=32kHz, TA=25℃		—	40	65	uA
	I_{DD4} (IDLE)	VDD=3V, f_{LFIRC}=32kHz, TA=25℃		—	2	5	uA
	I_{DD5}	STOP, VDD= 3V±10%, T_A= 25℃		—	0.5	3.0	uA
		STOP, VDD= 3V±10%, T_A= 25℃, LFIRC on		—	1.5	4.0	

NOTES:

1. Where the f_{HFIRC} is a high frequency internal RC oscillator and the f_{LFIRC} is a low frequency.
2. All supply current items don't include the current of an internal Watch-dog timer RC (WDTRC) oscillator and a peripheral block.
3. All supply current include the current of the power-on reset (POR) block.

21.12 Constant sink current electrical characteristics

Table 44. Constant Sink Current Electrical Characteristics

(T_A=-40℃ to +85℃, VDD=2.2V to 3.6V, VSS=0V)

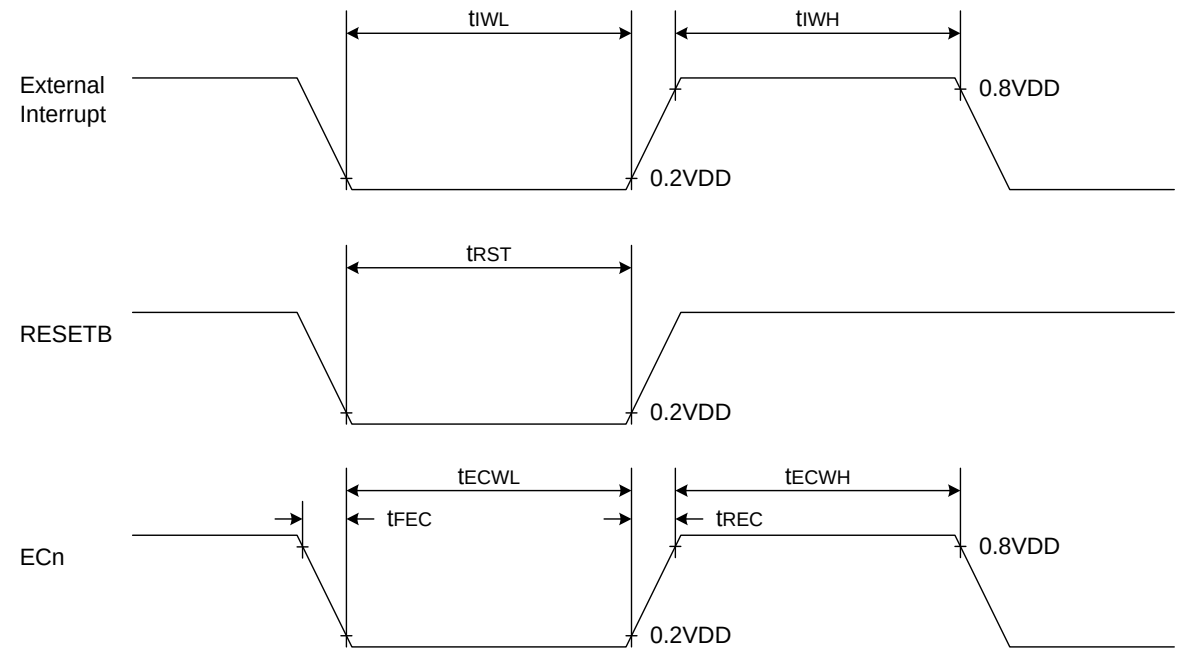
Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Constant sink current	I_{CS}	VDD=3V V_{ICS}=1.5V T_A=25℃	ICSDR[3:0] = 0	-7%	49	+7%	mA
			ICSDR[3:0] = 1	-7%	65	+7%	
			ICSDR[3:0] = 2	-7%	80	+7%	
			ICSDR[3:0] = 3	-7%	96	+7%	
			ICSDR[3:0] = 4	-7%	111	+7%	
			ICSDR[3:0] = 5	-7%	127	+7%	
			ICSDR[3:0] = 6	-7%	142	+7%	
			ICSDR[3:0] = 7	-7%	158	+7%	
			ICSDR[3:0] = 8	-7%	173	+7%	
			ICSDR[3:0] = 9	-7%	188	+7%	
			ICSDR[3:0] = 10	-7%	203	+7%	
			ICSDR[3:0] = 11	-7%	218	+7%	
			ICSDR[3:0] = 12	-7%	232	+7%	
			ICSDR[3:0] = 13	-7%	246	+7%	
			ICSDR[3:0] = 14	-7%	260	+7%	
			ICSDR[3:0] = 15	-7%	274	+7%	
		VDD=3V V_{ICS}=1V to 2.0V	ICSDR[3:0] = n n: 0 to 15	-15%	Typ.	+15%	
		VDD=2.7V to 3.6V V_{ICS}=1V to VDD-1.0V	ICSDR[3:0] = n n: 0 to 15	-20%	Typ.	+20%	

21.13 AC characteristics

Table 45. AC Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
RESETB input low width	t_{RST}	VDD = 3V	10	—	—	us
Interrupt input high, low width	t_{IWH}, t_{IWL}	All interrupt, VDD = 3V	200	—	—	ns
External counter input high, low pulse width	t_{ECWH}, t_{ECWL}	ECn, VDD = 3V Where n=0, 1, and 2	200	—	—	
External counter transition time	t_{REC}, t_{FEC}	ECn, VDD = 3V Where n=0, 1, and 2	20	—	—	
Wake-up from idle/stop mode	t_{WU0}	From HFIRC	—	—	16	us
	t_{WU1}	From LFIRC	—	—	1000	us



NOTE: n =0, 1 and 2.

Figure 100. AC Timing

21.14 SPI characteristics

Table 46. SPI Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Output clock pulse period	t_{SCK}	Internal SCK source	1000	—	—	ns
Input clock pulse period		External SCK source	1000	—	—	
Output clock high, low pulse width	t_{SCKH} t_{SCKL}	Internal SCK source	400	—	—	
Input clock high, low pulse width		External SCK source	400	—	—	
First output clock delay time	t_{FOD}	Internal/external SCK source	500	—	—	
Output clock delay time	t_{DS}	—	—	—	125	
Input setup time	t_{DIS}	—	500	—	—	
Input hold time	t_{DIH}	—	500	—	—	

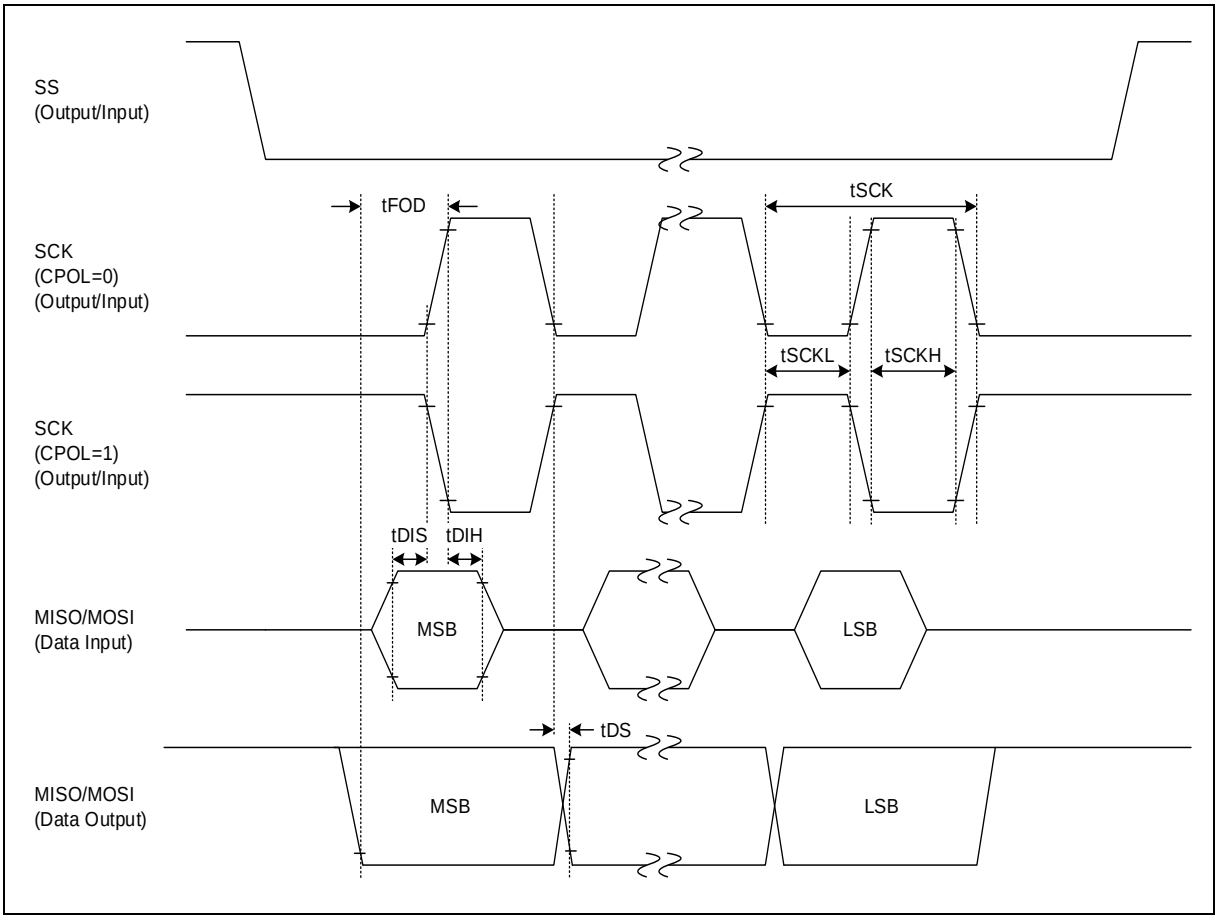


Figure 101. SPI Timing

21.15 UART timing characteristics

Table 47. UART Timing Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V, fx=1MHz)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Serial port clock cycle time	t_{SCK}	13.92	t_{CPU} x 16	18.08	us
Output data setup to clock rising edge	t_{S1}	6.5	t_{CPU} x 13	—	
Clock rising edge to input data valid	t_{S2}	—	—	6.5	
Output data hold after clock rising edge	t_{H1}	t_{CPU} – 0.1	t_{CPU}	—	
Input data hold after clock rising edge	t_{H2}	0	—	—	
Serial port clock High, Low level width	t_{HIGH}, t_{LOW}	5.5	t_{CPU} x 8	10.5	

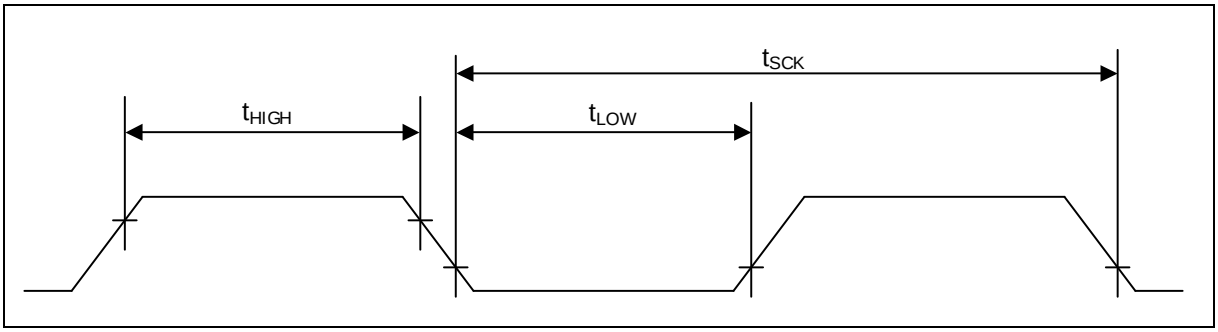


Figure 102. UART Timing Characteristics

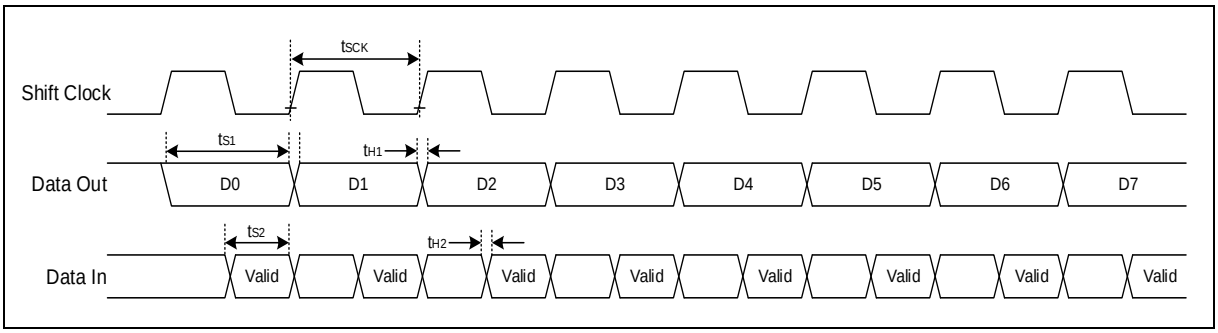


Figure 103. Timing Waveform of UART Module

21.16 I2C characteristics

Table 48. I2C Characteristics

(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V)

Parameter	Symbol	Standard Mode		High-speed Mode		Unit
		Min.	Max.	Min.	Max.	
Clock frequency	t_{CL}	0	100	0	400	kHz
Clock high pulse width	t_{SCLH}	4.0	—	0.6	—	us
Clock low pulse width	t_{SCLL}	4.7	—	1.3	—	
Bus free time	t_{BF}	4.7	—	1.3	—	
Start condition setup time	t_{STSU}	4.7	—	0.6	—	
Start condition hold time	t_{STHD}	4.0	—	0.6	—	
Stop condition setup time	t_{SPSU}	4.0	—	0.6	—	
Stop condition hold time	t_{SPHD}	4.0	—	0.6	—	
Output valid from clock	t_{VD}	0	—	0	—	
Data input hold time	t_{DIH}	0	—	0	1.0	
Data input setup time	t_{DIS}	250	—	100	—	ns

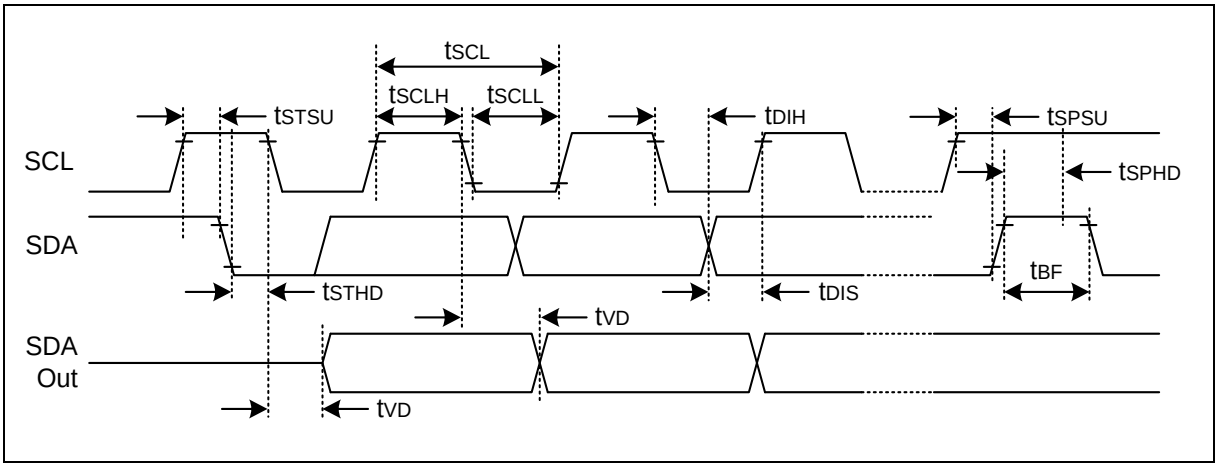


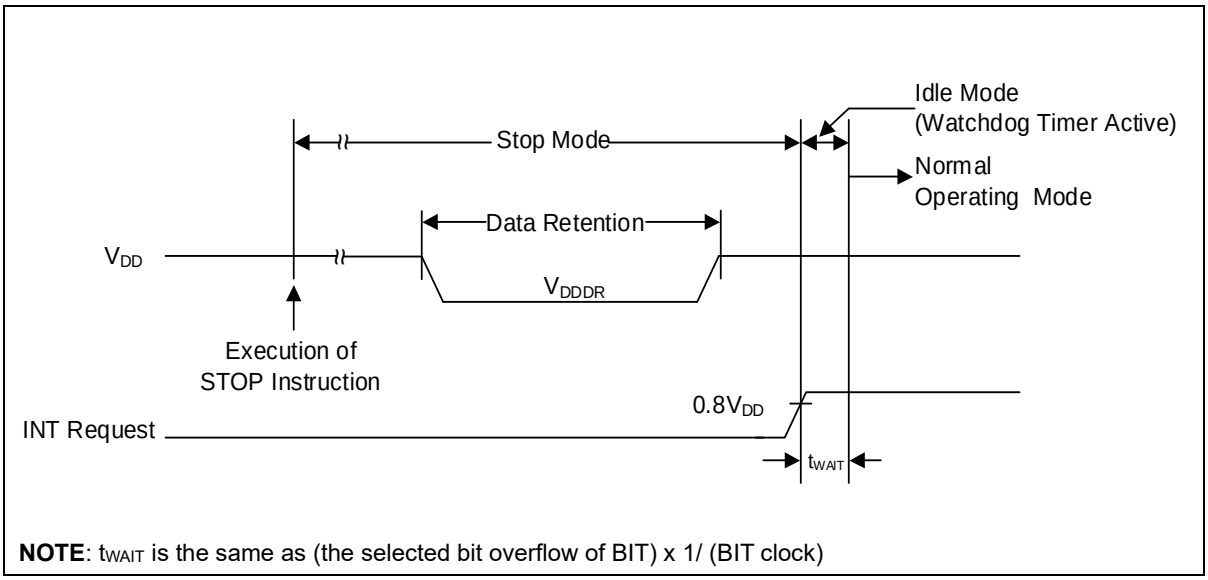
Figure 104. Timing Waveform of I2C

21.17 Data retention voltage in STOP mode

Table 49. Data Retention Voltage in STOP Mode

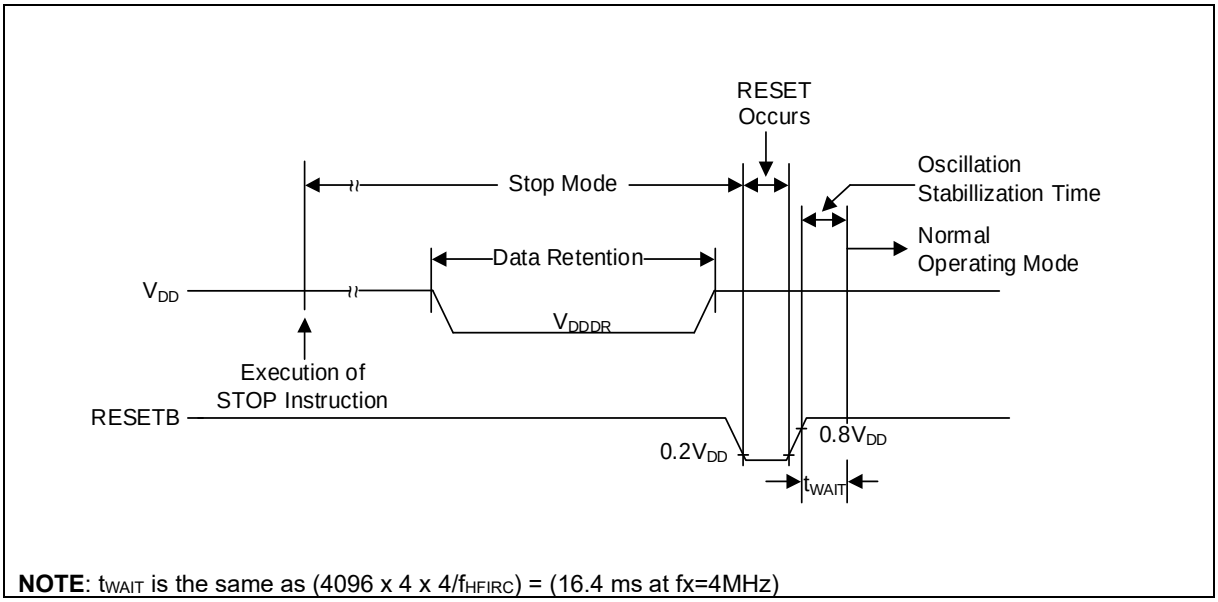
(T_A=-40℃ to +85℃, VDD=2.0V to 3.6V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Data retention supply voltage	V_{DDDR}	—	2.0	—	3.6	V
Data retention supply current	I_{DDDR}	VDDR = 2.0V (T_A= 25°C), STOP mode	—	—	1	uA



NOTE: t_{WAIT} is the same as (the selected bit overflow of BIT) x 1/ (BIT clock)

Figure 105. STOP Mode Release Timing when Initiated by an Interrupt



NOTE: t_{WAIT} is the same as (4096 x 4 x 4/f_{HFIRC}) = (16.4 ms at fx=4MHz)

'Figure 106. STOP Mode Release Timing when Initiated by RESETB

21.18 Internal Flash characteristics

Table 50. Internal Flash Characteristics

(T_A= +25℃, VDD=2.0V to 3.6V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Sector write time	t_{FSW}	—	—	2.5	2.7	ms
Sector erase time	t_{FSE}	—	—	2.5	2.7	
Code write protection time	t_{FHL}	—	—	2.5	2.7	
Page buffer reset time	t_{FBR}	—	—	—	5	us
Flash program Voltage	V_{PGM}	On erase/write	2.0	—	3.6	V
System clock frequency	f_{SCLK}	—	0.5	—	—	MHz
Endurance of write/erase	N_{FWE}	Sector erase, byte write	10,000	—	—	cycles

21.19 Internal Data Flash characteristics

Table 51. Internal Data Flash Characteristics

(T_A= +25℃, VDD=2.0V to 3.6V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Sector write time	t_{DFSW}	—	—	2.5	2.7	ms
Sector erase time	t_{DFSE}	—	—	2.5	2.7	
Page buffer reset time	t_{DFBR}	—	—	—	5	us
Data Flash program voltage	V_{DPGM}	On erase/write	2.0	—	3.6	V
System clock frequency	f_{SCLK}	—	0.5	—	—	MHz
Endurance of write/erase	NE_{WE}	Sector erase, byte write	100,000	—	—	cycles

21.20 Input/output capacitance characteristics

Table 52. I/O Capacitance Characteristics

(T_A=-40℃ to +85℃, VDD=0V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input capacitance	C_{IN}	fx=1MHz Unmeasured pins are connected to VSS.	—	—	10	pF
Output capacitance	C_{OUT}					
I/O capacitance	C_{IO}					

21.21 Recommended circuit and layout

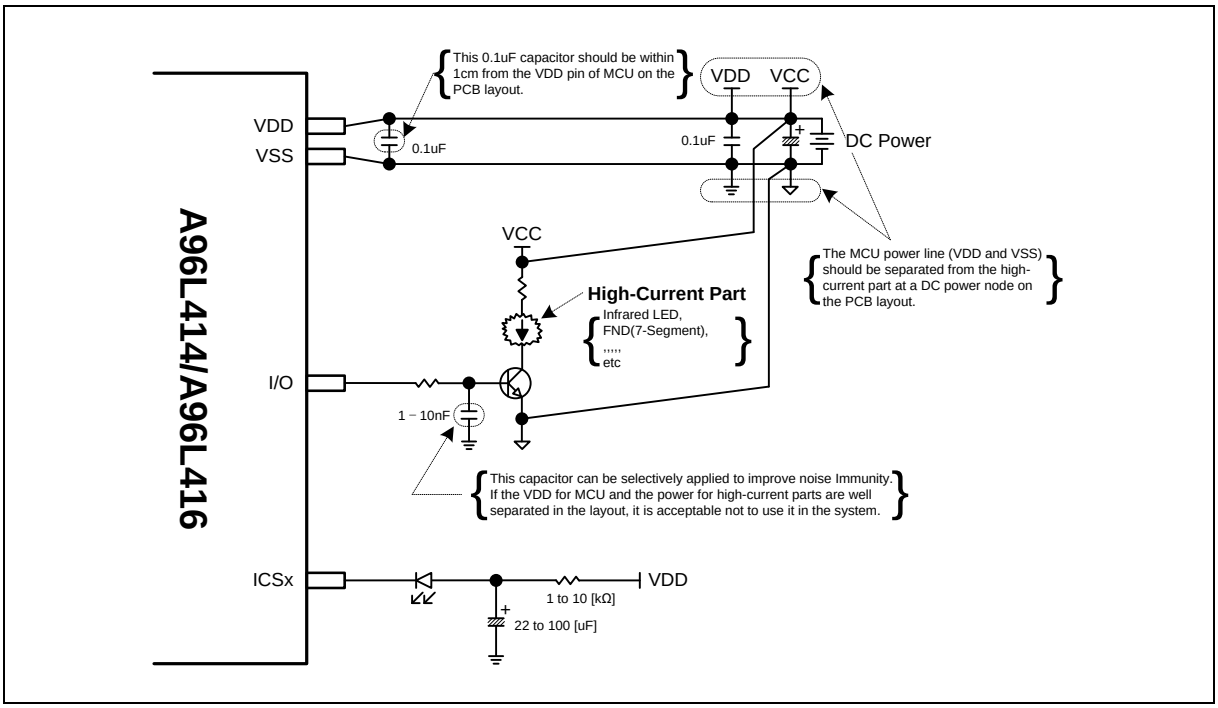
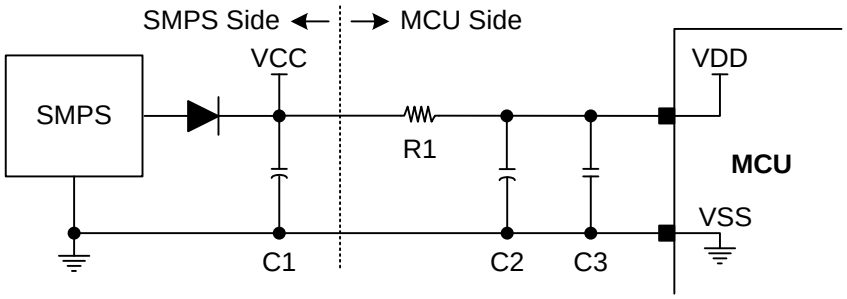


Figure 107. Recommended Circuit and Layout

21.22 Recommended circuit and layout with SMPS power



1. The C1 capacitor is to flatten out the voltage of the SMPS power, VCC.
   √ Recommended C1: 470uF/25V more.
2. The R1 and C2 are the RC filter for VDD and suppress the ripple of VCC.
   √ Recommended R1: 10Ω - 20Ω
   √ Recommended C2: 47uF/25V more
   √ The R1 and C2 should be as close by the C3 as possible.
3. The C3 capacitor is used for temperature compensation because an electrolytic capacitor becomes worse characteristics at low temperature.
   √ Recommended C3: ceramic capacitor 2.2uF more
   √ The C3 should be within 1cm from VDD pin of MCU on the PCB layout.
4. The above circuit is recommended to improve noise immunity (EFT, Surge, ESD, etc) when the SMPS supplies the VDD of MCU.

Figure 108. Recommended Circuit and Layout with SMPS Power

21.23 Typical characteristics

Figures and tables introduced in this chapter can be used only for design guidance, and are not tested or guaranteed. In graphs or tables some data may exceed specified operating range, and can be only for information. The device is guaranteed to operate properly only within the specified range.

The data presented in this chapter is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

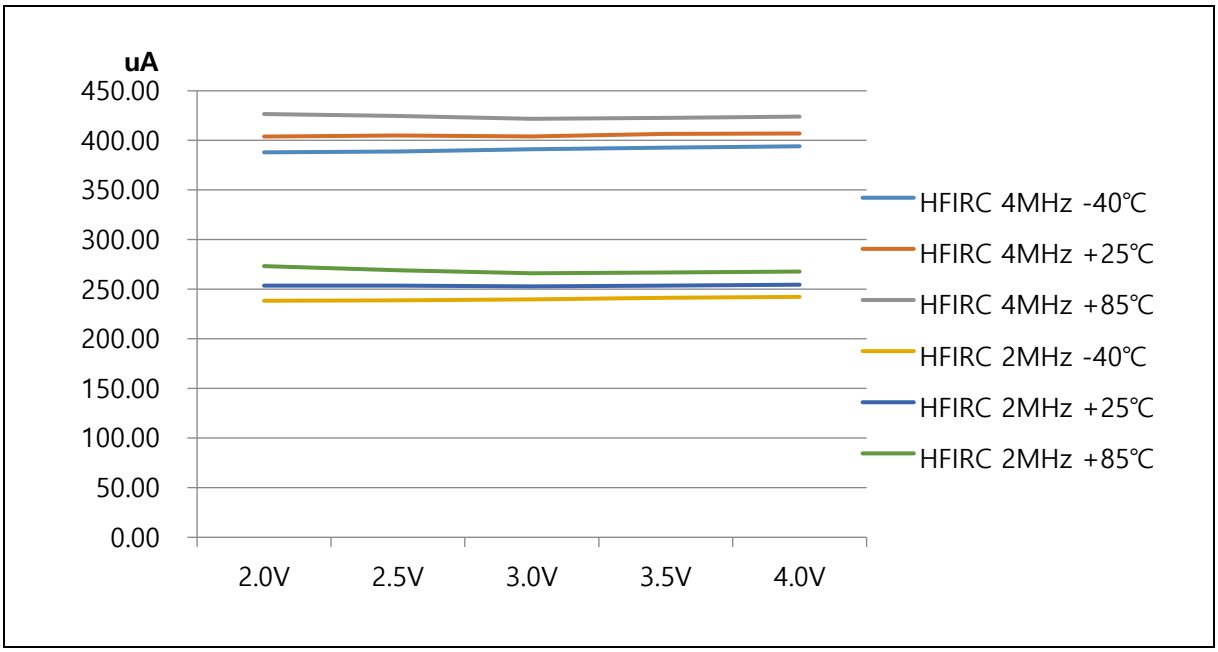


Figure 109. RUN (IDD1) Current

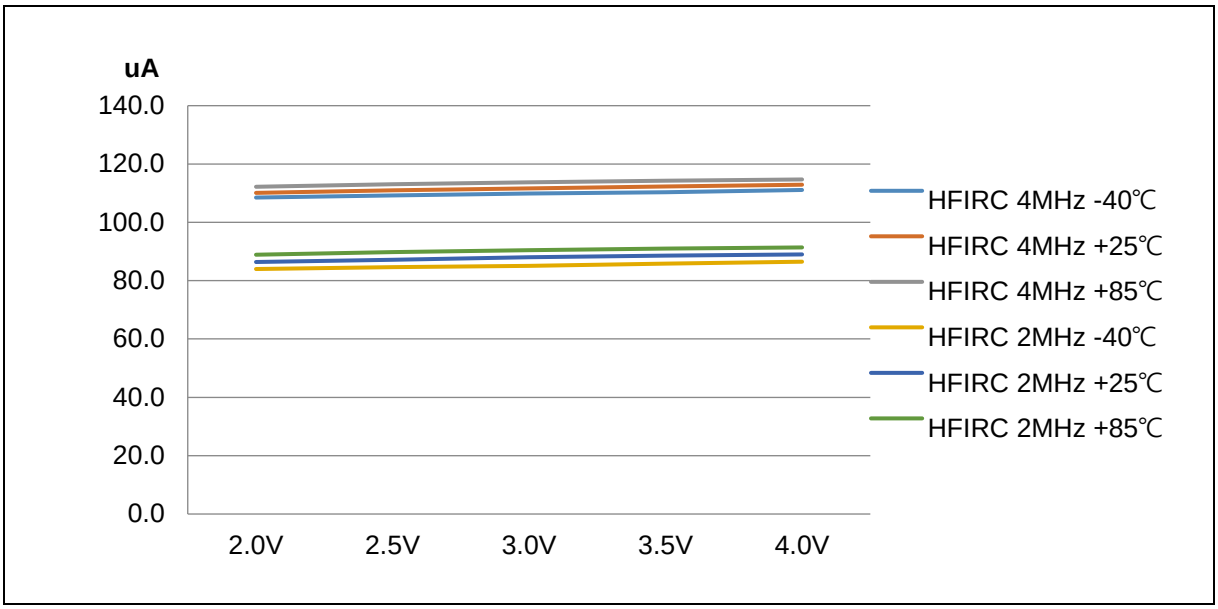


Figure 110. IDLE (IDD2) Current

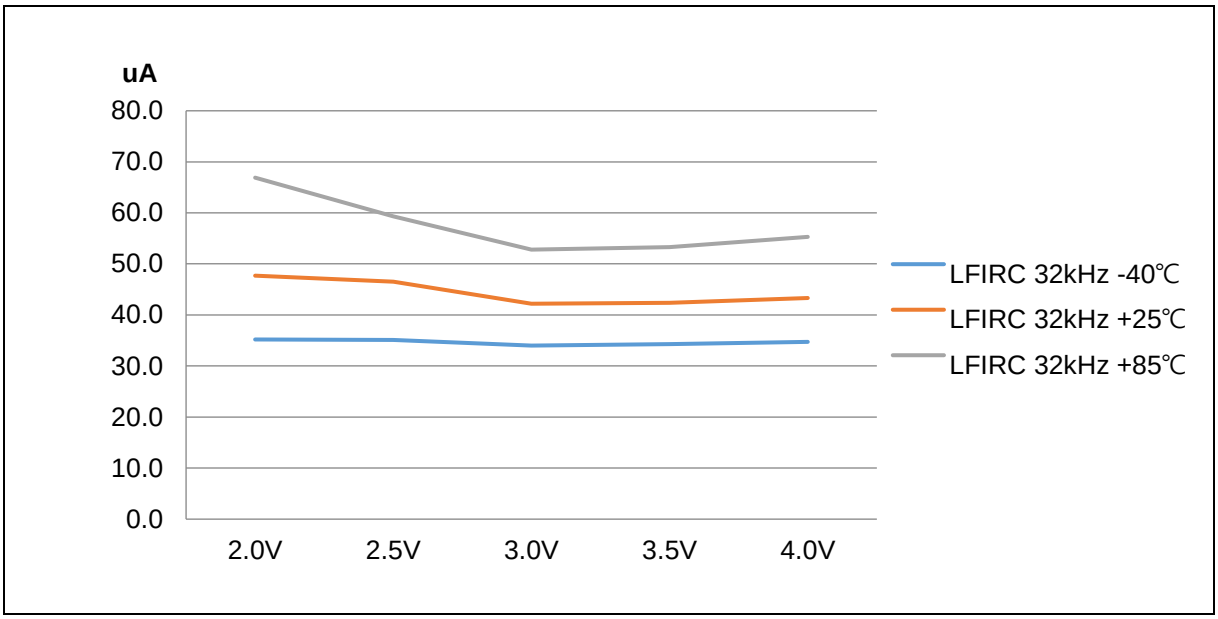


Figure 111. RUN (IDD3) Current

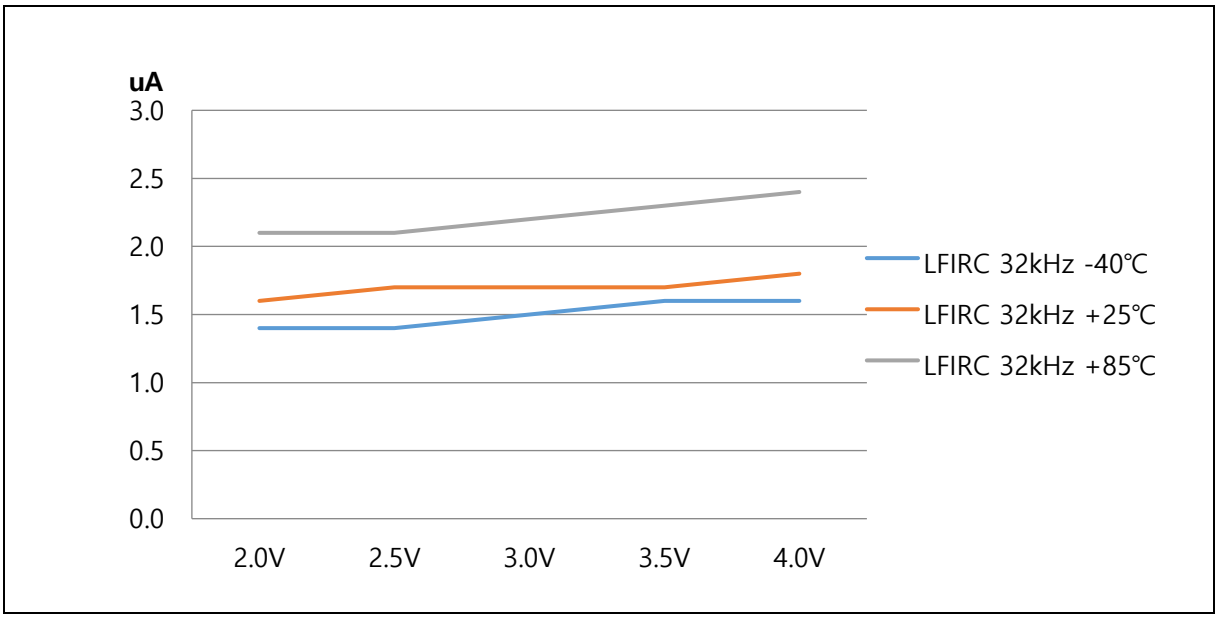


Figure 112. IDLE (IDD4) Current

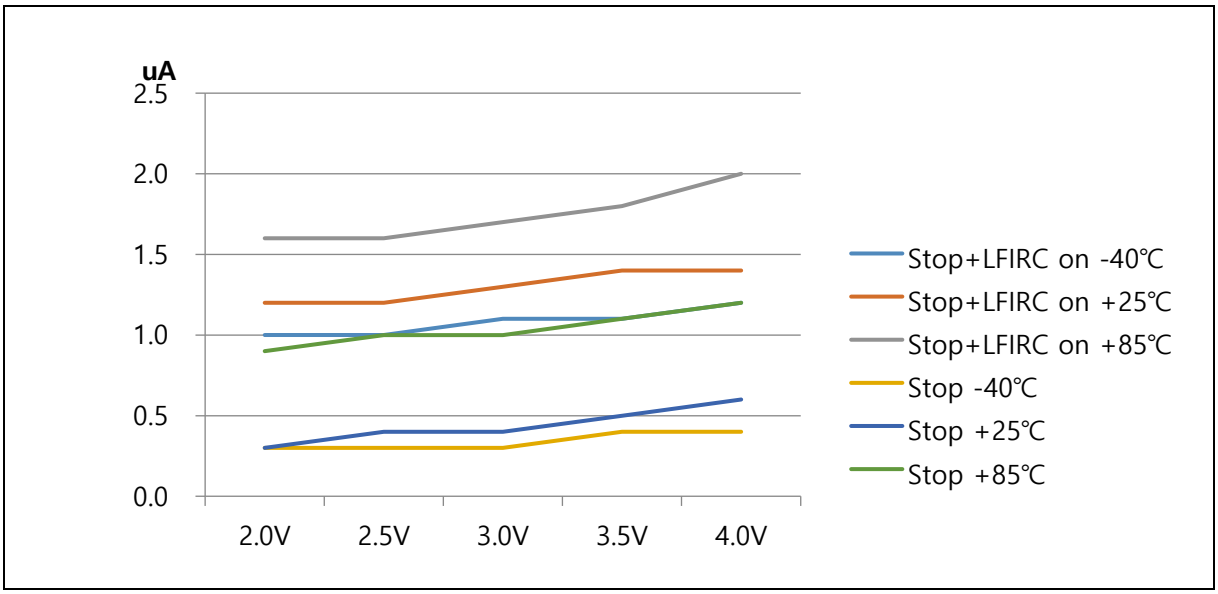


Figure 113. STOP (IDD5) Current

22 Package information

22.1 20 TSSOP package information

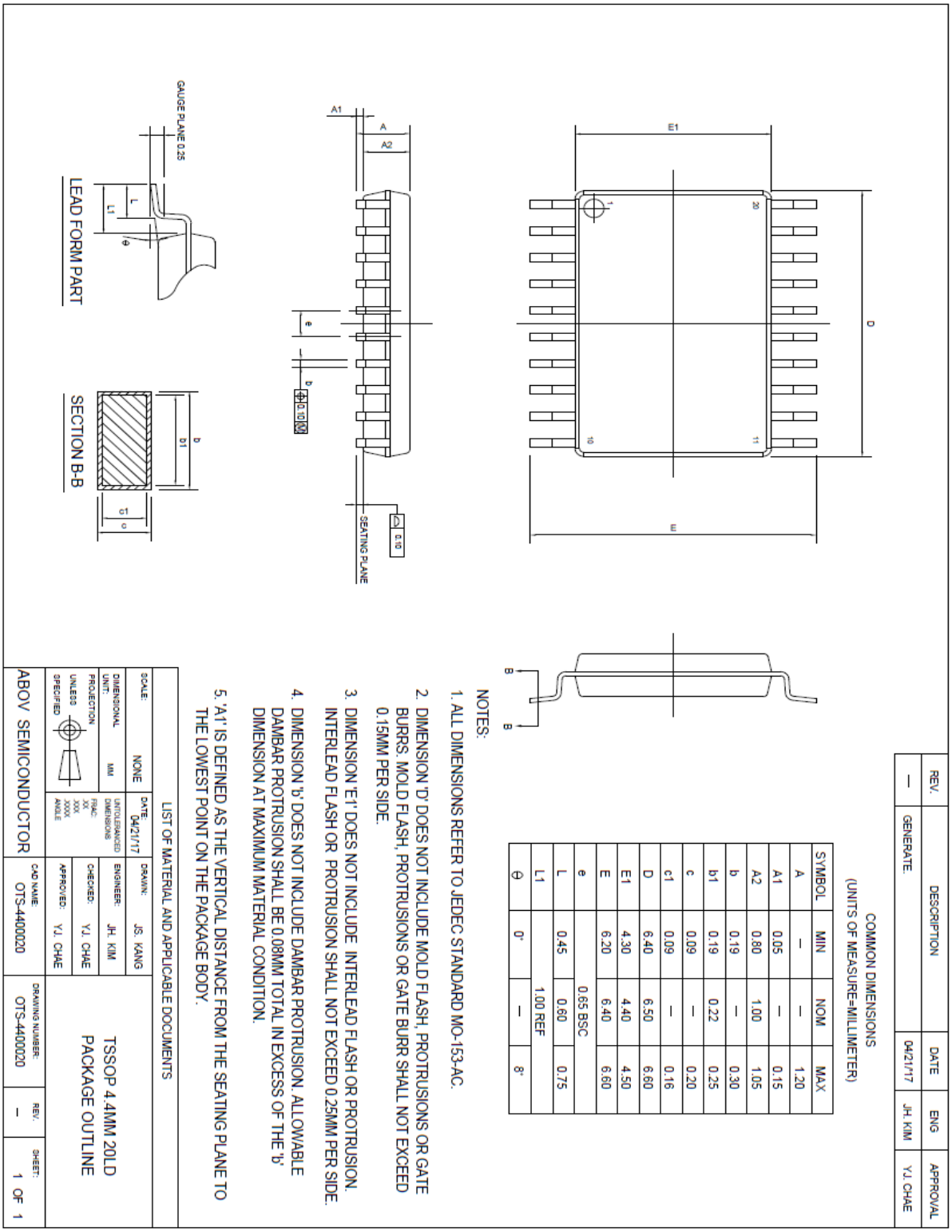


Figure 114. 20 TSSOP Package Outline

22.2 16 SOPN package information

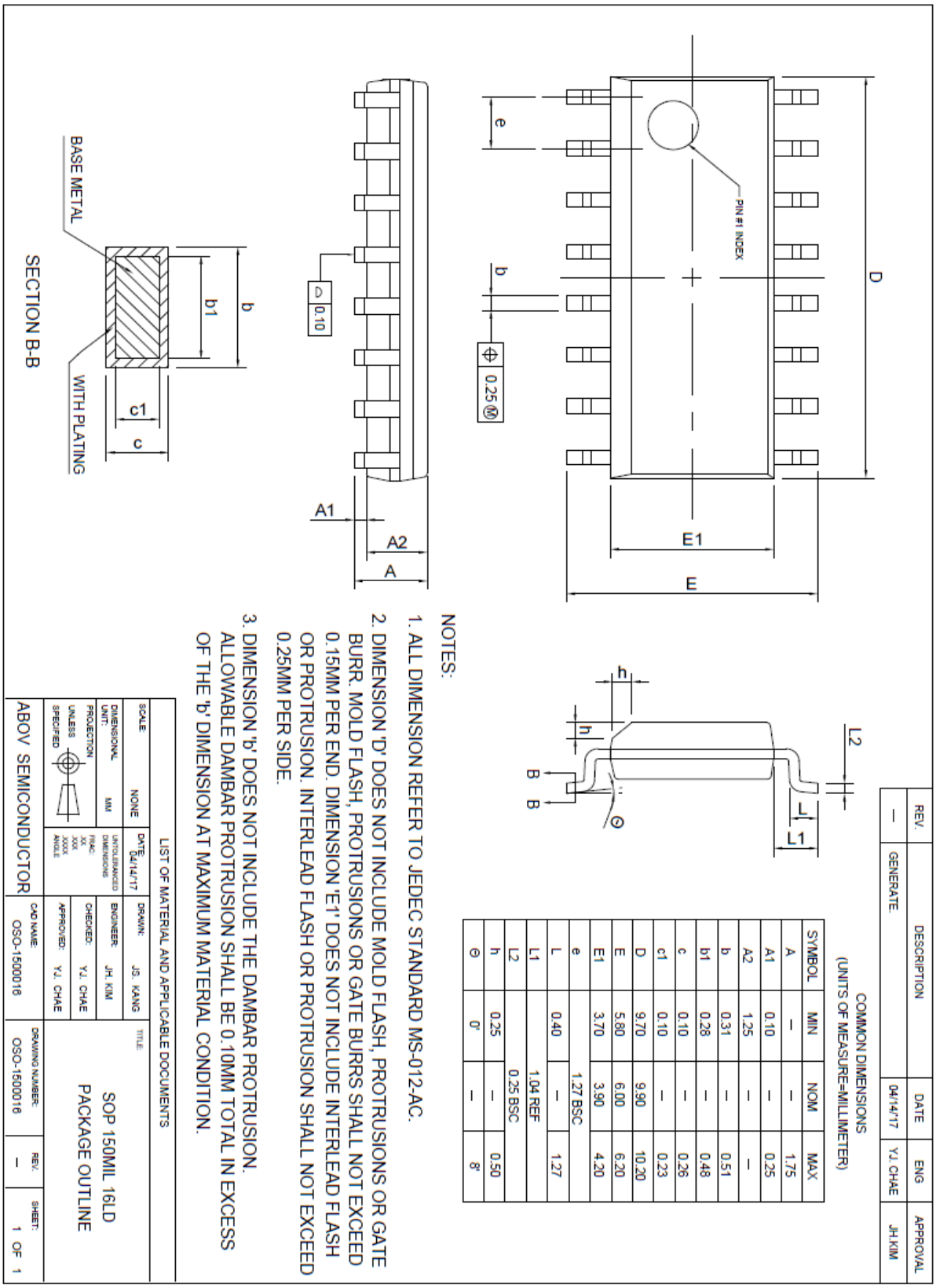


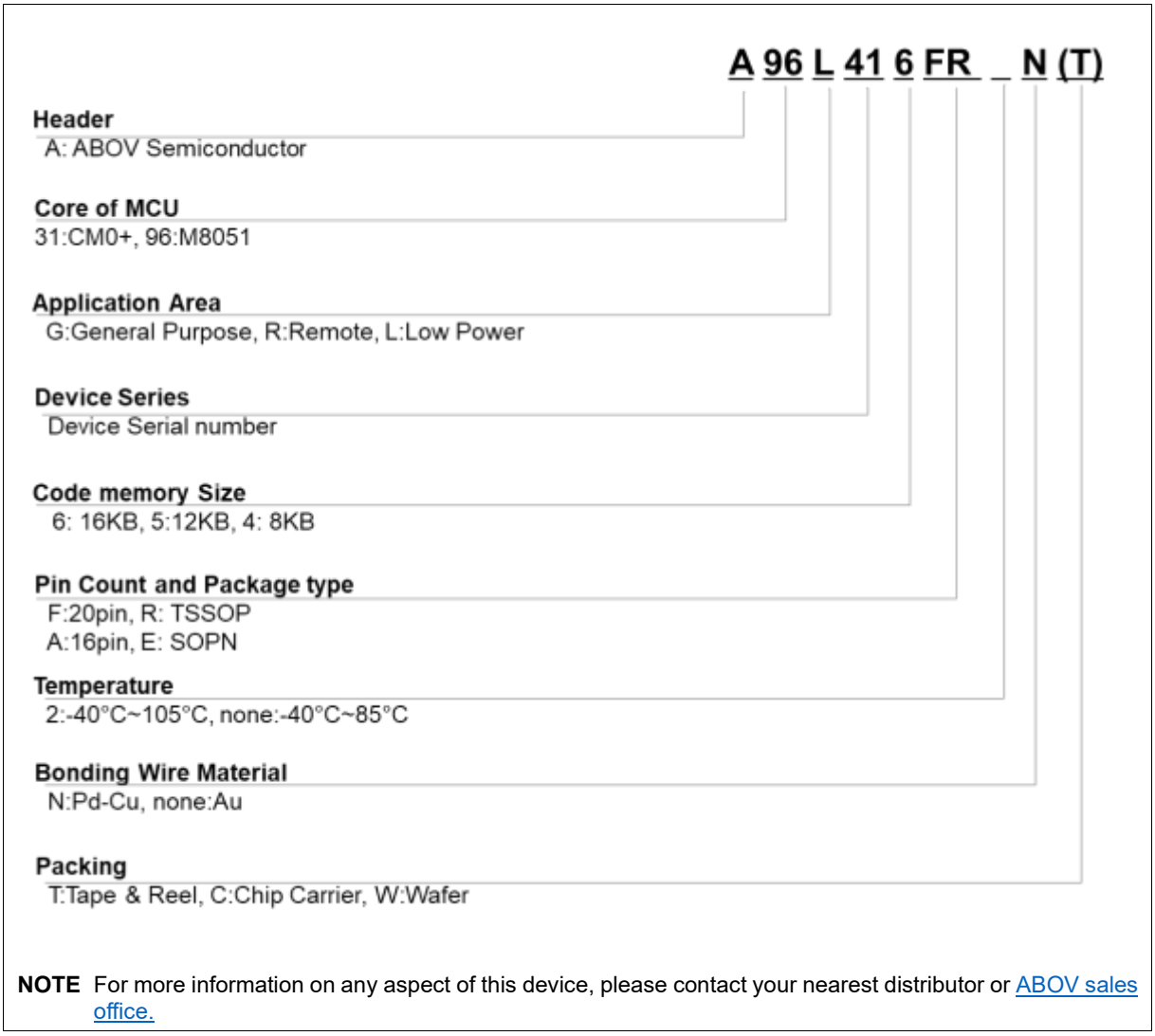
Figure 115. 16 SOPN Package Outline

23 Ordering information

Table 53. A96L414/A96L416 Device Ordering Information

Device name	Flash	IRAM/XRAM	Data Flash	ADC	I/O ports	Package type
A96L416FR	16 Kbyte	256/768 bytes	256 bytes	7 inputs	18	20 TSSOP
A96L414FR*	8 Kbyte	256/256 bytes	256 bytes	7 inputs	18	20 TSSOP
A96L416AE*	16 Kbyte	256/768 bytes	256 bytes	5 inputs	14	16 SOPN
A96L414AE*	8 Kbyte	256/256 bytes	256 bytes	5 inputs	14	16 SOPN

* For available options or further information on the device with an "*" mark, please contact the ABOV sales office.



NOTE For more information on any aspect of this device, please contact your nearest distributor or ABOV sales office.

Figure 116. A96L414/ A96L416 Device Numbering Nomenclature

24 Development tools

This chapter introduces wide range of development tools for A96L414/A96L416. ABOV offers software tools, debuggers, and programmers to help a user in generating right results to match target applications. ABOV supports entire development ecosystem of the customers.

24.1 Compiler

ABOV semiconductor does not provide any compiler for A96L414/A96L416. However, since A96L414/A96L416 has Mentor 8051 as its CPU core, you can use all kinds of third party's standard 8051 compiler such as Keil C Compiler. These compilers' output debug information can be integrated with our OCD emulator and debugger. Please visit our website www.abovsemi.com for more information regarding the OCD emulator and debugger.

24.2 OCD (On-Chip Debugger) emulator and debugger

The OCD emulator supports ABOV Semiconductor's 8051 series MCU emulation. The OCD uses two wires interfacing between PC and MCU, which is attached to user's system. The OCD can read or change the value of MCU's internal memory and I/O peripherals. In addition, the OCD controls MCU's internal debugging logic. This means OCD controls emulation, step run, monitoring and many more functions regarding debugging.

The OCD debugger program runs underneath MS operating system such as MS-Windows NT/ 2000/ XP/ Vista/ 7/ 8/ 8.1/ 10 (32-bit, 64-bit).

Programming information using the OCD is provided in section **24.5 Circuit design guide** later part in this chapter. More detailed information about the OCD, please visit our website www.abovsemi.com and download the debugger S/W and documents.

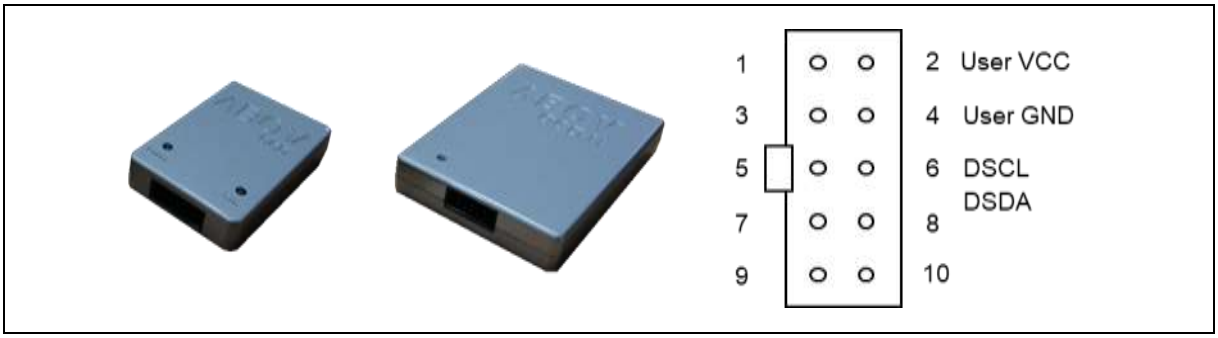


Figure 117. OCD and Pin Descriptions

Following is the OCD mode connections:

- DSCL (A96L414/A96L416 P12 port)
- DSDA (A96L414/A96L416 P13 port)

24.3 Programmer

24.3.1 E-PGM+

E-PGM+ is a single programmer, and allows a user to program on the device directly.

- Support ABOV / ADAM devices
- 2~5 times faster than S-PGM+
- Main controller : 32-bit MCU @ 72MHz
- Buffer memory : 1MB

Figure 118. E-PGM+ (Single Writer) and Pin Descriptions

24.3.2 OCD emulator

OCD emulator allows a user to write code on the device too, since OCD debugger supports ISP (In System Programming). It doesn't require additional H/W, except developer's target system.

24.3.3 Gang programmer

E-Gang4 and E-Gang6 allows a user to program on multiple devices at a time. They run not only in PC controlled mode but also in standalone mode without PC control. USB interface is available and it is easy to connect to the handler.

Table 54. Specification of E-Gang4 and E-Gang6

Gang programmer	E-Gang4	E-Gang6
Dimension (x, y, h)	33.5 x 22.5 x35mm	148.2 x 22.5 x35mm
Weight	2.0kg	2.8kg
Input voltage	DC Adaptor 15V/2A	DC Adaptor 15V/2A
Operating temperature	-10 ~ 40℃	-10 ~ 40℃
Storage temperature	-30 ~ 80℃	-30 ~ 80℃
Water proof	No	No

Figure 119. E-Gang4 and E-Gang6 (for Mass Production)

24.4 MTP programming

Program memory of A96L414/A96L416 is an MTP Type. This Flash is accessed through four pins such as DSCL, DSDA, VDD, and VSS in serial data format. Table 55 introduces each pin and corresponding I/O status.

Table 55. Pins for MTP Programming

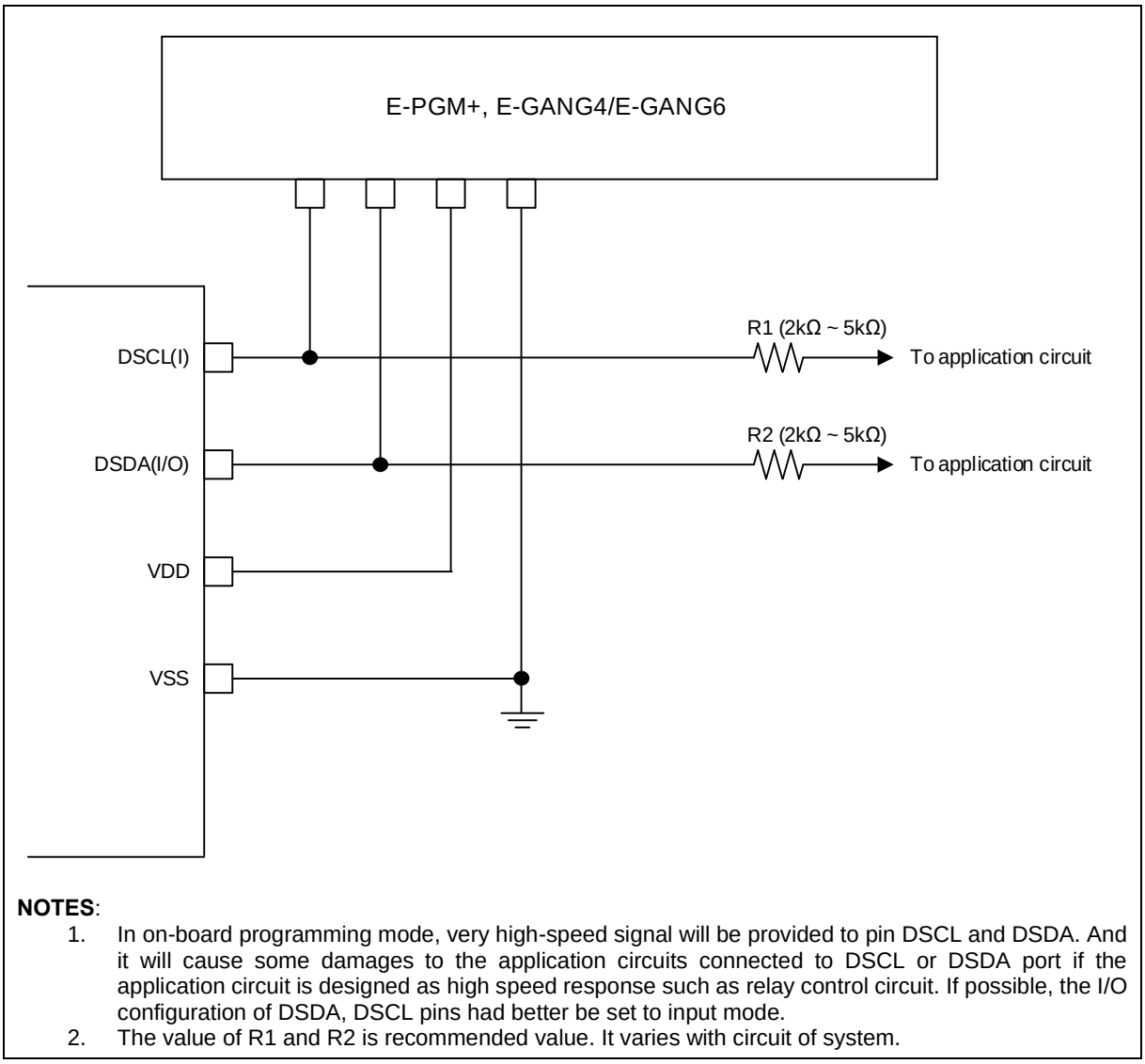
Pin name	Main chip pin name	During programming	
		I/O	Description
DSCL	P12	I	Serial clock pin. Input only pin.
DSDA	P13	I/O	Serial data pin. Output port when reading and input port when programming. Can be assigned as input/push-pull output port.
VDD, VSS	VDD, VSS	—	Logic power supply pin.

24.4.1 On-board programming

The A96L414/A96L416 needs only four signal lines including VDD and VSS pins for programming Flash with serial protocol. Therefore the on-board programming is possible if the programming signal lines are considered when the PCB of application board is designed.

24.5 Circuit design guide

When programming Flash memory, the programming tool needs 4 signal lines, DSCL, DSDA, VDD, and VSS. When you design a PCB circuit, you should consider the usage of these 4 signal lines for the on-board programming.



NOTES:

1. In on-board programming mode, very high-speed signal will be provided to pin DSCL and DSDA. And it will cause some damages to the application circuits connected to DSCL or DSDA port if the application circuit is designed as high speed response such as relay control circuit. If possible, the I/O configuration of DSDA, DSCL pins had better be set to input mode.
2. The value of R1 and R2 is recommended value. It varies with circuit of system.

Figure 120. PCB Design Guide for On-Board Programming

24.5.1 On-Chip Debug system

Detail descriptions for programming via the OCD interface can be found in the following figures. Table 56 introduces features of OCD and Figure 121 shows a block diagram of the OCD interface and the On-chip Debug system.

Table 56. Features of OCD

Two wire external interface	• 1 for serial clock input • 1 for bi-directional serial data bus
Debugger accesses	• All internal peripherals • Internal data RAM • Program Counter • Flash memory and EEPROM memory
Extensive On-Chip Debugging supports for Break Conditions	• Break instruction • Single step break • Program memory break points on single address • Programming of Flash, Data Flash, Fuses, and Lock bits through the two-wire interface • On-Chip Debugging supported by Dr. Choice®
Operating frequency	The maximum frequency of a target MCU.

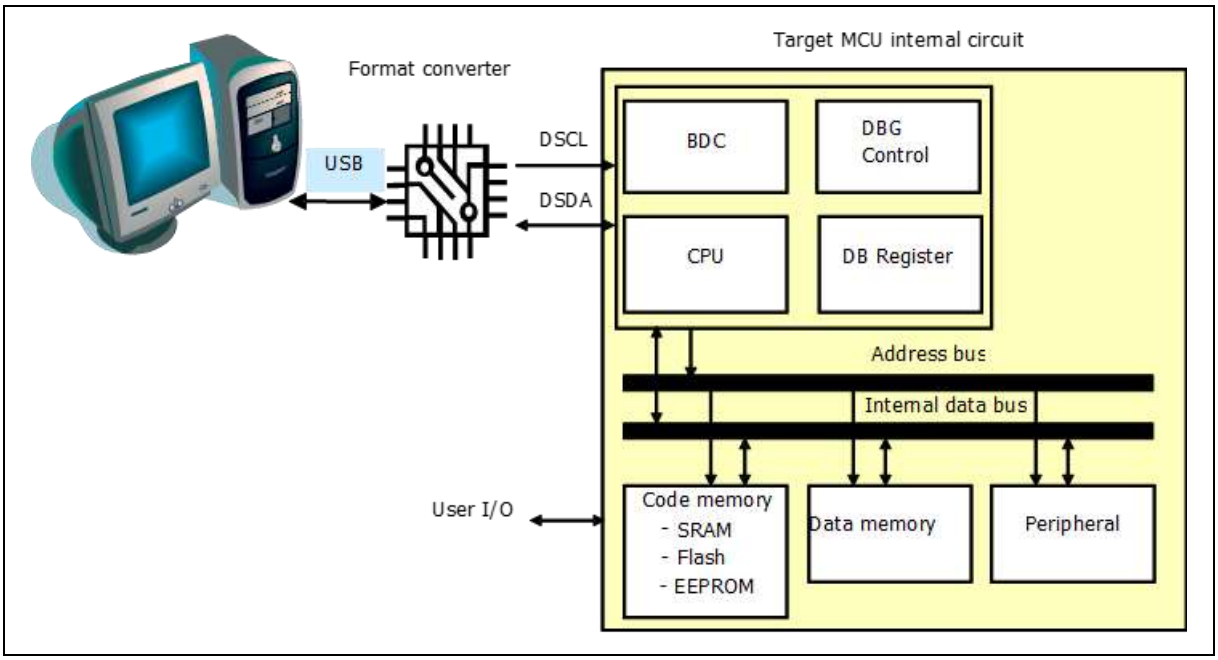


Figure 121. On-Chip Debugging System in Block Diagram

24.5.2 Two-pin external interface

Basic transmission packet

- 10-bit packet transmission using two-pin interface.
- 1-packet consists of 8-bit data, 1-bit parity and 1-bit acknowledge.
- Parity is even of '1' for 8-bit data in transmitter.
- Receiver generates acknowledge bit as '0' when transmission for 8-bit data and its parity has no error.
- When transmitter has no acknowledge (Acknowledge bit is '1' at tenth clock), error process is executed in transmitter.
- When acknowledge error is generated, host PC makes stop condition and transmits command which has error again.
- Background debugger command is composed of a bundle of packet.
- Start condition and stop condition notify the start and the stop of background debugger command respectively.

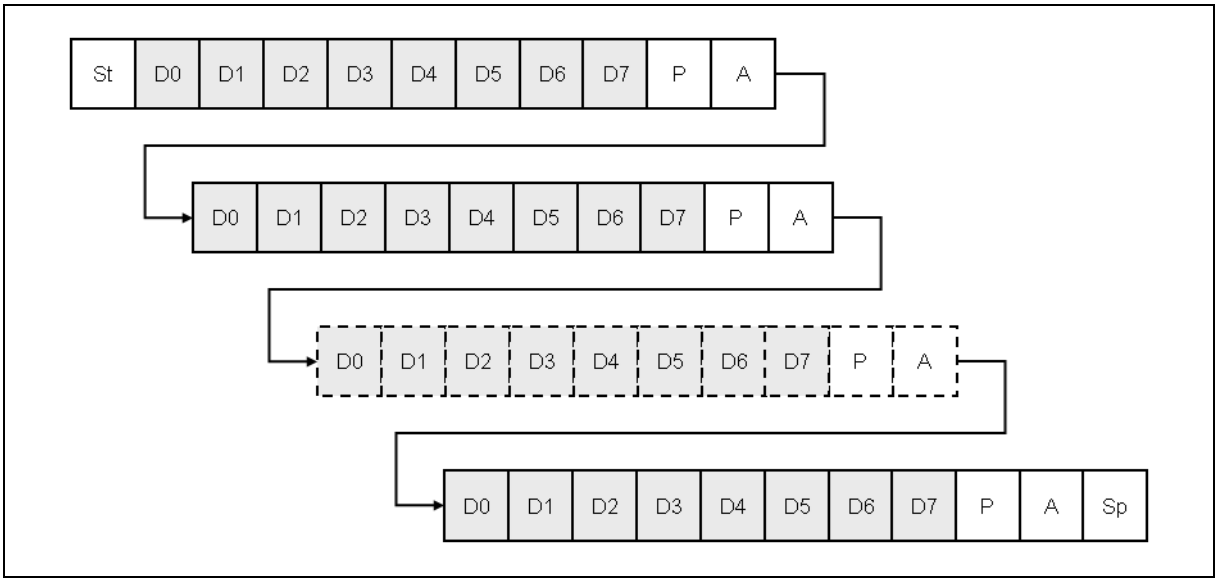


Figure 122. 10-bit Transmission Packet

Packet transmission timing

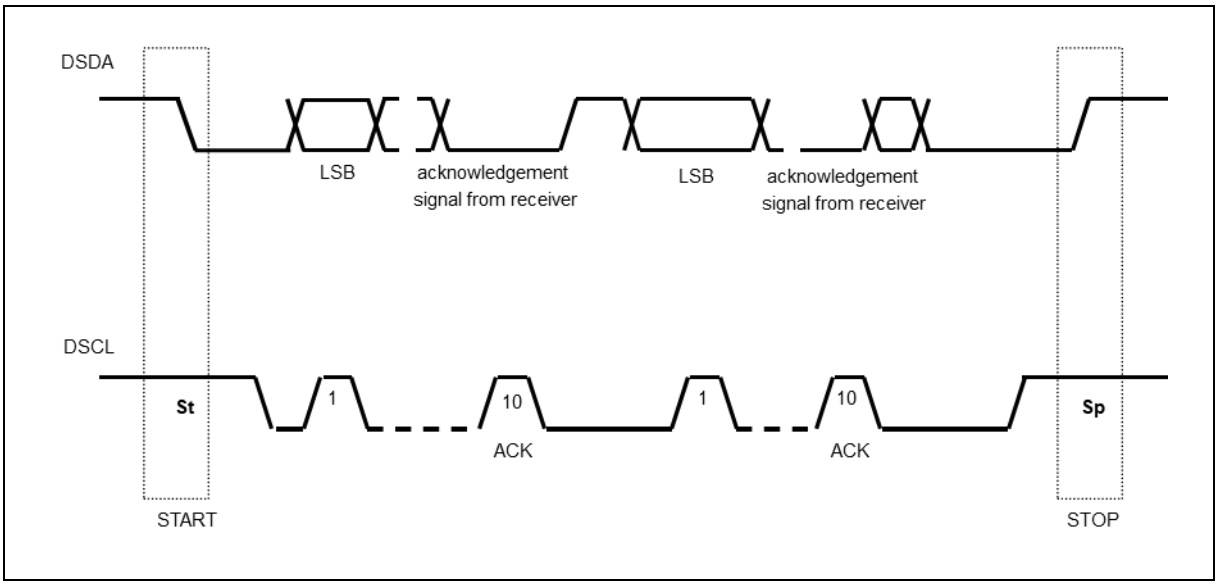


Figure 123. Data Transfer on Twin Bus

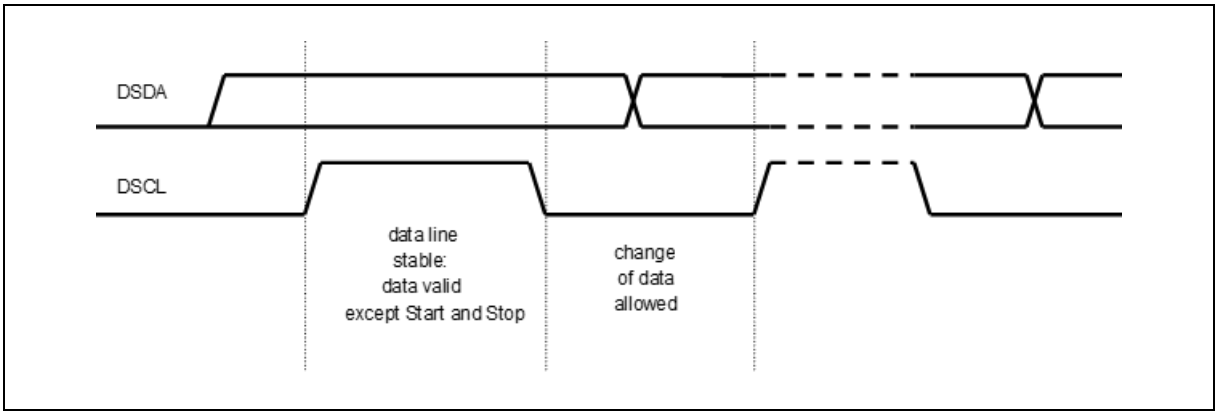


Figure 124. Bit Transfer on Serial Bus

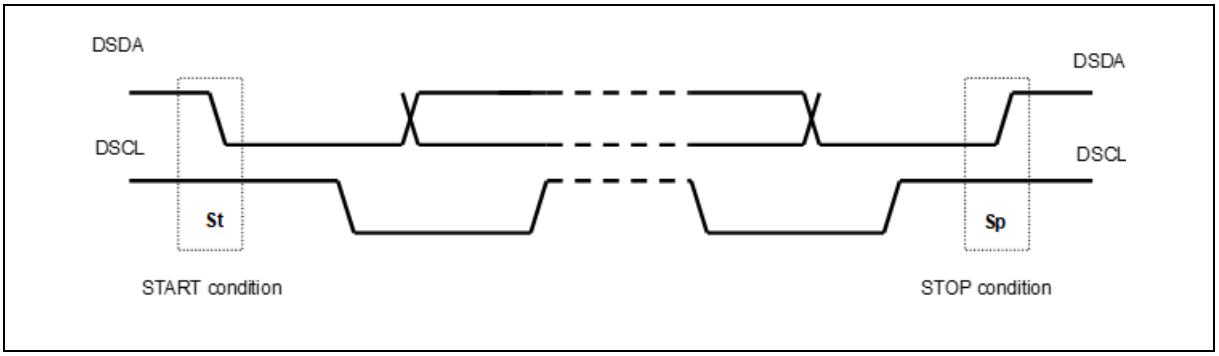


Figure 125. Start and Stop Condition

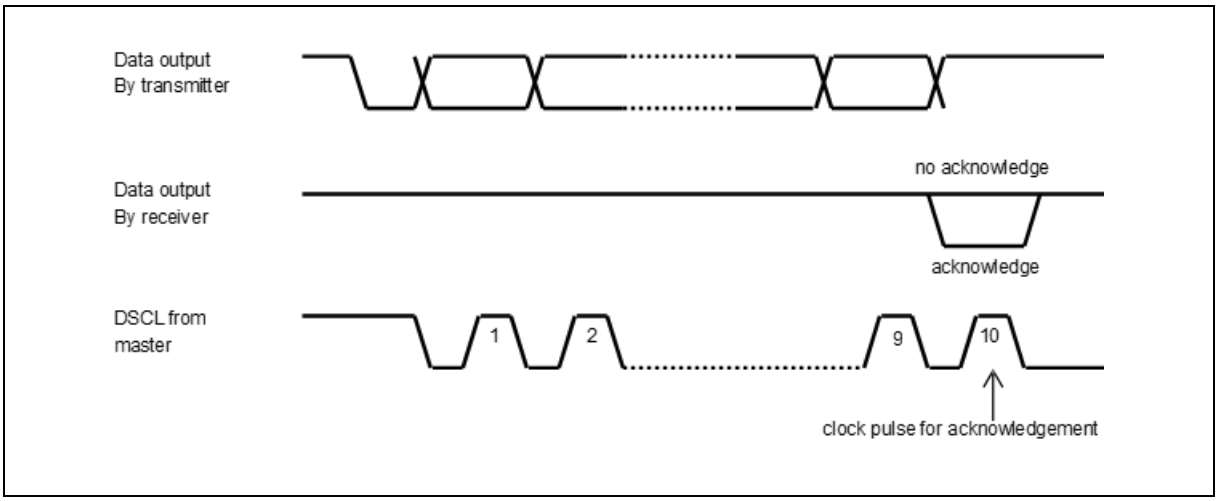


Figure 126. Acknowledge on Serial Bus

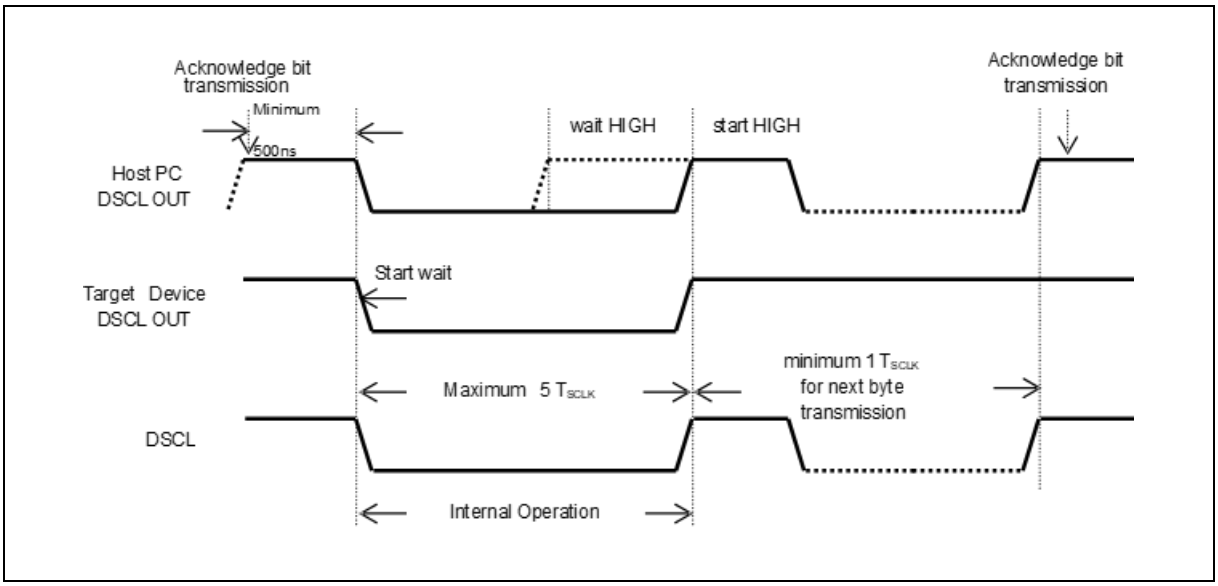


Figure 127. Clock Synchronization during Wait Procedure

24.5.3 Connection of transmission

Two-pin interface connection uses open-drain (wire-AND bidirectional I/O).

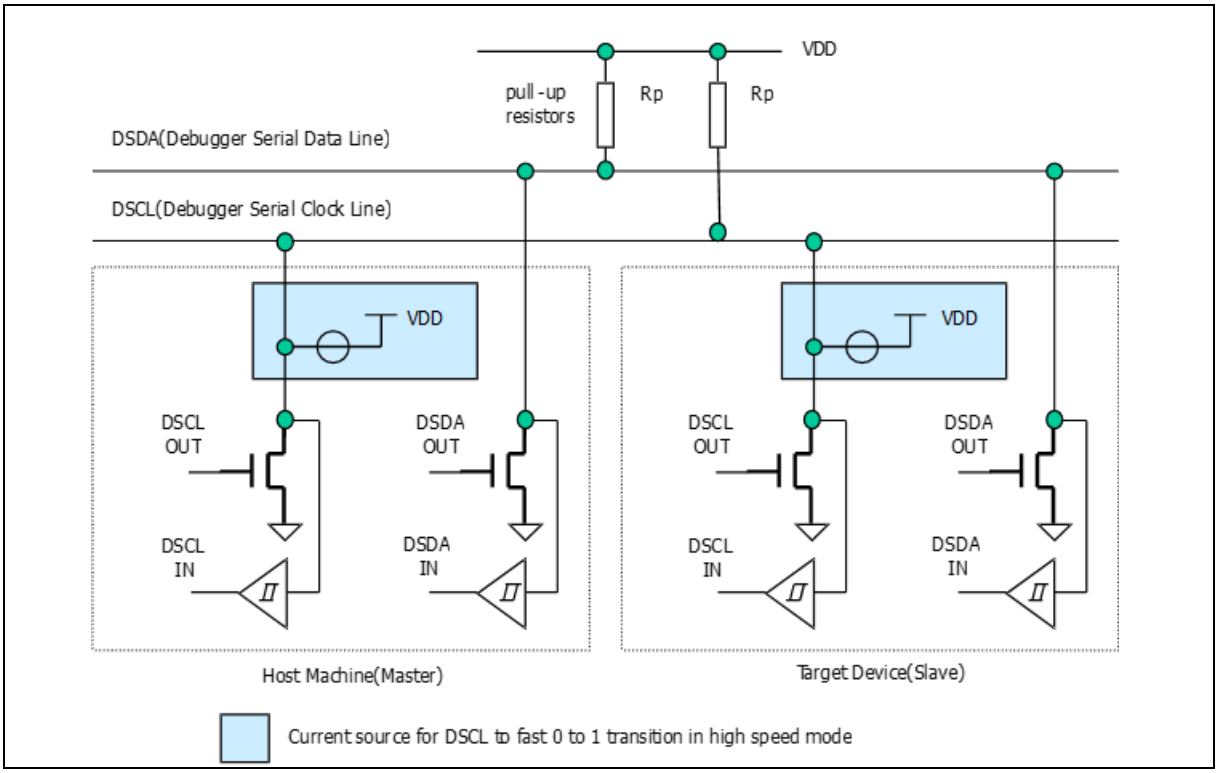


Figure 128. Connection of Transmission

Appendix

A. Configure option

Register description: configure option control

CONFIGURE OPTION 1: ROM Address 001FH

7	6	5	4	3	2	1	0
R_P	HL	–	VAPEN	–	–	–	RSTS

Initial value: 00H

R_P	Code Read Protection	
	0	Disable
	1	Enable
HL	Code Write Protection	
	0	Disable
	1	Enable
VAPEN	Vector Area (00H – FFH) Write Protection	
	0	Disable Protection (Erasable by instruction)
	1	Enable Protection (Not erasable by instruction)
RSTS	Select RESETB pin	
	0	Disable RESETB pin (P10)
	1	Enable RESETB pin

CONFIGURE OPTION 2 for 16-kBytes Flash memory: ROM Address 001EH

7	6	5	4	3	2	1	0
–	–	–	–	PAEN	PASS2	PASS1	PASS0

Initial value: 00H

PAEN	Enable Specific Area Write Protection			
	0	Disable (Erasable by instruction)		
	1	Enable (Not erasable by instruction)		
PASS [2:0]	Select Specific Area for Write Protection			
	NOTE: When PAEN = '1', it is applied.			
	PASS2	PASS1	PASS0	
	0	0	0	0.7Kbytes (Address 0100H – 03FFH)
	0	0	1	1.7Kbytes (Address 0100H – 07FFH)
	0	1	0	2.7Kbytes (Address 0100H – 0BFFH)
	0	1	1	3.8KBytes (Address 0100H – 0FFFH)
	1	0	0	13.7Kbytes (Address 0100H – 37FFH)
	1	0	1	14.7Kbytes (Address 0100H – 3BFFH)
	1	1	0	15.2Kbytes (Address 0100H – 3DFFH)
	1	1	1	15.5KBytes (Address 0100H – 3EFFH)

CONFIGURE OPTION 2 for 8-kBytes Flash memory: ROM Address 001EH

7	6	5	4	3	2	1	0
–	–	–	–	PAEN	PASS2	PASS1	PASS0

Initial value: 00H

PAEN Enable Specific Area Write Protection
- 0 Disable (Erasable by instruction)
- 1 Enable (Not erasable by instruction)

PASS [2:0] Select Specific Area for Write Protection

NOTE: When PAEN = '1', it is applied.

PASS2	PASS1	PASS0	
0	0	0	0.7Kbytes (Address 0100H – 03FFH)
0	0	1	1.7Kbytes (Address 0100H – 07FFH)
0	1	0	2.7Kbytes (Address 0100H – 0BFFH)
0	1	1	3.8KBytes (Address 0100H – 0FFFH)
1	0	0	5.7Kbytes (Address 0100H – 17FFH)
1	0	1	6.7Kbytes (Address 0100H – 1BFFH)
1	1	0	7.2Kbytes (Address 0100H – 1DFFH)
1	1	1	7.5KBytes (Address 0100H – 1EFFH)

B. Instruction table

- Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes' column in tables shown below.
- Each instruction takes either 1, 2 or 4 machine cycles to execute as listed in the following tables in this section.
- 1 machine cycle comprises 2 system clock cycles.

Table 57. Instruction Table: Arithmetic

Arithmetic				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ADD A,Rn	Add register to A	1	1	28-2F
ADD A,dir	Add direct byte to A	2	1	25
ADD A,@Ri	Add indirect memory to A	1	1	26-27
ADD A,#data	Add immediate to A	2	1	24
ADDC A,Rn	Add register to A with carry	1	1	38-3F
ADDC A,dir	Add direct byte to A with carry	2	1	35
ADDC A,@Ri	Add indirect memory to A with carry	1	1	36-37
ADDC A,#data	Add immediate to A with carry	2	1	34
SUBB A,Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	1	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	1	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	1	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18-1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16-17
INC DPTR	Increment data pointer	1	2	A3
MUL AB	Multiply A by B	1	4	A4
DIV AB	Divide A by B	1	4	84
DA A	Decimal Adjust A	1	1	D4

Table 58. Instruction Table: Logical

Logical				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65
XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through carry	1	1	13

Table 59. Instruction Table: Data Transfer

Data transfer				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOV A,Rn	Move register to A	1	1	E8-EF
MOV A,dir	Move direct byte to A	2	1	E5
MOV A,@Ri	Move indirect memory to A	1	1	E6-E7
MOV A,#data	Move immediate to A	2	1	74
MOV Rn,A	Move A to register	1	1	F8-FF
MOV Rn,dir	Move direct byte to register	2	2	A8-AF
MOV Rn,#data	Move immediate to register	2	1	78-7F
MOV dir,A	Move A to direct byte	2	1	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	2	85
MOV dir,@Ri	Move indirect memory to direct byte	2	2	86-87
MOV dir,#data	Move immediate to direct byte	3	2	75
MOV @Ri,A	Move A to indirect memory	1	1	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	1	76-77
MOV DPTR,#data	Move immediate to data pointer	3	2	90
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	2	93
MOVC A,@A+PC	Move code byte relative PC to A	1	2	83
MOVX A,@Ri	Move external data(A8) to A	1	2	E2-E3
MOVX A,@DPTR	Move external data(A16) to A	1	2	E0
MOVX @Ri,A	Move A to external data(A8)	1	2	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	2	F0
PUSH dir	Push direct byte onto stack	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A,Rn	Exchange A and register	1	1	C8-CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6-D7

Table 60. Instruction Table: Boolean

Boolean				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	1	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	1	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	1	B2
ANL C,bit	AND direct bit to carry	2	2	82
ANL C,/bit	AND direct bit inverse to carry	2	2	B0
ORL C,bit	OR direct bit to carry	2	2	72
ORL C,/bit	OR direct bit inverse to carry	2	2	A0
MOV C,bit	Move direct bit to carry	2	1	A2
MOV bit,C	Move carry to direct bit	2	2	92

Table 61. Instruction Table: Branching

Branching				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ACALL addr 11	Absolute jump to subroutine	2	2	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	2	22
RETI	Return from interrupt	1	2	32
AJMP addr 11	Absolute jump unconditional	2	2	01→E1
LJMP addr 16	Long jump unconditional	3	2	02
SJMP rel	Short jump (relative address)	2	2	80
JC rel	Jump on carry = 1	2	2	40
JNC rel	Jump on carry = 0	2	2	50
JB bit,rel	Jump on direct bit = 1	3	2	20
JNB bit,rel	Jump on direct bit = 0	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	2	60
JNZ rel	Jump on accumulator ≠0	2	2	70
CJNE A,dir,rel	Compare A,direct jne relative	3	2	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	2	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	2	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	2	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	2	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5

Table 62. Instruction Table: Miscellaneous

Miscellaneous				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
NOP	No operation	1	1	00

Table 63. Instruction Table: Additional Instructions

Additional instructions (selected through EO[7:4])				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOVC @(DPTR++),A	M8051W/M8051EW-specific instruction supporting software download into program memory	1	2	A5
TRAP	Software break command	1	1	A5

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, and the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

The CJNE instructions use the abbreviation #d for immediate data; other instructions use #data.

C. Flash protection for invalid erase/ write

Appendix C shows example code to prevent code or data from being changed by abnormal operations such as noise, unstable power, and malfunction.

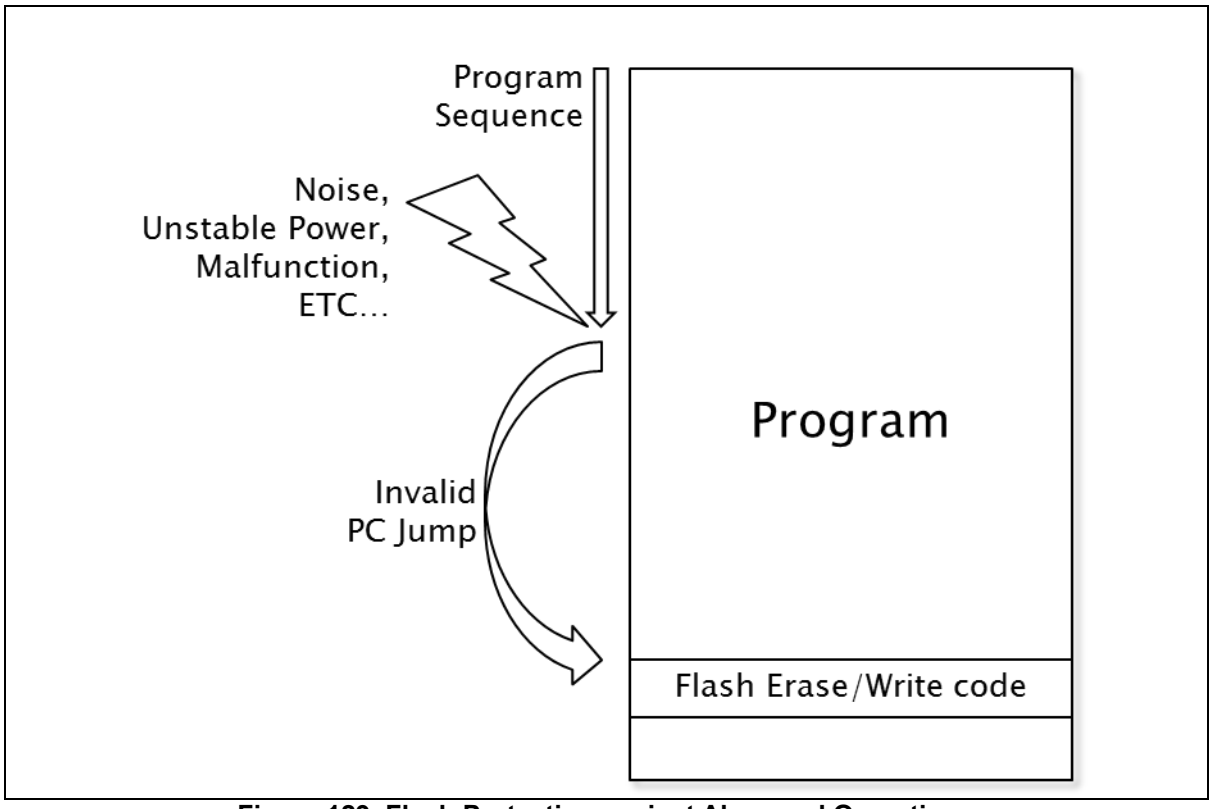


Figure 129. Flash Protection against Abnormal Operations

How to protect the Flash

- Divide into decision and execution to Erase/Write in Flash.
  - Check the program sequence from decision to execution in order of precedence about Erase/Write.
  - Setting the flags in program and check the flags in main loop at the end
  - When the Flash Erase/Write is executed, check the flags. If not matched, do not execute.
- Check the range of Flash Sector Address
  - If the Flash sector address is outside of specific area, do not execute.
- Use the Dummy Address
  - Set the Flash sector address to dummy address in usually run time.
  - Change the Flash sector address to real area range shortly before Erase/Write.
  - Even if invalid Erase/Write occurred, it will be Erase/Write in dummy address in Flash.
- Use the LVR/LVI
  - Unstable or low powers give an adverse effect on MCU. So use the LVR/LVI

Protection flow description

The Flash protection procedure is described in flowchart in Figure 130, and each step in this figure is described in the following lists:

1. Initialization
   - Set the LVR/LVI. Check the power by LVR/LVI and do not execute under unstable or low power.
   - Initialize User_ID1/2/3
   - Set Flash Sector Address High/Middle/Low to Dummy address. Dummy address is set to unused area range in Flash.
2. Decide to Write
   - When the Erase/Write are determined, set flag. Do not directly Erase/Write in Flash.
   - Make the user data.
3. Check and Set User_ID1/2/3
   - In the middle of source, insert code which can check and set the flags.
   - By setting the User_ID 1/2/3 sequentially and identify the flow of the program.
4. Set Flash Sector Address
   - Set address to real area range shortly before Erase/Write in Flash.
   - Set to Dummy address after Erase/Write. Even if invalid work occurred, it will be Erase/Write in Dummy address in Flash.
5. Check Flags
   - If every flag (User_ID1/2/3, LVI, Flash Address Min/Max) was set, than do Erase/Write.
   - If the Flash Sector Address is outside of Min/Max, do not execute
   - Address Min/Max is set to unused area.

6. Initialize Flags
   - Initialize User_ID1/2/3
   - Set Flash Sector Address to Dummy Address

- Sample Source
  - Refer to the ABOV website (www.abovsemi.com).
  - It is created based on the MC97F2664.
  - Each product should be modified according to the Page Buffer Size and Flash Size

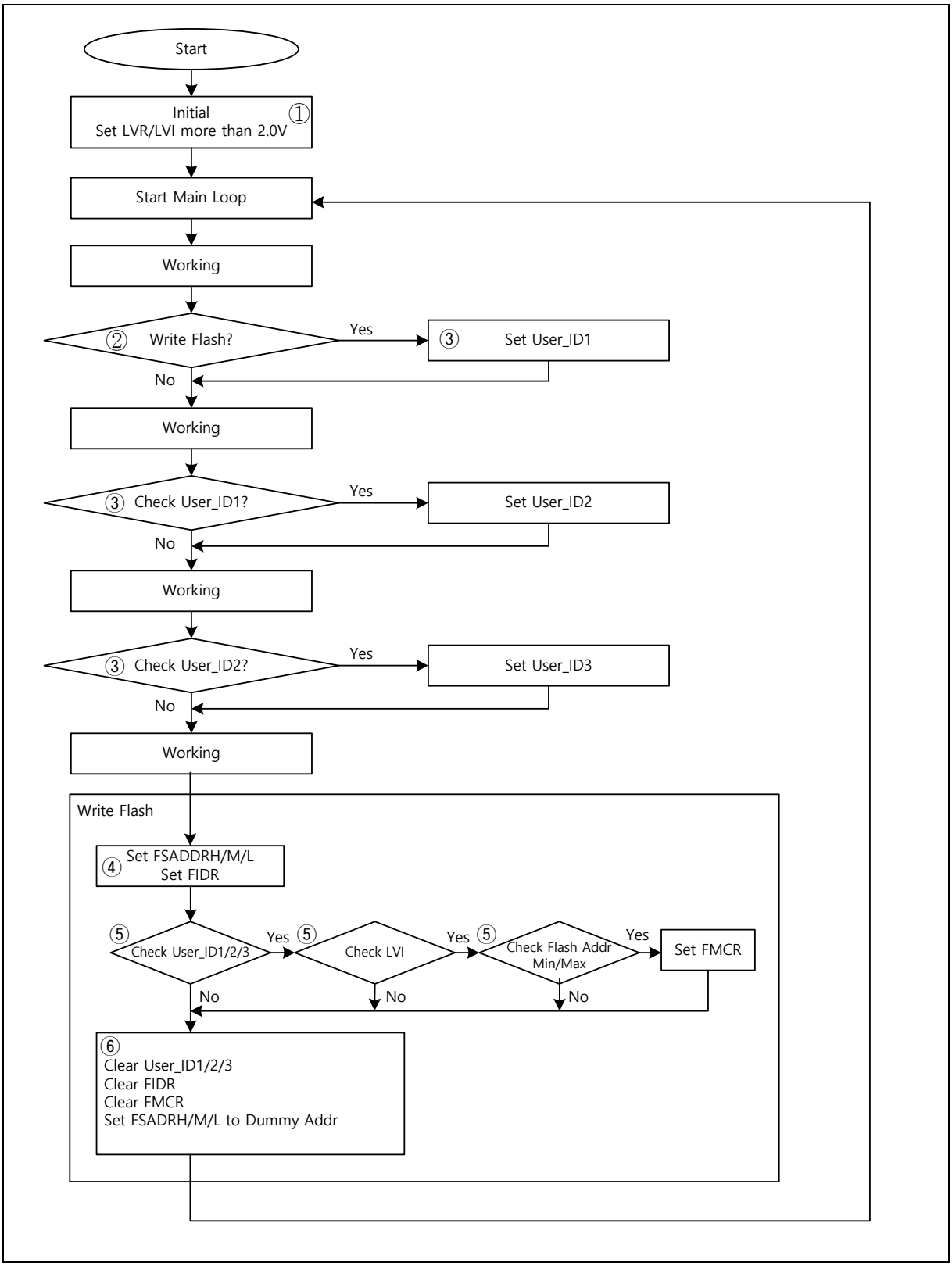


Figure 130. Flowchart of Flash Protection

Other protection by the configure options

- Protection by Configure option
  - — Set Flash protection by MCU Write Tool (OCD, PGM+, etc.)
    - ▪ Vector Area:

      00H~FFH
    - ▪ Specific Area (A96L414):

      0.7KBytes (Address 0100H – 03FFH)

      1.7KBytes (Address 0100H – 07FFH)

      2.7KBytes (Address 0100H – 0BFFH)

      3.8KBytes (Address 0100H – 0FFFH)

      5.7KBytes (Address 0100H – 17FFH)

      6.7KBytes (Address 0100H – 1BFFH)

      7.2KBytes (Address 0100H – 1DFFH)

      7.5KBytes (Address 0100H – 1EFFH)
    - ▪ Specific Area (A96L416):

      0.7KBytes (Address 0100H – 03FFH)

      1.7KBytes (Address 0100H – 07FFH)

      2.7KBytes (Address 0100H – 0BFFH)

      3.8KBytes (Address 0100H – 0FFFH)

      13.7KBytes (Address 0100H – 37FFH)

      14.7KBytes (Address 0100H – 3BFFH)

      15.2KBytes (Address 0100H – 3DFFH)

      15.5KBytes (Address 0100H – 3EFFH)
  - — The range of protection may be different each product.

D. Example circuit

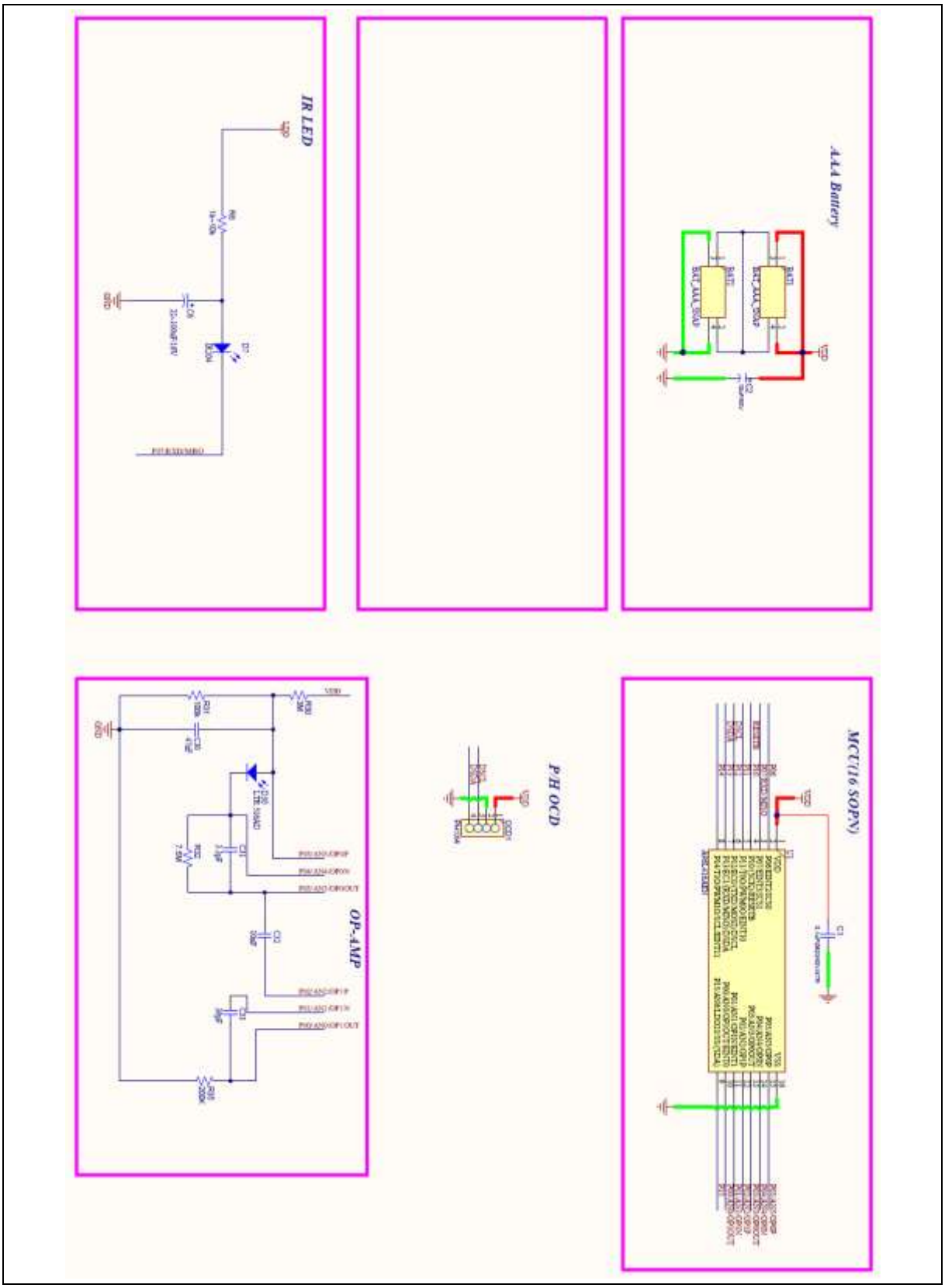


Figure 131. Example circuit using only IR LED(16 SOPN)

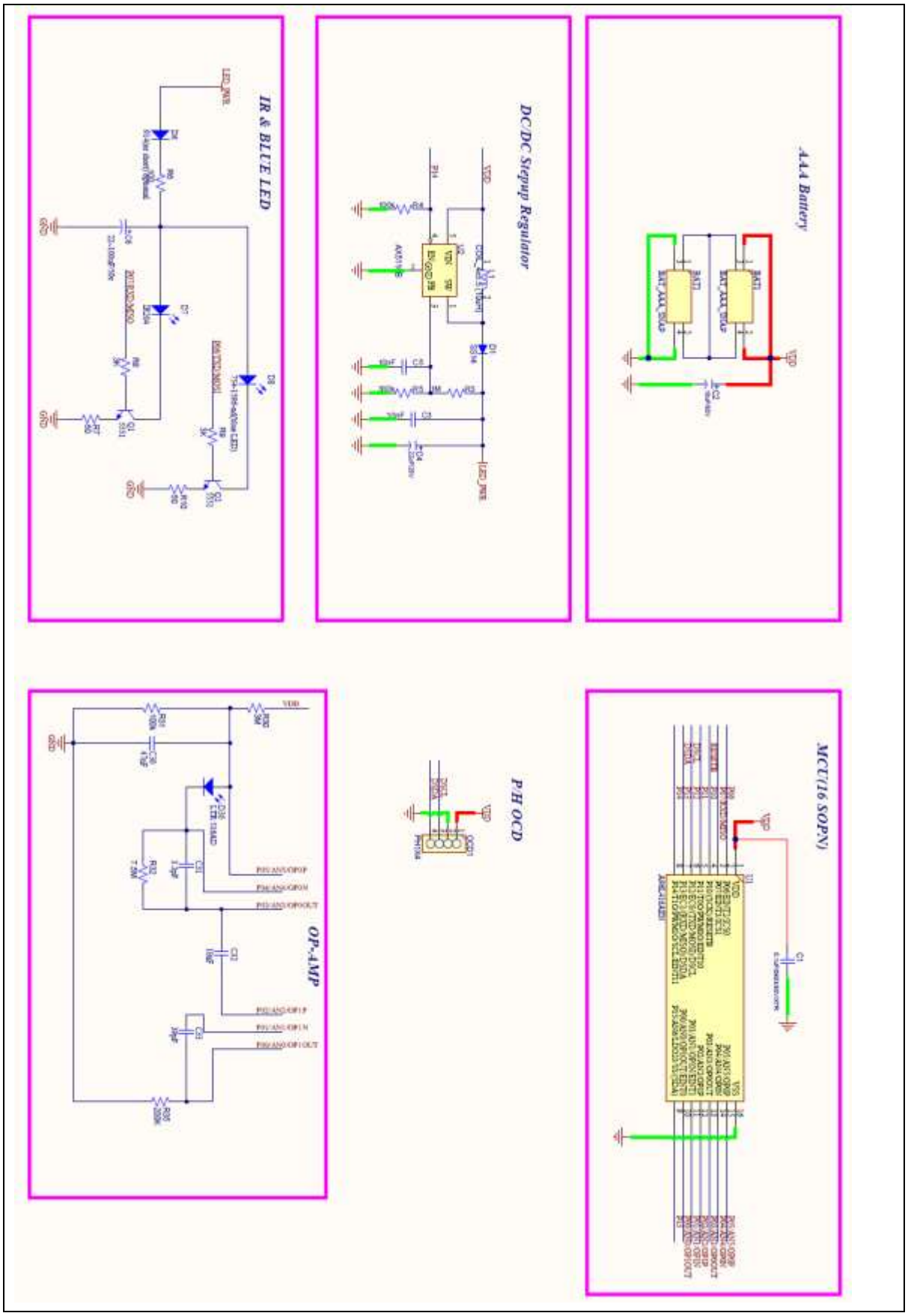


Figure 132. Example circuit using IR LED and Blue LED(16 SOPN)

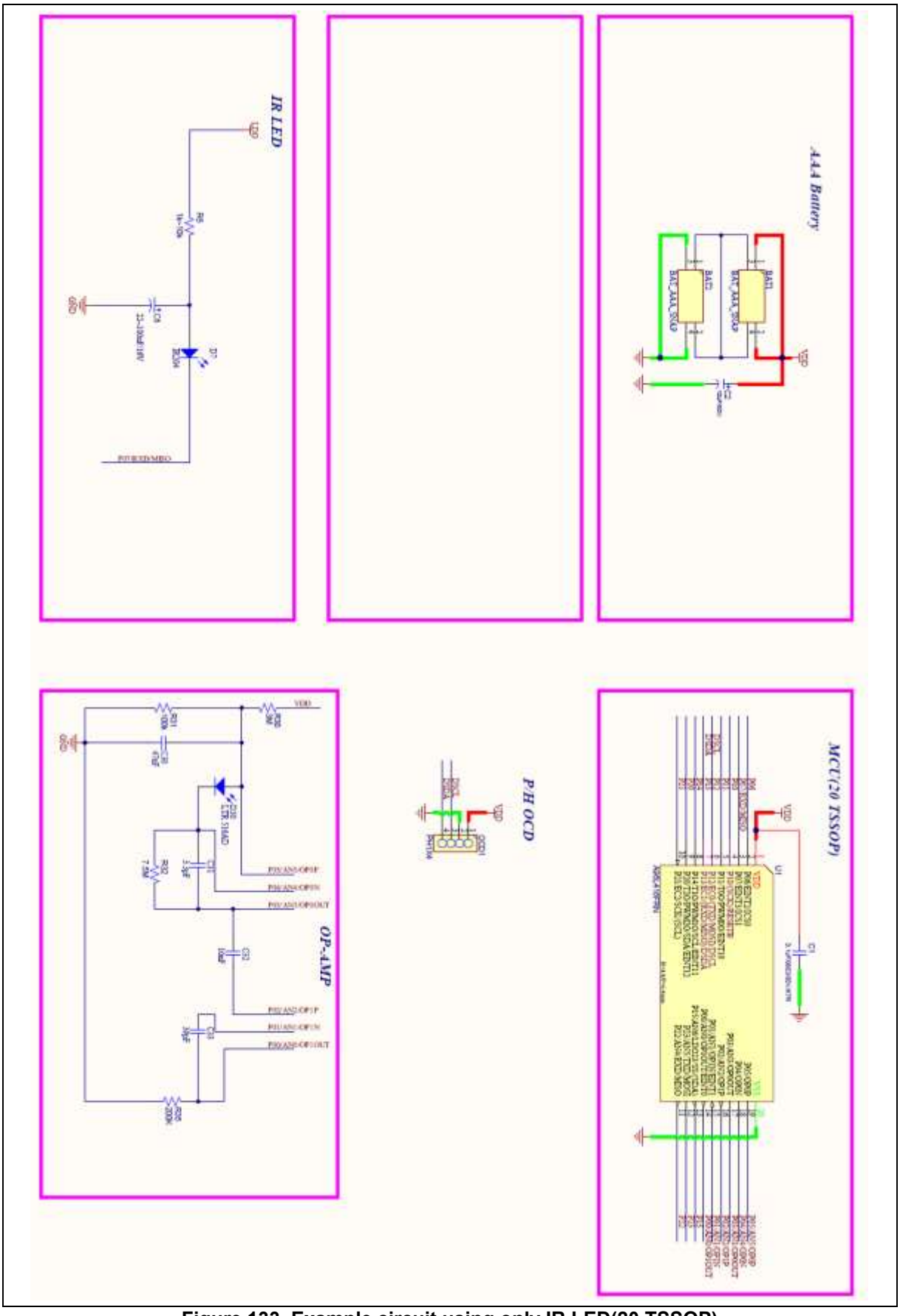


Figure 133. Example circuit using only IR LED(20 TSSOP)

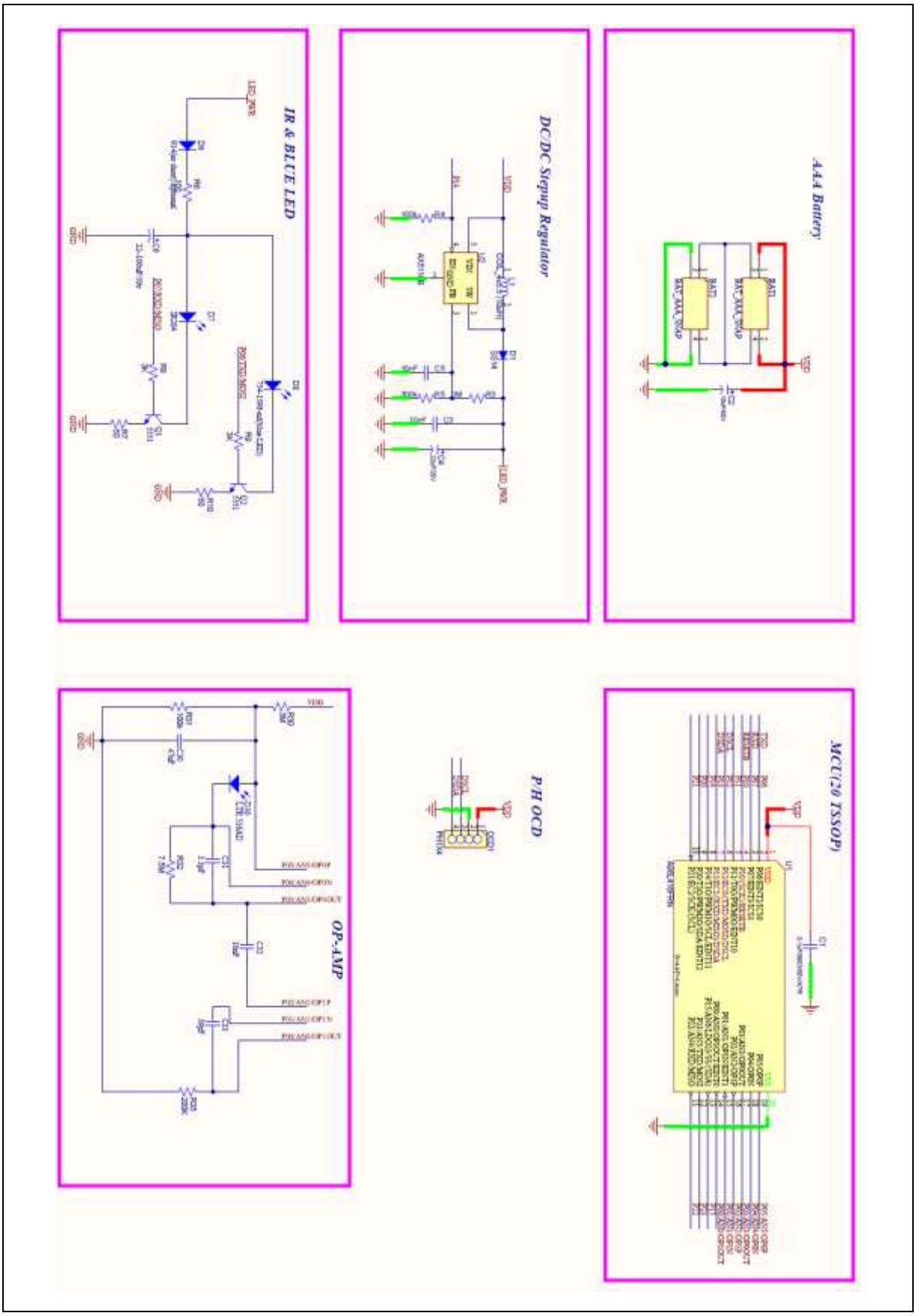


Figure 134. Example circuit using IR LED and Blue LED (20 TSSOP)

Revision history

Date	Version	Description
July.3, 2020	1.00	First creation
Sep.15. 2020	1.01	Updated example schematics
Oct.19. 2021	1.10	Changed the current and gain error of OP-Amp
May.18. 2022	1.20	Changed Recommend Circuits of OP-Amp
Dec. 27. 2022	1.30	Modify a font

Korea

Regional Office, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
www.abovsemi.com

HQ, Ochang
R&D, QA, and Test Center
93, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,
28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
www.abovsemi.com

Domestic Sales Manager
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

Global Sales Manager
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

China Sales Manager
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr