



# Preliminary Databook

## **TG471**

Preliminary

**Ver 0.97**

May 30, 2008

*History*

Ver 0.1 <b>Feb. 15, 2007</b>	<b>1st version released</b>
Ver 0.2 <b>April 16, 2007</b>	<b>Add UART 2Ch and SPI 1Ch</b>
Ver 0.3 <b>May 08, 2007</b>	<b>Add DMA Request Selection Register</b>
Ver 0.4 <b>May 16, 2007</b>	<b>ETC</b>
Ver 0.5 <b>May 22, 2007</b>	<b>Changed I/O Driving</b>
Ver 0.7 <b>November 22, 2007</b>	<b>Changed Pinout</b>
Ver 0.8 <b>November 26, 2007</b>	<b>Changed I2S with ADPCM</b>
Ver 0.9 <b>December 14, 2007</b>	<b>Changed Electrical Characteristics Changed Power Management</b>
Ver 0.95 <b>January 24, 2008</b>	<b>Add ISP...</b>
Ver 0.96 <b>April 16, 2008</b>	<b>Changed Multiplex Pin Name</b>
Ver 0.97 <b>May 30, 2008</b>	<b>Changed PWML bit (TMCON Register)</b>

## CONTENTS

1	Descriptions and features .....	8
1.1	General Description .....	8
1.2	Feature .....	8
2	Block Diagram & PIN Descriptions.....	10
2.1	Block Diagram.....	10
2.2	Pin Configurations.....	11
2.2.1	Pinout .....	11
2.2.2	Pin Definitions .....	12
2.2.3	Pin Descriptions .....	14
3	SE3208 EISC Core.....	16
3.1	Introduction .....	16
3.2	SE3208 Features.....	16
3.3	SE3208 Block Diagram .....	17
3.4	SE3208 Registers.....	18
3.4.1	General Purpose 32 Bit Registers (R0 – R7).....	18
3.4.2	Program Counter.....	19
3.4.3	Status Register .....	19
3.4.4	Stack Pointer.....	20
3.4.5	Extension Register .....	20
3.5	SE3208 Instruction Set .....	21
4	Memory Map.....	23
4.1	Flash Memory .....	25
4.1.1	TG471A (64KBytes) .....	25
4.1.2	TG471B (128KBytes).....	26
4.1.3	TG471C (512KBytes).....	27
4.1.4	Flash Access .....	29
4.1.5	Flash Control Registers .....	30
4.2	External Static Memory .....	32
5	System Power and Reset .....	33
5.1	Voltage Regulator .....	33
5.2	Reset .....	34
5.2.1	Power On Reset .....	34
5.2.2	External Reset.....	34
5.2.3	Watchdog Reset .....	35
5.2.4	Reset Control Registers.....	36
6	Clock Control and Power Management.....	37
6.1	Clock Control .....	37
6.2	Power Management .....	38
6.3	Power and Clock Control Registers .....	39
7	Alternate Port and I/O Ports .....	41
7.1	Alternate Port Function .....	41
7.1.1	PIN MUX Control Registers .....	42
7.2	I/O Ports.....	46
7.2.1	I/O Ports Registers .....	47
8	Interrupts .....	50
8.1	Interrupt Vector and Priority.....	51

8.2	External Interrupt (INT 0 ~ 7) .....	52
8.3	Internal Interrupt Mode.....	53
8.4	Interrupt Pending and Interrupt Pending Clear .....	53
8.5	Interrupt Enable .....	53
8.6	Interrupt Mask Set/Clear Register .....	53
8.7	Interrupt Control Registers.....	54
9	Timers .....	62
9.1	10bit Prescaler with clock source selection .....	63
9.2	Timer/Counter.....	64
9.3	Pulse Width Modulation (PWM) .....	66
9.4	Capture .....	68
9.5	Output Compare Mode .....	70
9.6	Timer Registers .....	71
10	SPI(Serial Peripheral Interface).....	74
10.1	SPI Registers Summary .....	75
10.2	SPI Pins .....	76
10.3	SPI Operating Modes .....	76
10.4	SCK Phase and Polarity Control.....	78
10.5	Data Transfer Timing .....	78
10.6	SPI Serial Clock Baud Rate.....	80
10.7	Open-Drain Output for Wired-OR .....	80
10.8	Transfer Size and Direction.....	80
10.9	Write Collision.....	80
10.10	MODE Fault .....	80
10.11	Interrupt .....	81
10.12	SPI Control Registers .....	82
11	TWI (Two Wired Interface).....	85
11.1	DATA TRANSFER FORMAT .....	86
11.2	START AND STOP CONDITION .....	86
11.3	ACK SIGNAL TRANSMISSION .....	87
11.4	READ-WRITE OPERATION.....	87
11.5	BUS ARBITRATION PROCEDURES .....	88
11.6	ABORT CONDITIONS (arbitration 이 발생하지 않은 경우).....	89
11.7	Operational Flow Diagrams .....	89
11.7.1	TWI initialization .....	89
11.7.2	Master Transmit /Receive .....	90
11.7.3	Slave Mode.....	92
11.8	TWI Control Registers .....	93
12	UART .....	96
12.1	Serial Data Format .....	98
12.2	UART Baud Rate .....	100
12.3	UART Control Registers .....	101
13	Direct Memory Access (DMA) .....	106
13.1	Function Description.....	107
13.1.1	Transfer mode.....	107
13.1.2	Address mode .....	107
13.1.3	Reloading mode .....	107
13.1.4	Data Transfer Width 와 Transfer Counter .....	107
13.1.5	Software Enable DMA and Hard-wire Enable DMA .....	108
13.2	DMA Control Registers.....	109

14	I2S with ADPCM.....	112
14.1	I2S SCLK, LRCK, MCLK (CODEC clock) Frequency .....	113
14.2	I2S Control Registers.....	114
15	USB Device.....	118
15.1	USB Register Summary .....	119
15.2	USB Function Address Register .....	120
15.3	USB Power Management Register .....	120
15.4	USB Interrupt Registers .....	120
15.5	USB Interrupt Enable Registers .....	120
15.6	Frame Number Registers.....	120
15.7	Index Register.....	120
15.8	MAXP Register.....	120
15.9	EPO Control Register .....	121
15.10	IN Control Registers .....	121
15.11	Out Control Registers .....	121
15.12	Out Write Count Registers.....	121
15.13	Endpoint FIFO Access Registers .....	121
15.14	USB Control Registers.....	122
16	ADC.....	134
16.1	ADC Control Registers.....	135
17	DAC.....	136
17.1	DAC Control Registers.....	136
18	In System Programmer (ISP) .....	137
18.1	ISP Command Set.....	138
18.2	Device Identification and Status Commands .....	139
18.3	Data Read Command.....	140
18.4	Programming Commands .....	141
18.5	Erase Commands.....	142
19	Electrical Data .....	143
19.1	DC ELECTRICAL CHARACTERISTICS .....	143
19.2	Voltage Regulator Electrical Characteristics .....	144
19.3	POR ELECTRICAL CHARACTERISTICS .....	144
19.4	RC OSC ELECTRICAL CHARACTERISTICS .....	144
19.5	PLL ELECTRICAL CHARACTERISTICS .....	144
19.6	ADC ELECTRICAL CHARACTERISTICS.....	145
19.7	DAC ELECTRICAL CHARACTERISTICS.....	145
20	Package Dimension .....	146

## FIGURES

FIGURE 2-1 TG471 BLOCK DIAGRAM.....	10
FIGURE 2-2 TG471 PINOUT DIAGRAM.....	11
FIGURE 3-1 SE3208 BLOCK DIAGRAM .....	17
FIGURE 3-2 SE3208 MCU REGISTERS.....	18
FIGURE 3-3 E-FLAG AND INSTRUCTION .....	20
FIGURE 4-1 FLASH ACCESS CYCLES .....	29
FIGURE 4-2 EXTERNAL STATIC MEMORY INTERFACE TIMING.....	32
FIGURE 5-1 POWER ON RESET TIMING DIAGRAM.....	34
FIGURE 6-1 TG471 CLOCK DISTRIBUTION.....	37
FIGURE 8-1 EXTERNAL INTERRUPT MODE .....	52
FIGURE 9-1 PRE-SCALER BLOCK DIAGRAM .....	63
FIGURE 9-2 TIMER OPERATION .....	64
FIGURE 9-3 PWM OPERATION.....	67
FIGURE 9-4 CAPTURE MODE OPERATION.....	68
FIGURE 9-5 TIMING DIAGRAM OF OUTPUT COMPARE OPERATION.....	70
FIGURE 10-1 SPI BLOCK DIAGRAM .....	75
FIGURE 10-2 SCK PHASE AND POLARITY .....	78
FIGURE 10-3 TRANSFER TIMING WHEN CPHA = '0' .....	79
FIGURE 10-4 TRANSFER TIMING WHEN CPHA = '1' .....	79
FIGURE 10-5 1-BYTE TRANSFER VS. STATUS AND INTERRUPT.....	81
FIGURE 10-6 N-BYTES TRANSFER VS. STATUS AND INTERRUPT.....	81
FIGURE 11-1 TWI BLOCK DIAGRAM .....	85
FIGURE 11-2 TWI-BUS INTERFACE DATA FORMAT .....	86
FIGURE 11-3 DATA TRANSFER ON THE TWI-BUS .....	86
FIGURE 11-4 ACKNOWLEDGEMENT OF TWI .....	87
FIGURE 11-5 BUS ARBITRATION 1 OF TWI.....	88
FIGURE 11-6 BUS ARBITRATION 2 .....	88
FIGURE 11-7 TWI INITIALIZATION FLOW CHAR .....	89
FIGURE 11-8 MASTER TRANSMIT FLOW CHAR .....	90
FIGURE 11-9 MASTER RECEIVE FLOW CHAR .....	91
FIGURE 11-10 SLAVE MODE FLOW CHAR .....	92
FIGURE 12-1 UART BLOCK DIAGRAM .....	96
FIGURE 12-2 UART LCR REGISTER SETTING AND SERIAL DATA FORMAT .....	99
FIGURE 13-1 DMA BLOCK DIAGRAM.....	106
FIGURE 14-1 I2S AND ADC / DAC BLOCK DIAGRAM.....	112
FIGURE 16-1 ADC CONTROL TIMING .....	134
FIGURE 18-1 SPI MODES SUPPORTED.....	137
FIGURE 18-2 ISP FLOW CHART.....	137
FIGURE 18-3 READ ID/STATUS TIMING AND FLOW CHART .....	139
FIGURE 18-4 ISP STATUS SETTING TIMING AND FLOW CHART .....	140
FIGURE 18-5 ISP DATA READ TIMING AND FLOW CHART .....	140
FIGURE 18-6 BYTE PROGRAMMING TIMING AND FLOW CHART.....	141
FIGURE 18-7 AAI PROGRAMMING TIMING AND FLOW CHART .....	141
FIGURE 18-8 ERASE COMMANDS TIMING AND FLOW CHART.....	142
FIGURE 20-1 PACKAGE DIMENSION .....	146

## TABLES

TABLE 2-1 TG471 PIN DEFINITIONS .....	12
TABLE 3-1 SE3208 INSTRUCTION SET SUMMERY .....	21
TABLE 4-1 MEMORY MAP.....	23
TABLE 4-2 MEMORY MAPPED I/O REGISTERS .....	24
TABLE 4-3 READ REGION IN ON-CHIP FLASH MEMORY (64KBYTES).....	25
TABLE 4-4 WRITE REGION IN ON-CHIP FLASH MEMORY (64KBYTES) .....	25
TABLE 4-5 READ REGION IN ON-CHIP FLASH MEMORY (128KBYTES).....	26
TABLE 4-6 WRITE REGION IN ON-CHIP FLASH MEMORY (128KBYTES) .....	26
TABLE 4-7 READ REGION IN ON-CHIP FLASH MEMORY (512KBYTES).....	27
TABLE 4-8 PROGRAMMING REGION IN ON-CHIP FLASH MEMORY (512KBYTES) .....	28
TABLE 4-9 EXTERNAL MEMORY INTERFACE TIMING PARAMETERS .....	32
TABLE 5-1 POWER ON RESET CHARACTERISTICS .....	34
TABLE 5-2 EXTERNAL RESET CHARACTERISTICS .....	34
TABLE 6-1 SYSTEM CLOCK.....	38
TABLE 6-2 ACTIVE CLOCK DOMAINS AND WAKE UP SOURCES .....	38
TABLE 7-1 INTERNAL PULL-UP CHARACTERISTICS .....	46
TABLE 8-1 INTERRUPT VECTOR & PRIORITY .....	51
TABLE 10-1 SPI PIN FUNCTIONS.....	76
TABLE 12-1 UART REGISTER SUMMERY .....	97
TABLE 12-2 UART BAUD RATE .....	100
TABLE 12-3 UART INTERRUPT CONTROL FUNCTION.....	102
TABLE 14-1 SAMPLING FREQUENCY AND CODEC CLOCK .....	113
TABLE 14-2 SAMPLING FREQUENCY AND SERIAL BIT CLOCK .....	113
TABLE 15-1. ENDPOINT LIST.....	118
TABLE 15-2. USB CORE REGISTER LIST.....	119
TABLE 18-1 ISP COMMAND SET .....	138
TABLE 18-2 ISP STATUS REGISTER .....	139
TABLE 19-1 DC ELECTRICAL CHARACTERISTICS .....	143
TABLE 19-2 ELECTRICAL CHARACTERISTICS.....	144
TABLE 19-3 POR ELECTRICAL CHARACTERISTICS .....	144
TABLE 19-4 RC OSC ELECTRICAL CHARACTERISTICS .....	144
TABLE 19-5 PLL ELECTRICAL CHARACTERISTICS.....	144
TABLE 19-6 ADC ELECTRICAL CHARACTERISTICS .....	145
TABLE 19-7 DAC ELECTRICAL CHARACTERISTICS .....	145

# 1 DESCRIPTIONS AND FEATURES

## 1.1 General Description

TG471 is a low-power 32bit micro-controller base on ADChips' EISC embedded microprocessor architecture. TG471 is fully integrated analog/digital peripherals, MCU and memories, 64KBytes~512KBytes Flash, 8KBytes SRAM. With on-chip Power On Reset, Voltage Regulator, Timer, UART, ADC, DAC. TG471 is truly System-on-a-Chip solution. The ISP allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an on-chip boot program running on the MCU core. TG471 is a powerful micro-controller that provides a highly flexible and cost effective solution to many embedded control applications.

## 1.2 Feature

- High-performance, Low-power 32-bit EISC Microprocessor
- 32-bit EISC Architecture
  - Internal 32-bit and External 8-bit Data Bus
  - 8 x 32-bit General Purpose Working Register + Peripheral Control Register
  - On-chip 1-cycle 32bit Multiplier
  - Up to 33MIPS Throughput at 33MHz
  - 1KB Instruction Cache
  - 8 x 32-bit Break Point Register
- Program and Data Memories
  - 64KBytes ~512KBytes of Re-programmable NOR Flash  
Endurance: 50,000 Write/Erase Cycles  
In System Programming
  - 8KBytes Internal SRAM
  - Static Memory Interface
- Peripherals
  - 26bit Watchdog Timer
  - 8 External Interrupts
  - 8Channel 16-bit Timer/Counter with 10bit Pre-scaler, Capture mode, PWM mode, and Output Compare Mode
  - 4 Channel UART with 4Bytes FIFO, Functionally compatible with the 16550
  - 2 Channel Master/Slave SPI with 8Bytes FIFO
  - Two Wire Interface
  - DMA
  - I2S with ADPCM
  - USB 2.0 Full Speed Device
  - 4 Channel 500KSPs 10bit ADC
  - 12bit DAC
  - 35 Port In/Out : 4mA(24), 8mA(11)



**▫ Special Features**

- Voltage Regulator(3.3V to 2.5V)
- Power On Reset
- On Chip RC Oscillator
- PLL
- 4Ch 10bit ADC
- 12bit DAC
- 3 Power Mode

**▫ Process**

- 0.25um CMOS VLSI
- 3.0V ~ 3.6V Operation
- 64 Pin LQFP Package

**▫ Speed**

- 0 ~ 33MHz

Preliminary

## 2 BLOCK DIAGRAM & PIN DESCRIPTIONS

### 2.1 Block Diagram

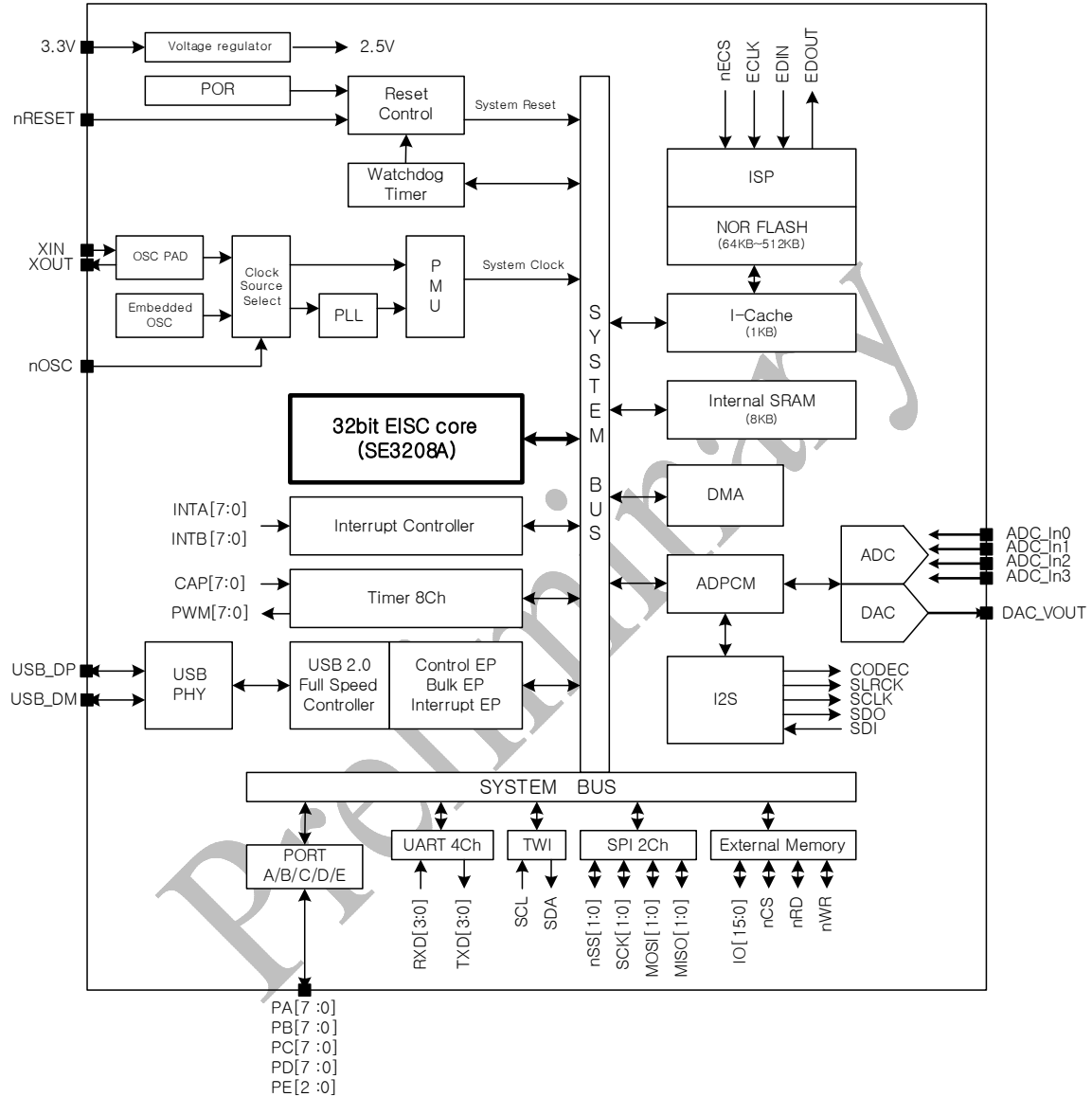


Figure 2-1 TG471 Block Diagram

## 2.2 Pin Configurations

### 2.2.1 Pinout

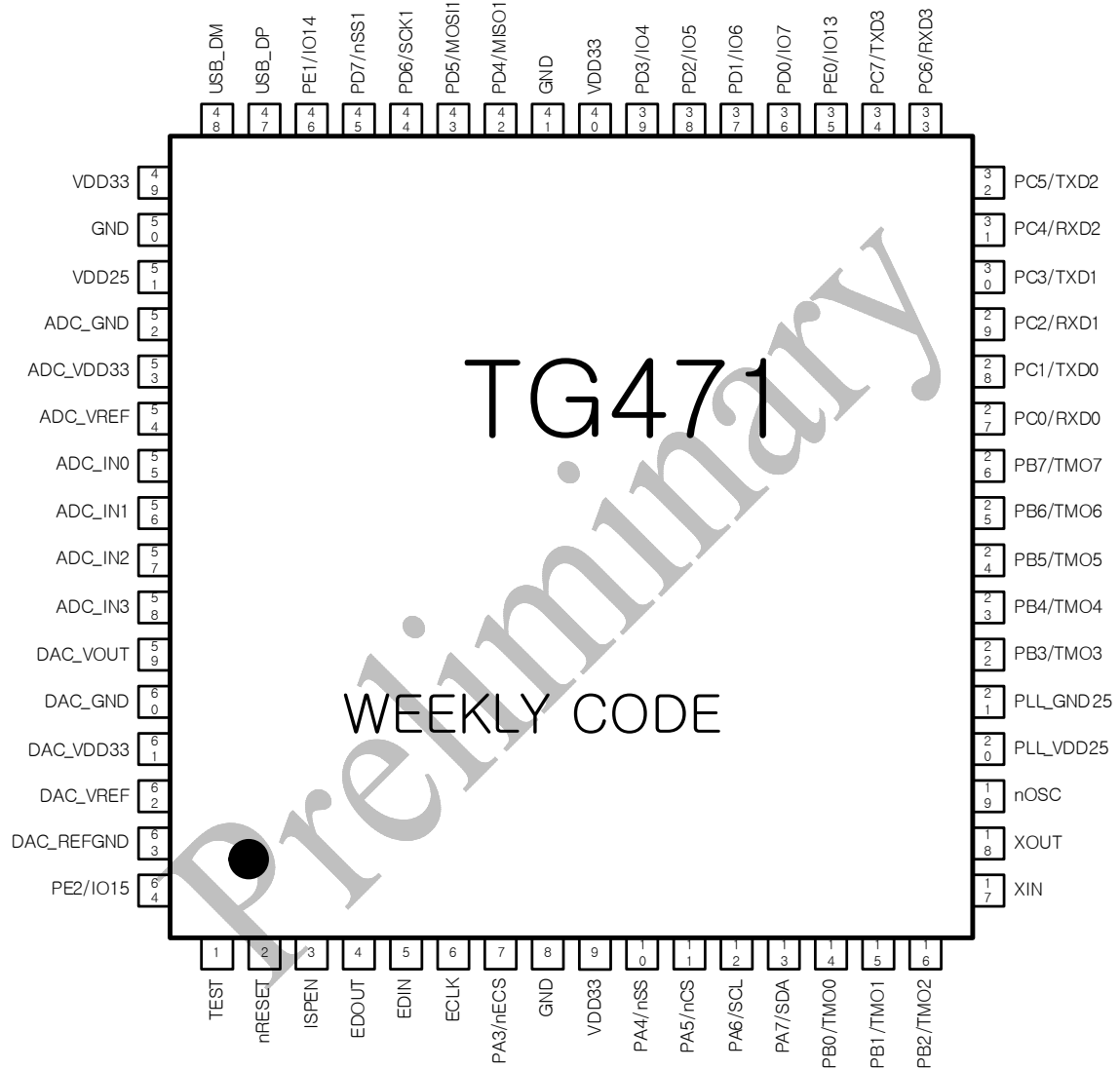


Figure 2-2 TG471 Pinout Diagram

## 2.2.2 Pin Definitions

Table 2-1 TG471 Pin Definitions

<i>Name</i>	<i>No.</i>	<i>Type</i>	<i>Output Drive Current</i>	<i>Pull-Up / Pull-Down</i>
TEST	1	Schmitt, 5V tolerant, Input	-	Pull-Down
nRESET	2	Schmitt, 5V tolerant, Input	-	Pull-Up
ISPEN	3	Schmitt, 5V tolerant, Input	-	Pull-Down
EDOUT	4	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
EDIN / nWR	5	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
ECLK / nRD	6	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
nECS	7	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
GND	8	Power Input	-	-
VDD33	9	Power Input	-	-
nSS0 / IO10 / PA4	10	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
nCS / PA5	11	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
SCL / IO11 / INT2B / PA6	12	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
SDA / IO12 / INT3B / PA7	13	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
TMO0 / CAP0 / INT0A / PB0	14	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
TMO1 / CAP1 / INT1A / PB1	15	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
TMO2 / CAP2 / INT2A / PB2	16	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
XIN	17	Analog Input	-	-
XOUT	18	Analog Output	-	-
nOSC	19	Schmitt, 5V tolerant, Input	-	Pull-Down
PLL_VDD25	20	Power Input	-	-
PLL_GND	21	Power Input	-	-
TMO3 / CAP3 / INT3A / PB3	22	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
TMO4 / CAP4 / INT4A / PB4	23	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
TMO5 / CAP5 / INT5A / PB5	24	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
TMO6 / CAP6 / INT6A / PB6	25	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
TMO7 / CAP7 / INT7A / PB7	26	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
RXD0 / PC0	27	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
TXD0 / PC1	28	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
RXD1 / INT0B / PC2	29	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
TXD1 / INT1B / PC3	30	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
SLRCK / RXD2 / INT4B / PC4	31	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
SCLK / TXD2 / INT5B / PC5	32	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
SDO / RXD3 / INT6B / PC6	33	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
SDI / TXD3 / INT7B / PC7	34	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
CODEC / PE0 / IO13	35	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
IO7 / PD0	36	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
IO6 / PD1	37	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
IO5 / PD2	38	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
IO4 / PD3	39	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
VDD33	40	Power Input	-	-
GND	41	Power Input	-	-

IO3 / MISO1 / PD4	42	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
IO2 / MOSI1 / PD5	43	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
IO1 / SCK1 / PD6	44	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
IO0 / nSS1 / PD7	45	Schmitt, 5V tolerant, Bidir	4mA	Pull-Up
IO14 / PE1	46	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up
USB_DP	47	Analog Out	-	-
USB_DM	48	Analog Out	-	-
VDD33	49	Power Input	-	-
GND	50	Power Input	-	-
VDD25	51	Power Output	-	-
ADC_GND	52	Power Input	-	-
ADC_VDD33	53	Power Input	-	-
ADC_VREF	54	Analog Input	-	-
ADC_IN0	55	Analog Input	-	-
ADC_IN1	56	Analog Input	-	-
ADC_IN2	57	Analog Input	-	-
ADC_IN3	58	Analog Input	-	-
DAC_VOUT	59	Analog Output	-	-
DAC_GND	60	Power Input	-	-
DAC_VDD33	61	Power Input	-	-
DAC_VREF	62	Analog Input	-	-
DAC_REFGND	63	Analog Input	-	-
IO15 / PE2	64	Schmitt, 5V tolerant, Bidir	8mA	Pull-Up

\*\*\* VDD25는 내부 Voltage regulator 의 출력이기 때문에 외부에서 2.5V를 공급할 필요가 없다. 또한 PLL\_VDD25에도 VDD25를 통해 2.5V 전압을 공급한다.

### 2.2.3 Pin Descriptions

VDD33	: 3.3V Supply voltage.
GND	: Ground.
VDD25	: Voltage regulator output 2.5V.
ADC_VDD33	: ADC supply voltage.
ADC_GND	: ADC ground.
ADC_VREF	: ADC Reference Input.
ADC_IN0	: ADC Analog Input 0.
ADC_IN1	: ADC Analog Input 1.
ADC_IN2	: ADC Analog Input 2.
ADC_IN3	: ADC Analog Input 3.
DAC_VDD33	: DAC digital supply voltage.
DAC_GND	: DAC ground.
DAC_VREF	: DAC Reference input.
DAC_REFGND	: DAC Reference ground.
DAC_VOUT	: DAC Voltage Output.
PLL_VDD25	: PLL 2.5V supply voltage.
PLL_GND	: PLL ground.
TEST	: TEST pin. Default value is pull-down(Typ. 130k $\Omega$ ). 0 : Normal operation mode      1 : Test mode
nOSC	: Clock Source Selection pin. Default value is pull-down(Typ. 130k $\Omega$ ). 0 : Embedded RC Oscillator section      1 : XIN section
ISPEN	: ISP Mode Selection pin. Default value is pull-down(Typ. 120k $\Omega$ ). 0 : Normal operation mode      1 : ISP mode
nRESET	: Active low Reset Input. Default value is pull-up(Typ. 120k $\Omega$ ).
XIN	: Crystal Input.(1MHz~10MHz)
XOUT	: Crystal output.
EDOUT	: ISP Data Out
EDIN	: ISP Data In
ECLK	: ISP Clock In
nECS	: ISP Chip Select In
PA[7:0]	: PIOA is an 8-bit bi-directional I/O port with internal pull-up resistors. 4mA driver.
PB[7:0]	: PIOB is an 8-bit bi-directional I/O port with internal pull-up resistors. 8mA driver.
PC[7:0]	: PIOC is an 8-bit bi-directional I/O port with internal pull-up resistors. 4mA driver.
PD[7:0]	: PIOD is an 8-bit bi-directional I/O port with internal pull-up resistors. 4mA driver.
PE[2:0]	: PIOE is an 3-bit bi-directional I/O port with internal pull-up resistors. 8mA driver.

MISO[1:0]	: SPI Master In / Slave Out
MOSI[1:0]	: SPI Master Out / Slave In
SCK[1:0]	: SPI Clock
nSS [1:0]	: SPI Chip Select
RXD[3:0]	: UART channel 0/1/2/3 receive data
TXD[3:0]	: UART channel 0/1/2/3 transmitter data
INTA[7:0]	: External Interrupts.
INTB[7:0]	: External Interrupts.
TMO[7:0]	: Timer outputs. PWM wave outputs or Timer wave outputs.
CAP[7:0]	: Timer Capture Input.
SCL	: TWI clock.
SDA	: TWI data.
nCS	: Static memory Chip Select.
nRD	: Static memory read.
nWR	: Static memory write.
IO[15:0]	: Static memory data.
CODEC	: Codec Clock Output.
SLRCK	: I2S Left/Right Signal Output.
SCLK	: I2S Serial Clock Output.
SDO	: I2S Data Output.
SDI	: I2S Data Input.

## 3 SE3208 EISC CORE

### 3.1 Introduction

TG471에는 SE3208 MCU가 내장되었다. SE3208은 RISC구조를 가지면서도 16 bit의 고정된 Instruction과 확장 가능한 Addressing을 이용하여 높은 성능을 갖도록 하였으며, 코드 밀도 또한 기존 RISC의 140-220%, 기존 CISC의 120-140%로 현격하게 높은 장점을 가지고 있다. 16 비트 축약 명령어 RISC보다 프로그램 크기가 5-15% 작고, 로드 스토어 빈도가 15% 이상 낮아서 데이터 전송 폭을 적게 요구 하므로 차세대 프로세서로 적합한 구조이고, 프로그램 메모리 크기가 작아지므로 Embedded Micro Processor로 폭 넓게 활용할 수 있다.

SE3208은 32비트 ALU와 32 비트의 General Purpose Register(GPR) 8개, Special Purpose Register(SPR) 5개를 가지고 있으며 메모리 시스템과 MCU가 연속적으로 동작하도록 3단의 파이프 라인을 사용하였다. 3단의 파이프 라인은 Instruction Fetch Cycle, Instruction Decoding/Execution, Memory Read/Write Cycle으로 구성되며 파이프 라인 제어를 하드웨어로 수행하여 부가적인 NOP 명령어의 삽입이 필요하지 않는다.

확장 플래그를 두어서 긴 오프셋과 긴 상수 오퍼랜드로 확장하는 구조를 채용함으로써 4G Byte Address도 16비트의 고정된 명령어으로써 표현 가능하다.

ALU, Shift 연산뿐만 아니라 Multiply 연산도 32bit 1-Cycle Signed Multiplier를 채용하여 대부분의 산술을 1-Cycle 에 처리할 수 있도록 하였다.

Exception Processing, Sub-routine Call/Return의 경우에 현재의 상태나 데이터를 저장하거나 복원하기 위해 PUSH/POP 명령어를 사용하는데 일반적으로 다수의 레지스터를 읽거나 써야 하는 특징이 있으므로 SE3208은 이러한 특징을 이용하여 한 개의 명령어로 11개의 레지스터에 대한 PUSH/POP을 수행할 수 있다.

SE3208은 Extension Register과 Extension Flag를 두어서 Immediate operand, Offset이 확장 가능하도록 하였고 Push/Pop list 명령어를 두어서 연속적인 Push/Pop이 가능하도록 하였으며, 또한 NOP Operation을 제거 함으로써 코드를 압축률을 높여 MCU가 사용하는 메모리의 크기를 줄여 실장 제어용 프로세서로서 적합한 구조를 가진다.

### 3.2 SE3208 Features

- **General Purpose Registers : 32 Bit \* 8**  
Register 0 ~ Register 7
- **Special Purpose Register : 32 bit \* 5**
  - PC : 32 bit program counter. Bit 0 is always 0
  - SP : 32 bit user stack pointer. Bit 1-0 are always 0
  - OSI : 32 bit OSI register. Bit 1-0 are always 0
  - ER : 32 bit extension register
  - SR : 32 bit status register
- **32 Bit ALSU (Adder , Shifter, Logic Block)**
- **16 Bit Fixed instruction set architecture.**  
Simple But Powerful instruction set



Simple hardware, High Performance  
Simple exception Processor

- **Extendable instruction set.**  
4G Byte linear Addressing,  
Operand could be extended up to full word length
- **Room for Co-processor.**
- **2 or 3 Stage Pipeline**
- **32 Bit EISC Processor (32 bit data & Address bus)**
- **32bit Signed Multiplier**  
32 X 32 -> 32 : Executed in one clock
- **In Case of Load or Branch Instructions, solve delay slot resolution by hardware**  
Don't need to use NOP instruction for delay slot.
- **Load/Store instruction set**  
Index addressing
- **Push/Pop Register List**  
Push/pop registers up to eleven.
- **14 Kinds of conditional branch instruction**  
JNV, JV, JP, JM, JNZ, JZ .....

### 3.3 SE3208 Block Diagram

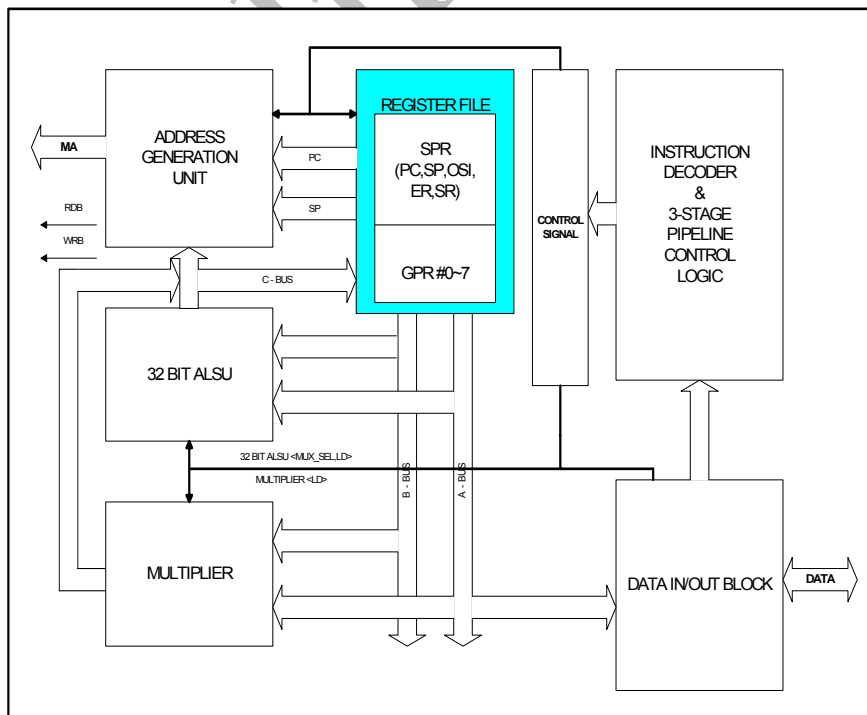


Figure 3-1 SE3208 Block Diagram

### 3.4 SE3208 Registers

SE3208은 8개의 32 bits general purpose register와 5개의special purpose register로써 PC, SP, OSI, ER, SR 을 가지고 있다. SE3208은 Little-endian 방식의 word ordering을 한다.

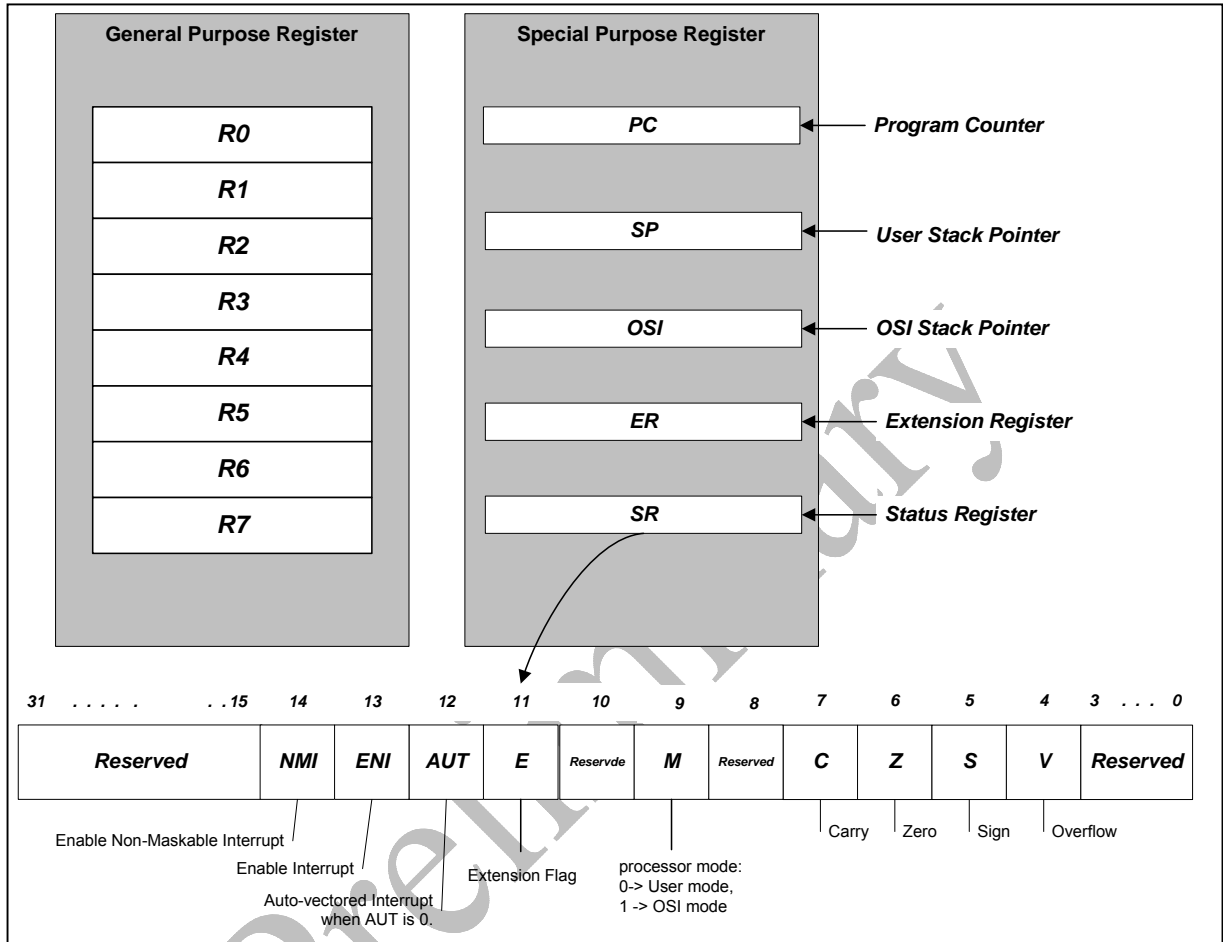


Figure 3-2 SE3208 MCU Registers

#### 3.4.1 General Purpose 32 Bit Registers (R0 – R7)

GPR 레지스터들은 일반적인 용도로 사용된다. 덧셈과 뺄셈에 대한 연산 과 shift 연산에 사용된다. 필요에 따라 8 비트, 16 비트, 32비트 연산을 행할 수 있으며 64비트 연산까지 처리한다. 즉 SE3208은 기본적으로 C 혹은 Assembler를 사용하여 64비트 연산까지 지원해 줄 수 있다. 이들 레지스터는 또한 R0를 제외하고 Index Register (혹은 base register)로 사용할 수 있다. GPR 레지스터는 아무런 제한이 없이 사용할 수 있도록 되어 있다. 하지만 사용자가 C언어를 사용하면서 inline 어셈블러를 사용할 때 레지스터 사용상에 주의를 기울여야 한다.

##### %R0-%R1

C에서 함수의 반환 값을 저장하는데 사용한다. 따라서 이 레지스터를 사용하는데 주의를 기울여야 한다.

**%R7**

C에서 frame pointer 레지스터로 사용한다. 따라서 이 레지스터를 사용할 경우에는 주의를 기울여야 한다. 그러나 optimization option을 주어 compile 할 경우 frame pointer를 사용하지 않기 때문에 이 경우는 %R7을 사용하는데 문제가 없다.

**3.4.2 Program Counter**

Program Counter는 현재 실행하고 있는 명령어에 대한 address를 가지고 있다. SE3208이 32bit Processor이지만 16bit fixed length 명령어를 사용하기 때문에 메모리에서 하나의 명령어를 가지고 올 때 마다 이 Program Counter는 자동으로 +2씩(2Bytes) 증가해 간다. Branch나 Call에서는 Program Counter를 사용하여 PC Relative addressing mode를 지원한다.

**3.4.3 Status Register**

Status Register는 현재 Core의 상태에 대한 정보를 각각의 비트에 담아 둔다. 연산의 결과에 따라 그 값이 설정되는 하위 4개의 Bit들과 Core의 상태 및 인터럽트에 대한 금지 여부 등을 가지는 상위 5개 비트로 구성된다.

**V : Overflow**

Set if an arithmetic overflow occurs implying that the result cannot be represented in the operand size; otherwise clear

**S : Sign flag**

Set if the Most Significant Bit of the result is set; otherwise clear

**Z : Zero flag**

Set if the result equals zero; otherwise clear

**C : Carry flag**

Set if a carryout of the Most Significant Bit of the operand occurs for an addition, or if a borrow occurs in a subtraction; otherwise clear

**E : Extension Flag**

필요한 Offset이나 데이터 값을 임시적으로 저장하여 의도하는 값을 만들어 내기 위해 Extension Register를 사용할 때 세트 된다. 명령어가 Extension Register를 사용하면 자동으로 '0'으로 설정 된다.

**AUT : Auto-vectored interrupt mode**

인터럽트를 Auto-vectored로 처리할 것인지 또는 Vectored로 처리할 것인지를 결정한다.

**ENI : Enable Interrupt**

NMI를 제외한 인터럽트를 Enable혹은 Disable시킬지 여부에 대한 결정을 한다.

**NMI : Enable non-maskable interrupt**

Non-maskable interrupt를 enable 혹은 disable 할 것인지에 대한 정보 비트이다. TG471에서는 NMI를 사용하지 않는다.

### 3.4.4 Stack Pointer

Stack pointer는 interrupt나 push, pop과 같은 경우 Stack영역의 주소를 나타낸다.

### 3.4.5 Extension Register

필요한 크기의 offset 혹은 immediate register를 만들기 위하여 임시로 사용되는 레지스터이다. 이 레지스터에 유효한 값을 기록하면 그때 Status Register에서 E Flag가 설정이 되고 뒤에 이어 오는 명령어에 의해서 Extension Register가 사용되는 명령어면 E Flag가 clear 된다. 즉 뒤에 이어 오는 명령어가 Extension Register를 사용하지 않는 명령어 ( ex, push, pop 등등) 일 경우에는 E Flag는 clear 되지 않고 그 상태를 지속시킨다. 이때 E Flag를 clear 시키기 위해서는 강제로 clear 시켜 주어야 한다. 단 Interrupt가 들어오면 자동으로 clear 된다.

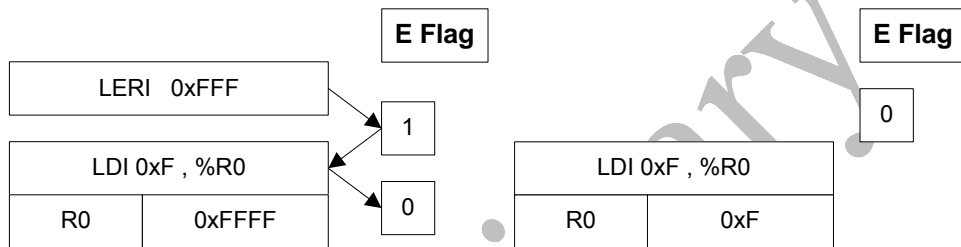


Figure 3-3 E-flag and Instruction

위 그림은 어셈블러에서 LERI 명령어를 사용했을 때의 결과와 사용하지 않았을 때의 결과를 보여준다. 좌측의 그림이 LERI를 사용해서 Extension Register에 값을 저장하고 다시 이 값을 사용해서 LDI 명령어에 의해서 Register 0에 0xFFFF를 기록하는 것이다. 이때 LERI 명령어를 사용했으므로 E-Flag는 1로 설정이 되고 다시 LDI 명령어에서 Extension Register 값을 사용했기 때문에 다시 '0'으로 설정된다. 우측의 그림은 같은 "LDI 0xF, %R0"이지만 이전에 LERI 명령어에 의해 E-Register에 값을 기록하지 않았으므로 Register 0에 기록되는 값은 단지 0xF이다. 이러한 과정은 실제 어셈블러를 사용할 경우 혹은 C를 사용한다고 해도 Assembler가 자동적으로 LERI 명령어를 사용한다. 어셈블러 혹은 C 프로그래밍을 할 경우에도 기본적으로 E Flag에 대한 고려는 하지 않아도 무방하다. 실질적으로 프로그래머에게 보여지거나 사용되는 register가 아니기 때문이다.

### 3.5 SE3208 Instruction Set

SE3208 MCU에 대한 구체적인 명령어는 SE3208 Core Manual 을 참고한다.

Table 3-1 SE3208 Instruction Set Summary

Operation		Assembler	Action
<b>Load</b>	Word	LD (%Ri/%SP, imm), %Rd	%Rd = [%Ri/%SP + imm]
	Byte	LDBU (%Ri/%SP, imm), %Rd	%Rd = ZeroExtent[Byte from (%Ri/%SP + imm)]
	signed	LDB (%Ri/%SP, imm), %Rd	%Rd = SignExtent[Byte from (%Ri/%SP + imm)]
	Half word (short)	LDSU (%Ri/%SP, imm), %Rd	%Rd = ZeroExtent[Short from (%Ri/%SP + imm)]
	signed	LDS (%Ri/%SP, imm), %Rd	%Rd = SignExtent[Short from (%Ri/%SP + imm)]
<b>Load Multiple</b>	Pop	POP <reg list>	while (all regs in reg list is popped) <lower num unpopped register in reg list> = [%SP], SP = SP + 4
<b>Store</b>	Word	ST %Rs, (%Ri/%SP, imm)	[%Ri/%SP + imm] = %Rs
	Byte	STB %Rs, (%Ri/%SP, imm)	[%Ri/%SP + imm][7:0] = %Rs[7:0]
	Half word (short)	STS %Rs, (%Ri/%SP, imm)	[%Ri/%SP + imm][15:0] = %Rs[15:0]
	Push	PUSH <reg list>	while (all regs in reg list is popped) [%SP = %SP - 4] = <higher num unpushed register in reg list>
<b>Move</b>	Move	MOV %Rs, %Rd	%Rd = %Rs
	immediate	LDI imm, %Rd	%Rd = imm
	with add	LEA (%Rs/%SP, imm), %Rd/%SP	%Rd/%SP = %Rs/%SP + imm
<b>Arithmetic</b>	Add	ADD %Rs1, %Rs2/imm, %Rd	%Rd = %Rs1 + %Rs2/imm
	with carry	ADC %Rs1, %Rs2/imm, %Rd	%Rd = %Rs1 + %Rs2/imm + C
	Subtract	SUB %Rs1, %Rs2/imm, %Rd	%Rd = %Rs1 - %Rs2/imm
	with carry	SBC %Rs1, %Rs2/imm, %Rd	%Rd = %Rs1 - %Rs2/imm - C
<b>Logical</b>	Multiply	MUL %Rs1, %Rs2, %Rd	%Rd = %Rs1 * %Rs2
	Test	TST %Rs1/imm, %Rs2	Update SR flag on %Rs1 AND %Rs2
	AND	AND %Rs1, %Rs2/imm, %Rd	%Rd = %Rs1 AND %Rs2/imm
	OR	OR %Rs1, %Rs2/imm, %Rd	%Rd = %Rs1 OR %Rs2/imm
	XOR	XOR %Rs1, %Rs2/imm, %Rd	%Rd = %Rs1 XOR %Rs2/imm
<b>Compare</b>	NEG(2's Complement)	NEG %Rs, %Rd	%Rd = - %Rs
	Compare	CMP %Rs1, %Rs2	Update SR flag on %Rs1 - %Rs2
<b>Shift</b>	Arithmetic shift right	ASR %Rc/imm_5, %Rd	{%Rd >> (%Rc/imm_5), %Rd@(MSB) = %Rd@(MSB)}
	Logical shift right	LSR %Rc/imm_5, %Rd	%Rd >> (%Rc/imm_5)
	Arithmetic shift left	ASL %Rc/imm_5, %Rd	%Rd << (%Rc/imm_5)
<b>Format Conversion</b>	Extension from byte to word	EXTB %Rd	SignExtent(%Rd[7:0])
	Extension from short to word	EXTS %Rd	SignExtent(%Rd[15:0])
<b>SR control</b>	Set a bit in SR	SET imm_4	SR.bit<imm_4> = 1 depend on processor mode
	Clear a bit in SR	CLR imm_4	SR.bit<imm_4> = 0 depend on processor mode

<b>Branch</b>	Jump on overflow clear on overflow set on sign clear / positive or zero on sign set / negative on non-zero / not equal on zero / equal on carry clear / unsigned higher or equal on carry set / unsigned lower on signed greater on signed less on signed greater or equal on signed less or equal on unsigned higher on unsigned lower or equal register indirection Call register indirection	JMP <label> JNV <label> JV <label> JP <label> JM <label> JNZ <label> JZ <label> JNC <label>  JC <label> JGT <label> JLT <label> JGE <label> JLE <label> JHI <label> JLS <label> JR %Rs CALL <label> CALLR %Rs	PC = address of <Label> if (V == 0) PC = address of <Label> if (V == 1) PC = address of <Label> if (S == 0) PC = address of <Label> if (S == 1) PC = address of <Label> if (Z == 0) PC = address of <Label> if (Z == 1) PC = address of <Label> if (C == 0) PC = address of <Label>  if (C == 1) PC = address of <Label> if ((Z+S^V) == 0) PC = address of <Label> if ((S^V) == 1) PC = address of <Label> if ((S^V) == 0) PC = address of <Label> if ((Z+S^V) == 1) PC = address of <Label> if ((C+Z) == 0) PC = address of <Label> if ((C+Z) == 1) PC = address of <Label> PC = %Rs { %SP = PC, PC = address of <Label> } { %SP = PC, PC = %Rs }
<b>Coprocessor</b>	Move to GPR from coproc Move to coproc from GPR	MVFC %Rs@CP MVTC %Rd@CP	%R0 = %Rs@CP %Rd@CP = %R0
<b>Soft Interrupt</b>	Software Interrupt	SWI imm 4	Software interrupt processor exception
<b>Halt</b>		HALT imm 4	Halt for low power

## 4 MEMORY MAP

TG471 내부에는 Program Memory와 Data Memory의 Main Memory 영역을 가진다.

Program Memory는 Flash 내장 형태에 따라 64K X 8, 128K X 8, 256K X 8, 512K X 8 구조의 64KB ~ 512KB In-System Programmable Flash Memory를 내장하였다.

Data Memory는 8KB SRAM을 내장하였다.

External Memory는 I/O와 Peripheral은 Memory Mapped I/O Register 영역에 위치한다.

Table 4-1 Memory Map

<i>Offset Address</i>	<i>Block</i>	<i>Remark</i>
0000_0000h ~ 0007_FFFFh	Embedded Flash (64KBytes)	Read Region of Flash Memory
0080_0000h ~ 0087_FFFFh	(128KBytes) (256KBytes) (512KBytes)	Programming Region of Flash Memory
0100_0000h ~ 0100_1FFFh	Internal SRAM (8KBytes)	Data Memory
0200_0000h ~ 02FF_FFFFh	Reserved	-
0300_0000h ~ 03FF_FFFFh	Memory Mapped I/O Register	Memory Mapped I/O
0400_0000h ~ 07FF_FFFFh	Reserved	-
0800_0000h ~ 08FF_FFFFh	External Static Memory	Static Memory
0900_0000h ~ FFFF_FFFFh	Reserved	-

Memory Mapped I/O Register는 Offset Address가 0x0300\_0000 ~ 0x0300\_0FFF 영역에서 256Bytes 단위로 나누어져 있다.

Table 4-2 Memory Mapped I/O Registers

<i>Offset Address</i>	<i>Block</i>
0300_0000h	PIN Multiplexer
0300_0100h	Reserved
0300_0200h	Power Management Unit
0300_0300h	NOR Flash Controller
0300_0400h	Watchdog Timer
0300_0500h	Interrupt Controller
0300_0600h	Timer 8Ch
0300_0700h	SPI 2Ch
0300_0800h	TWI
0300_0900h	UART 4Ch
0300_0A00h	GPIO
0300_0B00h	ADC
0300_0C00h	DAC
0300_0D00h	DMA
0300_0E00h	I2S with ADPCM
0300_0F00h	USB Device

\*\*\* The address boundary is 0x100 (256 Bytes)



## 4.1 Flash Memory

### 4.1.1 TG471A (64KBytes)

TG471A는 64KBytes Flash Memory 로 구성되어 있고, 16 Sector로 구성되어 있다. 각 Sector는 4KBytes의 Memory로 구성된다.

- Flash Size : 1 block = 16 sectors = 64 KBytes
- 1 sector = 4 KBytes

Flash Memory는 물리적으로는 한 개의 64KBytes Memory로 구성되지만, 논리적으로 Read Region과 Programming Region으로 분리되어있다. 즉, 0000\_0000h ~ 0000\_FFFFh 의 Read Region Flash Memory 와 0080\_0000h ~ 0080\_FFFFh 의 Programming Region Flash 는 논리적 주소는 다르지만 물리적으로 동일한 Memory 이기 때문에 Programming Region Flash의 sector0 ~ sector15에 Programming 한 데이터는 Region Flash의 sector0 ~ sector15을 통해 Read 된다.

Table 4-3 Read Region in On-chip Flash Memory (64KBytes)

<i>Address Range (64KBytes)</i>	<i>Sector Number</i>	<i>Block Number</i>
0000 0000h - 0000 0FFFh	Sector 0 (4KBytes)	Block (64KBytes)
.....	.....	
0000 F000h - 0000 FFFFh	Sector 15 (4KBytes)	

Table 4-4 Write Region in On-chip Flash Memory (64KBytes)

<i>Address Range (64KBytes)</i>	<i>Sector Number</i>	<i>Block Number</i>
0080 0000h - 0080 0FFFh	Sector 0 (4KBytes)	Block (64KBytes)
.....	.....	
0080 F000h - 0080 FFFFh	Sector 15 (4KBytes)	

**4.1.2 TG471B (128KBytes)**

TG471B는 128KBytes Flash Memory 로 구성되어 있다. 2개의 블록으로 구성되며, 각 블록당 16 Sector로 나누어져 있다. 각 Sector는 4KBytes의 Memory로 구성된다.

- Flash Size : 2 blocks = 32 sectors = 128 KBytes
- 1 sector = 4 KBytes

Flash Memory는 물리적으로는 한 개의 128KBytes Memory로 구성되지만, 논리적으로 Read Region과 Programming Region으로 분리되어있다. 즉, 0000\_0000h ~ 0001\_FFFFh 의 Read Region Flash Memory 와 0080\_0000h ~ 0081\_FFFFh 의 Programming Region Flash 는 논리적 주소는 다르지만 물리적으로 동일한 Memory 이기 때문에 Programming Region Flash의 sector0 ~ sector31에 Programming 한 데이터는 Read Region Flash의 sector0 ~ sector31을 통해 Read 된다.

Table 4-5 Read Region in On-chip Flash Memory (128KBytes)

<i>Address Range (128KBytes)</i>	<i>Sector Number</i>	<i>Block Number</i>
0000 0000h – 0000 0FFFh	Sector 0 (4KBytes)	Block 0 (64KBytes)
.....	.....	
0000 F000h – 0000 FFFFh	Sector 15 (4KBytes)	Block 1 (64KBytes)
0001 0000h – 0001 0FFFh	Sector 16 (4KBytes)	
.....	.....	
0001 F000h – 0001 FFFFh	Sector 31 (4KBytes)	

Table 4-6 Write Region in On-chip Flash Memory (128KBytes)

<i>Address Range (128KBytes)</i>	<i>Sector Number</i>	<i>Block Number</i>
0080 0000h – 0080 0FFFh	Sector 0 (4KBytes)	Block 0 (64KBytes)
.....	.....	
0080 F000h – 0080 FFFFh	Sector 15 (4KBytes)	Block 1 (64KBytes)
0081 0000h – 0081 0FFFh	Sector 16 (4KBytes)	
.....	.....	
0081 F000h – 0081 FFFFh	Sector 31 (4KBytes)	

**4.1.3 TG471C (512KBytes)**

TG471D는 512KBytes를 내장하고 있으며, 8개의 블록으로 구성된다. 각 블록당 4 Sector로 구성되어 있다. 각 Sector는 16KBytes의 Memory로 구성된다.

- Flash Size = 8 blocks = 32 sectors = 512 KBytes
- 1 sector = 16 KBytes

Flash Memory는 물리적으로는 한 개의 512KBytes Memory로 구성되지만, 논리적으로 Read Region과 Programming Region으로 분리되어있다. 즉, 0000\_0000h ~ 0007\_FFFFh 의 Read Region Flash Memory 와 0080\_0000h ~ 0087\_FFFFh 의 Programming Region Flash 는 논리적 주소는 다르지만 물리적으로 동일한 Memory 이기 때문에 Programming Region Flash의 sector0 ~ sector31에 Programming 한 데이터는 Read Region Flash의 sector0 ~ sector31을 통해 Read 된다.

Table 4-7 Read Region in On-chip Flash Memory (512KBytes)

<b>Address Range (512KBytes)</b>	<b>Sector Number</b>	<b>Block Number</b>
0000 0000h – 0000 3FFFh	Sector 0 (16KBytes)	Block 0 (64KBytes)
.....	.....	
0000 C000h – 0000 FFFFh	Sector 3 (16KBytes)	Block 1 (64KBytes)
0001 0000h – 0001 3FFFh	Sector 4 (16KBytes)	
.....	.....	Block 2 (64KBytes)
0001 C000h – 0001 FFFFh	Sector 7 (16KBytes)	
0002 0000h – 0002 3FFFh	Sector 8 (16KBytes)	Block 3 (64KBytes)
.....	.....	
0002 C000h – 0002 FFFFh	Sector 11 (16KBytes)	Block 4 (64KBytes)
0003 0000h – 0003 3FFFh	Sector 12 (16KBytes)	
.....	.....	Block 5 (64KBytes)
0003 C000h – 0003 FFFFh	Sector 15 (16KBytes)	
0004 0000h – 0004 3FFFh	Sector 16 (16KBytes)	Block 6 (64KBytes)
.....	.....	
0004 C000h – 0004 FFFFh	Sector 19 (16KBytes)	Block 7 (64KBytes)
0005 0000h – 0005 3FFFh	Sector 20 (16KBytes)	
.....	.....	Block 7 (64KBytes)
0005 C000h – 0005 FFFFh	Sector 23 (16KBytes)	
0006 0000h – 0006 3FFFh	Sector 24 (16KBytes)	Block 7 (64KBytes)
.....	.....	
0006 C000h – 0006 FFFFh	Sector 27 (16KBytes)	Block 7 (64KBytes)
0007 0000h – 0007 3FFFh	Sector 28 (16KBytes)	
.....	.....	Block 7 (64KBytes)
0007 C000h – 0007 FFFFh	Sector 31 (16KBytes)	

Table 4-8 Programming Region in On-chip Flash Memory (512KBytes)

<i>Address Range (512KBytes)</i>	<i>Sector Number</i>	<i>Block Number</i>
0080 0000h – 0080 3FFFh	Sector 0 (16KBytes)	Block 0 (64KBytes)
.....	.....	
0080 C000h – 0080 FFFFh	Sector 3 (16KBytes)	Block 1 (64KBytes)
0081 0000h – 0081 3FFFh	Sector 4 (16KBytes)	
.....	.....	Block 2 (64KBytes)
0081 C000h – 0081 FFFFh	Sector 7 (16KBytes)	
0082 0000h – 0082 3FFFh	Sector 8 (16KBytes)	Block 3 (64KBytes)
.....	.....	
0082 C000h – 0082 FFFFh	Sector 11 (16KBytes)	Block 4 (64KBytes)
0083 0000h – 0083 3FFFh	Sector 12 (16KBytes)	
.....	.....	Block 5 (64KBytes)
0083 C000h – 0083 FFFFh	Sector 15 (16KBytes)	
0084 0000h – 0084 3FFFh	Sector 16 (16KBytes)	Block 6 (64KBytes)
.....	.....	
0084 C000h – 0084 FFFFh	Sector 19 (16KBytes)	Block 7 (64KBytes)
0085 0000h – 0085 3FFFh	Sector 20 (16KBytes)	
.....	.....	Block 7 (64KBytes)
0085 C000h – 0085 FFFFh	Sector 23 (16KBytes)	
0086 0000h – 0086 3FFFh	Sector 24 (16KBytes)	Block 7 (64KBytes)
.....	.....	
0086 C000h – 0086 FFFFh	Sector 27 (16KBytes)	Block 7 (64KBytes)
0087 0000h – 0087 3FFFh	Sector 28 (16KBytes)	
.....	.....	Block 7 (64KBytes)
0087 C000h – 0087 FFFFh	Sector 31 (16KBytes)	

**4.1.4 Flash Access**

TG471에 내장된 Flash Memory는 최대 80nsec의 접근속도를 가지기 때문에 system clock의 속도에 따라 Access Cycle을 변경하여야 한다. System clock이 12MHz 이하의 속도에서 Flash Memory를 Read할 경우에는 FACC레지스터의 FRACC를 “00”으로 설정하면 1-clock으로 Flash Memory를 접근한다. 12MHz 이상일 경우에는 (system clock)/12MHz 단위로 FRACC를 설정하여야 한다.

예) 24MHz system clock : FRACC => 2-clock

12MHz 이상일 경우에도 Flash Memory의 속도에 독립적으로 MCU가 매 Clock 마다 명령을 처리할 수 있도록 I-Cache를 내장하였다. I-Cache는 Flash Memory에 대한 Cache로써 MCU가 Flash Memory의 속도에 독립적으로 동작할 수 있도록 한다.

I-Cache는 CAEN레지스터의 CAEN 비트에 의해 설정될 수 있고, I-Cache가 Enable되면 FRACC 값에 관계없이 대부분의 명령을 1-clock으로 접근하게 된다.

Flash Memory를 programming 하는 경우에는 FPACC를 추가로 설정하여야 하고, FRACC와 동일하게 12MHz 단위로 설정된다. 내장된 Flash Memory는 8비트 구조를 가지고 있기 때문에 Flash programming 시에는 8비트 단위로 접근하여야 한다.

Flash Memory Program Cycle 이 결정된 후, Flash Memory에 programming 하기 위해서는 FPEN 비트를 설정하여 Flash Memory에 대한 programming을 가능하도록 설정한 후, FMOD 비트를 설정하여 Flash의 동작모드를 결정한다.

Flash Memory에 대한 동작 중 상태는 FST 레지스터를 통해 확인할 수 있다. Flash Memory는 구조상 Program 시 충분한 지연시간이 필요하다.

I-Cache가 Enable된 상태에서 Flash Memory에 대한 programming을 수행한다면 CAIV bit를 설정하여 programming된 Flash의 내용과 I-Cache의 내용을 다시 일치시켜야 한다.

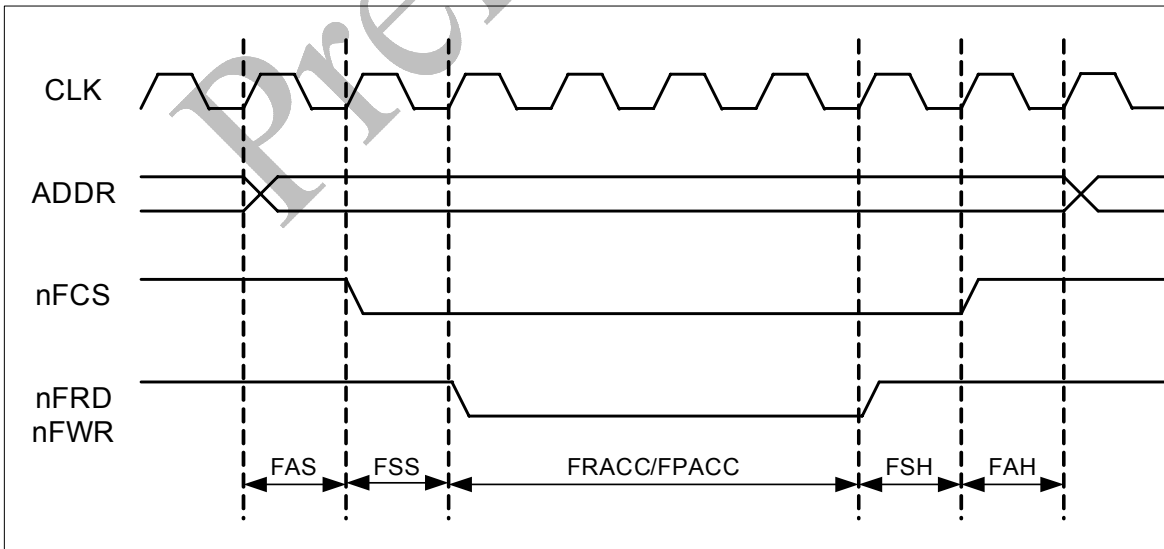


Figure 4-1 Flash Access Cycles

#### 4.1.5 Flash Control Registers

##### Flash Access Control Register (FACC)

Address : 0x0300\_0300h

Bit	R/W	Description	Default Value
31 : 14	R	Reserved	-
13 : 12	RW	FRACC : Flash read access cycle 00 : 1 clock(~12MHz) 01 : 2 clock(~24MHz) 10 : 3 clock(25MHz~) 11 : 4 clock(25MHz~)	11
11 : 10	RW	FAS : Flash address set-up cycle 00 : 0 clock            01 : 1 clock 10 : 2 clock            11 : 4 clock	01
9 : 8	RW	FSS : Flash select set-up cycle 00 : 0 clock            01 : 1 clock 10 : 2 clock            11 : 4 clock	00
7 : 6	R	Reserved	-
5 : 4	RW	FPACC : Flash programming access cycle 00 : 1 clock(~12MHz) 01 : 2 clock(~24MHz) 10 : 3 clock(25MHz~) 11 : 4 clock(25MHz~)	00
3 : 2	RW	FSH : Flash selection hold cycle 00 : 0 clock            01 : 1 clock 10 : 2 clock            11 : 4 clock	00
1 : 0	RW	FAH : Flash address hold cycle 00 : 0 clock            01 : 1 clock 10 : 2 clock            11 : 4 clock	00

##### Flash Operation Mode Control Register (FMOD)

Address : 0x0300\_0304h

Bit	R/W	Description	Default Value
31 : 5	R	Reserved	-
4	RW	FPEN : Flash programming enable bit 0 : Disable            1 : Enable Flash memory map에서 Programming Region에 대한 접근을 설정한다. Read Region에는 영향을 주지 않는다.	0
3	R	Reserved	-
2 : 0	RW	FMOD : Flash operation mode 000 : Read mode 001 : Programming mode 010 : Sector erase mode 011 : Block erase mode 100 : Chip erase mode	000

**Flash Status Register (FST)**

Address : 0x0300\_0308h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 7	R	Reserved	-
6 : 4	R	FMODST : Flash Operation State bit 000 : Read mode 001 : Byte program mode 010 : Sector erase mode 011 : Block erase mode 100 : Chip erase mode	000
3 : 1	R	Reserved	-
0	R	FRDY : Flash Ready bit 0 : Busy, indicate the progress of the operation 1 : Ready, indicate the completion of the operation	00

**Instruction Cache Enable Register (CAEN)**

Address : 0x0300\_0310h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 1	R	Reserved	-
0	W	CAEN : Instruction Cache Enable bit 0 : Disable                      1 : Enable	0

**Instruction Cache Invalidation Enable Register (CAIV)**

Address : 0x0300\_0314h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 1	R	Reserved	-
0	W	CAIV : Instruction Cache Invalidation Enable bit 0 : Disable                      1 : Enable	0

### 4.2 External Static Memory

External Static Memory와의 Interface를 위해 nCS, nRD, nWR, 그리고 D[15:0]을 지원하며 항상 4 cycle의 system clock으로 접근한다.

Memory의 Address는 별도로 할당되어 있지 않기 때문에 GPIO를 통하여 생성하여야 한다. 그러나 NAND Flash는 Address가 필요 없기 때문에 CEb, ALE, CLE 를 GPIO를 통하여 생성하면 제어할 수 있다.

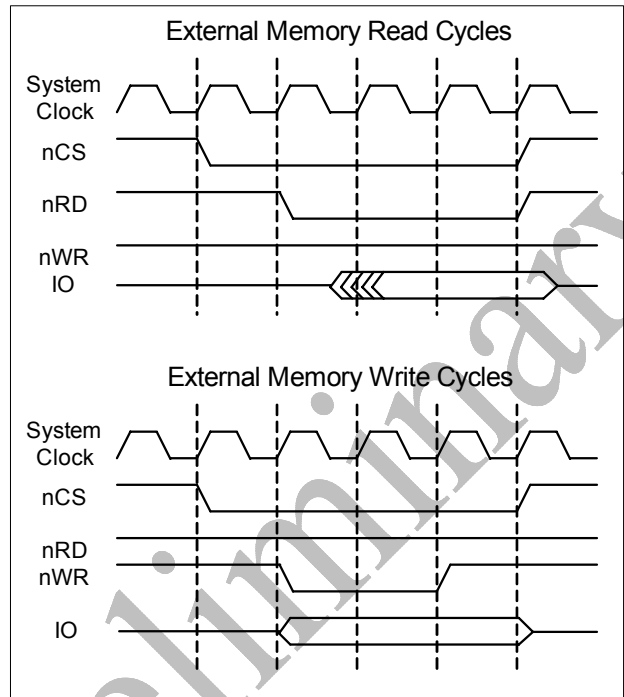


Figure 4-2 External Static Memory Interface timing

Table 4-9 External Memory Interface Timing Parameters

<i>System Clock</i>	<i>nCS width</i>	<i>nRD width</i>	<i>nWR width</i>	<i>Unit</i>
33MHz	Max. 120	Max. 90	Max. 60	ns
24MHz	Max. 166	Max. 125	Max. 83	ns
12MHz	Max. 332	Max. 250	Max. 166	ns
6MHz	Max. 664	Max. 500	Max. 332	ns



## 5 SYSTEM POWER AND RESET

### 5.1 Voltage Regulator

Voltage Regulator는 3.3V전압을 입력 받아 내부 회로에 2.5V 전원을 공급하기 때문에 외부에 3.3V 단일 전원을 사용할 수 있도록 한다.

PLL\_VDD25로는 Voltage Regulator의 출력인 VDD25를 칩 외부에서 연결하여 공급한다.

Preliminary

## 5.2 Reset

TG471의 Reset Sources는 다음과 같다.

- Power on reset
- External reset
- Watchdog reset

### 5.2.1 Power On Reset

Power On 후 내부 전압이 1.8V 이상이 되면 300usec의 Reset을 유지한다. Power On Reset이 풀린 후에는 External Reset에 의해 내부 Reset이 처리된다.

Table 5-1 Power On Reset Characteristics

Parameter	Min	Typ	Max	Units
Power on reset threshold point		1.8		V

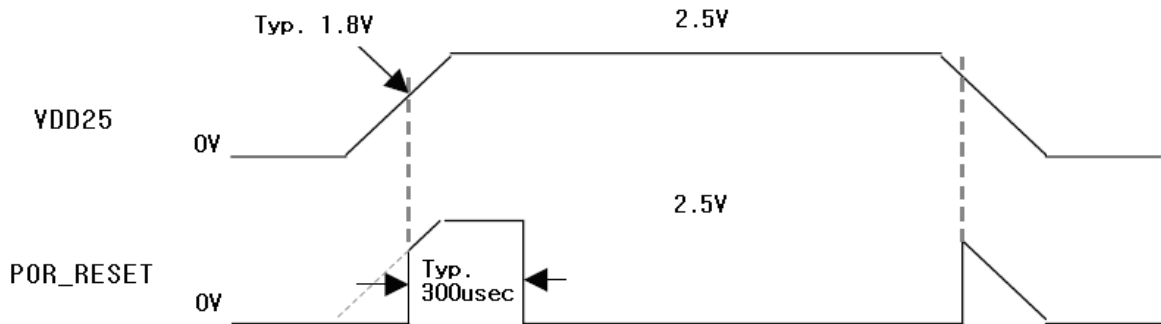


Figure 5-1 Power On Reset Timing Diagram

### 5.2.2 External Reset

외부에서 인가된 nRESET 신호가 “LOW” 일 때 발생한다. nRESET이 “HIGH”로 상승한 후 XIN 클럭이 1024 cycle이 될 때까지 내부 Reset 을 유지한 후, MCU가 0x0000\_0000 주소에 접근하여 Reset Vector Address를 읽어 들인다.

Table 5-2 External Reset Characteristics

Parameter	Min	Typ	Max	Units
External reset threshold point (rising)	1.44	1.50	1.56	V
External reset threshold point (falling)	0.89	0.94	0.99	V
Width of Reset		1024		XIN clock

### 5.2.3 Watchdog Reset

Watchdog Timer는 시스템 에러, 정상적으로 응답하지 않는 장치 또는 noise와 같은 이유로 MCU가 정상적인 동작을 하지 않을 때, 정상상태로 복귀시키는 역할을 한다.

Watchdog Reset 이 Enable (WDREN = '1')되면 WDCNT에 설정된 값에서 1024 System Clock 마다 '1'씩 감소하여 WDCNT 값이 '0'이 되면 Watchdog Reset 이 발생한다. Watchdog Timer는 10bit pre-scaler를 가지고 있기 때문에 Watchdog counter은 1024 Clock 마다 감소한다.

Watchdog Reset이 발생하면 WDST 비트에 Watchdog Reset이 발생한 상태가 저장된다. 모든 레지스터의 초기값은 Watchdog Reset이 발생하면 초기화되지만, WDST 비트는 RSCON 레지스터가 Read 될 때만 자동으로 Clear 된다.

WDREN을 설정된 상태에서 Watchdog Reset이 걸리지 않게 하기 위해서는 16 비트의 Watchdog Counter 값이 '0'이 되지 않도록 주기적으로 WDCNT를 재설정하여 Watchdog Reset이 발생하지 않도록 해야 한다.

Program Counter Address Register와 Stack Point Address Register는 Watchdog Reset이 발생한 시점에서 MCU가 수행하던 값을 저장하고 있기 때문에 Watchdog 한 시점에서의 MCU의 Program Counter Address 값과 Stack Point Address 값을 확인할 수 있기 때문에 디버깅하는데 유용하게 사용할 수 있다.

## 5.2.4 Reset Control Registers

### Reset Control Register (RSCON)

Address : 0x0300\_0400h

Bit	R/W	Description	Default Value
31 : 4	R	Reserved.	-
4	R	WDST : Watchdog Status bit 0 : No Watchdog Reset 1 : Watchdog Reset Clear at Read	0
3	R	Reserved	-
2	RW	USB Reset 0 : USB Reset Disable 1 : USB Reset Enable	0
1	RW	PLL Reset 0 : PLL Reset Enable 1 : PLL Reset Disable	1
0	RW	WDREN : Watchdog Timer Reset Enable bit 0 : Disable Watchdog Timer Reset 1 : Enable Watchdog Timer Reset	0

### Watchdog Timer Counter Value Register (WDCNT)

Address : 0x0300\_0404h

Bit	R/W	Description	Default Value
31: 16	R	Reserved	-
15 : 0	RW	Watchdog Timer Counter 16bit Value.	0xFFFFh

### Program Counter Address Register (PCADR)

Address : 0x0300\_0408h

Bit	R/W	Description	Default Value
31: 0	R	Program Counter Address Watchdog Reset 이 발생한 시점의 PC 값.	0x0000_0000h

### Stack Point Address Register (SPADR)

Address : 0x0300\_040Ch

Bit	R/W	Description	Default Value
31: 0	R	Stack Point Address Watchdog Reset 이 발생한 시점의 SP 값.	0x0000_0000h

## 6 CLOCK CONTROL AND POWER MANAGEMENT

### 6.1 Clock Control

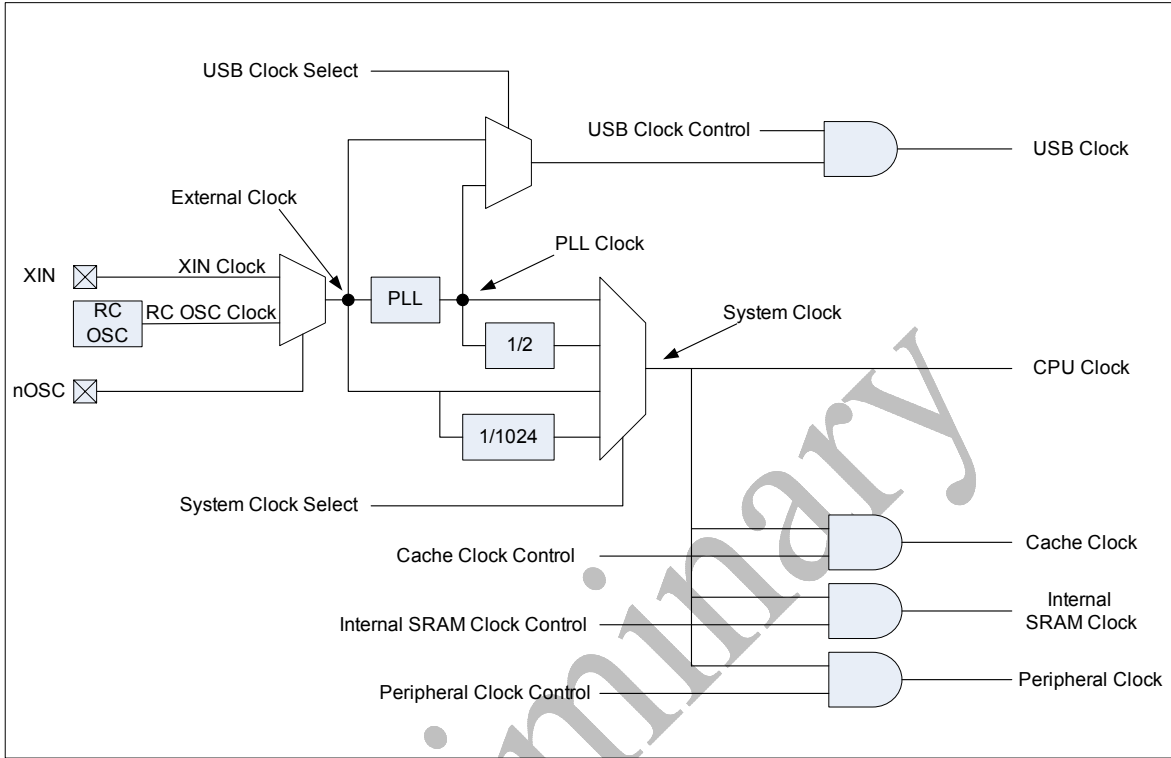


Figure 6-1 TG471 Clock Distribution

TG471는 External Clock을 외부로부터 받아들이며 “MCU Clock”, “Flash Clock”, “Cache Clock”, “Internal SRAM Clock”, “Peripheral Clock” 그리고 “USB Clock”으로 분류하여 제어한다.

## 6.2 Power Management

“CPU Clock” 은 nOSC, OSCCON, SCLKSEL 비트의 조합에 의해 다음과 같이 제어된다.

Table 6-1 System Clock

<i>nOSC</i>	<i>OSCCON[1:0]</i>	<i>CLKCON[1:0]</i>	<i>System Clock</i>	<i>Mode</i>
1	0x	xx	Clock OFF	STOP
1	1x	00	XIN Clock	External
1	1x	01	XIN Clock / 1024	Sleep
1	1x	10	XIN's PLL Clock	PLL
1	1x	11	XIN's PLL Clock / 2	PLL
0	x0	xx	Clock OFF	STOP
0	x1	00	RC OSC Clock	External
0	x1	01	RC OSC Clock / 1024	Sleep
0	x1	10	RC OSC's PLL Clock	PLL
0	x1	11	RC OSC's PLL Clock / 2	PLL

CLKCON 레지스터는 해당 블록이 사용되지 않을 경우에는 언제든지 ON/OFF 제어가 가능하다.

Table 6-2 Active Clock Domains and Wake Up Sources

<i>Sleep mode</i>	<i>Active clock</i>	<i>Stop clock</i>	<i>Wake up sources</i>
PLL clock mode	External Clock PLL Clock CPU Clock Cache Clock Internal SRAM Clock Peripheral Clock		All Interrupt Reset
External clock mode	External Clock CPU Clock Cache Clock Internal SRAM Clock Peripheral Clock	PLL Clock	All Interrupt Reset
Sleep mode (Ext/1024)	External Clock CPU Clock(Ext/1024) Cache Clock(Ext/1024) Internal SRAM Clock (Ext/1024) Peripheral Clock (Ext/1024)	PLL Clock	All Interrupt Reset
Stop mode (XIN OFF)		External Clock CPU Clock Cache Clock Internal SRAM Clock Peripheral Clock	Reset

### 6.3 Power and Clock Control Registers

#### RC Oscillator Control Register (OSCCON)

Address : 0x0300\_0200h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7 : 4	RW	OSCCLK : Embedded RC OSC Clock Selection 0000 : 5.00MHz      0001 : 6.10MHz 0010 : 8.08MHz      0011 : 9.81MHz 0100 : 13.01MHz     0101 : 15.72MHz 0110 : 17.55MHz     0111 : 21.12MHz 1000 : 22.99MHz     1001 : 25.58MHz 1010 : 28.58MHz     1011 : 30.42MHz 1100 : 33.10MHz     1101 : 35.12MHz 1110 : 37.73MHz     1111 : 39.89MHz RC OSC는 ±5%의 오차가 발생할 수 있기 때문에 정확한 클럭이 필요한 시스템에는 사용하기 힘들다.	1000
3 : 2	R	Reserved	-
1	RW	XIN Enable bit 0 : Disable            1 : Enable XIN을 통하여 인가되는 클럭을 제어한다. Disable시 XIN/XOUT을 사용한 Crystal의 발진이 멈추게 된다.	1
0	RW	RC OSC Enable bit 0 : Disable            1 : Enable 내장된 RC Oscillator의 동작을 제어한다.	1

#### PLL Control Register (PLLCON)

Address : 0x0300\_0204h

Bit	R/W	Description	Default Value
31 : 13	R	Reserved	-
12 : 8	RW	N-Counter[4:0]	00011
7 : 6	RW	O-Counter[1:0]	0
5 : 4	RW	M-Counter[1:0]	0
3 : 2	R	Reserved	-
1	R	PLL Lock bit 0 : Unlocking          1 : Locking PLL 에 의해 발생된 클럭이 안정된 상태임을 나타 내는 비트이다. PLL 클럭을 사용하고자 할 때는 반드시 이 비트가 Locking 된 것을 확인한 후 PLL 클럭으 로 전환해야 한다. Lock bit는 PLL Enable bit 와 N/O/M Counter 비트 설정이 바뀔 때 마다 Unlocking 된 후 최대 200usec 이후에 다시 Locking 된다.	0
0	RW	PLL Enable bit 0 : Disable            1 : Enable	0

$$PLL \text{ Multiplication Factor} = \frac{N+1}{2^M \times 2^{O+1}}$$

**Clock Control Register (CLKCON)**

Address : 0x0300\_0208h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 10	R	Reserved	-
8	RW	UCLKSEL : USB Clock Select bit 0 : PLL Clock      1 : EXT Clock	0
7	RW	UCLKEN : USB Clock Control bit 0 : Clock Disable      1 : USB Clock	0
6	RW	PCLKEN : Peripheral Clock Control bit 0 : Clock Disable      1 : System Clock	1
5	RW	DCLKEN : Internal Data SRAM Clock Control bit 0 : Clock Disable      1 : System Clock	1
4	RW	CCLKEN : Cache Clock Control bit 0 : Clock Disable      1 : System Clock	1
3 : 2	R	Reserved	-
1 : 0	RW	SCLKSEL : System Clock Select bit 00 : External Clock      01 : External Clock / 1024 10 : PLL Clock      11 : PLL / 2	00

\*\*\* External Clock : RC OSC Clock or XIN Clock

**I2S External Clock Selection Register (I2SCLK)**

Address : 0x0300\_020Ch

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 2	R	Reserved	-
1 : 0	RW	I2SCLK : I2S Source Clock Select bit 00 : XIN Clock      01 : RC OSC Clock 10 : CAP[0]      11 : CAP[1] I2S에 사용될 클럭을 선택한다.	00



## 7 ALTERNATE PORT AND I/O PORTS

### 7.1 Alternate Port Function

TG471의 모든 I/O Port 핀들의 초기값은 PIO Input 이다. 또한 PIO의 모든 입력 핀이 내부적으로 Pull-up 되어있어 입력을 unknown으로 받아들이는 사항을 방지하였다.

PA[3:0] Ports의 경우 내부 Flash Download를 할 수 있는 ISP 관련 신호가 연결되어 있기 때문에 ISPEN = "High" 에서 PA[3:0] Ports는 nECS, ECLK, EDIN, EDOUT 신호로 자동으로 전환된다.

Preliminary

### 7.1.1 PIN MUX Control Registers

#### **PIN MUX Select 0 Register (POMUX)**

Address : 0x0300\_0020h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 16	R	Reserved	-
15 : 14	RW	PA7 : PA7 Port Selection bit 00 : SDA(TWI)      01 : IO12 10 : INT3B          11 : PA7	11
13 : 12	RW	PA6 : PA6 Port Selection bit 00 : SCL(TWI)      01 : IO11 10 : INT2B          11 : PA6	11
11 : 10	RW	PA5 : PA5 Port Selection bit 00 : Reserved      01 : nCS 10 : Reserved      11 : PA5	11
9 : 8	RW	PA4 : PA4 Port Selection bit 00 : nSS0(SPI Ch0) 01 : IO10 10 : Reserved      11 : PA4	11
7 : 6	RW	PA3 : PA3 Port Selection bit 00 : Reserved      01 : IO9 10 : Reserved      11 : PA3 ISP Mode에서 nECS 핀으로 설정된다.	11
5 : 4	RW	PA2 : PA2 Port Selection bit 00 : SCK0(SPI Ch0) 01: nRD 10 : Reserved      11: PA2 ISP Mode에서 ECLK 핀으로 설정된다.	11
3 : 2	RW	PA1 : PA1 Port Selection bit 00 : MOSIO(SPI Ch0) 01 : nWR 10 : Reserved      11 : PA1 ISP Mode에서 EDIN 핀으로 설정된다.	11
1 : 0	RW	PA0 : PA0 Port Selection bit 00 : MISO0(SPI Ch0) 01 : IO8 10 : Reserved      11 : PA0 ISP Mode에서 EDOUT 핀으로 설정된다.	11

**PIN MUX Select 1 Register (P1MUX)**

Address : 0x0300\_0024h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 16	R	Reserved	-
15 : 14	RW	PB7 : PB7 Port Selection bit 00 : TMO7            01 : CAP7 10 : INT7A           11 : PB7	11
13 : 12	RW	PB6 : PB6 Port Selection bit 00 : TMO6            01 : CAP6 10 : INT6A           11 : PB6	11
11 : 10	RW	PB5 : PB5 Port Selection bit 00 : TMO5            01 : CAP5 10 : INT5A           11 : PB5	11
9 : 8	RW	PB4 : PB4 Port Selection bit 00 : TMO4            01 : CAP4 10 : INT4A           11 : PB4	11
7 : 6	RW	PB3 : PB3 Port Selection bit 00 : TMO3            01 : CAP3 10 : INT3A           11 : PB3	11
5 : 4	RW	PB2 : PB2 Port Selection bit 00 : TMO2            01 : CAP2 10 : INT2A           11 : PB2	11
3 : 2	RW	PB1 : PB1 Port Selection bit 00 : TMO1            01 : CAP1 10 : INT1A           11 : PB1	11
1 : 0	RW	PB0 : PB0 Port Selection bit 00 : TMO0            01 : CAP0 10 : INT0A           11 : PB0	11

\*\*\* External Interrupt 는 INTxA 와 INTxB 가 AND 된 조건으로 입력된다. INTxA 와 INTxB 중 PIN MUX에 의해 선택되지 않은 신호는 내부적으로 “High”로 정의된다.

**PIN MUX Select 2 Register (P2MUX)**

Address : 0x0300\_0028h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 16	R	Reserved	-
15 : 14	RW	PC7 : PC7 Port Selection bit 00 : SDI(I2S)      01 : TXD3 10 : INT7B          11 : PC7	11
13 : 12	RW	PC6 : PC6 Port Selection bit 00 : SDO(I2S)      01 : RXD3 10 : INT6B          11 : PC6	11
11 : 10	RW	PC5 : PC5 Port Selection bit 00 : SCLK(I2S)      01 : TXD2 10 : INT5B          11 : PC5	11
9 : 8	RW	PC4 : PC4 Port Selection bit 00 : SLRCK(I2S)    01 : RXD2 10 : INT4B          11 : PC4	11
7 : 6	RW	PC3 : PC3 Port Selection bit 00 : TXD1            01 : Reserved 10 : INT1B          11 : PC3	11
5 : 4	RW	PC2 : PC2 Port Selection bit 00 : RXD1            01: Reserved 10 : INT0B          11: PC2	11
3 : 2	RW	PC1 : PC1 Port Selection bit 00 : TXD0            01 : Reserved 10 : Reserved       11 : PC1	11
1 : 0	RW	PC0 : PC0 Port Selection bit 00 : RXD0            01 : Reserved 10 : Reserved       11 : PC0	11

**PIN MUX Select 3 Register (P3MUX)**

Address : 0x0300\_002Ch

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 16	R	Reserved	-
15 : 14	RW	PD7 : PD7 Port Selection bit 00 : IO0                      01 : nSS1(SPI Ch1) 10 : Reserved                11 : PD7	11
13 : 12	RW	PD6 : PD6 Port Selection bit 00 : IO1                      01 : SCK1(SPI Ch1) 10 : Reserved                11 : PD6	11
11 : 10	RW	PD5 : PD5 Port Selection bit 00 : IO2                      01 : MOSI1(SPI Ch1) 10 : Reserved                11 : PD5	11
9 : 8	RW	PD4 : PD4 Port Selection bit 00 : IO3                      01 : MISO1(SPI Ch1) 10 : Reserved                11 : PD4	11
7 : 6	RW	PD3 : PD3 Port Selection bit 00 : IO4                      01 : Reserved 10 : Reserved                11 : PD3	11
5 : 4	RW	PD2 : PD2 Port Selection bit 00 : IO5                      01: Reserved 10 : Reserved                11: PD2	11
3 : 2	RW	PD1 : PD1 Port Selection bit 00 : IO6                      01 : Reserved 10 : Reserved                11 : PD1	11
1 : 0	RW	PD0 : PD0 Port Selection bit 00 : IO7                      01 : Reserved 10 : Reserved                11 : PD0	11

**PIN MUX Select 4 Register (P4MUX)**

Address : 0x0300\_0030h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 6	R	Reserved	-
5 : 4	RW	PE2 : PE2 Port Selection bit 00 : Reserved                01: IO15 10 : Reserved                11: PE2	11
3 : 2	RW	PE1 : PE1 Port Selection bit 00 : Reserved                01 : IO14 10 : Reserved                11 : PE1	11
1 : 0	RW	PE0 : PE0 Port Selection bit 00 : CODEC_CLK(I2S) 01 : IO13 10 : Reserved                11 : PE0	11

## 7.2 I/O Ports

I/O Ports는 8-bit으로 구성된 4개의 블록과 3-bit으로 구성된 1개의 블록으로 구성되어 총 35개의 IN/OUT Ports를 제공한다. 각 Ports는 레지스터 설정으로 쉽게 구성될 수 있으며, 다양한 입출력 응용과 시스템 구성에 사용된다.

I/O Ports는 PxODIR 레지스터를 통해 각 Port별로 Output mode로 설정되고 또한 PxIDIR 레지스터에 의해 각 Port별로 Input mode로 설정된다. 각 Port의 설정상태는 PxDIR 레지스터를 통해 확인할 수 있다. PxODIR 레지스터와 PxIDIR 레지스터 설정시 “1”인 비트만 해당 동작으로 설정되고, “0”인 비트는 어떠한 영향을 미치지 못한다.

I/O Ports의 출력 레벨은 Output mode로 설정된 상태에서 PxOHIGH 레지스터를 통해 High Level로 설정되고, PxOLOW 레지스터를 통해 Low Level로 설정된다. Output level의 설정상태는 PxOLEV 레지스터를 통해 확인할 수 있다.

I/O Ports의 입력 레벨은 PxILEV 레지스터를 통해 확인할 수 있다. 각 Ports에는 Pull-up 저항이 내부적으로 연결되어 있어 입력에 대한 unknown 상태를 방지 하였다.

Table 7-1 Internal Pull-up Characteristics

<i>Parameter</i>	<i>Min</i>	<i>Typ</i>	<i>Max</i>	<i>Unit</i>
Pull-up resistance	90	120	195	K $\Omega$

## 7.2.1 I/O Ports Registers

### Port Direction Registers ( PxDIR )

Address : 0x0300\_0A00h / 0x0300\_0A20h / 0x0300\_0A40h/ 0x0300\_0A60h / 0x0300\_0A80h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7 : 0	R	PxyDIR : Pxy Direction bit 0 : Input                      1 : Output	0x00h

\*\*\* 현재 설정된 Port의 Direction 을 확인할 수 있다.

### Port Direction Output Mode Setting Registers ( PxODIR )

Address : 0x0300\_0A00h / 0x0300\_0A20h / 0x0300\_0A40h/ 0x0300\_0A60h / 0x0300\_0A80h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	W	Px7ODIR : Px7 Direction Output Mode Setting bit	0
6	W	Px6ODIR : Px6 Direction Output Mode Setting bit	0
5	W	Px5ODIR : Px5 Direction Output Mode Setting bit	0
4	W	Px4ODIR : Px4 Direction Output Mode Setting bit	0
3	W	Px3ODIR : Px3 Direction Output Mode Setting bit	0
2	W	Px2ODIR : Px2 Direction Output Mode Setting bit	0
1	W	Px1ODIR : Px1 Direction Output Mode Setting bit	0
0	W	Px0ODIR : Px0 Direction Output Mode Setting bit	0

\*\*\* Port Direction Output Mode Setting bit

0 : No effect

1 : Set to output mode the corresponding bit in the PxDIR registers

### Port Direction Input Mode Setting Registers ( PxIDIR )

Address : 0x0300\_0A04h / 0x0300\_0A24h / 0x0300\_0A44h/ 0x0300\_0A64h / 0x0300\_0A84h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	W	Px7IDIR : Px7 Direction Input Mode Setting bit	0
6	W	Px6IDIR : Px6 Direction Input Mode Setting bit	0
5	W	Px5IDIR : Px5 Direction Input Mode Setting bit	0
4	W	Px4IDIR : Px4 Direction Input Mode Setting bit	0
3	W	Px3IDIR : Px3 Direction Input Mode Setting bit	0
2	W	Px2IDIR : Px2 Direction Input Mode Setting bit	0
1	W	Px1IDIR : Px1 Direction Input Mode Setting bit	0
0	W	Px0IDIR : Px0 Direction Input Mode Setting bit	0

\*\*\* Port Direction Input Mode Setting bit

0 : No effect

1 : Set to input mode the corresponding bit in the PxDIR registers

**Port Output Data Level Registers ( PxOLEV )**

Address : 0x0300\_0A08h / 0x0300\_0A28h / 0x0300\_0A48h/ 0x0300\_0A68h / 0x0300\_0A88h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7 : 0	R	PxyOLEV : Pxy Output Level bit 0 : Low Level            1 : High Level	0x00h

\*\*\* 현재 설정된 Port의 Output Level 을 확인할 수 있다.

**Port Output Data High Level Setting Registers ( PxOHIGH )**

Address : 0x0300\_0A08h / 0x0300\_0A28h / 0x0300\_0A48h/ 0x0300\_0A68h / 0x0300\_0A88h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	W	Px7OH : Px7 Output Data High Level Setting bit	0
6	W	Px6OH : Px6 Output Data High Level Setting bit	0
5	W	Px5OH : Px5 Output Data High Level Setting bit	0
4	W	Px4OH : Px4 Output Data High Level Setting bit	0
3	W	Px3OH : Px3 Output Data High Level Setting bit	0
2	W	Px2OH : Px2 Output Data High Level Setting bit	0
1	W	Px1OH : Px1 Output Data High Level Setting bit	0
0	W	Px0OH : Px0 Output Data High Level Setting bit	0

\*\*\* Port Output Data High Level Setting bit

0 : No effect

1 : Set to high level output data the corresponding bit in the PxOLEV registers

**Port Output Data Low Level Setting Registers ( PxOLOW )**Address : 0x0300\_0A0Ch / 0x0300\_0A2Ch / 0x0300\_0A4Ch  
/ 0x0300\_0A6Ch / 0x0300\_0A8Ch

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	W	Px7OL : Px7 Output Data Low Level Setting bit	0
6	W	Px6OL : Px6 Output Data Low Level Setting bit	0
5	W	Px5OL : Px5 Output Data Low Level Setting bit	0
4	W	Px4OL : Px4 Output Data Low Level Setting bit	0
3	W	Px3OL : Px3 Output Data Low Level Setting bit	0
2	W	Px2OL : Px2 Output Data Low Level Setting bit	0
1	W	Px1OL : Px1 Output Data Low Level Setting bit	0
0	W	Px0OL : Px0 Output Data Low Level Setting bit	0

\*\*\* Port Output Data Low Level Setting bit

0 : No effect

1 : Set to low level output data the corresponding bit in the PxOLEV registers



**Port Input Data Level Registers ( PxILEV )**

Address : 0x0300\_0A10h / 0x0300\_0A30h / 0x0300\_0A50h  
 / 0x0300\_0A70h / 0x0300\_0A90h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7	R	Px7ILEV : Px7 Input Level bit 0 : Low Level      1 : High Level	Pull-up
6	R	Px6ILEV : Px6 Input Level bit 0 : Low Level      1 : High Level	Pull-up
5	R	Px5ILEV : Px5 Input Level bit 0 : Low Level      1 : High Level	Pull-up
4	R	Px4ILEV : Px4 Input Level bit 0 : Low Level      1 : High Level	Pull-up
3	R	Px3ILEV : Px3 Input Level bit 0 : Low Level      1 : High Level	Pull-up
2	R	Px2ILEV : Px2 Input Level bit 0 : Low Level      1 : High Level	Pull-up
1	R	Px1ILEV : Px1 Input Level bit 0 : Low Level      1 : High Level	Pull-up
0	R	Px0ILEV : Px0 Input Level bit 0 : Low Level      1 : High Level	Pull-up

\*\*\* 현재 Port의 Input Level 을 확인할 수 있다.

## 8 INTERRUPTS

TG471는 32개 채널의 인터럽트 입력을 가지며, 이 입력들은 Timer, SPI, TWI, UART 등과 같은 내부 장치에서 발생하는 24개의 인터럽트와 외부 8개의 인터럽트로 구성된다.

### *Interrupt Features*

- 32 채널의 인터럽트 ( 8 채널의 외부 인터럽트와 24채널의 내부 인터럽트 )
- 외부 인터럽트에 대한 동작 조건 설정 (5가지)
- 내부 인터럽트에 대한 동작 조건 설정 (2가지)
- 채널 별 Mask/Unmask 기능
- 채널 별 Enable 기능
- 개별적으로 프로그램 가능한 인터럽트 우선순위
- 다양한 Interrupt vector table 지원

Interrupt의 처리는 다음과 같은 과정을 통하여 이루어진다.

1. Interrupt Resource가 Interrupt Controller에 인가
2. Interrupt Enable Register에 의해 선별된 후, Interrupt Pending Register에 저장
3. Interrupt Priority를 판단하여 MCU에 인터럽트를 요청
4. MCU는 Interrupt 요청을 받아들이기 위해 Machine Cycle 에 진입하여, Interrupt Controller의 special machine cycle register를 읽어서 Interrupt Controller 에 Interrupt Acknowledge 상태를 알림
5. Interrupt Controller 는 Interrupt Vector Address를 준비
6. MCU는 interrupt vector address 영역을 읽어서 해당 Interrupt Service Routine으로 진입
7. Interrupt Service Routine의 기능 수행
8. Interrupt Pending Clear Value 쓰기 후, Interrupt Service Routine 을 빠져 나옴

## 8.1 Interrupt Vector and Priority

인터럽트 우선순위는 INT0가 가장 높다. 인터럽트 벡터 주소는 SE3208 MCU가 32bit Addressing을 하기 때문에 각각 4bytes의 크기를 가진다.

HVA(High 22bit Interrupt Vector Address)는 인터럽트 벡터 주소의 상위 22bit으로 인터럽트 벡터 주소를 변경할 때 사용되며, 인터럽트 벡터 테이블을 다양하게 가질 수 있도록 한다.

Table 8-1 Interrupt Vector & Priority

<i>Vector No.</i>	<i>Description</i>	<i>Vector Address</i>
0x3F	UART Ch3 Interrupt	HVA + 0x000000FC
0x3E	SPI Ch1 Interrupt	HVA + 0x000000F8
0x3D	Timer 7 Interrupt	HVA + 0x000000F4
0x3C	INT7 Interrupt	HVA + 0x000000F0
0x3B	UART Ch2 Interrupt	HVA + 0x000000EC
0x3A	ADC Interrupt	HVA + 0x000000E8
0x39	Timer 6 Interrupt	HVA + 0x000000E4
0x38	INT6 Interrupt	HVA + 0x000000E0
0x37	Output Compare 4A Interrupt	HVA + 0x000000DC
0x36	UART Ch1 Interrupt	HVA + 0x000000D8
0x35	Timer 5 Interrupt	HVA + 0x000000D4
0x34	INT5 Interrupt	HVA + 0x000000D0
0x33	Output Compare 3B Interrupt	HVA + 0x000000CC
0x32	USB Interrupt	HVA + 0x000000C8
0x31	Timer 4 Interrupt	HVA + 0x000000C4
0x30	INT4 Interrupt	HVA + 0x000000C0
0x2F	Output Compare 3A Interrupt	HVA + 0x000000BC
0x2E	TWI Interrupt	HVA + 0x000000B8
0x2D	Timer 3 Interrupt	HVA + 0x000000B4
0x2C	INT3 Interrupt	HVA + 0x000000B0
0x2B	Output Compare 2B Interrupt	HVA + 0x000000AC
0x2A	UART Ch0 Interrupt	HVA + 0x000000A8
0x29	Timer 2 Interrupt	HVA + 0x000000A4
0x28	INT2 Interrupt	HVA + 0x000000A0
0x27	Output Compare 2A Interrupt	HVA + 0x0000009C
0x26	SPI Ch0 Interrupt	HVA + 0x00000098
0x25	Timer 1 Interrupt	HVA + 0x00000094
0x24	INT1 Interrupt	HVA + 0x00000090
0x23	DMA Interrupt	HVA + 0x0000008C
0x22	I2S Interrupt	HVA + 0x00000088
0x21	Timer 0 Interrupt	HVA + 0x00000084
0x20	INT0 Interrupt (Highest Priority)	HVA + 0x00000080

### 8.2 External Interrupt (INT 0 ~ 7)

External Interrupt는 EIMR 레지스터의 설정에 의해 5가지 형태의 외부 인터럽트를 받아들인다.

- Low Level Mode에서는 External Interrupt 신호가 “Low”를 유지하는 동안에 매 System Cycle 마다 인터럽트 발생시킨다.
- High Level Mode에서는 External Interrupt 신호가 “High”를 유지하는 동안에 매 System Cycle 마다 인터럽트를 발생시킨다.
- Falling Edge Mode에서는 External Interrupt 신호가 “High->Low”로 바뀔 때 인터럽트를 발생시킨다.
- Rising Edge Mode에서는 External Interrupt 신호가 “Low->High”로 바뀔 때 인터럽트를 발생시킨다.
- Any Edge Mode에서는 External Interrupt 신호가 “High->Low” 또는 “Low-> High”로 바뀔 때 인터럽트를 발생시킨다.

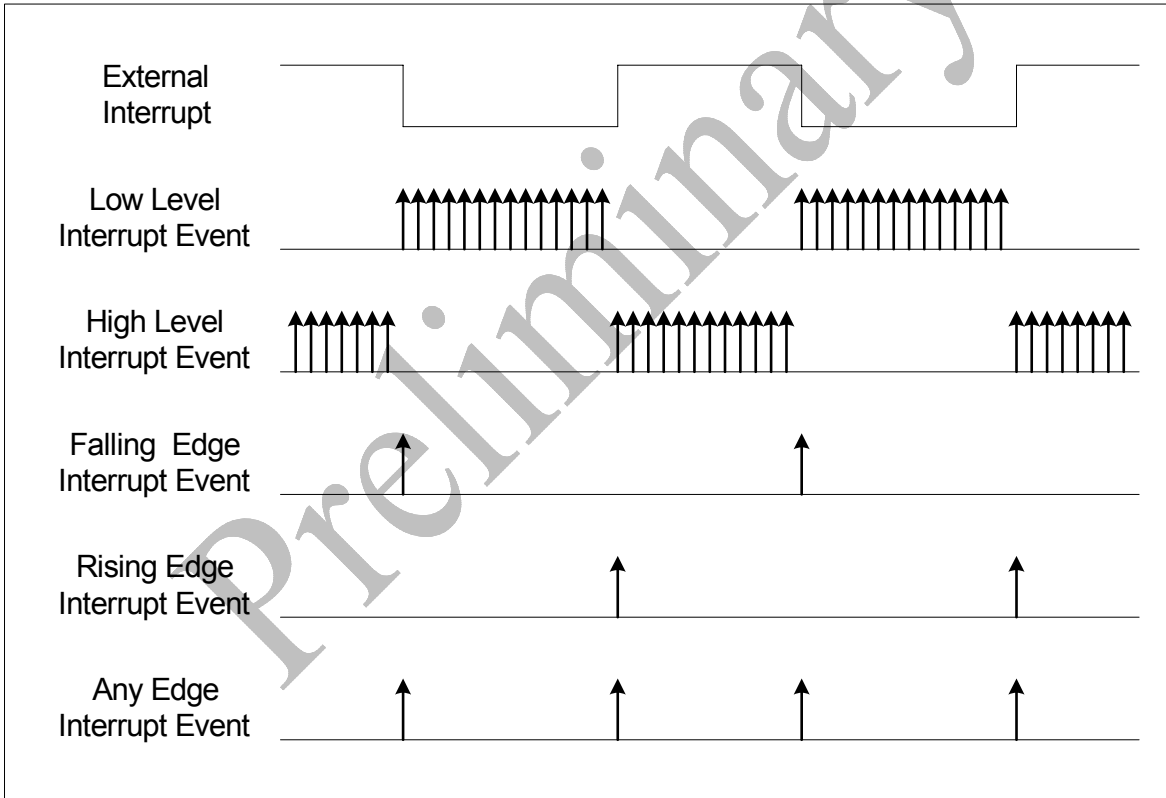


Figure 8-1 External Interrupt Mode

### 8.3 Internal Interrupt Mode

내부 인터럽트는 모두 “High Level”로 동작한다. 그러나 사용자가 특별히 “High Edge”로 인터럽트를 처리를 원할 경우에 Internal Interrupt Mode Registers를 설정할 수 있다.

### 8.4 Interrupt Pending and Interrupt Pending Clear

각 인터럽트의 발생상태는 Interrupt Pending Registers를 통해 확인할 수 있다. 일단 한번 발생한 인터럽트는 Interrupt Pending Clear Register에 의해 Clear 되기 전까지는 계속 Interrupt Pending Register에 저장된다. 또한 현재 발생한 인터럽트보다 높은 우선순위의 인터럽트가 Masking되지 않은 상태로 Interrupt Pending Registers에 저장되어 있을 경우에는 높은 우선순위의 인터럽트가 모두 Clear될 때까지 Interrupt Pending Registers에 저장되어 자신의 우선순위가 되기를 기다린다.

Interrupt Pending Registers에 저장된 인터럽트들을 Clear하기 위해서는 Interrupt Pending Clear Register를 통해 해당 인터럽트 벡터 번호 값을 Write하면 된다.

### 8.5 Interrupt Enable

Interrupt Mask Registers에 의해 Mask되어 있는 인터럽트는 Interrupt Pending Registers에 계속 저장되는데 비해, Interrupt Enable Registers(IENR)에 의해 Disable된 인터럽트는 Interrupt Pending Registers에 저장되지 않는다. 따라서 이 레지스터는 전혀 받아들이고 싶지 않은 인터럽트에 대해 Disable 하는데 사용한다.

### 8.6 Interrupt Mask Set/Clear Register

Set 이면 Request 가 Enable 되고, Clear 이면 Request 가 Disable 된다.

각 인터럽트는 Interrupt Mask Registers에 의해 해당 인터럽트에 대한 Request를 수행할 수 있다. Interrupt Mask Set bit 가 “1”일 경우에는 Interrupt Pending Register에 저장된 Interrupt를 CPU로 요청하고, Interrupt Mask Clear bit가 “1” 일 경우에는 Interrupt Pending Register에 저장되어 있는 Interrupt를 CPU로 요청하지 못한다.

Mask bit가 “0”으로 설정된 인터럽트라도 Interrupt Pending Registers(IPR)에는 저장되기 때문에 Mask bit을 “1”로 재설정하면 Interrupt Pending Registers에 저장되어 있는 인터럽트가 우선순위에 의해 인터럽트를 다시 요청한다.

## 8.7 Interrupt Control Registers

### **Interrupt Pending Clear Register (PENDCLR)**

Offset Address : 0x0300\_0500h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7 : 0	W	Interrupt Pending Register Clear Value (0x20 ~ 0x3F)	0xFFh

\*\*\* Interrupt Pending Register를 Clear 하기 위해서는 Interrupt Vector No. 값으로 clear 해야 한다. (Interrupt Vector No. 참고)

Preliminary

**External Interrupt Mode and External PIN Level Register (EINTMOD)**

Offset Address : 0x0300\_0504h

Bit	R/W	Description	Default Value
31	R	INT7ST : INT7 PIN Level	-
30 : 28	RW	INT7MOD : INT7 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010
27	R	INT6ST : INT6 PIN Level	-
26 : 24	RW	INT6MOD : INT6 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010
23	R	INT5ST : INT5 PIN Level	-
22 : 20	RW	INT5MOD : INT5 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010
19	R	INT4ST : INT4 PIN Level	-
18 : 16	RW	INT4MOD : INT4 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010
15	R	INT3ST : INT3 PIN Level	-
14 : 12	RW	INT3MOD : INT3 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010
11	R	INT2ST : INT2 PIN Level	-
10 : 8	RW	INT2MOD : INT2 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010
7	R	INT1ST : INT1 PIN Level	-
6 : 4	RW	INT1MOD : INT1 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010
3	R	INT0ST : INT0 PIN Level	-
2 : 0	RW	INT0MOD : INT0 Active State 000 : Low Level      001 : High Level 010 : Falling Edge    011 : Rising Edge 1xx : Any Edge	010

\*\*\* INTxA 와 INTxB 중 PIN MUX를 통하여 선택된 핀의 값을 External Interrupt 로 받아들인다. 만약 PIN MUX에서 INTxA 와 INTxB를 모두 선택할 경우에는 두 값의 “AND”가 INTx로 인가된다.

**Internal Interrupt Mode Register (IINTMOD)**

Offset Address : 0x0300\_0508h

<b>Bit</b>	<b>R/W</b>	<b>Description</b>	<b>Default Value</b>
31	RW	UART Ch3 Interrupt Mode bit	0
30	RW	SPI Ch1 Interrupt Mode bit	0
29	RW	Timer 7 Interrupt Mode bit	0
28	-	Reserved	-
27	RW	UART Ch2 Interrupt Mode bit	0
26	RW	ADC Interrupt Mode bit	0
25	RW	Timer 6 Interrupt Mode bit	0
24	-	Reserved	-
23	RW	Output Compare 4A Interrupt Mode bit	0
22	RW	UART Ch1 Interrupt Mode bit	0
21	RW	Timer 5 Interrupt Mode bit	0
20	-	Reserved	-
19	RW	Output Compare 3B Interrupt Mode bit	0
18	RW	USB Interrupt Mode bit	0
17	RW	Timer 4 Interrupt Mode bit	0
16	-	Reserved	-
15	RW	Output Compare 3A Interrupt Mode bit	0
14	RW	TWI Interrupt Mode bit	0
13	RW	Timer 3 Interrupt Mode bit	0
12	-	Reserved	-
11	RW	Output Compare 2B Interrupt Mode bit	0
10	RW	UART Ch0 Interrupt Mode bit	0
9	RW	Timer 2 Interrupt Mode bit	0
8	-	Reserved	-
7	RW	Output Compare 2A Interrupt Mode bit	0
6	RW	SPI Ch0 Interrupt Mode bit	0
5	RW	Timer 1 Interrupt Mode bit	0
4	-	Reserved	-
3	RW	DMA Interrupt Mode bit	0
2	RW	I2S Interrupt Mode bit	0
1	RW	Timer 0 Interrupt Mode bit	0
0	-	Reserved	-

\*\*\* Internal Interrupt Mode bit

0 : Level Mode

1 : Edge Mode



**Interrupt Pending Register (INTPEND)**

Offset Address : 0x0300\_050Ch

<b>Bit</b>	<b>R/W</b>	<b>Description</b>	<b>Default Value</b>
31	R	UART Ch3 Interrupt Pending bit	0
30	R	SPI Ch1 Interrupt Pending bit	0
29	R	Timer 7 Interrupt Pending bit	0
28	R	INT7 Interrupt Pending bit	0
27	R	UART Ch2 Interrupt Pending bit	0
26	R	ADC Interrupt Pending bit	0
25	R	Timer 6 Interrupt Pending bit	0
24	R	INT6 Interrupt Pending bit	0
23	R	Output Compare 4A Interrupt Pending bit	0
22	R	UART Ch1 Interrupt Pending bit	0
21	R	Timer 5 Interrupt Pending bit	0
20	R	INT5 Interrupt Pending bit	0
19	R	Output Compare 3B Interrupt Pending bit	0
18	R	USB Interrupt Pending bit	0
17	R	Timer 4 Interrupt Pending bit	0
16	R	INT4 Interrupt Pending bit	0
15	R	Output Compare 3A Interrupt Pending bit	0
14	R	TWI Interrupt Pending bit	0
13	R	Timer 3 Interrupt Pending bit	0
12	R	INT3 Interrupt Pending bit	0
11	R	Output Compare 2B Interrupt Pending bit	0
10	R	UART Ch0 Interrupt Pending bit	0
9	R	Timer 2 Interrupt Pending bit	0
8	R	INT2 Interrupt Pending bit	0
7	R	Output Compare 2A Interrupt Pending bit	0
6	R	SPI Ch0 Interrupt Pending bit	0
5	R	Timer 1 Interrupt Pending bit	0
4	R	INT1 Interrupt Pending bit	0
3	R	DMA Interrupt Pending bit	0
2	R	I2S Interrupt Pending bit	0
1	R	Timer 0 Interrupt Pending bit	0
0	R	INT0 Interrupt Pending bit	0

\*\*\* Interrupt Pending Register의 각 비트의 값은 해당 인터럽트가 발생하였음을 나타낸다. Interrupt Pending Register의 값은 Interrupt Pending Clear 레지스터에 의해 Clear된다. 일반적으로 해당 Interrupt 가 끝날 때 Clear한다.

**Interrupt Enable Register (INTEN)**

Offset Address : 0x0300\_0510h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31	RW	UART Ch3 Enable bit	0
30	RW	SPI Ch1 Enable bit	0
29	RW	Timer 7 Enable bit	0
28	RW	INT7 Enable bit	0
27	RW	UART Ch2 Enable bit	0
26	RW	ADC Enable bit	0
25	RW	Timer 6 Enable bit	0
24	RW	INT6 Enable bit	0
23	RW	Output Compare 4A Enable bit	0
22	RW	UART Ch1 Enable bit	0
21	RW	Timer 5 Enable bit	0
20	RW	INT5 Enable bit	0
19	RW	Output Compare 3B Enable bit	0
18	RW	USB Enable bit	0
17	RW	Timer 4 Enable bit	0
16	RW	INT4 Enable bit	0
15	RW	Output Compare 3A Enable bit	0
14	RW	TWI Enable bit	0
13	RW	Timer 3 Enable bit	0
12	RW	INT3 Enable bit	0
11	RW	Output Compare 2B Enable bit	0
10	RW	UART Ch0 Enable bit	0
9	RW	Timer 2 Enable bit	0
8	RW	INT2 Enable bit	0
7	RW	Output Compare 2A Enable bit	0
6	RW	SPI Ch0 Enable bit	0
5	RW	Timer 1 Enable bit	0
4	RW	INT1 Enable bit	0
3	RW	DMA Enable bit	0
2	RW	I2S Enable bit	0
1	RW	Timer 0 Enable bit	0
0	RW	INT0 Enable bit	0

\*\*\* Interrupt Enable bit

0 : Interrupt Disable

1 : Interrupt Enable

**Interrupt Mask Register (INTMASK)**

Offset Address : 0x0300\_0514h

Bit	R/W	Description	Default Value
31 : 0	R	Interrupt Mode Register	0x0000_0000h

\*\*\* 모든 Mask bit의 상태를 확인할 수 있다.

**Interrupt Mask Set Register (MASKSET)**

Offset Address : 0x0300\_0514h

Bit	R/W	Description	Default Value
31	W	UART Ch3 Interrupt Request Set bit	0
30	W	SPI Ch1 Interrupt Request Set bit	0
29	W	Timer 7 Interrupt Request Set bit	0
28	W	INT7 Interrupt Request Set bit	0
27	W	UART Ch2 Interrupt Request Set bit	0
26	W	ADC Interrupt Request Set bit	0
25	W	Timer 6 Interrupt Request Set bit	0
24	W	INT6 Interrupt Request Set bit	0
23	W	Output Compare 4A Interrupt Request Set bit	0
22	W	UART Ch1 Interrupt Request Set bit	0
21	W	Timer 5 Interrupt Request Set bit	0
20	W	INT5 Interrupt Request Set bit	0
19	W	Output Compare 3B Interrupt Request Set bit	0
18	W	USB Interrupt Request Set bit	0
17	W	Timer 4 Interrupt Request Set bit	0
16	W	INT4 Interrupt Request Set bit	0
15	W	Output Compare 3A Interrupt Request Set bit	0
14	W	TWI Interrupt Request Set bit	0
13	W	Timer 3 Interrupt Request Set bit	0
12	W	INT3 Interrupt Request Set bit	0
11	W	Output Compare 2B Interrupt Request Set bit	0
10	W	UART Ch0 Interrupt Request Set bit	0
9	W	Timer 2 Interrupt Request Set bit	0
8	W	INT2 Interrupt Request Set bit	0
7	W	Output Compare 2A Interrupt Request Set bit	0
6	W	SPI Ch0 Interrupt Request Set bit	0
5	W	Timer 1 Interrupt Request Set bit	0
4	W	INT1 Interrupt Request Set bit	0
3	W	DMA Interrupt Request Set bit	0
2	W	I2S Interrupt Request Set bit	0
1	W	Timer 0 Interrupt Request Set bit	0
0	W	INT0 Interrupt Request Set bit	0

\*\*\* Interrupt Request Set bit

0 : No Effect interrupt Mask.

1 : Pending interrupt is allowed to become active (interrupts sent to CPU).

**Interrupt Mask Clear Register (MASKCLR)**

Offset Address : 0x0300\_0518h

<b>Bit</b>	<b>R/W</b>	<b>Description</b>	<b>Default Value</b>
31	W	UART Ch3 Interrupt Request Clear bit	0
30	W	SPI Ch1 Interrupt Request Clear bit	0
29	W	Timer 7 Interrupt Request Clear bit	0
28	W	INT7 Interrupt Request Clear bit	0
27	W	UART Ch2 Interrupt Request Clear bit	0
26	W	ADC Interrupt Request Clear bit	0
25	W	Timer 6 Interrupt Request Clear bit	0
24	W	INT6 Interrupt Request Clear bit	0
23	W	Output Compare 4A Interrupt Request Clear bit	0
22	W	UART Ch1 Interrupt Request Clear bit	0
21	W	Timer 5 Interrupt Request Clear bit	0
20	W	INT5 Interrupt Request Clear bit	0
19	W	Output Compare 3B Interrupt Request Clear bit	0
18	W	USB Interrupt Request Clear bit	0
17	W	Timer 4 Interrupt Request Clear bit	0
16	W	INT4 Interrupt Request Clear bit	0
15	W	Output Compare 3A Interrupt Request Clear bit	0
14	W	TWI Interrupt Request Clear bit	0
13	W	Timer 3 Interrupt Request Clear bit	0
12	W	INT3 Interrupt Request Clear bit	0
11	W	Output Compare 2B Interrupt Request Clear bit	0
10	W	UART Ch0 Interrupt Request Clear bit	0
9	W	Timer 2 Interrupt Request Clear bit	0
8	W	INT2 Interrupt Request Clear bit	0
7	W	Output Compare 2A Interrupt Request Clear bit	0
6	W	SPI Ch0 Interrupt Request Clear bit	0
5	W	Timer 1 Interrupt Request Clear bit	0
4	W	INT1 Interrupt Request Clear bit	0
3	W	DMA Interrupt Request Clear bit	0
2	W	I2S Interrupt Request Clear bit	0
1	W	Timer 0 Interrupt Request Clear bit	0
0	W	INT0 Interrupt Request Clear bit	0

\*\*\* Interrupt Request Clear bit

0 : No Effect Interrupt Mask.

1 : Pending interrupt is masked from becoming active (interrupts not sent to CPU).

**High 22bit Interrupt Vector Address Register (HVA)**

Offset Address : 0x0300\_051Ch

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 10	RW	Interrupt Vector Address[31:10]	0x0000_00
9 : 0	R	Reserved	-

\*\*\* 인터럽트 벡터 주소의 상위 22bit 값이다. HVA 레지스터 설정을 통하여 다양한 Interrupt Vector Table을 가질 수 있다.

Preliminary

## 9 TIMERS

TG471는 Timer/Counter, Capture, PWM, 그리고 Output Compare 기능을 가진 16bit Timer/Counter를 8채널 내장하였다.

### *Timer Features*

- 10bit Pre-scale
- 16bit Timer/Counter
- 16bit Capture
- 16bit PWM
- 16bit Output Compare
- 16bit Timer Counter Wave Out

Preliminary

### 9.1 10bit Prescaler with clock source selection

Pre-scaler는 System Clock과 External Clock 핀을 통해 외부로부터 받아들인 입력을 CLKSEL 비트를 통해 선택한 후, 10bit Pre-scaler를 통해 1/2 ~ 1/1024배 분주된 클럭을 생성하여 Timer/Counter 로 전달한다. Timer/Counter는 Pre-scaler를 통해 분주된 클럭을 선택하여 16bit Counter 를 구동한다.

Pre-scaler에서 분주되는 클럭의 정확한 위상이 필요할 경우에는 CNTCLR 비트를 통하여 Pre-scaler counter를 초기화한 후 사용한다.

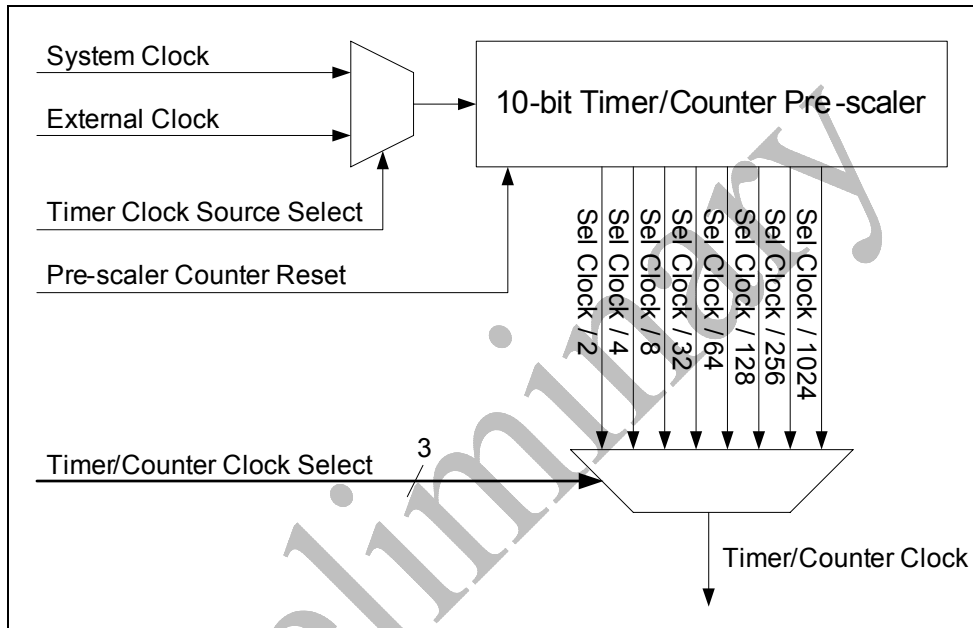


Figure 9-1 Pre-scaler Block Diagram

### 9.2 Timer/Counter

Pre-scaler에 의해 분주된 Clock을 사용하여 0x0000 의 초기값에서 매 클럭마다 카운터 값을 “1” 씩 증가하여 사용자가 설정한 Timer Counter 레지스터 값에 도달하면 다시 0x0000 이 되면서 인터럽트를 발생한다.

Pre-scaler의 입력으로 32.768KHz를 인가하면 Real Time Counter로 동작한다.

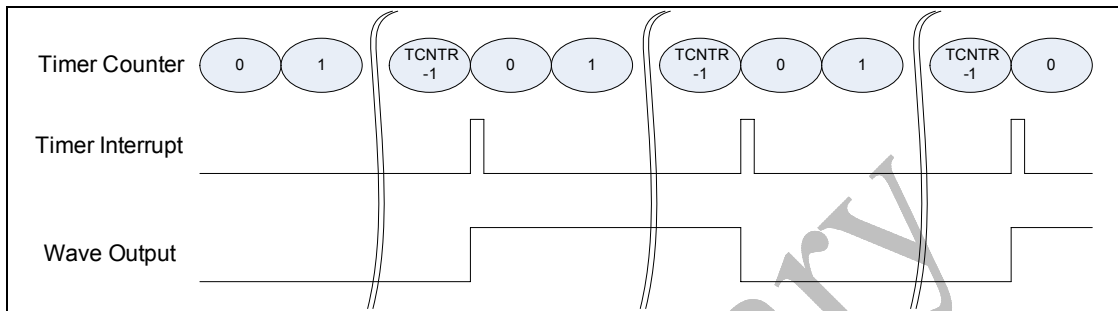


Figure 9-2 Timer Operation

Timer 주기는 선택된 클럭, Pre-scaler 그리고 Timer Counter 에 의해 결정된다.

$$\begin{aligned}
 \text{Timer Period} &= \frac{1}{\text{Clock Source Frequency}} \times \frac{1}{\text{Pre-scaler Factor}} \times (\text{TMCNT}) \text{ [sec]} && \{ \text{TMCNT} \geq 3 \} \\
 \text{Timer Period} &= \frac{1}{\text{Clock Source Frequency}} \times \frac{1}{\text{Pre-scaler Factor}} \times (\text{TMCNT} + 1) \text{ [sec]} && \{ \text{TMCNT} < 3 \}
 \end{aligned}$$

Timer Period Example :

- Clock Source Frequency : 12MHz System Clock
  - Pre-scaler Factor : 1 / 1024
  - Timer Counter Value (TMCNT) : 1000
- => 1/12MHz X 1024 X 1000 = 85.333msec = 11.718Hz

Timer Counter로 동작시키기 위하여 설정되어야 하는 레지스터는 다음과 같다.

- TPCON : Pre-scaler의 클럭 입력을 결정하고, 필요에 따라 Pre-scaler를 clear 한다.
- TMCON's TMOD : Timer Counter 모드를 설정한다.
- TMCON's WAVE : Timer Counter의 주기로 생성된 클럭을 출력유무를 결정한다.
- TMCON's PFSEL : Timer Counter에서 사용할 Clock을 결정한다.
- TMCON's TMEN : Timer Counter를 Enable 한다.
- TMCNT : Timer Counter의 최대 Counter 값을 결정한다.

Timer Counter는 다음 순서로 설정하여 동작시킨다.

- TPCON 설정
- TMCNT 설정
- TMCON 설정
- 필요에 따라 TPCON's CNTCLR 비트 설정



Timer Mode에서 WAVE를 Enable 하면 Timer 주기보다 2배 느린 클럭이 생성되어 TMO로 출력된다.

Preliminary

### 9.3 Pulse Width Modulation (PWM)

PWM은 programmable한 duty와 주기의 펄스 신호를 출력하기 위한 Device이다.

Pre-scaler에서 설정한 Clock을 통해 동작하며 PWM Period 레지스터 값의 주기로 카운트를 반복하면서 사용자가 설정한 형태의 파형을 출력한다.

PWM의 출력 펄스는 16비트 카운터의 값이 PWM Duty, PWM Period 레지스터 값에 이를 때마다 레벨이 반전되어 출력 파형이 만들어진다. PWM의 출력횟수는 PWM Pulse Number 레지스터에 의해 결정되며 펄스의 출력횟수에 다르면 PWM Interrupt를 발생한다. 그러나 PWM Interrupt가 발생하더라도 별도의 설정이 없을 경우에는 PWM 출력이 계속 생성된다. 따라서 PWM Pulse를 멈추기 위해서는 Timer Interrupt 에서 PWM을 Disable 해야 한다.

$$PWM \text{ Pulse Period} = \frac{1}{\text{Clock Source Frequency}} \times \frac{1}{\text{Pre-scaler Factor}} \times (TMCNT) \text{ [sec]} \quad \{TMCNT \geq 3\}$$

$$PWM \text{ Pulse Period} = \frac{1}{\text{Clock Source Frequency}} \times \frac{1}{\text{Pre-scaler Factor}} \times (TMCNT + 1) \text{ [sec]} \quad \{TMCNT < 3\}$$

PWM Period Example :

- Clock Source Frequency : 12MHz System Clock
  - Pre-scaler Factor : 1 / 1024
  - PWM Period Value(TMCNT) : 10
  - PWM Duty Value : 6
- => 1/12MHz X 1024 X 10 = 0.853msec = 1.171KHz

PWM으로 동작시키기 위하여 설정되어야 하는 레지스터는 다음과 같다.

- TPCON : Pre-scaler의 클럭 입력을 결정하고, 필요에 따라 Pre-scaler를 clear 한다.

- TMCON's TMOD : PWM 모드로 설정한다.
- TMCON's PWML : PWM 출력의 시작 레벨을 결정한다.
- TMCON's PFSEL : PWM에서 사용할 Clock을 결정한다.
- TMCON's TMEN : PWM을 Enable 한다.
- TMCNT : PWM의 주기를 결정한다.
- OCA : PWM의 Duty를 결정한다.
- OCB : PWM의 Pulse 횟수를 결정한다. PWM Pulse의 횟수가 이

레지스터 값에 도달하면 Timer Interrupt가 발생시킨다. 그러나 PWM Pulse는 중지되지 않고 계속 생성된다.

PWM은 다음 순서로 설정하여 동작시킨다.

- TPCON 설정
- TMCNT 설정
- OCA 설정
- OCB 설정
- 필요에 따라 TPCON's CNTCLR 설정

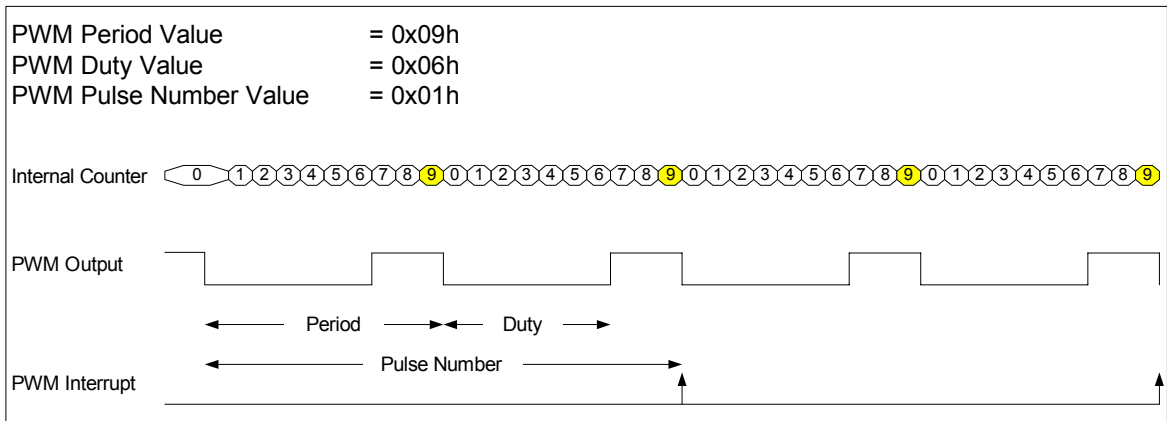


Figure 9-3 PWM Operation

Preliminary

### 9.4 Capture

Capture 기능은 Pre-scale에서 설정한 Clock을 기준으로 하여 외부 입력을 측정한다.

외부입력은 Low/High Pulse, Only Low Pulse, Only High Pulse, Falling to Falling Period, Rising to Rising Period 의 5가지 형태의 펄스 주기를 측정할 수 있다.

Capture mode로 Timer를 Enable할 때는 첫 번째로 Capture하는 값은 신호가 변하는 중간의 값이기 때문에 무시하여야 한다.

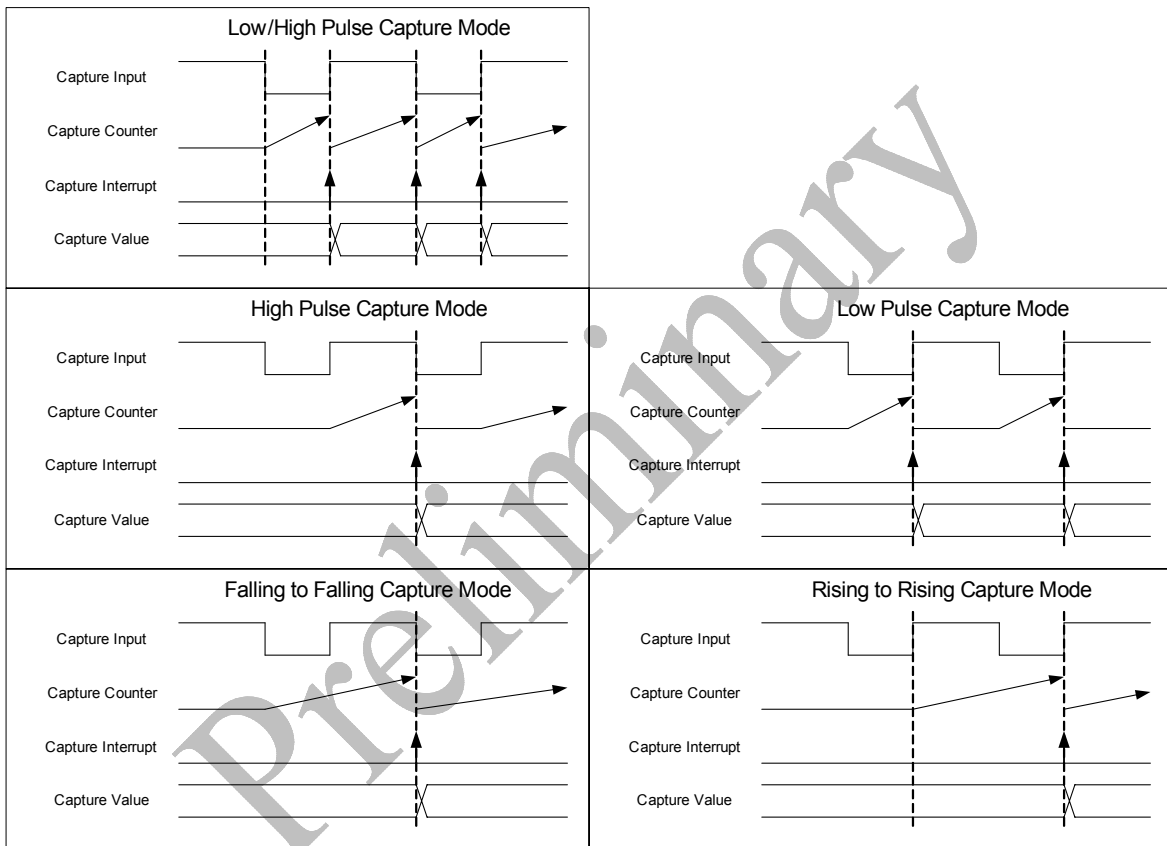


Figure 9-4 Capture Mode Operation

Capture 주기는 다음과 같이 측정된다.

$$Capture\ Signal\ Width\ Time = \frac{1}{Clock\ Source\ Frequency} \times \frac{1}{Pre - scaler\ Factor} \times (OCA + 1) [sec]$$

Capture Time Example :

- Clock Source Frequency : 12MHz System Clock
  - Pre-scaler Factor : 1 / 1024
  - Capture Value (OCA) : 9
- => 1/12MHz X 1024 X 10 = 0.853msec

Capture 모드로 동작시키기 위하여 설정되어야 하는 레지스터는 다음과 같다.

- TPCON : Pre-scaler의 클럭 입력을 결정하고, 필요에 따라 Pre-scaler를 clear 한다.
- TMCON's TMOD : Capture 모드로 설정한다.
- TMCON's CAPMOD : Capture Pulse 형태를 결정한다.
- TMCON's PFSEL : Capture에서 사용할 Clock을 결정한다.
- TMCON's TMEN : Capture를 Enable 한다.

Capture는 다음 순서로 설정하여 동작시킨다.

- TPCON 설정
- TMCON 설정
- 필요에 따라 TPCON's CNTCLR 설정
- OCA 을 읽어서 Capture 주기 확인
- TMCON's OVST를 읽어서 Overflow 유무 확인

Preliminary

### 9.5 Output Compare Mode

Timer/Counter는 각 채널 당 2개의 Output Compare 레지스터를 가지고 있기 때문에 Timer의 Counter 값이 Overflow 되기 전에 2번의 Output Compare를 수행할 수 있다.

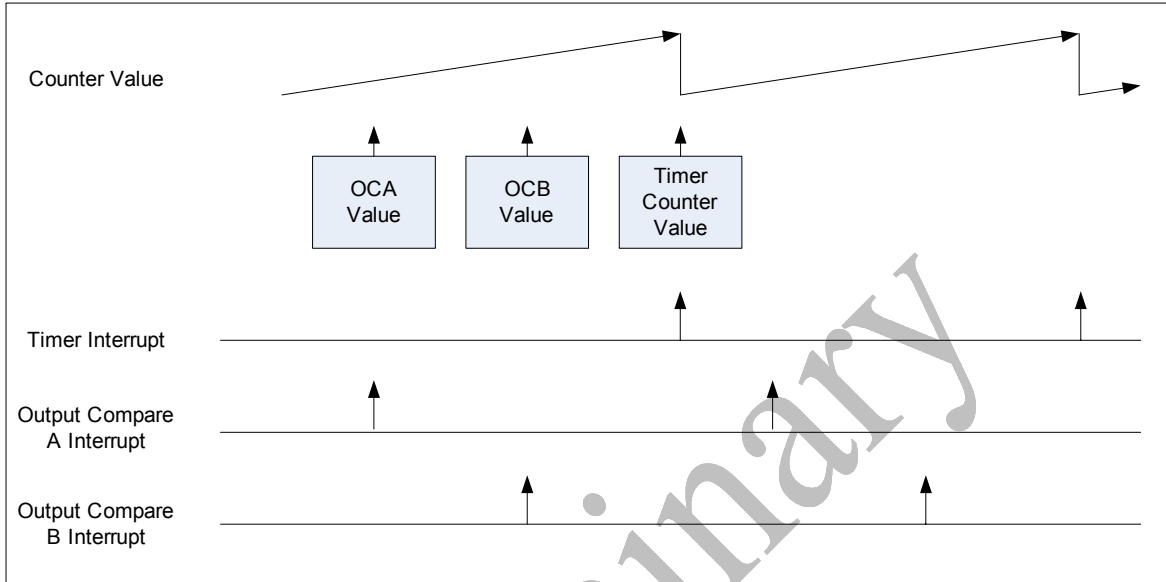


Figure 9-5 Timing Diagram of Output Compare Operation

Preliminary

## 9.6 Timer Registers

### Timer Pre-scale Control Registers ( TPxCON )

Address : 0x0300\_0600h / 0x0300\_0620h / 0x0300\_0640h / 0x0300\_0660h /  
0x0300\_0680h / 0x0300\_06A0h / 0x0300\_06C0h / 0x0300\_06E0h

Bit	R/W	Description	Default Value
31 : 2	R	Reserved	-
1	R/W	CNTCLR : Pre-scale Counter and Timer Counter Reset When this bit is "1", the Timer Pre-scale and Counter will be reset.	0
0	R/W	CLKSEL : Pre-scale Clock Selection 0 : System clock      1 : CAP[7:0]	0

\*\*\* CAP[7:0]은 Timer 채널 별로 할당되어 있다.

**Timer Control Registers ( TMxCON)**

Address : 0x0300\_0604h / 0x0300\_0624h / 0x0300\_0644h / 0x0300\_0664h /  
0x0300\_0684h / 0x0300\_06A4h / 0x0300\_06C4h / 0x0300\_06E4h

Bit	R/W	Description	Default Value
31 : 16	R	Reserved	-
15 : 14	RW	TMOD : Timer/Counter Mode 00 : Timer                    01 : PWM 1x : Capture	00
13	R	OCEN : Output Compare Mode Enable bit 0 : Disable                    1 : Enable	0
12	R	Reserved	0
11	RW	OVST : Capture Overflow Status bit Overflow status bit를 clear하기 위해서는 “0”을 write해야 한다.	0
10 : 8	RW	CAPMOD : Capture Mode Selection 00x : Low/High Pulse Capture mode 010 : Low Pulse Capture mode 011 : High Pulse Capture mode 10x : Failing to Failing Period Capture mode 11x : Rising to Rising Period Capture mode	000
7 : 6	R	Reserved	-
5	RW	PWML : PWM Output Start Level 0 : Start Level is High 1 : Start Level is Low	0
4	RW	WAVE : Waveform Generation 0 : Disable                    1 : Enable	0
3 : 1	RW	PFSEL : Pre-scale Factor Selection 000 : clock / 2                001 : clock / 4 010 : clock / 8                011 : clock / 32 100 : clock / 64               101 : clock / 128 110 : clock / 256              111 : clock / 1024	111
0	RW	TMEN : Timer/Counter or PWM Enable 0 : Disable                    1 : Enable	0



**Timer Counter / PWM Period Registers ( TMxCNT )**

Address : 0x0300\_0608h / 0x0300\_0628h / 0x0300\_0648h / 0x0300\_0668h /  
0x0300\_0688h / 0x0300\_06A8h / 0x0300\_06C8h / 0x0300\_06E8h

Bit	R/W	Description	Default Value
31 : 16	R	Reserved	-
15 : 0	R/W	(Timer mode) - Write : Timer Counter Value - Read : Current Up-counter Value  (PWM mode) - Read/Write : PWM Period Value	0xFFFFh

**Capture Counter Registers / PWM Duty Registers / Output Compare A Registers ( TMxDUT )**

Address : 0x0300\_060Ch / 0x0300\_062Ch / 0x0300\_064Ch / 0x0300\_066Ch /  
0x0300\_068Ch / 0x0300\_06ACh / 0x0300\_06CCh / 0x0300\_06ECh

Bit	R/W	Description	Default Value
31 : 16	R	Reserved	-
15 : 0	RW	(Capture mode) - Read : Result value of counting at the sampling period  (PWM mode) - Read/Write : PWM Duty Value  (Output Compare Mode) - Read/Write : Output Compare A Value	0xFFFFh

\*\*\* PWM Duty : First Halt Duty of PWM Pulse

**PWM Pulse Count Registers / Output Compare B Registers ( TMxPUL )**

Address : 0x0300\_0610h / 0x0300\_0630h / 0x0300\_0650h / 0x0300\_0670h /  
0x0300\_0690h / 0x0300\_06B0h / 0x0300\_06D0h / 0x0300\_06F0h

Bit	R/W	Description	Default Value
31 : 16	R	Reserved	-
15 : 0	RW	(PWM mode) - Read/Write : PWM Pulse Number Value  (Output Compare Mode) - Read/Write : Output Compare B Value	0xFFFFh

## 10 SPI(SERIAL PERIPHERAL INTERFACE)

TG471에 내장된 SPI는 동기 직렬 버스를 통해 외부의 장치나 다른 MCU와 데이터 교환을 한다. 이 SPI는 모토로라 M68HC11, M68HC05와 MC68HC16 계열의 SPI와 호환을 이루어, Full duplex 3-wire 전송이나 Half duplex 2-wire를 수행할 수 있다.

고속 SPI 전송을 위해 8Bytes의 FIFO를 내장하여 Mbps 속도의 전송에서도 MCU에 부담을 주지 않고 수행할 수 있다.

TG471의 SPI는 Master Mode와 Slave Mode를 모두 지원한다.

### *SPI Features*

- 2Ch SPI
- Full duplex mode. Three-wired synchronous Transfer
- Master or Slave Operation
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability
- 8Bytes FIFO

Clock 컨트롤 회로에서 Clock의 극성 선택과 두 개의 Clock 프로토콜의 선택을 통해서 대부분의 동기 직렬 주변장치와의 호환이 이루어진다. SPI가 Master로 설정되면 소프트웨어적으로 256개의 다양한 시리얼 Clock을 만들 수 있다.

SPI는 데이터 전송동작과 데이터 수신동작이 동시에 이루어진다. 두 시리얼 데이터 라인에서 정보의 샘플링(sampling)과 쉬프팅(shifting)은 시리얼 Clock 라인에 의해 동기된다. Slave SPI 디바이스의 개별적인 선택은 Slave 선택 라인을 통해 할 수 있다. 선택되지 않은 Slave 디바이스는 SPI 버스의 동작에 영향을 주지 않는다. Master SPI 디바이스에서는 Slave 선택 라인은 다중 Master 버스 충돌을 나타내는데 사용 되어질 수 있다.

에러검출 회로는 프로세스끼리의 연결을 위해 사용된다. 전송 동작 중에 시리얼 쉬프터 레지스터에 데이터를 쓰게 되면 쓰기 충돌이 발생한다. 다중 Master 모드 실패 검출은 한 개 이상의 MCU가 동시에 버스 Master가 되려고 시도할 때 출력 드라이버를 disable 시킨다.

### 10.1 SPI Registers Summery

SPI Control Register(SPICONR) : SPI 컨트롤 레지스터는 SPI 설정에 관계된 Parameter를 포함한다. 이 레지스터는 언제든지 읽기와 쓰기가 가능하다.

SPI Baud Register(SPIBRR) : SPI baud register는 SCK를 만들기 위한 baud rate를 설정한다. .

SPI Status Register(SPISTR) : SPI 상태 레지스터는 SPI 상태 정보를 포함한다. SPI 는 이 레지스터 비트들의 값을 설정만 할 수 있다. MCU는 현재 SPI 상태를 확인하기 위해 상태 레지스터를 읽으면 된다.

SPI Data Register (SPIDR) : SPI 데이터 레지스터는 시리얼 버스에 데이터 전송과 수신을 위해 사용된다. TX data 레지스터와 RX buffer 레지스터로 구성 되어 있다. 쓰기는 바로 TX data 레지스터에 쓰여진다. 바이트 또는 워드 전송 후에 Master 디바이스와 Slave 디바이스의 SPIF 상태 비트가 설정된다.

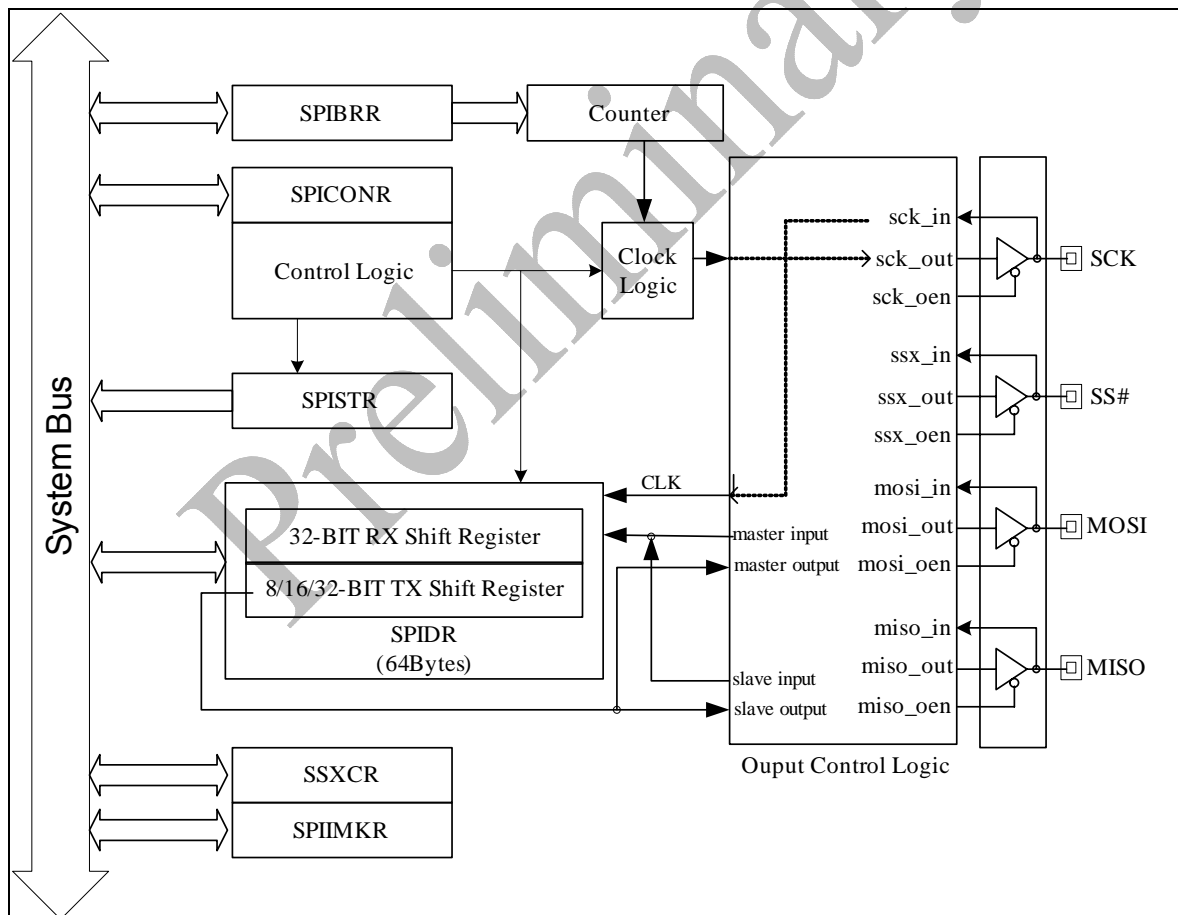


Figure 10-1 SPI Block Diagram

## 10.2 SPI Pins

SPI에는 MISO, MOSI, SCK, SSX, 네 개의 양방향 핀이 있다. SPI 컨트롤 레지스터의 WOMP 비트가 각각의 핀의 출력 동작에 대해 Open Drain 출력 이나 CMOS 출력을 결정하게 된다.

SPI 컨트롤 레지스터의 MSTR 비트에 의해 Master 또는 Slave 동작이 결정되어지고 이에 따라 핀의 동작이 결정되어진다.

Table 10-1 SPI Pin Functions

<i>Pin Name</i>	<i>Mode</i>	<i>Function</i>
Master in, slave out(MISO)	Master	Provides serial data input to the SPI
	Slave	Provides serial data output from the SPI
Master out, slave in (MOSI)	Master	Provides serial output from the SPI
	Slave	Provides serial input to the SPI
serial clock(SCK)	Master	Provides clock output from the SPI
	Slave	Provides clock input to the SPI
Slave select(SSX)	Master	Output : Selects slave devices
	Slave	Input : chip select for SPI

## 10.3 SPI Operating Modes

SPI는 Master나 Slave 모드에서 동작한다. Master 모드는 MCU가 데이터 전송을 주관할 경우에 사용된다. Slave 모드는 외부 디바이스에 의해 MCU에 시리얼 전송이 이루어지는 경우에 사용된다. 컨트롤 레지스터의 MSTR 비트에 의해 Master나 Slave 동작이 선택된다.

### *Master Mode*

SPCR의 MSTR 비트를 설정하면 Master 모드 동작이 선택된다. Master 모드에서는 시리얼 전송을 초기화 할 수 있고 외부에서의 초기화된 전송에 응답하지 않는다.

SPI를 Master 모드에서 사용하려면 다음 과정을 거쳐야 한다.

Pin MUX 레지스터에서 다음의 핀이 SPI 할당되어 있는지 확인한다. : MISO and MOSI. Master 모드에서는 MISO는 시리얼 데이터 입력으로 사용되어지고 MOSI는 시리얼 데이터 출력으로 사용되어진다. 특정한 응용 분야에 따라 하나 또는 둘 다 필요할 수 있다.

컨트롤레지스터에 BAUD, CPHA, CPOL, SIZE, MSBF, WOMP의 값을 할당한다. Master동작을 위해 MSTR 비트 설정한다. SPI를 enable하도록 SPIEN 비트를 설정한다.

Slave 디바이스를 enable 한다.

전송을 시작하기 위해 적당한 데이터를 SPI 데이터 레지스터에 쓰기를 한다.

SPI는 전송이 끝나면 상태레지스터의 SPIF 플래그를 H/W적으로 설정한다. SPIF가 인가되면 인터럽트 요청이 발생된다. SPIF가 설정되어 있는 상태에서 상태레지스터를 읽고 데이터 레지스터에 쓰기 나 읽기 동작이 일어나면 SPIF 플래그는 자동적으로 클리어 된다.

데이터 전송은 내부에서 만든 시리얼 클럭(SCK)에 동기 되어 진다. 컨트롤 레지스터의 CPHA 와 CPOL 비트들은 클럭의 위상과 극성을 컨트롤 한다. Master MCU가 MOSI 핀에서 데이터를 보내는 SCK edge와 MISO 핀을 통해 들어오는 데이터의 latch하는 SCK edge는 CPHA와 CPOL에 의해 결정된다.

### *Slave Mode*

컨트롤 레지스터의 MSTR 비트를 클리어 시키면 Slave 모드로 동작한다. Slave 모드에서는, SPI는 시리얼 전송을 초기화 할 수 없다. 전송은 외부 버스 Master에 의해 초기화 된다. Slave 모드는 특히 다중Master SPI 버스에서 사용된다. 주어진 시간에 단 하나의 디바이스만이 버스Master가 될 수 있기 때문이다.

SPI에 MISO 와 MOSI 핀이 할당되도록 PIN MUX 레지스터를 설정한다. Slave 모드에서는 시리얼 데이터 출력을 위해 MISO 핀이 사용되고, 시리얼 데이터 입력을 위해 MOSI핀이 사용된다. 특정한 응용분야에 따라서 둘 다 또는 하나의 핀만 필요하다. SCK 는 입력 시리얼 클럭이다. SSX가 인가되면 Slave로 선택된다.

데이터 전송을 위해 데이터 레지스터에 쓰기를 한다. Slave 모드에서는 SCK, MOSI, and SSX 는 입력이고 MISO는 출력이 된다.

CPHA, CPOL, SIZE, MSBF, WOMP 의 설정을 위해 컨트롤 레지스터에 값을 쓴다. MSTR 비트 클리어 함으로써 Slave 동작을 선택한다. SPIEN를 셋하여 SPI를 enable시킨다. Slave 모드의 디바이스에서는 BAUD의 값은 SPI 동작에 영향을 미치지 않는다.

SPIEN가 설정되고 MSTR이 클리어되면, SSX핀 입력의 “Low” 상태가 Slave 모드 동작을 초기화 한다. SSX 핀 은 오로지 입력으로만 사용된다.

데이터의 바이트나 워드 전송 후에 SPI는 SPIF 플래그를 셋한다. 컨트롤 레지스터의 SPIE 비트가 설정되어 있으면, SPIF가 인가되면 인터럽트요청이 발생한다.

전송은 외부에서 발생된 SCK에 동기되어 진다. CPHA와 CPOL은 Slave MCU가 MOSI 핀을 통해 들어오는 데이터를 래치하거나 MISO 핀을 통해 나가는 데이터의 클럭의 Edge를 결정한다.

### 10.4 SCK Phase and Polarity Control

컨트롤 레지스터의 두 개의 비트가 SCK의 위상과 극성을 결정한다. 클럭 극성(CPOL) 비트는 클럭의 극성 (High or Low)을 선택한다. 클럭 위상 비트 CPHA는 전송 타이밍에 영향을 주는 두 가지 전송 형태 중 하나를 선택한다. 클럭의 위상과 극성은 Master와 Slave 모두 동일하여야 한다. 어떤 경우에는 전송 사이에 위상과 극성을 바꾸어 Master 디바이스가 Slave 디바이스와 다른 조건으로 데이터를 주고 받을 수도 있다. SPI의 이러한 유연성은 거의 모든 동기 시리얼 주변장치와의 직접적인 연결을 가능하게 한다.

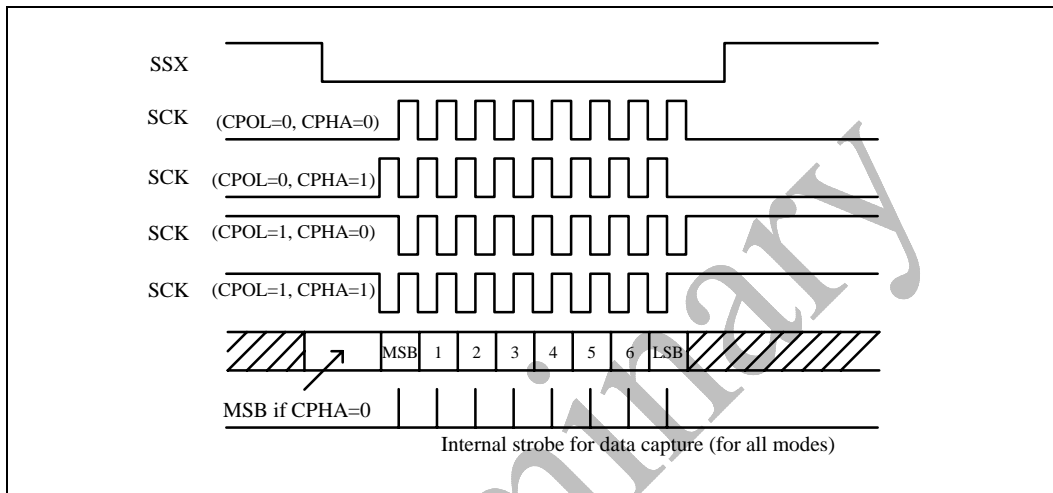


Figure 10-2 SCK Phase and Polarity

### 10.5 Data Transfer Timing

CPHA='0'이고 MSB 시작인 모드에서 1Byte 데이터 전송 타이밍도를 보여준다. SCK의 두 개의 형태의 파형을 나타나 있다. 하나는 CPOL인 '0'인 경우이고 다른 하나는 CPOL이 '1'인 경우이다. 이 타이밍도는 Master와 Slave가 SCK, MISO와 MOSI핀으로 직접 연결되어 있으므로 Master 타이밍도 또는 Slave 타이밍도로 볼 수 있다. MISO 신호는 Slave에서의 출력이고 MOSI 신호는 Master의 출력 신호이다. SSX신호는 Slave로의 칩 선택 신호이다.

Master일 때 SPDR에 데이터를 쓰면 전송이 초기화 된다. Slave는 SSX가 falling edge 일 때 전송을 초기화 한다. SCK 신호는 첫 번째 SCK cycle의 반주기까지 inactive 상태로 남아 있다. 전송 완료를 나타내는 SPIF비트는 8번째 SCK cycle의 끝에서 set된다. CPHA='0'일 때는 SSX가 low였다가 1바이트 전송 후 짧은 시간 내에 Inactive (High)된다. Slave는 SSX가 low일 때 data register에 값을 write 하면 write collision error가 발생한다.

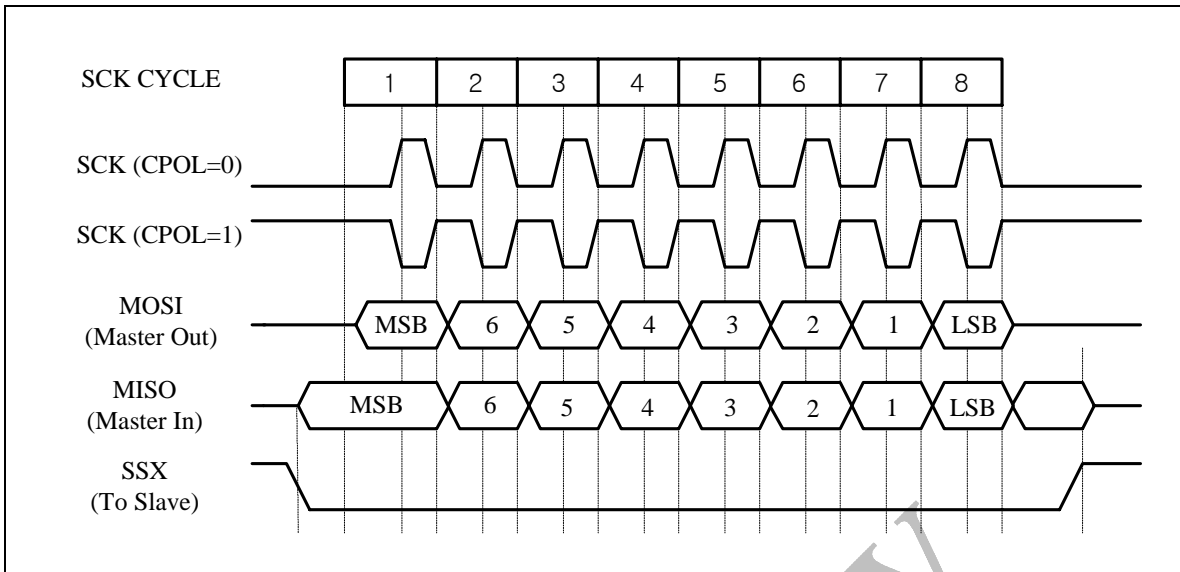


Figure 10-3 Transfer Timing when CPHA = '0'

CPHA='1'인 경우의 전송 타이밍도이다. SCK는 마지막 8번째 cycle의 반주기에서 inactive 된다. SPIF 비트는 8번째 SCK cycle의 중단에서 set된다. 8번째 SCK cycle의 중간주기에 마지막 edge가 생기기 때문에 Slave는 8번째 SCK cycle의 중간에서 마지막 데이터를 sample 한 후에 수신 완료한다. SSX는 1Byte 전송 마치고 어느 정도 시간 동안 충분히 low를 유지한다. 따라서 CPU가 전송 상태를 Polling하여 연속으로 전송하는 경우에는 계속 low 상태를 유지하게 된다.

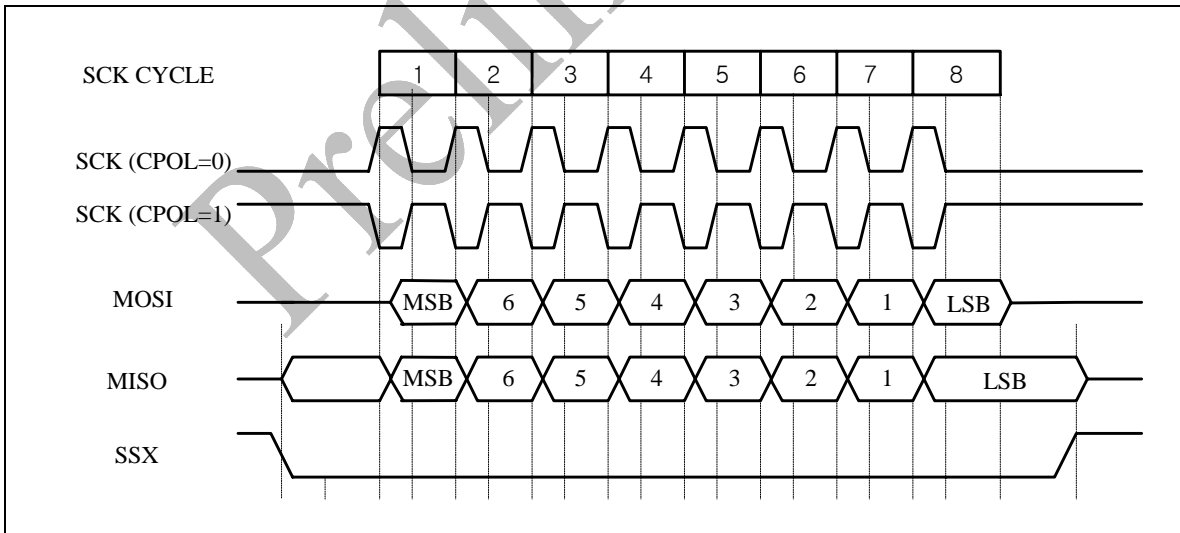


Figure 10-4 Transfer Timing when CPHA = '1'

## 10.6 SPI Serial Clock Baud Rate

SPI Baud rate는 SPBR 레지스터에 1에서 255까지의 값을 저장하여 설정할 수 있다.

Slave Mode에서의 외부 SPI Master가 제공하는 SCK를 받아들이기 때문에 SPIBRR 레지스터의 값의 설정에 영향을 받지 않는다. 그러나 Slave Mode에서 동작할 수 있는 최대 속도는 System Clock 에 영향을 받는다.

$$SCK \text{ Baud Rate} = \frac{\text{System Clock}}{2 \times (\text{SPIBRR}[7:0] + 1)}$$

or

$$\text{SPIBRR}[7:0] = \frac{\text{System Clock}}{2 \times SCK \text{ Baud Rate}}$$

## 10.7 Open-Drain Output for Wired-OR

Multiple SPI Master가 아니면 SPI 버스 출력이 Open-Drain을 지원할 필요는 없다. Open-Drain 출력이 필요할 경우 SPIxCON 레지스터의 WOMP 비트를 설정하여 Open-Drain 출력을 제공하도록 할 수 있다. Open-Drain으로 설정할 경우 각각의 출력 라인에 반드시 pull-up 저항을 달아야 한다.

## 10.8 Transfer Size and Direction

SPICONR 레지스터의 SPISIZE 비트는 전송크기 8/16/32비트를 선택한다. SPICONR 레지스터의 MSBF비트는 데이터 전송의 시작을 MSB(MSBF=1)나 LSB부터 하도록 한다.

## 10.9 Write Collision

전송 진행 중에 SPDR 레지스터에 쓰기를 시도하면 쓰기충돌이 발생한다.

## 10.10 MODE Fault

SPI system 이 Master로 설정되고, SS signal input line이 assert 되었을 때, mode fault error 가 발생하면, SPSR 의 MODF bit이 set 된다. Master device 만 MODF를 발생시킬 수 있으며, 다른 SPI device가 master가 되려고 할 때 발생한다.



### 10.11 Interrupt

#### SPIF Interrupt

FIFO에 저장된 데이터와 TX Shift 레지스터가 모두 비워지면 발생하고, SPI 전송이 완료되었음을 의미한다. SPI 전송이 완료되었음을 확인할 수 있는 Interrupt 이다.

#### MODF Interrupt

Mode fault 가 일어나면 발생한다. Mode fault란 여러 개의 Master가 존재하는 경우 둘 이상의 Master가 데이터를 전송하는 경우를 말한다.

#### SSX Interrupt

SSX port 신호를 감지하여 변화가 생기면 발생한다.

#### TX\_FIFO\_FULL, TX\_FIFO\_EMPTY, RX\_FIFO\_FULL, RX\_FIFO\_EMPTY

- TX\_FIFO\_FULL : 64Byte의 내부 FIFO 가 모두 채워졌음을 의미한다. 이 상태에서 TX FIFO에 데이터를 더 채우게 되면, 잘못된 데이터 전송이 이루어진다.
- TX\_FIFO\_EMPTY : TX FIFO에 채워졌던 데이터가 모두 전송되어 되었음을 의미한다. 그러나 아직 TX Shift 레지스터가 비워지지 않았기 때문에 SPI 전송이 완료된 것은 아니다.
- RX\_FIFO\_FULL : RX\_FIFO가 모두 채워졌음을 의미한다.
- RX\_FIFO\_EMPTY : RX\_FIFO가 모두 비워졌음을 의미한다.

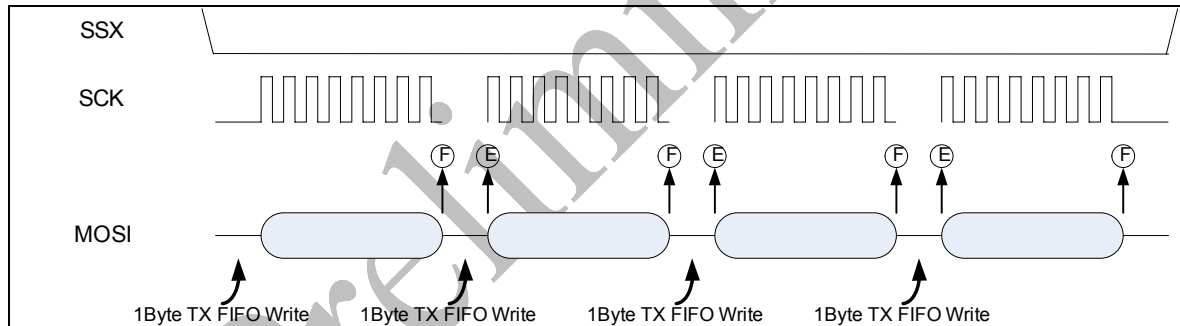


Figure 10-5 1-Byte Transfer vs. Status and Interrupt

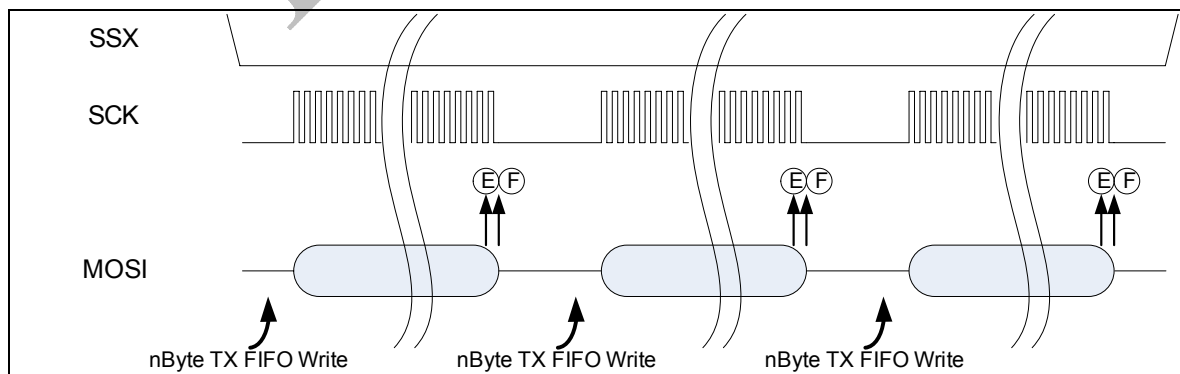


Figure 10-6 n-Bytes Transfer vs. Status and Interrupt

## 10.12 SPI Control Registers

### SPI Control Register (SPIxCON)

Address : 0x0300\_0700h / 0x0300\_0780h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	RW	SPIEN : SPI Enable 0 : SPI is disabled. 1 : SPI is enabled	0
6	RW	WOMP : Wired-OR Mode for SPI Pins 0 : Outputs have normal CMOS drivers. 1 : Open-drain drivers	0
5	RW	MSTR : Master/Slave Mode Select 0 : SPI is a slave device 1 : SPI is system master	0
4	RW	CPOL : Clock Polarity 0 : The inactive state value of SCK is logic level zero 1 : The inactive state value of SCK is logic level one.	0
3	RW	CPHA : Clock Phase 0 : Data captured on the leading edge of SCK and changed on the trailing edge of SCK. 1 : Data is changed on the leading edge of SCK and captured on the trailing edge of SCK.	0
2	RW	MSBF : Most Significant Bit First 0 : Serial data transfer starts with LSB. 1 : Serial data transfer starts with MSB.	0
1 : 0	RW	SPISIZE : Transfer Data Size 00 : 8-bit data transfer. 01 : 16-bit data transfer. 10 : 32-bit data transfer.	0

### SPI Baud Rate Register (SPIxBAUD)

Address : 0x0300\_0704h / 0x0300\_0784h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved.	-
7 : 0	RW	Serial Clock Baud Rate $SCK = \frac{\text{System Clock}}{2 \times (\text{SPIBRR} [7:0] + 1)}$ Master Mode SCK ≤ System Clock / 2 Slave Mode SCK ≤ System Clock / 4	0xFF

**SPI Status Register (SPIxSTAT)**

Address : 0x0300\_0708h / 0x0300\_0788h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
15 : 8	R	Reserved	-
7	R	SPIF : SPI Finished Flag 0 : SPI is not finished. 1 : SPI is finished.	0
6	R	WCOL : Write Collision 0 : No attempt to write to the SPDR happened during the serial transfer. 1 : Write collision occurred.	0
5	R	MODF : Mode Fault Flag 0 : Normal operation 1 : Another SPI node requested to become the network SPI master while the SPI was enabled in master mode	0
4	R	nSS : Slave Select Flag 0 : Current Value of nSS port is low 1 : Current Value of nSS port is high	0
3	R	STXF : TX FIFO Full Status bit 0 : FIFO_TX is not full 1 : FIFO_TX is full	0
2	R	STXE : TX FIFO Empty Status bit 0 : FIFO_TX is not empty 1 : FIFO_TX is empty	0
1	R	SRXF : RX FIFO Full Status bit 0 : FIFO_RX is not full 1 : FIFO_RX is full	0
0	R	SRXE : RX FIFO Empty Status bit 0 : FIFO_RX is not empty 1 : FIFO_RX is empty	0

**SPI Data Register (SPIxDATA)**

Address : 0x0300\_070Ch / 0x0300\_078Ch

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 0	RW	SPI Data At 32-bit transfer mode - MSB of Data is SPDR[31] At 16-bit transfer mode - MSB of Data is SPDR[15] At 8-bit transfer mode - MSB of Data is SPDR[7]  LSB of Data (received or transmit) is SPDR[0] in any transfer mode	0x0000_0000h

**SPI nSS Control Register (nSSxCON)**

Address : 0x0300\_0710h / 0x0300\_0790h

Bit	R/W	Description	Default Value
31 : 1	R	Reserved	-
0	RW	nSSCON : nSS Output Level	1

**SPI Interrupt Mask Register (SPiXINT)**

Address : 0x0300\_0714h / 0x0300\_0794h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	RW	SPIFE : SPIF Interrupt en/disable SPIF Interrupt occurs when transfer has completed. 0 : SPIF interrupt is disabled 1 : SPIF is enabled	0
6	RW	MODFE : MODFI Interrupt en/disable MODFI Interrupt occurs when two more master use data line. 0 : MODFI interrupt is disabled 1 : MODFI is enabled	0
5	R	Reserved	0
4	RW	nSSEN : nSS Interrupt en/disable nSS Interrupt occurs when nSS signal has changed. 0 : nSS Interrupt is disabled 1 : nSS Interrupt is enabled	0
3	RW	STXFE : FIFO_TX_FULL Interrupt en/disable FIFO_TX_FULL Interrupt occurs when FIFO_TX is full 0 : FIFO_TX_FULL Interrupt is disabled 1 : FIFO_TX_FULL Interrupt is enabled	0
2	RW	STXEE : FIFO_TX_EMPTY Interrupt en/disable FIFO_TX_EMPTY Interrupt occurs when FIFO_TX is empty 0 : FIFO_TX_EMPTY Interrupt is disabled 1 : FIFO_TX_EMPTY Interrupt is enabled	0
1	RW	SRXFE : FIFO_RX_FULL Interrupt en/disable FIFO_RX_FULL Interrupt occurs when FIFO_RX is full 0 : FIFO_RX_FULL Interrupt is disabled 1 : FIFO_RX_FULL Interrupt is enabled	0
0	RW	SRXEE : FIFO_RX_EMPTY Interrupt en/disable FIFO_RX_EMPTY Interrupt occurs when FIFO_RX is empty 0 : FIFO_RX_EMPTY Interrupt is disabled 1 : FIFO_RX_EMPTY Interrupt is enabled	0

## 11 TWI (Two WIRED INTERFACE)

범용 TWI 버스와 인터페이스를 위해 TWI 컨트롤러를 내장한다. TWI는 SCL(serial clock)과 SDA(serial data) 를 가진다.

### *TWI Features*

- Master transmitter mode
- Master receive mode
- Slave transmitter mode
- Slave receive mode
- Software programmable clock frequency
- Software programmable acknowledge bit
- Interrupt driven data-transfers
- Start/Stop/Repeated Start/Acknowledge generation
- Multi master operation

*Note*  
2.2K $\Omega$  to 4.7 K $\Omega$  pull-ups are required on the SDA and SCL lines.

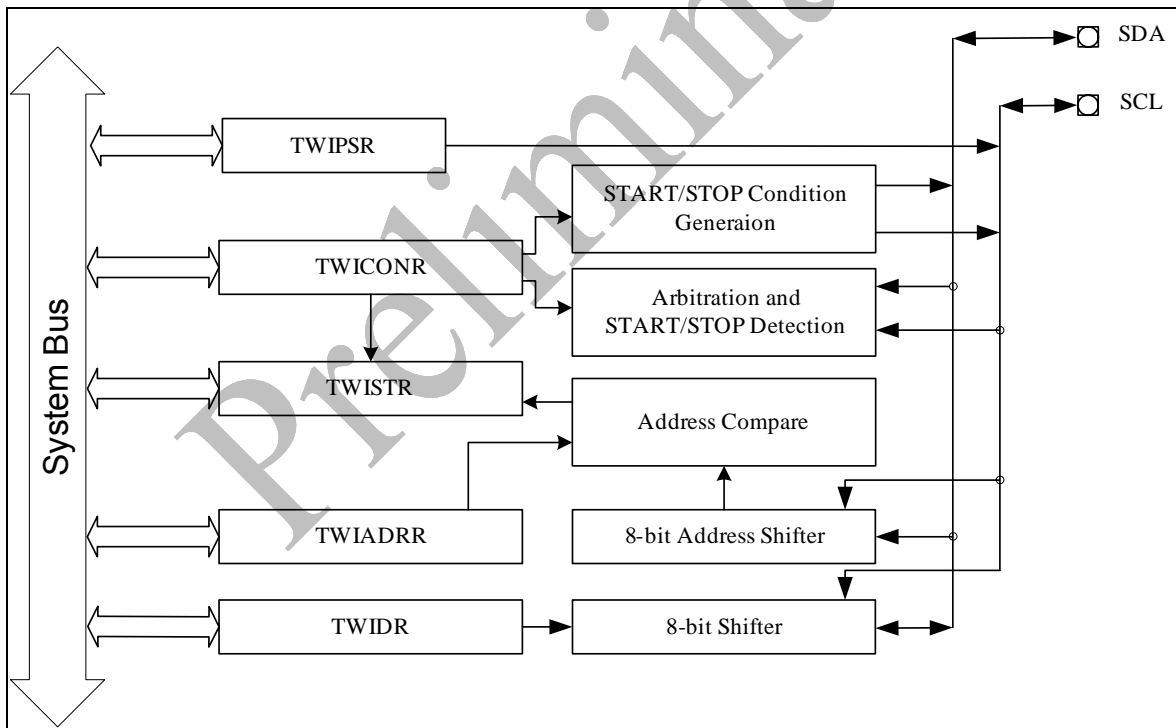


Figure 11-1 TWI Block Diagram

### 11.1 DATA TRANSFER FORMAT

SDA 라인에는 모든 데이터 길이는 8비트다. 매 전송마다 전송되어 질 수 있는 바이트 수는 제한되어 있지 않다. 시작조건 다음의 첫 바이트는 주소필드이다. TWI-bus 가 Master로 모드로 동작 할 때 Master에 의해 주소필드가 전송된다. 모든 바이트는 다음에는 ACK 비트가 따라온다. 항상 데이터와 주소의 MSB 비트부터 전송이 시작된다.

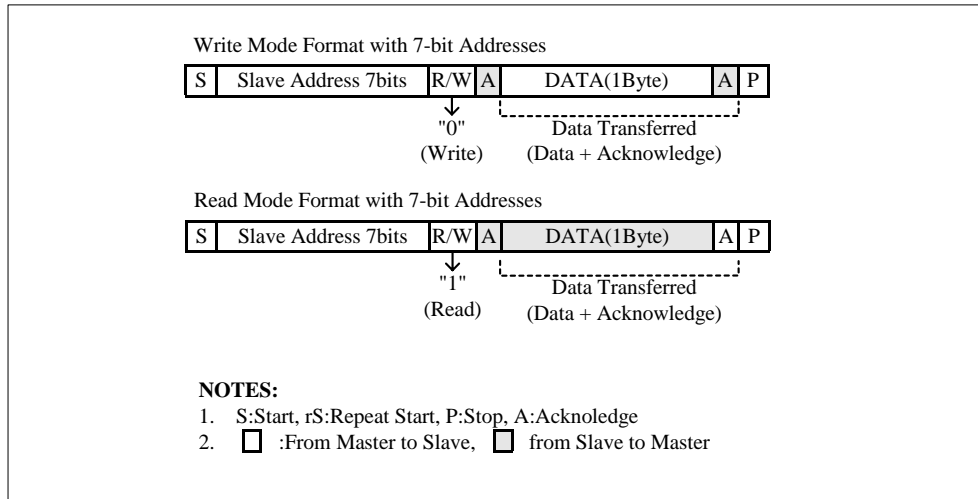


Figure 11-2 TWI-Bus Interface Data Format

### 11.2 START AND STOP CONDITION

Start condition은 1 Byte 의 data를 전송 할 수 있다. 그리고, Stop condition은 data 전송을 종료한다. Start condition은 SCL 이 high 일 때 SDA line이 high-to-low 로 transition 한다. A stop condition은 SCL 이 high 일 때 SDA line 이 low-to-high로 transition 한다. Start condition 이 발생하면, I2C bus 는 busy가 된다. Stop condition이 발생한 후, I2C bus는 free 가 된다.

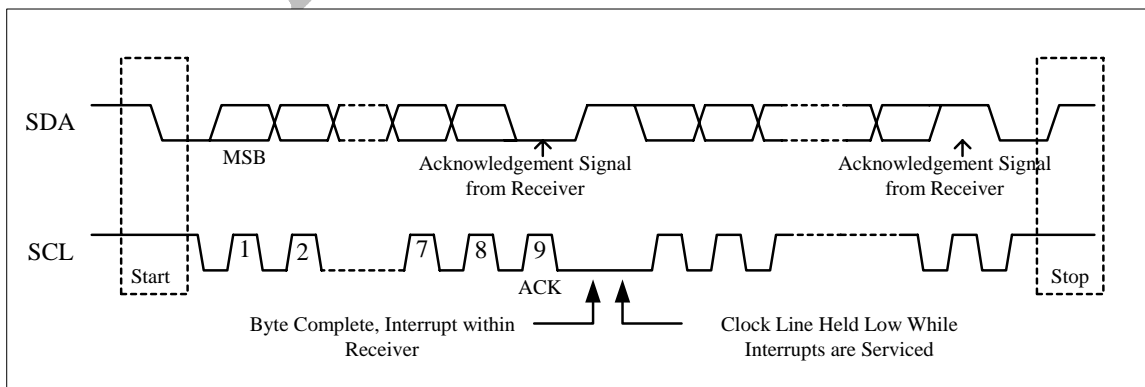


Figure 11-3 Data Transfer on the TWI-Bus

### 11.3 ACK SIGNAL TRANSMISSION

한 바이트 전송을 완전히 끝내기 위해서는 수신단은 송신단에 ACK 비트를 보내야 한다. ACK 펄스는 SCL 라인의 9 번째 클럭에서 발생해야 한다. 한 바이트 데이터를 전송을 위해 8 개의 클럭이 필요하다. Master는 ACK 비트 전송을 위한 클럭 펄스를 생성해야 한다.

송신단은 ACK 클럭 펄스를 입력 받을 때 SDA 라인을 “High”로 만들기 위해 SDA 라인을 놓아줘야 한다. 또한 수신기는 ACK 펄스 때 SDA 라인을 “Low”로 유지하여 SCL의 아홉 번째의 “High” 구간에서 SDA를 “Low”로 만든다.

ACK 비트 전송 동작은 소프트웨어적으로 enable 또는 disable 될 수 있다(컨트롤 레지스터의 TXACK비트). 그러나 한 바이트 데이터 전송 동작을 완료하기 위해 SCL의 아홉 번째에서의 ACK 펄스가 필요하다.

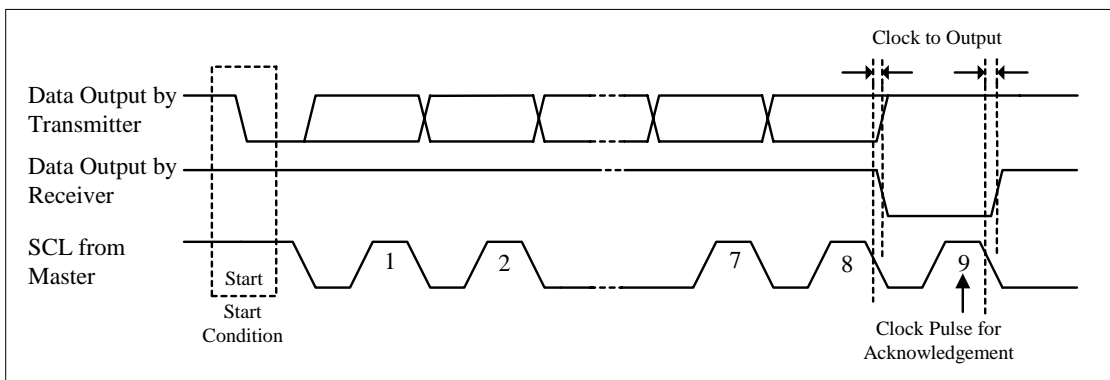


Figure 11-4 Acknowledgement of TWI

### 11.4 READ-WRITE OPERATION

송신 동작 모드에서 데이터 전송 후에 TWI-bus 인터페이스는 데이터 쉬프트 레지스터에 데이터가 준비될 때 까지 기다려야 한다. 데이터 쓰여질 때까지 SCL 라인은 Low로 유지될 것이다. 새로운 데이터가 데이터 쉬프트 레지스터에 쓰여지고 나서 SCL은 release 된다. 즉, TWI는 현재 데이터 전송 후 interrupt를 요청한다. CPU는 interrupt 요청을 받은 뒤에 새로운 데이터를 버퍼에 쓴다.

수신 동작 모드에서 데이터를 수신한 후에, TWI bus는 data를 읽어 갈 때 까지 기다린다. 수신된 데이터가 읽어 갈 때 까지 SCL을 LOW로 유지 된다. New data가 읽혀지고 난 다음에 SCL은 release 된다. TWI는 데이터를 수신한 후 interrupt를 발생하고, interrupt request를 받은 CPU는 data를 읽어 간다.

### 11.5 BUS ARBITRATION PROCEDURES

두 개의 master가 Bus를 제어 하는 것을 방지한다. SDA를 HIGH level로 내보낸 master가 또 다른 master가 내보낸 LOW level의 SDA를 인식하면, Bus가 자신의 제어영역이 아니기 때문에 data전송을 하지 못한다.

Device1과 Device2과 동시에 master mode 로 동작하는 경우에 SCL 라인에서 발생하는 클럭의 모양을 보면 다음과 같이 동기화가 이루어진다.

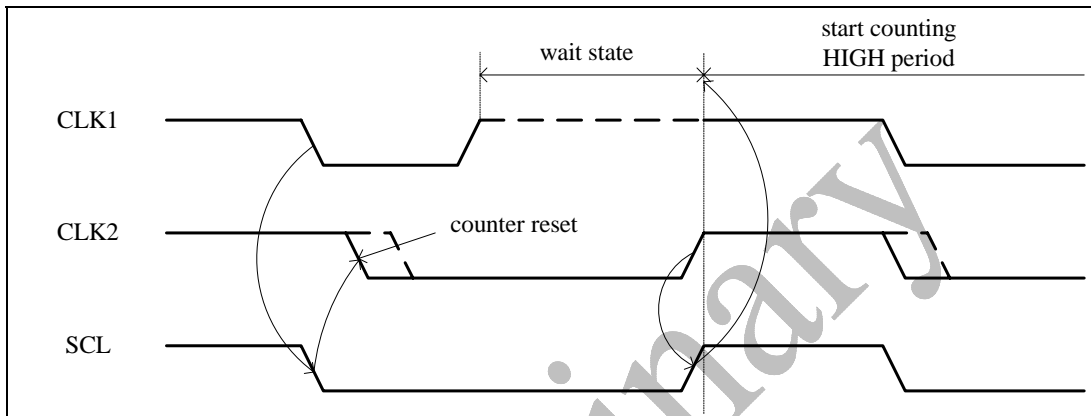


Figure 11-5 Bus arbitration 1 of TWI

위 상황에서 SDA 라인에 나타나는 data 값에 따라 Device1, Device2중 하나가 우선권을 갖는 과정은 다음과 같다.

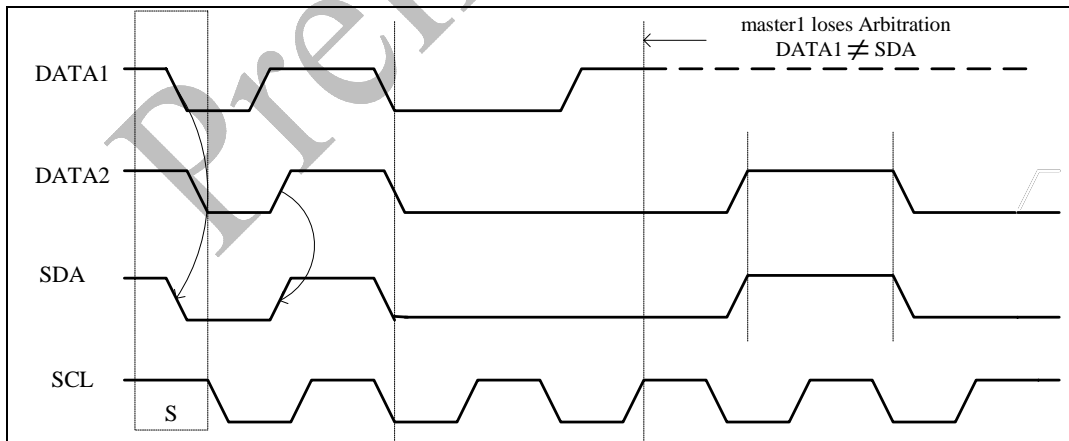


Figure 11-6 Bus arbitration 2



## 11.6 ABORT CONDITIONS (arbitration 이 발생하지 않은 경우)

1. 소프트웨어적으로 컨트롤 레지스터의 MSTR 비트을 클리어 시키면 stop 조건이 발생한다.
2. NO ACK 발생하여 stop 조건이 발생한다. 즉 ACK 구간에서 SDA 신호가 “Low”가 아니면 발생한다.

Arbitration 발생에 의해 제어권을 잃은 경우 MSTR 비트은 클리어 되지만 이에 의한 스톱 조건은 발생하지 않는다. 지금 진행중인 클럭 발생은 한 바이트 전송 끝까지 진행되고 데이터 출력인 SDA는 HIGH 상태가 된다.

## 11.7 Operational Flow Diagrams

### 11.7.1 TWI initialization

TWI는 먼저 초기화가 이루어져야 한다.

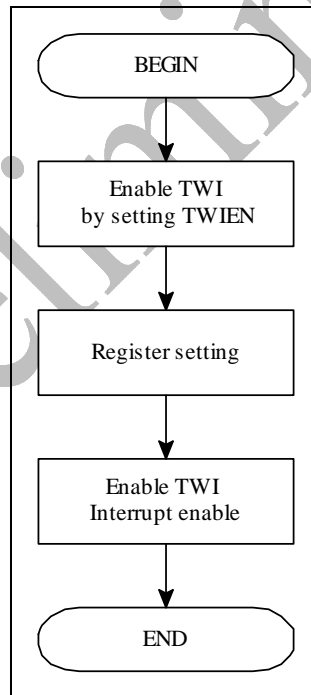


Figure 11-7 TWI Initialization Flow Char

11.7.2 Master Transmit /Receive

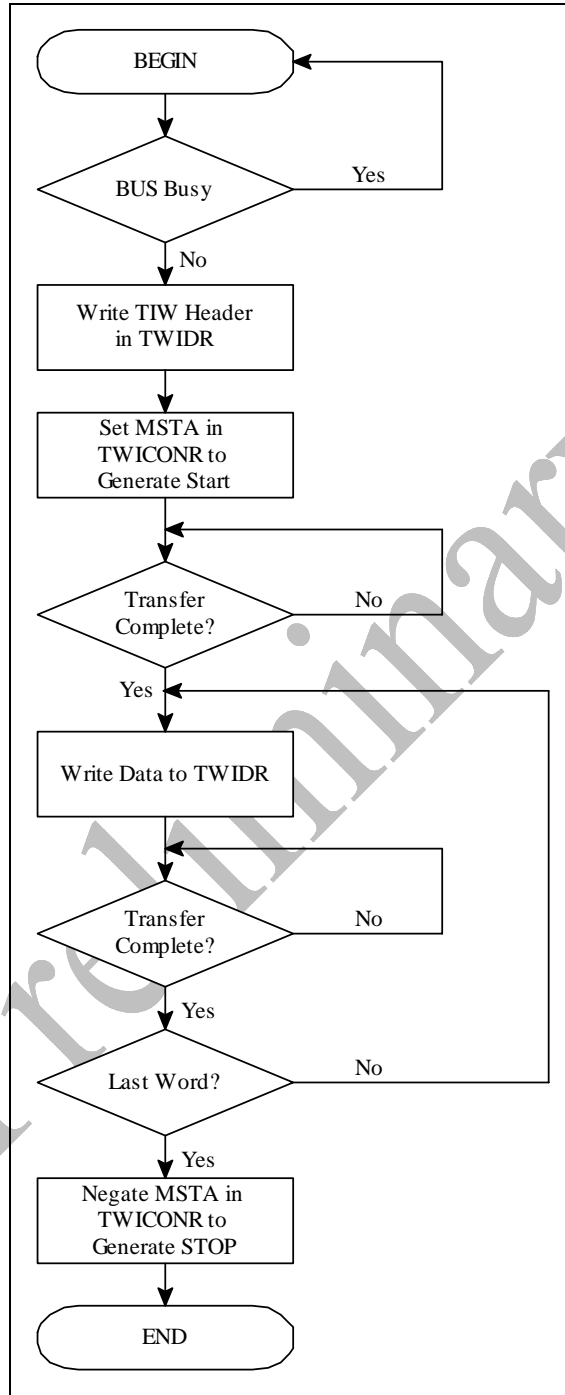


Figure 11-8 Master Transmit Flow Char

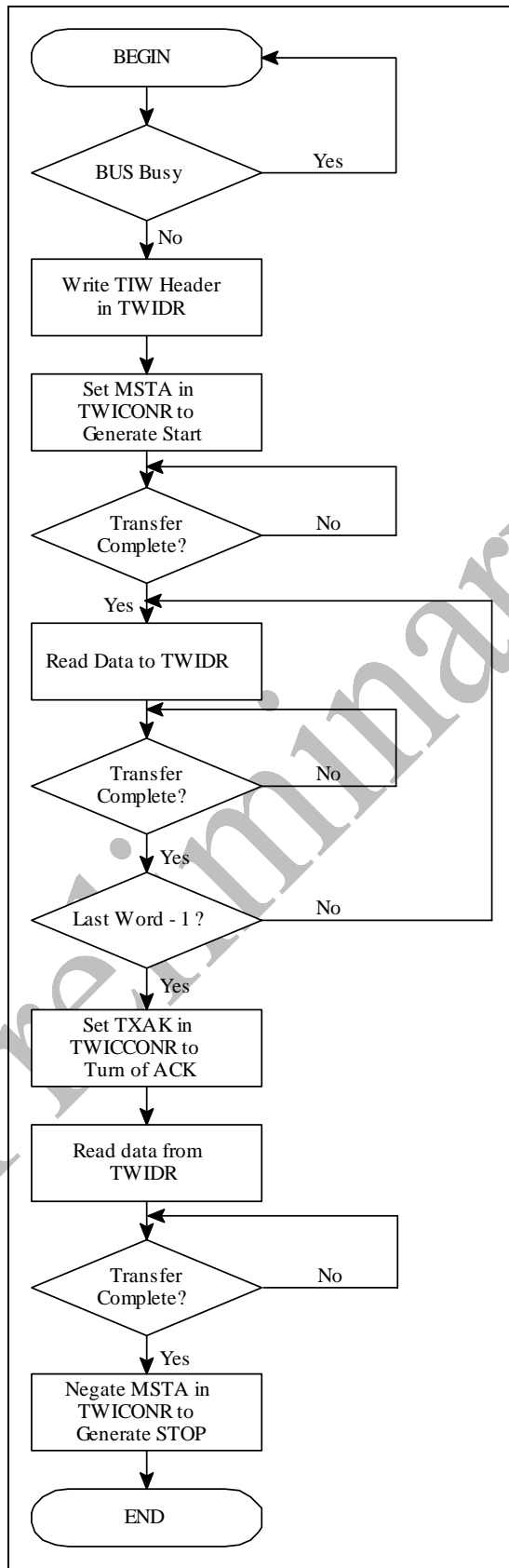


Figure 11-9 Master Receive Flow Char

11.7.3 Slave Mode

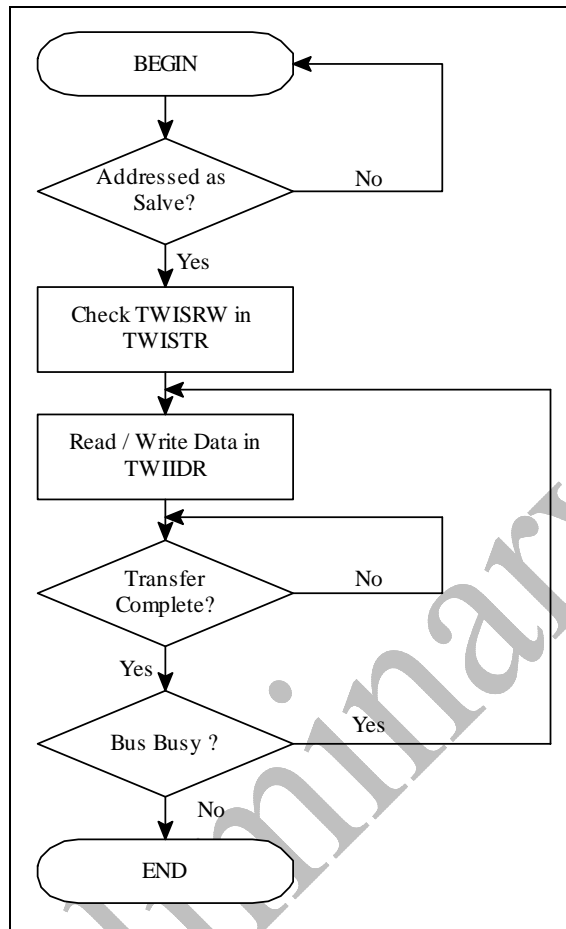


Figure 11-10 Slave Mode Flow Char

## 11.8 TWI Control Registers

### ***TWI Control Register (TWICON)***

Address : 0x0300\_0800

<b><i>Bit</i></b>	<b><i>R/W</i></b>	<b><i>Description</i></b>	<b><i>Default Value</i></b>
31 : 8	R	Reserved.	-
7	RW	TWIEN : TWI Controller Enable. This bit must be set before any other control register bits have any effect 0 : Resets and disables the TWI 1 : Enables the TWI	0
6	R	Reserved.	-
5	RW	TWIMOD : Master/Slave Mode Select. When the uC changes this bit from "0" to "1", the TWI controller generates a START condition in Master mode. When this bit is cleared, a STOP condition is generated and the TWI controller switches to Slave mode. If this bit is cleared, however, because arbitration for the bus has been lost, a STOP condition is not generated.	0
4	RW	TWITR : Transmit/Receive Mode Select. This bit selects the direction of Master transfers. Available only in master device. 0 : Master Receive 1 : Master Transmit	0
3	RW	TWIAK : Transmit Acknowledge Enable. This bit specifies the value driven onto the SDA line during acknowledge cycles for both Master and Slave receivers 0 : Acknowledge 1 : No acknowledge Since Master receivers indicate the end of data reception by not acknowledging the last byte of the transfer, this bit is the means of the uC to end a Master receiver transfer.	0
2	RW	REPST : Repeated Start. Writing a "1" to this bit generates a repeated START condition on the bus if the TWI controller is the current bus Master. This bit is always read as "0"	0
1	R/W	Transfer complete Interrupt enable bit 0 : Interrupt Disable 1 : Interrupt Enable	0
0	R/W	Lost arbitration Interrupt enable bit 0 : Interrupt Enable 1 : Master로 동작 시, 전송 권한을 잃으면 인터럽트 발생.	0

**TWI Status Register (TWISTAT)**

Address : 0x0300\_0804h

Bit	R/W	Description	Default Value
31 : 10	R	Reserved.	-
9	RW	TX_EMPTY. 송신 버퍼의 상태를 나타낸다. 0일 때, 원하는 값으로 Write할 수 있다. 0 : 송신 버퍼에 보낼 데이터가 있음 1 : 송신 버퍼가 비었음	1
8	RW	RX_FULL. 수신 버퍼의 상태를 나타낸다. 1일 때, 원하는 값으로 Write할 수 있다. 0 : 수신 버퍼가 비었음 1 : 수신 버퍼에 읽어갈 데이터가 있음	0
7	R	TWIDT : Data Transferring Bit. While one byte of data and one bit of ack are being transferred, this bit is cleared. It is set by the falling edge of the 9th clock of SCL and is cleared by reading status register. 0 : Transfer in progress 1 : Transfer is complete	0
6	R	TWIAS : Addressed as Slave Bit. When the address on the TWI bus matches the Slave address in the TWIADDR register, the TWI controller is being addressed as a Slave and switches to Slave mode.	0
5	R	TWIBUSY : Bus Busy Bit. This bit indicates the status of the TWI bus. This bit is set when a START condition is detected and cleared when a STOP condition is detected. 0 : Indicates the bus is idle 1 : Indicates the bus is busy	0
4	RW	TWILOST : Lost Arbitration Bit. This bit is set by hardware when arbitration for the TWI bus is lost. This bit must be cleared by MCU writing a '0' this bit.	0
3	R	TWISRW : Slave Read/Write Bit. When the TWI controller has been addressed as a Slave (AAS is set), this bit indicates the value of the read/write bit sent by the Master. This bit is only valid when a complete transfer has occurred and no other transfers have been initiated. 0 : Indicates Master writing to Slave 1 : Indicates Master reading from Slave	0
2	R	Reserved.	-
1	RW	Repeated start flag Repeated start condition이 발생하였는지 확인하는 flag bit. Repeated start condition이 발생하면 set되며, stop condition이 발생하거나, set된 상태에서 이 bit에 1을 write하면 clear된다	0

		0 : Repeated start condition was not occurred. 1 : Repeated start condition was occurred.	
0	R	TWIRAK : Received Acknowledge Bit. This bit reflects the value of the SDA signal during the acknowledge cycle of the transfer. 0 : Indicates that an acknowledge was received 1 : Indicates that no acknowledge was received	1

**TWI Address Register(TWIADR)**

Address : 0x0300\_0808h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved.	-
7 : 0	RW	(At only slave mode) 7-bit slave address, latched from the TWI bus. You can read the TWIADDR value at any time, regardless of the current serial output enables bit (TWICON) setting. bit[7:1] : Slave Address bit[0] : Not mapped	00h

**TWI Data Register (TWIDATA)**

Address : 0x0300\_080Ch

Bit	R/W	Description	Default Value
31 : 8	R	Reserved.	-
7 : 0	RW	TWI data	00h

**TWI Baud-Rate 0 Register (TWIBRO)**

Address : 0x0300\_0810h

Bit	R/W	Description	Default Value
31 : 4	R	Reserved.	-
7 : 0	RW	Baud Rate 0 Value. $TWIBRO \geq 3$	Fh

**TWI Pre-scale Counter Register (TWIBR1)**

Address : 0x0300\_0814h

Bit	R/W	Description	Default Value
31 : 9	R	Reserved.	-
8 : 0	RW	Baud Rate 1 Value. $TWIBR1 \geq 0$	FFh

$$TWIBRO = f_{system\_clock} \times 700ns + 3$$

$$SCL = \frac{f_{system\_clock}}{(2TWIBR1 + TWIBRO + 7)}$$

## 12 UART

TG471의 UART는 RS-232C 인터페이스의 기능을 보유한 일반적인 PC 및 I/O device와 직렬 비동기 통신을 위한 다양한 제어기능을 가진 4 채널 UART(Universal Asynchronous Receiver/ Transmitter) Controller 가 내장되어 있다.

### UART Features

- Compatible with standard 16450/16550 UARTs
- Fully programmable serial-interface protocols
  - 5,6,7,8-bit characters
  - Even, odd or no-parity, stick parity generation and detection
  - 1, 1.5, 2 stop bit generation
  - Baud rate generator
- Line break generation and detection
- False start bit detection
- Prioritized transmit, receive and line status control interrupts
- Independent 16 characters transmit and receive 4Bytes FIFOs
- Support 4 Ch. UARTs

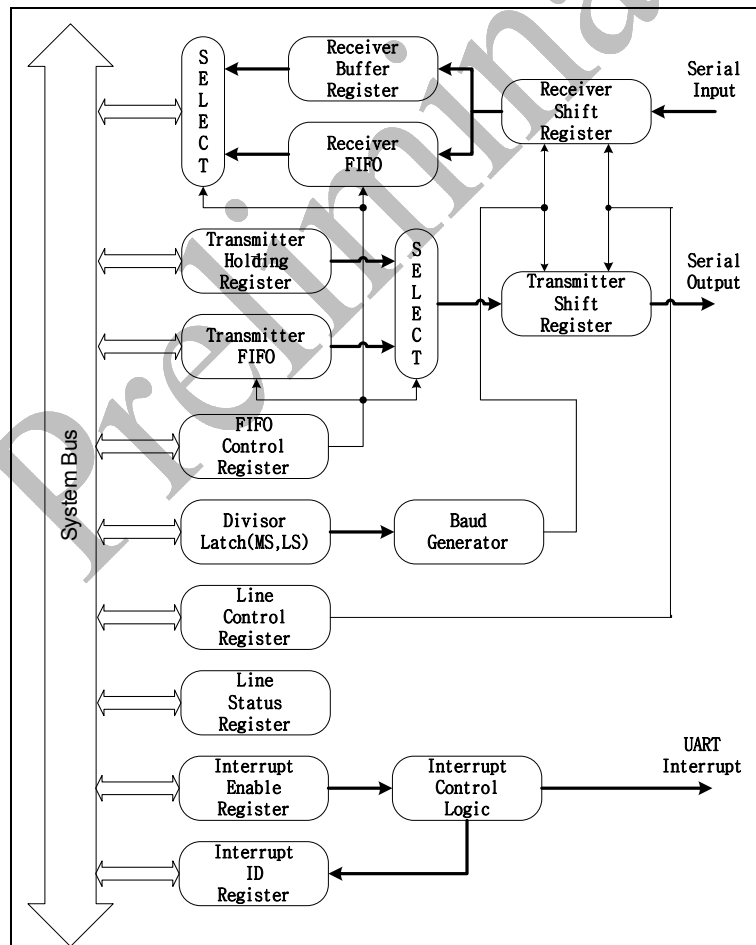


Figure 12-1 UART Block Diagram



UART Registers Summary

Table 12-1 UART Register Summary

Bit No.	DLAB = 0	DLAB = 0	DLAB = 0	DLAB = 0	DLAB = X	DLAB = X	DLAB = X	DLAB = 1	DLAB = 1
	0x00	0x00	0x04	0x08	0x08	0x0C	0x14	0x00	0x04
	Receiver Buffer Register	Transmitter Holding Register	Interrupt Enable Register	Interrupt Ident. Register	FIFO Control Register	Line Control Register	Line Status Register	Divisor Latch (LSB)	Divisor Latch (MSB)
	RBR	THR	IER	IIR	FCR	LCR	LSR	DLL	DLM
	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W
0	Data Bit 0	Data Bit 0	Enable Received Data Available Interrupt	"0" if Interrupt Pending	FIFO Enable	Word Length Select Bit 0	Data Ready	Bit 0	Bit 0
1	Data Bit 1	Data Bit 1	Enable Transmitter Holding Register Empty Interrupt	Interrupt ID Bit 0	RCVR FIFO Reset	Word Length Select Bit 1	Overrun Error	Bit 1	Bit 1
2	Data Bit 2	Data Bit 2	Enable Receiver Line Status Interrupt	Interrupt ID Bit 1	XMIT FIFO Reset	Number of Stop Bits	Parity Error	Bit 2	Bit 2
3	Data Bit 3	Data Bit 3	0	Interrupt ID Bit 2	0	Parity Enable	Framing Error	Bit 3	Bit 3
4	Data Bit 4	Data Bit 4	0	0	Reserved	Even Parity Select	Break Interrupt	Bit 4	Bit 4
5	Data Bit 5	Data Bit 5	0	0	Reserved	Stick Parity	Transmitter Holding Register	Bit 5	Bit 5
6	Data Bit 6	Data Bit 6	0	FIFOs Enabled	RCVR Trigger(LS B)	Set Break	Transmitter Empty	Bit 6	Bit 6
7	Data Bit 7	Data Bit 7	0	FIFOs Enabled	RCVR Trigger(MS B)	Divisor Latch Access Bit (DLAB)	Error in RCVR FIFO	Bit 7	Bit 7

\* DLAB = LCR[7](Divisor Latch Access Bit)  
 \* FIFO Control Register :  
   - DLAB = 0 : Register Write  
   - DLAB = 1 : Register Read  
 \* Address 0x10(0x30), 0x18(0x38), 0x1C(0x3C)는 16550 UART 표준과의 호환성을 위해 Reserved 되었다.

### 12.1 Serial Data Format

TG471의 UART에서는 ULCRn[4:0] 비트의 레지스터 설정으로 UART 통신 Serial Data Format에 대한 변경이 가능하다. 다음 표는 ULCRn[4:0] bit의 Register 설정으로 변경 가능한 데이터 포맷에 대한 설명이다.

ULCRn[4:0]	Description
00010 No Parity / 1 Stop bit / 7 Data bit	<p>시작비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 정지비트</p>
00011 No Parity / 1 Stop bit / 8 Data bit	<p>시작비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 정지비트</p>
00110 No Parity / 2 Stop bit / 7 Data bit	<p>시작비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 정지비트, 정지비트</p>
00111 No Parity / 2 Stop bit / 8 Data bit	<p>시작비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 정지비트, 정지비트</p>
11010 Even Parity / 1 Stop bit / 7 Data bit	<p>시작비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 패리티비트, 정지비트</p>
11011 Even Parity / 1 Stop bit / 8 Data bit	<p>시작비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 데이터비트, 패리티비트, 정지비트</p>

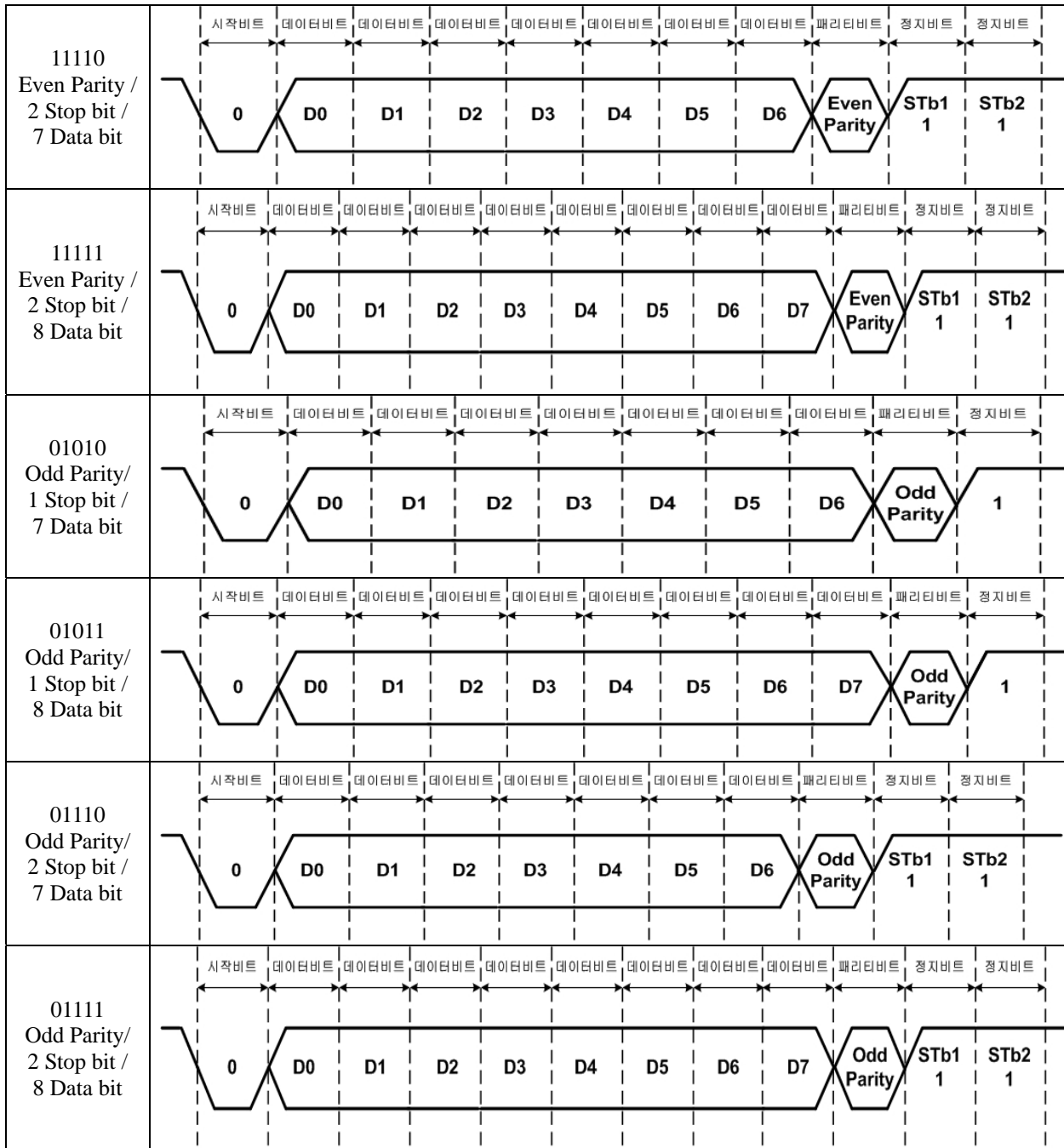


Figure 12-2 UART LCR Register Setting and Serial Data Format

## 12.2 UART Baud Rate

TX/RX Baud Rate은 아래 식으로 계산된다.

$$\text{Baud Rate} = \frac{\text{System Clock}}{\text{UDL} \times 16}$$

UART Divisor Latch Value (UDL) = UDLM[7:0] << 8 + UDLL[7:0]

Table 12-2 UART Baud Rate

<i>Baud Rate(MHz)</i>		<b>4</b>	<b>6</b>	<b>8</b>	<b>10</b>	<b>12</b>	<b>16</b>	<b>20</b>	<b>24</b>	<b>30</b>
2400 bps	UDL	104	156	208	260	313	417	521	625	781
	ERR(%)	0.16	0.16	0.16	0.16	0.16	0.08	0.03	0.00	0.03
4800 bps	UDL	52	78	104	130	156	208	260	313	391
	ERR(%)	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.10
9600 bps	UDL	26	39	52	65	78	104	130	156	195
	ERR(%)	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
14400 bps	UDL	17	26	35	43	52	69	87	104	130
	ERR(%)	2.12	0.16	0.79	0.94	0.16	0.64	0.22	0.16	0.16
19200 bps	UDL	13	-	26	33	39	52	65	78	98
	ERR(%)	0.16	-	0.16	1.36	0.16	0.16	0.16	0.16	0.35
38400 bps	UDL	-	-	13	16	-	26	33	39	49
	ERR(%)	-	-	0.16	1.73	-	0.16	1.36	0.16	0.35
57600 bps	UDL	-	-	-	11	13	17	22	26	33
	ERR(%)	-	-	-	1.36	0.16	2.12	1.36	0.16	1.36
115200bps	UDL	-	-	-	-	-	-	11	13	16
	ERR(%)	-	-	-	-	-	-	1.36	0.16	1.73

\*\*\* ERR 이 2.2% 이상에서는 UART 동작의 안정성을 보장 받을 수 없다.

## 12.3 UART Control Registers

### UART Channel 0/1 Receiver Buffer Registers ( UxRB )

Address : 0x0300\_0900h / 0x0300\_0920h / 0x0300\_0980h / 0x0300\_09A0h

Bit	R/W	Description	Default Value
31:8	R	Reserved.	-
7:0	R	Receive Buffer Data	-

\*\*\* DLAB가 “0” 일 때 Access 가능하다.

### UART Channel Transmitter Holding Registers ( UxTH )

Address : 0x0300\_0900h / 0x0300\_0920h / 0x0300\_0980h / 0x0300\_09A0h

Bit	R/W	Description	Default Value
31:8	W	Reserved.	-
7:0	W	Transmit Holding Data	-

\*\*\* DLAB가 “0” 일 때 Access 가능하다.

### UART Channel Interrupt Enable Registers ( UxIE )

Address : 0x0300\_0904h / 0x0300\_0924h / 0x0300\_0984h / 0x0300\_09A4h

Bit	R/W	Description	Default Value
31:3	R	Reserved.	-
2	RW	RLSIEN : Receiver Line Status Interrupt Enable bit 0 : Disable                      1 : Enable	0
1	RW	THEIEN : Transmitter Holding Empty Interrupt Enable bit 0 : Disable                      1 : Enable	0
0	RW	RDAIEN : Received Data Available Interrupt Enable bit 0 : Disable                      1 : Enable	0

\*\*\* DLAB가 “0” 일 때 Access 가능하다.

**UART Channel Interrupt Identification Register ( UxII )**

Address : 0x0300\_0908h / 0x0300\_0928h / 0x0300\_0988h / 0x0300\_09A8h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved.	-
7 : 6	R	FIFOST : FIFOs Enabled Status bit. 00 : not in FIFO mode 11 : FIFO mode	00
5 : 4	R	Reserved	0
3 : 1	R	INTID : UART Interrupt ID ( Note, UART Interrupt Control Function)	000
0	R	INTP : UART Interrupt Pending bit When this bit is a logic 1, no interrupt is pending	1

\*\*\* DLAB가 “0” 일 때만 Read Mode로 Access 가능하다.

Table 12-3 UART Interrupt Control Function

Interrupt Identification Register				Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Condition
Bit 3	Bit 2	Bit 1	Bit 0				
0	0	0	1	-	None	None	-
0	1	1	0	Highest	Receiver Line Status	Overrun Error or Parity Error Framing Error or Break Interrupt	Reading the Line Status Register
0	1	0	0	Second	Received Data Available	Receiver Data Available or Trigger Level Reached	Reading the Receiver Buffer Register or the FIFO Drops Below the Trigger Level
1	1	0	0	Second	Character Timeout Indication	No Characters have been removed from or input to the RCVR FIFO during the last 4 Char. times, and there is at least 1 Char. in it during this Time	Reading the Receiver Buffer Register
0	0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if source of interrupt) or Writing into the Transmitter Holding Register

**UART Channel FIFO Control Register ( UxFC )**

Address : 0x0300\_0908h / 0x0300\_0928h / 0x0300\_0988h / 0x0300\_09A8h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved.	-
7 : 6	RW	RFTL : Receiver FIFO Trigger Level 00 : 1 Byte                      01 : 2 Byte 1x : 3 Byte	00
5 : 3	R	Reserved	-
2	RW	XFR : XMIT FIFO Reset XFR가 “1” 일 때, XMIT FIFO 내의 모든 데이터는 Reset 된다. 그러나 Shift Register 내의 데이터는 Reset 되지 않는다.	0
1	RW	RFR : RCVR FIFO Reset RFR가 “1” 일 때, RCVR FIFO 내의 모든 데이터는 Reset 된다. 그러나 Shift Register 내의 데이터는 Reset 되지 않는다.	0
0	RW	FIFOEN : FIFO Enable Bit 0 : 16450 UART Mode      1 : Enables FIFO	0

\*\*\* DLAB가 “0” 일 때는 Write Mode 이고, DLAB가 “1” 일 때는 Read Mode 이다.

**UART Channel Line Control Register ( UxLC )**

Address : 0x0300\_090Ch / 0x0300\_092Ch / 0x0300\_098Ch / 0x0300\_09ACh

Bit	R/W	Description	Default Value
31 : 8	R	Reserved.	-
7	RW	DLAB : Divisor Latch Access Bit DLAB이 “1” 일 때, Divisor Latch Registers의 Read/Write와 FIFO Control Register의 Read가 가능하다.	0
6	RW	SB : Set Break SB가 “1” 일 때, Serial Data Output에 Logic “0”이 출력된다. SB는 내부 Transmitter Logic에는 영향을 미치지 않으며, 단지 Serial Output에만 영향을 미친다.	0
5	RW	SP : Stick Parity 0 : Disables Stick Parity 1 : PEN, EPS, SP가 “1”일 때, Parity Bit “0” PEN, SP가 “1”이고, EPS가 “0” 일 때, Parity Bit “1”	0
4	RW	EPS : Even Parity Select 0 : Select Odd Parity      1 : Select Even Parity	0
3	RW	PEN : Parity Enable Bit 0 : Disables Parity      1 : Enables Parity	0
2	RW	STB : Number of Stop Bit 0 : 1 Stop bit 1 : 2 Stop bits(만약, WLS Bit에서 5 Bits/Character를 선택했다면, 1.5 Stop bits 을 갖는다.)	0
1 : 0	RW	WLS : Word Length Select 00 : 5 Bits/Character      01 : 6 Bits/Character 10 : 7 Bits/Character      11 : 8 Bits/Character	00

**UART Channel Line Status Register ( UxLS )**

Address : 0x0300\_0914h / 0x0300\_0934h / 0x0300\_0994h / 0x0300\_09B4h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved.	-
7	R	EIRF : Error in RCVR FIFO FIFO 모드가 아닌 경우 EIRF는 항상 “0”이다. FIFO 모드에서 EIRF는 RCVR FIFO 내에서 OE, PE, FE, BI 중 어느 하나라도 “1”이 설정되면, “1”이 된다. EIRF는 만약 FIFO 내에 연속적인 에러가 없다면, LSR 레지스터를 읽었을 때 Clear(“0”)된다.	0
6	R	TEMP : Transmitter Empty FIFO 모드가 아닌 경우 TEMT는 Transmitter Holding Register (THR)와 Transmitter Shift Register(TSR)이 모두 Empty일 때 “1”이 된다. THR 또는 TSR에 데이터가 있으면 Clear된다. FIFO 모드 에서는, TEMT는 Transmitter FIFO와 TSR이 모두 Empty일 때 “1”이 된다.	1
5	R	THRE : Transmitter Holding Register Empty FIFO 모드가 아닌 경우 THRE는 THR의 데이터가 TSR로 전송 되어 Empty가 되었을 때 “1”이 되며, THR에 전송을 위한 새로운 데이터를 쓸 수 있다. FIFO 모드에서는 Transmit FIFO가 Empty일 때 THRE가 “1”이 되며, 적어도 하나의 Byte라도 Transmit FIFO에 쓰이면 Clear된다. 만약 THRE interrupt(ETHREI) 가 “1”이고 THRE가 “1”이라면 Interrupt가 발생한다.	1
4	R	BINT : Break Interrupt : 수신되는 입력 데이터가 Full-word 전송 시간 동안 “0”일 때 BI는 “1”이 된다. Full-word 전송 시간은 Start, Data, Parity 그리고 Stop 비트 전송을 위한 전체 시간을 의미한다. FIFO 모드에서 이 에러는 FIFO 내의 각각의 Byte에 적용되며, BI가 발생했을 때 FIFO에는 “0”이 썬진다. CPU가 LSR을 읽어 올 때 Clear 된다.	0
3	R	FERR : Framing Error FE는 수신되는 입력 데이터가 유효한 Stop 비트를 가지지 않았을 때 “1”이 된다. FIFO 모드에서 이 에러는 FIFO 내의 각각의 Byte에 적용된다. CPU가 LSR을 읽어 올 때 Clear 된다.	0
2	R	PERR : Parity Error PE는 수신되는 입력 데이터가 LCR 레지스터에 의해 선택된 Parity 비트와 같지 않을 때 “1”이 된다. FIFO 모드에서 이 에러는 FIFO 내의 각각의 Byte에 적용된다. CPU가 LSR을 읽어 올 때 Clear 된다.	0
1	R	OERR : Overrun Error OE는, FIFO 모드가 아닌 경우, RBR 내의 데이터를 읽어가기 전에 새로운 데이터가 썬진 경우 “1”이 된다. FIFO 모드에서는 FIFO가 Full 상태에서 Receiver Shift	0



		Register(RSR)에 새로운 Full-word가 들어왔을 때 “1”이 된다. 이 경우 RSR은 새로운 데이터로 계속 갱신이 되지만, FIFO로 전송은 되지 않는다. CPU가 LSR을 읽어 올 때 Clear 된다.	
0	R	DRDY : Data Ready DR은 수신된 데이터가 RBR 또는 FIFO에 써졌을 때 “1”이 된다. RBR 또는 FIFO 내의 모든 데이터가 CPU에 의해 읽혀졌을 때 Clear된다.	0

**UART Channel Divisor Latch LSB Register ( UxDLL )**

Address : 0x0300\_0900h / 0x0300\_0920h / 0x0300\_0980h / 0x0300\_09A0h

Bit	R/W	Description	Default Value
31:8	R	Reserved.	-
7:0	RW	Divisor Latch Least Significant Byte	0x00

\*\*\* DLAB가 “1” 일 때 Access 가능하다.

**UART Channel Divisor Latch MSB Register ( UxDLM )**

Address : 0x0300\_0904h / 0x0300\_0924h / 0x0300\_0984h / 0x0300\_09A4h

Bit	R/W	Description	Default Value
31:8	R	Reserved.	-
7:0	RW	Divisor Latch Most Significant Byte	0x00

\*\*\* DLAB가 “1” 일 때 Access 가능하다.

**IRDA Control Register ( UxIRDA )**

Address : 0x0300\_0918h / 0x0300\_0938h / 0x0300\_0998h / 0x0300\_09B8h

Bit	R/W	Description	Default Value
31:8	R	Reserved.	-
1	RW	IRDA RX Enable bit	0
0	RW	IRDA TX Enable bit	0

## 13 DIRECT MEMORY ACCESS (DMA)

TG471은 1채널 범용 DMA를 내장하였다. DMA는 시스템 버스상에 위치하여 메모리와 I/O Device 또는 메모리 와 메모리 사이에서 고속 데이터 전송을 지원한다.

### DMA features :

- 1 채널 DMA
- 32bits Source/Destination Address
- Source와 Destination 에 대한 Increase Address /Decrease Address/Fixing Address
- Burst mode
- Address and counter auto reload
- 8 / 16 / 32 bits data transfer
- Software Enable DMA mode and Hard-wire Enable DMA mode

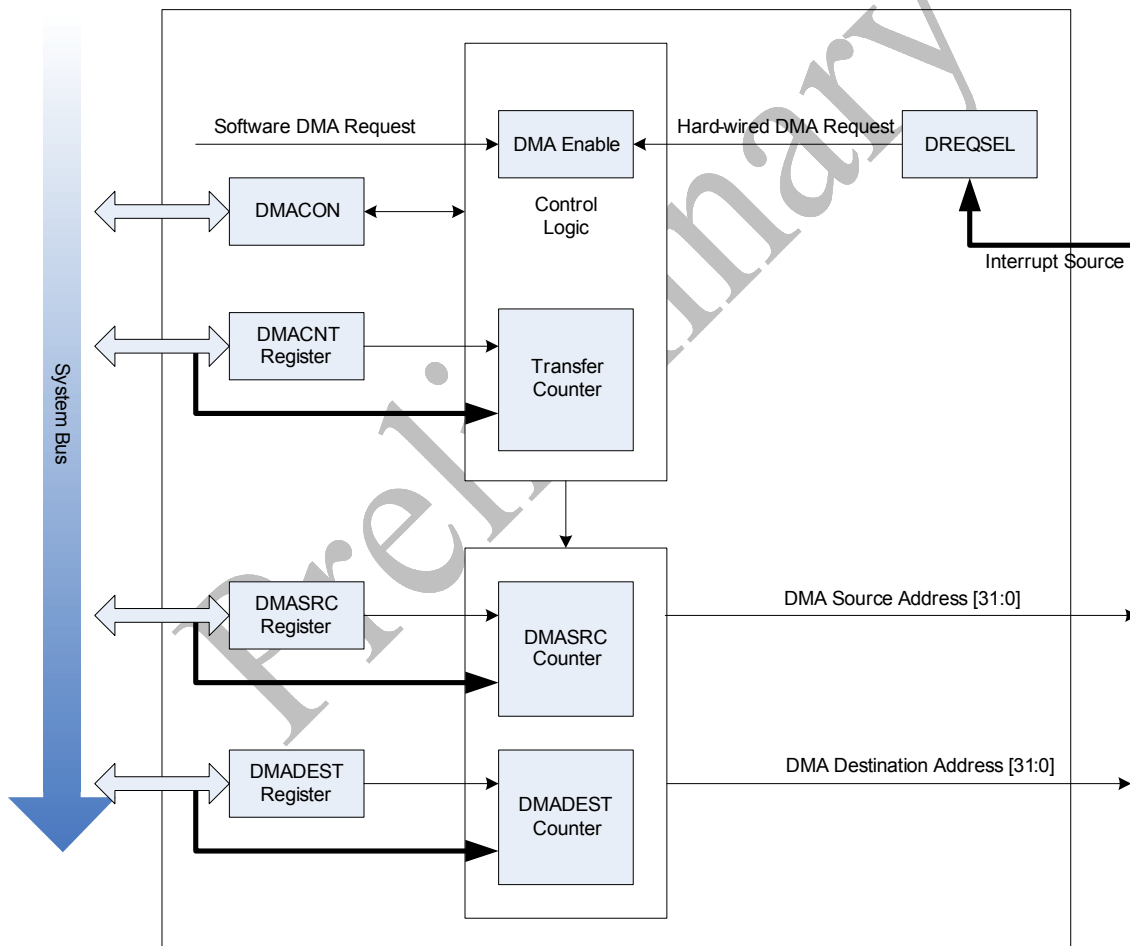


Figure 13-1 DMA Block Diagram

## 13.1 Function Description

### 13.1.1 Transfer mode

DMA의 동작은 내부 시스템 버스를 Burst mode로 접근하기 때문에 DMA가 Enable 되면 내부 시스템 버스 접근이 DMA로만 이루어지고, MCU에서는 접근하지 않는다. 따라서 DMA가 Enable 되면 DMA 전송이 끝날 때까지 DMA가 시스템 버스를 점유하기 때문에 새로운 명령은 DMA 전송이 완료된 이후에 접근할 수 있다.

Source Address와 Destination Address에 대한 설정은 각각 increase / decrease / fixing의 다양한 방법으로 사용될 수 있다

### 13.1.2 Address mode

DMA에 사용할 Source Address와 Destination Address에 대한 제어는 각각 increase / decrease / fixing의 다양한 방법으로 사용될 수 있다.

### 13.1.3 Reloading mode

DMA Source Address, DMA Destination Address, 그리고 DMA Transfer Counter 에는 MCU가 설정에 사용하는 레지스터와 전송이 진행될 때 내부적으로 제어되는 각각의 Counter가 존재한다.

DMA Source/Destination Address Register reload enable bit는 DMA가 Enable 될 때 내부 Counter가 사용할 초기값으로 내부 Counter 값을 계속 사용할 것인지 또는 Source/Destination Register로부터 다시 받아들일 것인지를 결정한다.

DMA Transfer Counter Register reload enable bit는 DMA가 Enable 될 때 내부 Counter가 사용할 초기값으로 내부 Counter 값을 계속 사용할 것인지 또는 Transfer counter Register로부터 다시 받아들일 것인지를 결정한다.

Reloading mode는 연속적인 DMA 전송시 Source/Destination Address Register 와 Transfer Counter Register를 재 설정할 필요없이 사용할 수 있는 편의를 제공한다.

### 13.1.4 Data Transfer Width 와 Transfer Counter

DMA의 Data Transfer Width는 DMA가 한번에 전송할 데이터의 크기를 의미하고 DMA Transfer Counter는 데이터의 처리횟수를 나타낸다.

예를 들어 Data Transfer Width가 8bit이고 Transfer Counter 가 10이면 10bytes의 데이터가 전송되고, Data Transfer Width가 32bit 이고 Transfer Counter 가 10이면 40bytes의 데이터가 전송된다.

### 13.1.5 Software Enable DMA and Hard-wire Enable DMA

DMA Control 레지스터의 DMA Enable bit를 통해 DMA를 Enable 하는 방식이 Software Enable DMA이다. Hard-wire Enable DMA는 DMA Request Source Select 레지스터를 통해 Request Source에서 인터럽트가 발생할 때, 인터럽트를 통한 전송이 아닌 DMA로 전송하는 방식이다.

Hard-wire Enable DMA를 사용할 때는 DMA Request Source로 설정한 장치가 DMA와 인터럽트 양쪽에서 사용하지 않도록 주의가 필요하다.

만약 2가지 이상의 DMA Request Source를 사용할 경우에는 DMA Source/Destination Address 레지스터와 DMA Transfer Counter 레지스터를 적절한 시점에서 재 설정할 필요가 있다.

Preliminary

## 13.2 DMA Control Registers

### DMA Control Register (DMACON)

Address : 0x0300\_0D00h

Bit	R/W	Description	Default Value
31 : 11	R	Reserved.	-
10	RW	DMA Register and Count Update mode 0 : Register only      1 : Register and count	1
9	RW	DMA Transfer Counter Register reload enable bit 0 : Internal Counter use at internal counter 1 : Transfer Counter Register uses at internal counter	0
8	RW	DMA Source/Destination Address Register reload enable bit 0 : Internal Counter use at internal counter 1 : Source/Destination Address Register uses at internal counter	0
7 : 6	RW	Source Address Direction 00 : Increase Address 01 : Decrease Address 1x : Fixing Address	0
5 : 4	RW	Destination Address Direction 00 : Increase Address 01 : Decrease Address 1x : Fixing Address	00
3 : 2	RW	Data Transfer Width 00 : 8bit 01 : 16bit 1x : 32bit	00
1	RW	Reserved	-
0	RW	Software DMA Enable bit 0 : Disable      1 : Software DMA Enable DMA가 완료되면 자동적으로 Disable된다. Software DMA Enable bit가 “0”인 상태에서 DREQ_SEL에 의해 선택된 인터럽트가 발생하면 Hardwire DMA가 실행된다.	0

**DMA Source Address Register (DMASRC)**

Address : 0x0300\_0D04h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 0	RW	DMA Source Address 32bit	0x0000_0000

**DMA Destination Address Register (DMADEST)**

Address : 0x0300\_0D08h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 0	RW	DMA Destination Address 32bit	0x0000_0000

**DMA Transfer Counter Register (DMATCNT)**

Address : 0x0300\_0D0Ch

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 16	R	Reserved	-
15 : 0	RW	16bit DMA Transfer Counter Register	0x0000

**DMA Request Source Select Register (DREQ\_SEL)**

Address : 0x0300\_0D10h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31	RW	OCR3B DMA Request	0
30	RW	OCR2B DMA Request	0
29	RW	OCR1B DMA Request	0
28	RW	OCR0B DMA Request	0
27	RW	OCR7A DMA Request	0
26	RW	OCR6A DMA Request	0
25	RW	OCR5A DMA Request	0
24	RW	OCR4A DMA Request	0
23	RW	OCR3A DMA Request	0
22	RW	OCR2A DMA Request	0
21	RW	OCR1A DMA Request	0
20	RW	OCR0A DMA Request	0
19	RW	Timer 7 DMA Request	0
18	RW	Timer 6 DMA Request	0
17	RW	Timer 5 DMA Request	0
16	RW	Timer 4 DMA Request	0
15	RW	Timer 3 DMA Request	0
14	RW	Timer 2 DMA Request	0
13	RW	Timer 1 DMA Request	0
12	RW	Timer 0 DMA Request	0
11	R	Reserved	0
10	R	Reserved	0
9	RW	ADC DMA Request	0
8	RW	TWI DMA Request	0
7	RW	UART Ch3DMA Request	0
6	RW	UART Ch2DMA Request	0
5	RW	UART Ch1 DMA Request	0
4	RW	UART Ch0 DMA Request	0
3	RW	SPI Ch1 DMA Request	0
2	RW	SPI Ch0 DMA Request	0
1	RW	USB DMA Request	0
0	RW	I2S DMA Request	0

\*\*\* DMA Request Source를 한번에 한가지만 선택하여야 한다.

## 14 I2S WITH ADPCM

TG471에 내장된 I2S with ADPCM은 외부의 I2S IC를 통한 audio 입출력과 내부 DAC을 통한 audio 출력, ADC를 통한 audio 입력이 가능하다.

ADPCM Engine은 16-bit PCM을 4-bit ADPCM으로의 Encoding과 압축된 4-bit ADPCM을 16-bit PCM으로의 Decoding을 지원한다.

### I2S Feature

- I2S-bus, MSB-justified format.
- DAC / ADC interface.
- ADPCM Engine (Encoding / Decoding).
- 4-bit ADPCM, 8 / 16-bit linear PCM data.
- 16, 32, 48, 64fs serial bit clock per channel. (fs : sampling frequency)
- 256, 384, 512fs master clock.
- 64 byte FIFO for transmits and receives.

### I2S Block Diagram

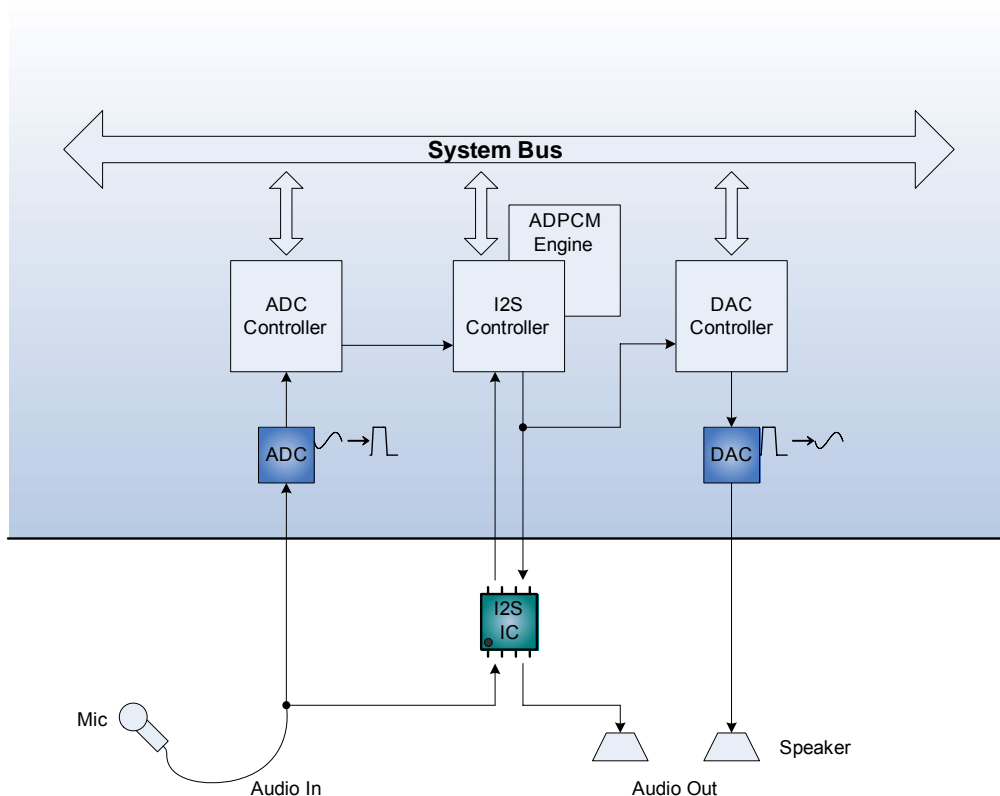


Figure 14-1 I2S and ADC / DAC Block Diagram



### 14.1 I2S SCLK, LRCK, MCLK (CODEC clock) Frequency

Table 14-1 sampling frequency and CODEC clock

LRCK (fs)	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
CODEC Clock (MHz)	256fs									
	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
	384fs									
	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640
	512fs									
	4.096	5.6448	8.1920	11.2896	16.3840	22.5792	24.5760	32.7680	45.1584	49.1520

예) MCLK (16.9344MHz) = 384 x LRCK (44.1kHz)

Table 14-2 sampling frequency and serial bit clock

Serial bit per channel	8-bit	16-bit
CODEC Clock	Serial clock frequency (SCK)	
256fs	16fs, 32fs	32fs
384fs	16fs, 32fs, 48fs	32fs, 48fs
512fs	16fs, 32fs, 48fs, 64fs	32fs, 48fs, 64fs

## 14.2 I2S Control Registers

### I2S Control Register ( I2SCON )

Address : 0x0300\_0E00h

Bit	R/W	Description	Default Value
31 : 6	R	Reserved	-
5	RW	Transmit interrupt active 0 : Disable 1 : Enable	0
4	RW	Receive interrupt active 0 : Disable 1 : Enable	0
3	RW	Transmit pause active 0 : Disable 1 : Enable	0
2	R	Reserved	-
1	RW	Receive ADC active 0 : Disable ( Default I2S or MSB format ) 1 : Enable	0
0	RW	I2S Enable 0 : Disable ( Stop ) 1 : Enable ( Start )	0

\*\*\* Transmit / Receive interrupt mode - FIFO의 half level Interrupt / Request signal를 발생시킨다.

\*\*\* Transmit pause active - Transmit 동작을 잠시 멈춘다.

\*\*\* Receive ADC active - 외부 ADC에서 data를 받을 때 사용된다.

\*\*\* I2S Enable - I2S의 동작에 필요한 LRCK와 SCLK를 발생시킨다.

**I2S Mode Register ( I2SMOD )**

Address : 0x0300\_0E04h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 14	R	Reserved	-
13	RW	Byte-Swap 32-bit 0 : Disable                      1 : Enable	0
12	RW	Byte-Swap 16-bit 0 : Disable                      1 : Enable	0
11	RW	Active level of left / right channel 0 : High for left channel ( Low for right channel ) 1 : Low for left channel ( High for right channel )	0
10	R	Reserved	-
9 : 8	RW	Transfer transmit / receive mode 00 : No transfer 01 : Receive 10 : Transmit 11 : Transmit and receive	0
7 : 6	RW	Quantization mode 00 : Not used 01 : 4-bit ADPCM 10 : 8-bit linear PCM 11 : 16-bit linear PCM	0
5	RW	Stereo mode 0 : Mono                      1 : Stereo	0
4	RW	Serial interface format 0 : I2S-bus format 1 : MSB(Left)-justified format	0
3 : 2	RW	Master clock frequency 00 : 256fs                      01 : 384fs 10 : 512fs                      11 : Not used	0
1 : 0	RW	Serial bit clock frequency 00 : 16fs                      01 : 32fs 10 : 48fs                      11 : 64fs	0

\*\*\* Byte-Swap - I2S data의 32-bit / 16-bit Byte-Swap을 할 때 사용된다.

\*\*\* Active level of left / right channel - Left / Right channel의 high와 low를 결정한다.

\*\*\* Transfer transmit / receive mode - I2S의 동작에서 transmit할 것인지 receive할 것인지 결정한다. 만약, 각각의 transfer를 off하면 해당 FIFO는 자동으로 clear된다.

\*\*\* Quantization mode - Transmit / Receive data의 quantization을 결정한다.

\*\*\* Stereo mode - Transmit / Receive data의 mono와 stereo를 결정한다.

\*\*\* Serial interface format / Master clock frequency / Serial bit clock frequency - I2S의 CODEC clock과 LRCK, SCLK를 결정한다.

**I2S Prescaler Register ( I2SPRE )**

Address : 0x0300\_0E08h

Bit	R/W	Description	Default Value
31 : 14	R	Reserved	-
4	RW	Master clock 0 : Source clock is system bus clock 1 : Source clock is external codec clock	0
3 : 0	RW	Prescaler Data value : 0 ~ 15 ( Division factor is N+ 1 )	0

\*\*\* Master clock – CODEC clock을 발생시키기 위해 source clock을 선택한다.

\*\*\* Prescaler – Source clock을 system bus clock으로 선택했을 경우, prescaler의 값으로 원하는 CODEC clock을 발생시킨다. External clock을 선택했을 경우는 이 설정 값은 reserved 된다.

**I2S Status Register ( I2SSTAT )**

Address : 0x0300\_0E0Ch

Bit	R/W	Description	Default Value
31 : 21	R	Reserved	-
20	R	Left / Right channel index When active level of left / right channel bit is '0', 0 : Right                    1 : Left	-
19	R	Transmit FIFO half level When transfer is transmit, 0 : more than half 1 : less than half	-
18	R	Receive FIFO half level When transfer is receive, 0 : less than half 1 : more than half	-
17	R	Transmit FIFO ready flag When transfer is transmit, 0 : Not ready ( Empty ) 1 : Ready ( Not empty )	-
16	R	Receive FIFO ready flag When transfer is receive, 0 : Not ready ( Full ) 1 : Ready ( Not full )	-
15 : 8	R	Transmit FIFO data count Data count value : 0 ~ 8	-
7 : 0	R	Receive FIFO data count Data count value : 0 ~ 8	-

\*\*\* Left / Right channel index – 현재 출력되고 있는 LRCK의 channel를 나타낸다.

**I2S Data Register ( I2SDATA )**

Address : 0x0300\_0E10h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 0	RW	I2S Data	0

Preliminary

## 15 USB DEVICE

TG471 에 내장된 USB Device는 2.0 Full-speed(12Mbps)를 지원하며, 5개의 endpoint으로 구성되어 있다.

하드웨어적으로 USB 프로토콜을 지원하며, 자동적인 data retry, data toggle 그리고 power management 기능(suspend와 resume)을 지원한다. 내부에 PHY 과 포함 되어 있다.

### USB Device Feature

- USB 2.0 Full Speed(12Mbps)
- 5 개의 Endpoint 지원
- 하드웨어적으로 USB 프로토콜 지원
- Suspend와 Resume signaling 지원

Table 15-1. Endpoint List

<i>Endpoint</i>	<i>Max Size (bytes)</i>	<i>Direction</i>	<i>Transaction Type</i>
0	16	IN/OUT	Control
1	64	OUT	Bulk
2	64	IN	Bulk
3	16	OUT	Interrupt
4	16	IN	Interrupt

## 15.1 USB Register Summary

Table 15-2. USB Core Register List

<i>Register</i>	<i>Address</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
USBFA	0x03000F00	R/W	Function address register	0x00
USBPM	0x03000F04	R/W	Power management register	0x00
USBEP1	0x03000F08	R/W	Endpoint interrupt register	0x00
USBINT	0x03000F10	R/W	USB interrupt register	0x00
USBEP1EN	0x03000F14	R/W	Endpoint interrupt enable register	0x1F
USBINTEN	0x03000F18	R/W	USB interrupt enable register	0x04
USBLBFN	0x03000F1C	R	Frame number1 register	0x00
USBHBFN	0x03000F20	R	Frame number2 register	0x00
USBIND	0x03000F24	R/W	Index register	0x00
USBMP	0x03000F28	R/W	MAXP register	0x00
USBEP0C	0x03000F2C	R/W	EP0 control register	0x00
USBIC1	0x03000F2C	R/W	EP2, 4 IN Control register1	0x00
USBIC2	0x03000F30	R/W	EP2, 4 IN Control register2	0x00
USBOC1	0x03000F38	R/W	EP1, 3 OUT Control register 1	0x00
USBOC2	0x03000F3C	R/W	EP1, 3 OUT Control register 2	0x00
USLBLBOWC	0x03000F40	R	Low Byte OEP Write count register	0x00
USBHBOWC	0x03000F44	R	High Byte OEP write count register	0x00
USBEP0D	0x03000F48	R/W	EP0 FIFO data register	0x00
USBEP1D	0x03000F4C	R/W	EP1 FIFO data register	0x00
USBEP2D	0x03000F50	R/W	EP2 FIFO data register	0x00
USBEP3D	0x03000F54	R/W	EP3 FIFO data register	0x00
USBEP4D	0x03000F58	R/W	EP4 FIFO data register	0x00

## 15.2 USB Function Address Register

USBFAR 레지스터에는 호스트에 의해 할당된 USB 디바이스 주소가 저장된다. MCU는 SET\_ADDRESS Descript 수행을 통해 받은 값을 이 레지스터에 저장한다. 이 값은 다음 토큰에서 사용된다.

## 15.3 USB Power Management Register

Power Management 레지스터는 Suspend, Resume 그리고 reset 신호에 의해 사용된다. Suspend와 Reset 상태는 USB\_INTERRUPT Register에 저장된다.

## 15.4 USB Interrupt Registers

USB Host의 요청상태와 각 Endpoint 의 상태와 알려준다.

## 15.5 USB Interrupt Enable Registers

각 Endpoint의 인터럽트를 Enable 한다. 대부분의 인터럽트는 초기값이 Enable상태이나, Suspend 인터럽트는 Disable 이다.

## 15.6 Frame Number Registers

Frame Packet의 끝에서 frame 번호를 저장한다.

## 15.7 Index Register

인덱스 레지스터는 각각의 endpoint에 해당하는 컨트롤 레지스터를 선택할 때 사용한다.

## 15.8 MAXP Register

8byte 배수 단위로 사용할 FIFO 크기를 조절할 수 있다. 그러나 각 Endpoint 에서 지원하는 최대 FIFO 사이즈보다 크기는 설정 할 수 없다.



## 15.9 EP0 Control Register

Endpoint 0의 제어와 상태를 나타낸다.

## 15.10 IN Control Registers

IN Endpoint의 제어와 상태를 나타낸다.

## 15.11 Out Control Registers

Out Endpoint의 제어와 상태를 나타낸다.

## 15.12 Out Write Count Registers

두 개의 레지스터로 이루어져 write count 값을 가지다. OUT endpoint에서 OPOPR 비트가 set 되면, 이 레지스터에는 MCU에 의해 가져간 packet의 수를 가지고 있다.

## 15.13 Endpoint FIFO Access Registers

FIFO에 register 이다.

## 15.14 USB Control Registers

### USB Function Address Register (USBFA)

Address : 0x0300\_0F00h

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 8			Reserved	
7	R/W	R/ Clear	ADDUP : ADDR_UPDATE bit. Function 주소가 업데이트 되면 MCU 가 이 비트를 1로 설정한다. Endpoint 0 CSR의 DATA_END 비트를 clear 에 의해 알 수 있는 제어 전송의 status 단계 이후에 FUNCTION_ADDR 부분은 사용된다.	0
6 : 0	R/W	R	FUNADD : FUNCTION_ADDR bit. MCU가 주소를 여기에 write 한다.	0

### USB Power Management Register (USBPM)

Address : 0x0300\_0F04h

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 4			Reserved	
3	R	Set	UBRST : USB_RESET bit. 호스트로부터 Reset 신호를 받으면 USB가 이 비트를 설정한다. Reset 신호가 버스상에서 유지되는 한 ,이 비트는 set 상태를 유지한다.	0
2	W/R	R	UBRSUM : USB_RESUME bit. Resume 신호를 초기화 하기 위해 MCU가 10ms ( 최대 15ms)동안 이 비트를 설정한다. Suspend 모드에서 이 비트가 설정되어 있는 동안 USB 가 Resume 신호를 발생한다.	0
1	R	R/W	UBSPDMOD : SUSPEND_MODE bit. Suspend모드로 들어가게 되면 USB 가 이 비트를 설정한다. 다음 조건에 의해 clear 가 된다. -Resume 신호를 끝내기 위해서 MCU가 MUC_RESUME 를 clear 하는 경우 -USB_RESUME 인터럽트 발생 때 MCU가 인터럽트 레지스터 3 을 읽게 되는 경우.	0
0	R/W	R	UBENSPD : ENABLE_SUSPEND bit = 1 Enable Suspend mode = 0 Disable Suspend mode (Default) 이 비트가 zero 이면, 디바이스는 suspend 모드 상태로 들어 가지 않는다.	0

**USB Endpoint Interrupt Register (USBEP1)**

Address : 0x0300\_0F08h

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 5			Reserved	
4	R/ Clear	Set	EP4INT : EP4 Interrupt bit. (Interrupt in mode) 이 비트는 endpoint4 인터럽트에 해당된다. (USBIC1R, USBIC2R 의 bit 참고) - ICIPR(In Control 1 In Packet Ready bit) 비트가 clear 될 때 - FIFO가 flush 되었을 때 - ICSTSTAL(In Control 1 Sent Stall bit) 비트가 set 되었을 경우에	0
3	R/ Clear	Set	EP3INT : EP3 Interrupt bit. (Interrupt out mode) 이 비트는 endpoint3 인터럽트에 해당된다. (USBOC1R, USBOC2R 의 bit 참고) - OCOPR(Out Control 1 Out Packet Ready bit) 비트를 set 할 때 - OCSTSTAL(Out Control 1 Sent Stall bit ) 비트를 set 할 때	0
2	R/ Clear	Set	EP2INT : EP2 Interrupt bit. (Bulk in mode) 이 비트는 endpoint2 인터럽트에 해당된다. (USBIC1R, USBIC2R 의 bit 참고) - ICIPR(In Control 1 In Packet Ready bit) 비트가 clear 될 때 - FIFO가 flush 되었을 때 - ICSTSTAL(In Control 1 Sent Stall bit) 비트가 set 되었을 경우에	0
1	R/ Clear	Set	EP1INT : EP1 Interrupt bit. (Bulk out mode) 이 비트는 endpoint1 인터럽트에 해당된다. (USBOC1R, USBOC2R 의 bit 참고) - OCOPR(Out Control 1 Out Packet Ready bit) 비트를 set 할 때 - OCSTSTAL(Out Control 1 Sent Stall bit ) 비트를 set 할 때	0
0	R/ Clear	Set	EPOINT : EP0 Interrupt bit. (Control mode) 이 비트는 endpoint0 인터럽트에 해당된다. (USBEP0CR 의 bit 참고) 1. EP0OPR bit is set. 2. EP0IPR bit is cleared 3. EPOSTSTAL bit is set 4. EPOSTED bit is set 5. EPODED bit is cleared(Indicates End of control transfer)	0

**USB Interrupt Register (USBINT)**

Address : 0x0300\_0F10h

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 3			Reserved	
2	R/ Clear	Set	RSTINT : USB Reset Interrupt bit. Reset신호가 입력되면 USB가 이 비트를 set 한다.	0
1	R/ Clear	Set	RSUMINT : Resume Interrupt bit. Suspend 모드 상태에서 Resume신호를 받으면 USB가 이 비트를 set한다. USB Reset에 의한 Resume 이면, Resume 인터럽트에 의해 MCU에 먼저 인터럽트가 걸린다. 일단 Clock이 다시 동작하고 SE0 상태가 3ms 동안 지속되면, USB Reset 인터럽트가 발생한다..	0
0	R/ Clear	Set	SPDINT : Suspend Interrupt bit Suspend 신호를 수신하면 USB는 이 비트를 set 한다. 버스상에서 3ms 동안 아무런 동작이 이루어지지 않으면 이 비트는 set 된다. 그래서 MCU가 첫 번째 suspend 인터럽트 이후에 Clock을 멈추지 않으면, USB 버스상에서 아무런 동작이 이루어지 않는 한 매 3ms 마다 인터럽트가 계속 발생한다. 디폴트로 이 인터럽트는 disable 이다.	0

**Endpoint Interrupt Enable Register (USBEPIN)**

Address : 0x0300\_0F14h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 5	R	Reserved	
4	R/W	EP4INTEN : Endpoint 4 Interrupt enable bit	1
3	R/W	EP3INTEN : Endpoint 3 Interrupt enable bit	1
2	R/W	EP2INTEN : Endpoint 2 Interrupt enable bit	1
1	R/W	EP1INTEN : Endpoint 1 Interrupt enable bit	1
0	R/W	EPOINTEN : Endpoint 0 Interrupt enable bit	1

**USB Interrupt Enable Register (USBINTEN)**

Address : 0x0300\_0F18h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 3	R	Reserved	
2	R/W	RSTINTEN : USB RESET Interrupt enable bit	1
1	R	Reserved	
0	R/W	SPDINTEN : SUSPEND Interrupt enable bit	0

**USB Low Byte Frame Number Register (USBLBFN)**

Address : 0x0300\_0F1Ch

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	
7 : 0	R/W	Frame Number 1 register	0x00

**USB High Byte Frame Number Register (USBHBFN)**

Address : 0x0300\_0F20h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	
7 : 0	R/W	Frame Number 2 register	0x00

**USB Index Register (USBIND)**

Address : 0x0300\_0F24h

Bit	R/W	Description	Default Value
31 : 3	R	Reserved	
2 : 0	R/W	Index register 000 : Endpoint 0      001 : Endpoint 1 010 : Endpoint 2      011 : Endpoint 3 100 : Endpoint 4      101 : Reserved 110 : Reserved        111 : Reserved	000

**USB MAXP Register (USBMP)**

Address : 0x0300\_0F28h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	
7 : 0	R/W	Max FIFO Size 0000_0000      MAXP=8 0000_0010      MAXP=16 0000_0100      MAXP=32 0000_1000      MAXP=64	0x00

**USB EPO Control Register (USBEP0C)**

Address : 0x0300\_0F2Ch

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 8	R		Reserved	
7	Clear		EPOSUEC : EPO Set Up End Clear bit. MCU가 EPOSTED 비트를 clear 하기 위해 1를 write 한다.	0
6	Clear		EPOOPRC : EPO Out Packet Ready Clear bit. MCU는 EPOOPR 비트를 clear하기 위해 이 비트에 1를 write한다.	0
5	Set	Clear	EPOSDSTAL : EPO Send Stall bit. MCU는 잘못된 token이라고 인식되면, EPOOPR 비트를 clear와 동시에 이 비트를 set 한다. USB는 STALL handshake를 현재 컨트롤 전송에 발생시킨다. MCU는 STALL 상황을 끝내기 위해 0를 write 한다.	0
4	R	Set	EPOSTED : EPO Setup End bit. 이 비트는 읽기 전용이다. EPODED 비트가 set되기 전에 컨트롤 전송이 끝났을 때 USB 가 이 비트를 set한다. USB가 이 비트를 set 할 때 MCU에 인터럽트가 전달된다. 이러한 상황이 발생했을 때 USB는 FIFO를 flush하고 MCU의 FIFO 접근을 무효화 한다. MCU의 FIFO 접근이 무효화 될 때 이 비트는 clear 된다.	0
3	Set/R	Clear	EPODED : EPO Data End bit. MCU는 다음과 같은 상황에서 이 비트 set한다: - 마지막 데이터 패킷을 가져온 후 EPOOPR 비트를 clear 할 때 - Zero length data 구간에서 EPOOPR 비트를 clear 하고 EPOIPR 비트를 set 할 때 - MCU가 FIFO에 대한 패킷 데이터를 load한 후에 EPOIPR 비트를 set함과 동시에 이 비트(EPODED) 를 set 한다.	0
2	Clear /R	Set	EPOSTSTAL : Sent Stall bit. 프로토콜 오류로 컨트롤 transaction이 끝나면 USB가 이 비트 set 한다. 이 비트가 set 되면 인터럽트가 발생한다.	0
1	Set/R	Clear	EPOIPR : EPO In Packet Ready bit. MCU는 endpoint 0 FIFO에 데이터 패킷을 write 한 후에 이 비트를 set 한다. 데이터 패킷이 성공적으로 호스트에 전달되면 USB가 이 비트를 clear 시킨다. USB가 이 비트를 clear시키면 인터럽트가 발생한다. 그래서 MCU는 계속해서 다음 데이터를 load 할 수 있게 된다. Zero length data phase에서는 MCU는 동시에 이 비트(EPOIPR)와 EPODED 비트를 set 한다.	0
0	R	Set	EPOOPR : EPO Out Packet Ready bit. 이 비트는 읽기 전용이다. 유효한 token이 FIFO에 쓰여	0

			지면 USB가 이 비트를 set 한다. USB가 set 하면 인터럽트가 발생한다. MCU는 EP0OPRC 비트에 1를 write 함으로써 이 비트를 clear 시킨다.	
--	--	--	---	--

Preliminary



**USB IN Control 1 Register (USBIC1)**

Address : 0x0300\_0F2Ch

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 7	R		Reserved	
6	Set	R/Clear	ICCDT : In Control 1 Clear Data Toggle bit. MCU가 이 비트에 1을 write하면 data toggle 비트가 clear 된다. 이 비트는 쓰기 전용이다.	0
5	R/Clear	Set	ICSTSTAL : In Control 1 Sent Stall bit. MCU가 ICSDSTAL 비트를 set 했기 때문에, IN token에 STALL handshake를 발생된다. 이 때 USB 가 이 비트를 set 한다. USB 가 STALL handshake를 발생 시키면 ICIPR 비트는 clear된다. MCU 가 0를 write함으로써 이 비트를 clear 시킨다.	0
4	R/W	R	ICS DSTAL : In Control 1 Send Stall bit. MCU가 USB에 STALL handshake를 발생시키기 위해 이 비트에 1를 write한다. STALL 상황을 끝내기 위해 MCU가 이 비트를 clear 한다.	0
3	R/Set	Clear	ICFFLU : In Control 1 FIFO Flush bit. IN FIFO를 flush하고자 하면 MCU가 이 비트를 set 한다. FIFO가 flush가 되면 USB 에 의해 이 비트는 clear 된다. 이런 상황이 발생하면 MCU에 인터럽트가 걸린다. Token이 진행 중이라면, USB는 FIFO가 flush 되기 전에 전송이 완료 될 때까지 기다린다. 만약에 두 개의 패킷이 FIFO에 load되어 있으면, 가장 상위의 패킷(호스트로 보내려고 하는 것)만 flush이 되고 그 패킷에 관련 있는 ICIPR 비트가 clear 된다.	0
2			Reserved	0
1	R	Set	ICFNE : In Control 1 FIFO Not Empty bit. FIFO에 적어도 한 개의 데이터 패킷이 있음을 나타 내다. 0 : FIFO에 패킷이 없다. 1 : FIFO에 패킷이 있다.	0
0	Set / R	Clear	ICIPR : In Control 1 In Packet Ready bit. FIFO에 데이터 패킷을 쓰고 난 뒤 MCU가 이 비트를 set 한다. 호스트로 데이터 패킷 전송이 성공적으로 끝나면 USB는 이 비트를 clear 한다. 이 비트를 USB 가 clear 하면 인터럽트가 발생하고, MCU는 다음 패킷을 로드 할 수 있게 된다. 이 비트가 set 되어 있는 동안에는 MCU는 FIFO에 쓰기를 할 수 없다. MCU에 의해 ICSDSTAL 비트가 set 되면, 이 비트는 set 될 수 없다.	0

**USB IN Control 2 Register (USBIC2)**

Address : 0x0300\_0F30h

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 8	R		Reserved	
7	R/W	R	ICASET : In Control 2 Auto Set bit. 이 비트가 set 되어 있으면, MCU가 MAXP만큼의 데이터를 쓰기를 하면 자동적으로 ICIPR 비트가 set 된다. MAXP데이터 보다 적은 데이터를 쓸 경우는 MCU가 ICIPR 비트를 set 해줘야 한다.	0
6			Reserved	0
5	R/W	R	ICMODIN : In Control 2 Mode In bit. Endpoint의 방향을 프로그래머블할 수 있게끔 해준다. 1 = endpoint의 방향을 IN으로 설정된다. 0 = endpoint의 방향을 OUT으로 설정된다.	1
4 : 0			Reserved	

**USB Out Control Register 1 (USBOC1)**

Address : 0x0300\_0F38h

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 8	R		Reserved	
7	R/W	R	OCDDT : Out Control 1 Clear Data Toggle bit. MCU가 이 비트에 1를 write하면, data toggle sequence 비트가 DATA0로 reset 된다.	0
6	Clear/ R	Set	OCSTSTAL : Out Control 1 Sent Stall bit. OUT token이 STALL handshake로 종료될 때 USB가 이 비트 set 한다. OUT Token에서 MAXP 데이터 보다 더 많은 데이터를 보낼 경우 USB가 host에 stall handshake를 발생 시킨다. MCU가 0를 write하면 clear 된다.	0
5	W/R	R	OCSDSTAL : Out Control 1 Send Stall bit. USB에 STALL handshake를 발생시키기 위해 MCU가 이 비트에 1를 write 한다. STALL 상황을 끝내기 위해 MCU가 이 비트에 0을 write 한다.	0
4	R/W	Clear	OCFFLU : Out Control 1 FIFO Flush bit. MCU가 FIFO를 flush하기 위해 1를 write 하고 flush를 멈추기 위해 0을 write 한다. OCOPR 비트가 set되어 있는 동안만 이 비트가 set 될 수 있다. MCU가 가져간 데이터 패킷은 flush가 될 것이다.	0
3	R	R/W	OCERR : Out Control 1 Data Error bit 전송받은 데이터에 에러(bit stuffing 또는 CRC)가 있음을 나타낸다. OCOPR 비트가 clear될 때 자동적으로 clear 된다.	0
2	R	R	Reserved	
1	R	R/W	OCFFUL : Out Control 1 FIFO Full bit. 더 이상의 패킷을 수용할 수 없음을 나타낸다. 0 : FIFO is not full. 1 : FIFO is full.	0
0	R/ Clear	Set	OCOPR : Out Control 1 Out Packet Ready bit. FIFO에 데이터 패킷이 load가 되면 USB가 이 비트를 set 한다. MCU가 패킷 전체를 읽고 나면 이 비트는 MCU에 의해 clear 되어야 한다. MCU가 0을 write 함으로써 clear 된다.	0

**USB OUT Control Register 2 (USBOC2)**

Address : 0x0300\_0F3Ch

Bit	R/W		Description	Default Value
	MCU	USB		
31 : 8	R		Reserved	
7	R/W	R	OCACLR : Out Control 2 Auto Clear bit. 이 비트가 set이면, MCU가 OUT FIFO에서 데이터를 읽을 때 마다 자동적으로 USB core에 의해 OCOPR 비트가 clear 된다.	0
6 : 0			Reserved	0

**USB Low Byte Out Write Count Register (USBLOWC)**

Address : 0x0300\_0F40h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	
7 : 0	R/W	(LBOWC) Low Byte OEP write count register	0x00

**USB High Byte Out Write Count Register (USBHBOWC)**

Address : 0x0300\_0F44h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	
7 : 0	R/W	(HBOWC) High Byte OEP write count register	0x00

**EP0 FIFO Data Register (USBEP0)**

Address : 0x0300\_0F48h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7 : 0	R/W	EP0 FIFO Data Register	0x00

**EP1 FIFO Data Register (USBEP1)**

Address : 0x0300\_0F4Ch

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7 : 0	R/W	EP1 FIFO Data Register	0x00

**EP2 FIFO Data Register (USBEP2)**

Address : 0x0300\_0F50h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7 : 0	R/W	EP2 FIFO Data Register	0x00

**EP3 FIFO Data Register (USBEP3)**

Address : 0x0300\_0F54h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7 : 0	R/W	EP3 FIFO Data Register	0x00

**EP4 FIFO Data Register (USBEP4)**

Address : 0x0300\_0F58h

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7 : 0	R/W	EP4 FIFO Data Register	0x00

## 16 ADC

500KSPS, 10bit ADC 를 4 채널 내장하였다. 그러나 내부적으로는 한 개의 ADC를 사용하여 제어하기 때문에 4 채널을 모두 사용하기 위해서는 ADC\_SEL을 통해 사용할 ADC 입력을 선택하여야 한다.

ADC를 통해 변환된 디지털 데이터는 ADC의 ADCDAT 레지스터와 I2S의 I2SDAT 레지스터로 동시에 전달된다. ADC를 제어용으로 사용할 때에는 ADCDAT 레지스터를 읽어서 변환된 데이터를 처리하고, 음성녹음을 위해 ADC를 사용할 경우에는 I2SDAT를 통해 ADPCM Format으로 압축한다.

### ADC Features

- Resolution : 10 bit
- Differential Linearity Error :  $\pm 1.0$  LSB
- Integral Linearity Error :  $\pm 2.0$  LSB
- Signal to Noise rate : 56dB
- Maximum Conversion Rate : 500KSPS(10 bit)

ADC를 동작시켜 Digital Data를 얻기 위해서는 다음 단계에 따라 ADC를 제어하여야 한다.

- ADC Input Selection, ADC Clock Selection
- ADC Power On
- ADC Start Of Conversion Enable
- ADC Data Ready Check or ADC Interrupt Event
- ADC Data Read

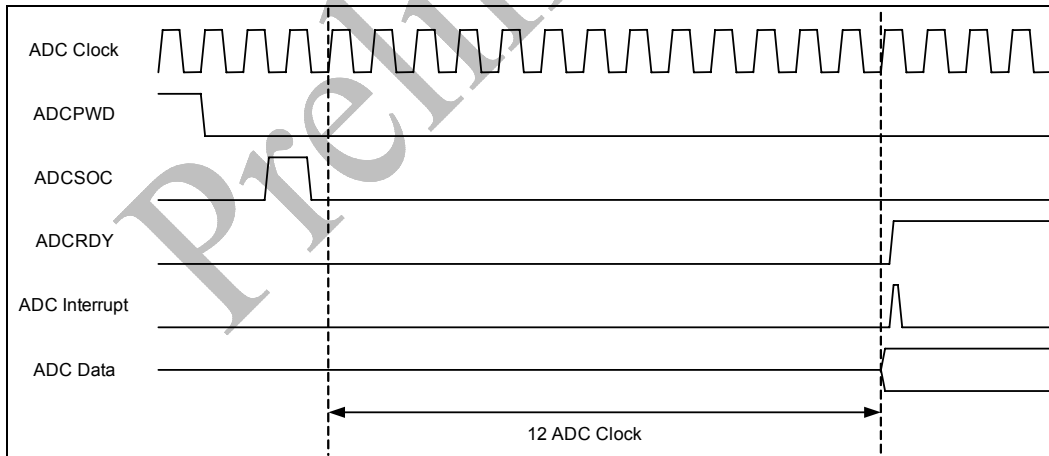


Figure 16-1 ADC Control Timing

## 16.1 ADC Control Registers

### ADC Control Register (ADCCON)

Address : 0x0300\_0B00h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	R	ADCRDY : ADC Data Ready Bit At Read, this bit is cleared	0
6	R	Reserved	-
5 : 4	RW	AINSEL : ADC Input Channel Selection 00 : ADC_IN0            01 : ADC_IN1 10 : ADC_IN2           11 : ADC_IN3	000
3 : 2	RW	ACLKSEL : ADC Clock Select bit 00 : System Clock / 4 01 : System Clock / 8 10 : System Clock / 16 11 : System Clock / 32	11
1	RW	ADCSOC : Start of Conversion 0 : No Conversion    1 : Start of Conversion	0
0	RW	ADCPWD : ADC Power Down 0 : ADC Power On    1 : ADC Power Down	1

### ADC Data Register (ADCDAT)

Address : 0x0300\_0B04h

Bit	R/W	Description	Default Value
31 : 10	R	Reserved	-
9 : 0	R	10bit ADC Data	0

## 17 DAC

ADPCM 음성출력과 제어용에 적합한 12bit DAC을 내장하였다. DAC으로 출력될 데이터는 MCU에서 제어한 데이터와 ADPCM에서 출력되는 데이터를 선택할 수 있다.

### DAC Features

- Resolution : 12 bit
- Differential Linearity Error :  $\pm 4.0$  LSB
- Integral Linearity Error :  $\pm 1.0$  LSB
- Maximum Conversion Rate : 2MHz

### 17.1 DAC Control Registers

#### DAC Control Register (DACCON)

Address : 0x0300\_0C00h

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
1	RW	DACSEL : DAC Data Select 0 : ADPCM Data      1 : MCU Data	0
0	RW	DACEN : DAC Power Down 0 : DAC Power Down   1 : DAC Power On	0

#### DAC Data Register (DACDAT)

Address : 0x0300\_0C04h

Bit	R/W	Description	Default Value
31 : 12	R	Reserved	-
11 : 0	RW	12bit DAC Data ( ADPCM Data or MCU Data)	0



# 18 IN SYSTEM PROGRAMMER (ISP)

ISP는 “SPI Mode 0” 과 “SPI Mode 3” 에 호환되는 interface를 가진다. ECLK 주기는 XIN 의 8배 이상의 주기를 가져야 한다.(Tcss/Tcsh/Tsck Period > 8 x XIN Period )

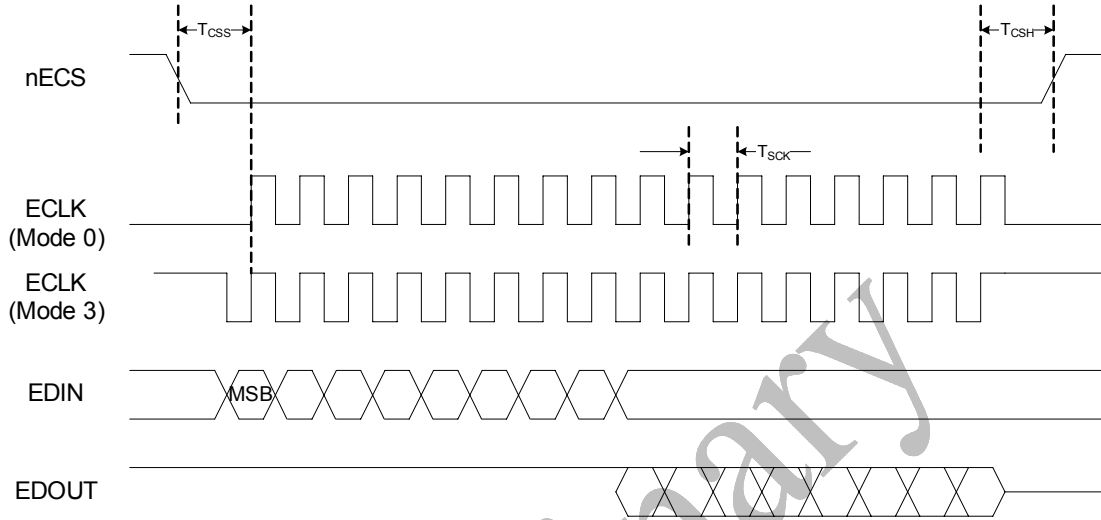


Figure 18-1 SPI Modes Supported

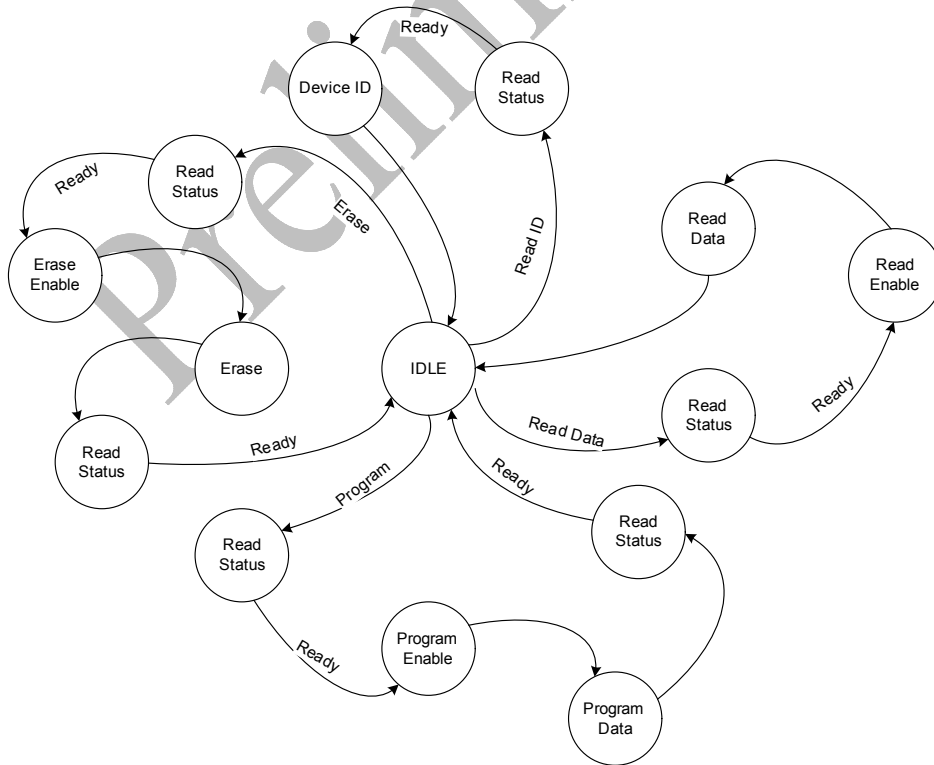


Figure 18-2 ISP Flow Chart

### 18.1 ISP Command Set

Flash Command Set 을 다룰 때는 다음 주의가 필요하다.

- Byte Programming과 Auto Address Increment Programming 을 수행하기 위해서는 Program Enable 이 먼저 실행되어야 한다.
- Sector Erase 을 수행하기 위해서는 Sector Erase Enable 이 먼저 수행되어야 한다.
- Block Erase 을 수행하기 위해서는 Block Erase Enable 이 먼저 수행되어야 한다.
- Flash Erase 을 수행하기 위해서는 Flash Erase Enable 이 먼저 수행되어야 한다.
- Programming과 Erase 가 완료되면 Programming Disable과 Erase Disable 명령을 수행하여 Flash 내용이 예기치 않은 동작에 의해 변경되는 것을 방지 하여야 한다.

Table 18-1 ISP Command Set

Bus cycle	1 cycle		2 cycle		3 cycle		4 cycle		5 cycle		6 cycle	
	SI	SO	SI	SO	SI	SO	SI	SO	SI	SO	SI	SO
Read Data	03h	Z	A2	Z	A1	Z	A0	Z	X	X	X	DO
Sector Erase	20h	Z	A2	Z	A1	Z	A0	Z	-	-	-	-
Block Erase	52h	Z	A2	Z	A1	Z	A0	Z	-	-	-	-
Chip Erase	60h	Z	A2	Z	A1	Z	A0	Z	-	-	-	-
Programming	02h	Z	A2	Z	A1	Z	A0	Z	DI	Z	-	-
Auto Address Increment Programming	AFh	Z	A2	Z	A1	Z	A0	Z	DI	Z	-	-
Read Enable	16h	Z	-	-	-	-	-	-	-	-	-	-
Read Disable	14h	Z	-	-	-	-	-	-	-	-	-	-
Chip Erase Enable	26h	Z	-	-	-	-	-	-	-	-	-	-
Block Erase Enable	27h	Z	-	-	-	-	-	-	-	-	-	-
Sector Erase Enable	28h	Z	-	-	-	-	-	-	-	-	-	-
Erase Disable	24h	Z	-	-	-	-	-	-	-	-	-	-
Programming Enable	36h	Z	-	-	-	-	-	-	-	-	-	-
Programming Disable	34h	Z	-	-	-	-	-	-	-	-	-	-
Read Status	05h	Z	X	ST	-	-	-	-	-	-	-	-
Read ID	90h	Z	X	ID	-	-	-	-	-	-	-	-

- SI : Serial Input,
- SO : Serial Output
- X : Dummy cycles
- : Cycles are not necessary
- A2 : Address[23:16]
- A1 : Address[15:8]
- A0 : Address[7:0]
- DI : Data input [7:0]
- DO : Data output[7:0]
- ID : Device Identification (0xAC)
- ST : Status

### 18.2 Device Identification and Status Commands

Read ID Command를 통해 TG471의 전원상태와 ISP Cable의 상태를 확인할 수 있으며 “0xAC”가 읽어지면 ISP의 연결이 정상적으로 이루어졌음을 의미한다.

ISP 동작시 ISP Controller의 상태는 Read Status Command를 통해 확인할 수 있다.

Table 18-2 ISP Status Register

Bit	R/W	Description	Default Value
31 : 7	R	Reserved	-
6	R	<b>Auto address increment programming mode bit</b> 0 : Random program mode 1 : Auto address increment programming mode	0
5	R	<b>Programming enable bit</b> 0 : Programming disabled 1 : Programming enabled	0
4	R	<b>Flash Erase enable bit</b> 0 : Flash erase disabled 1 : Flash erase enabled	0
3	R	<b>Block Erase enable bit</b> 0 : Block erase disabled 1 : Block erase enabled	0
2	R	<b>Sector Erase enable bit</b> 0 : Sector erase disabled 1 : Sector erase enabled	0
1	R	<b>Read enable bit</b> 0 : Read disabled 1 : Read enabled	0
0	R	<b>Flash Ready bit</b> 0 : Busy with an operation(erase/program) in progress 1 : Ready for next valid operation (read/erase/program)	1

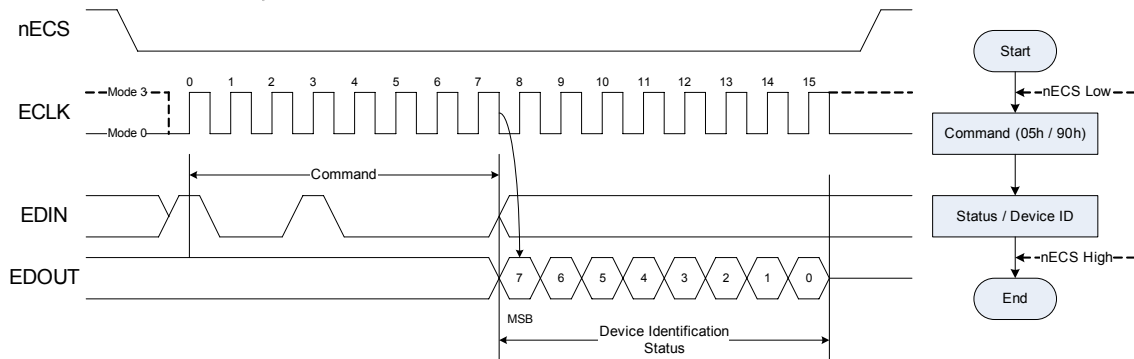


Figure 18-3 Read ID/Status Timing and Flow Chart

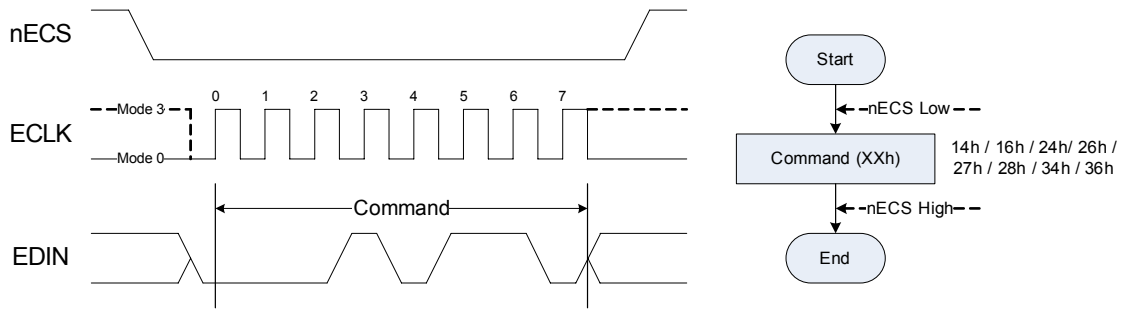


Figure 18-4 ISP Status Setting Timing and Flow Chart

### 18.3 Data Read Command

Flash에 저장된 Data는 Read Data Command를 통해 읽히며, 연속된 Flash영역을 접근할 경우에는 시작 주소를 첫 한번만 인가해주면 된다.

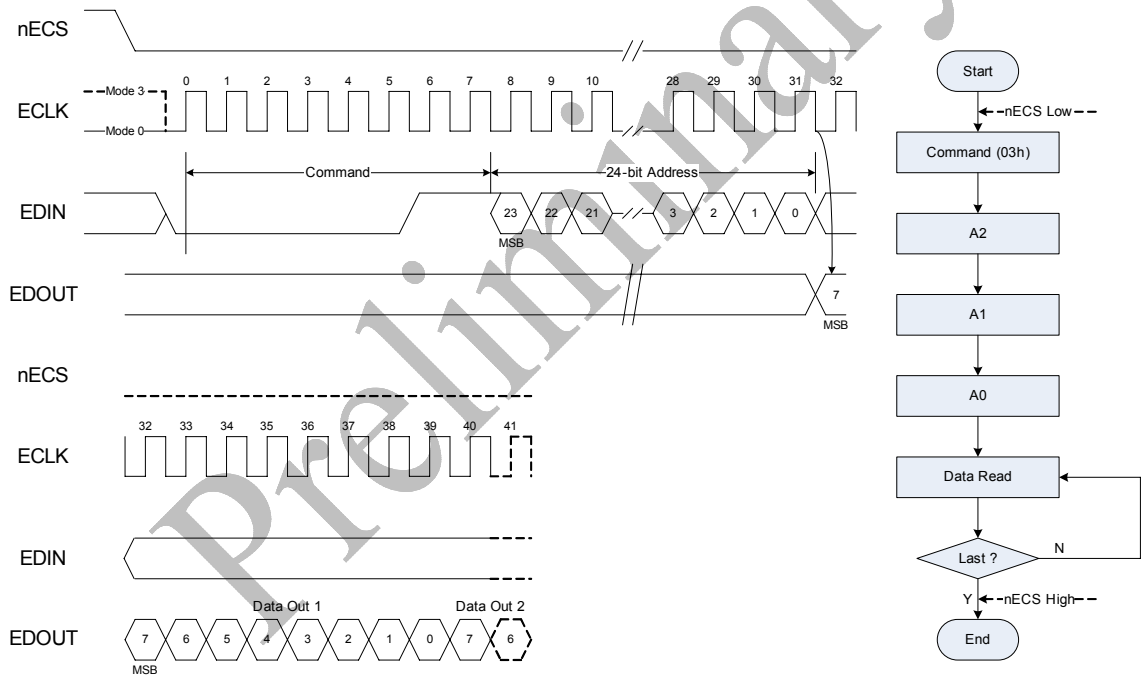


Figure 18-5 ISP Data Read Timing and Flow Chart

### 18.4 Programming Commands

Programming Command는 Byte 단위로 24bit Address를 인가하는 Byte Programming 방식과 첫 24bit Address를 인가한 후에는 Programming에 필요한 데이터만 인가하는 방식인 Auto Address Increment Programming 방식을 지원한다. Auto Address Increment Programming 방식을 사용하면 ISP 속도를 높일 수 있다.

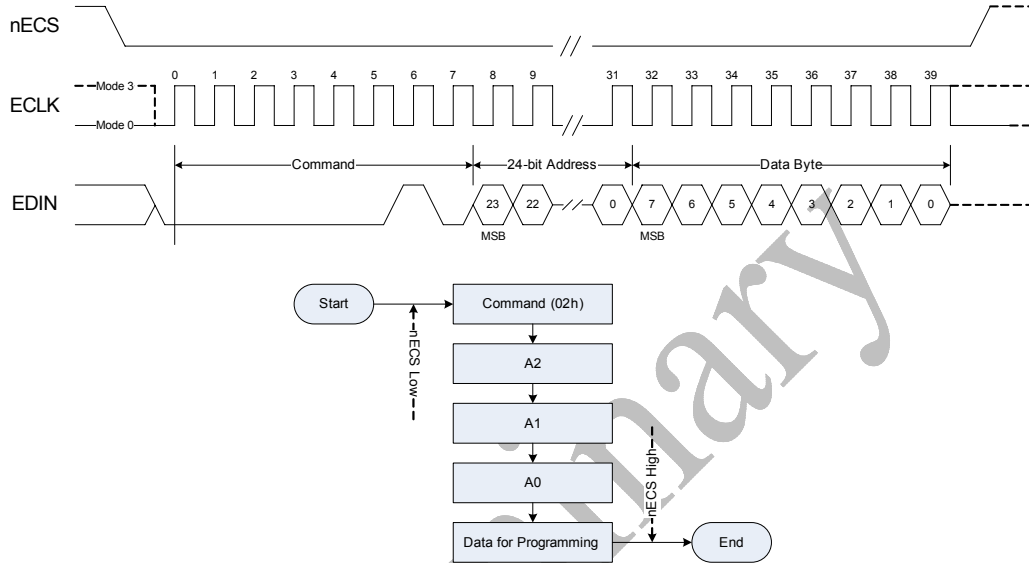


Figure 18-6 Byte Programming Timing and Flow Chart

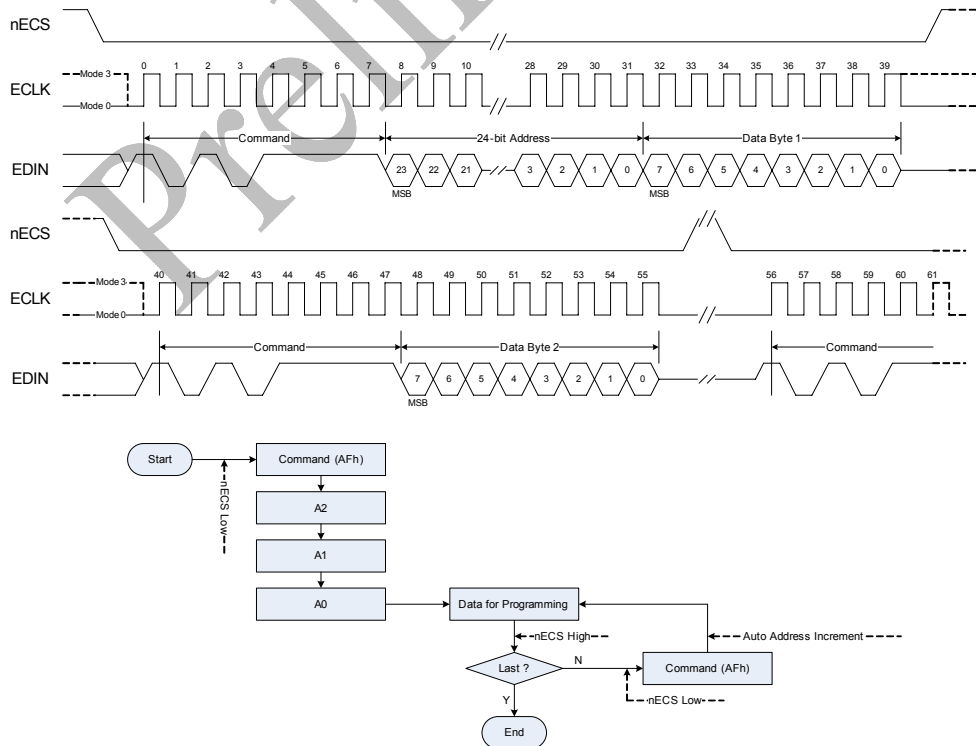


Figure 18-7 AAI Programming Timing and Flow Chart

### 18.5 Erase Commands

Flash에 대한 Erase Command는 Sector/Block/Chip 의 3가지 레벨로 나누어진다. TG471에 내장된 NOR Flash 는 Programming에 의해 Flash 데이터를 “1 ->0” 로 변경할 수 있지만, “0 ->1” 로는 변경할 수 없기 때문에 “0 -> 1”로 변경하기를 원하는 bit가 Flash내에 존재할 때에는 Erase Command를 사용하여야 한다.

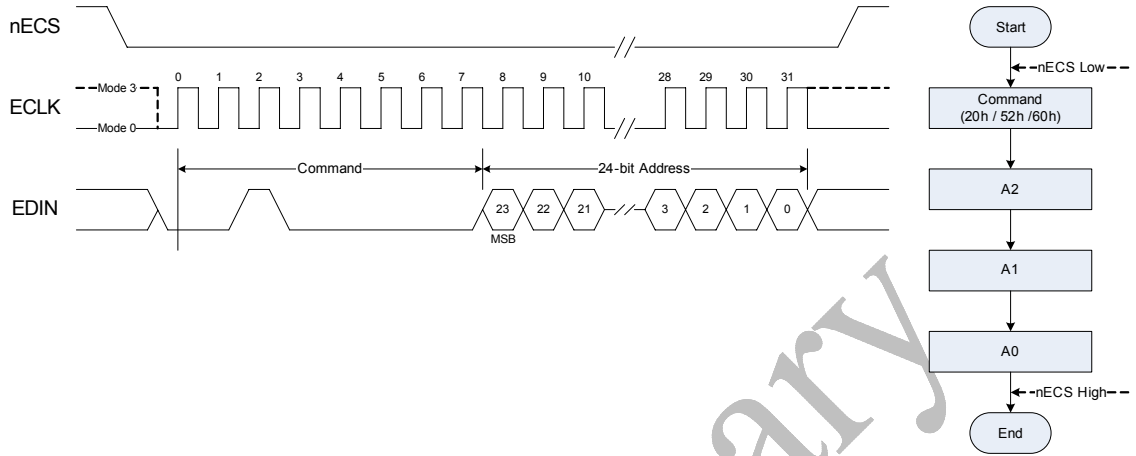


Figure 18-8 Erase Commands Timing and Flow Chart

## 19 ELECTRICAL DATA

### 19.1 DC ELECTRICAL CHARACTERISTICS

Table 19-1 DC Electrical Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Unit	
I/O Supply Voltage	VDD33		3.0	3.3	3.6	V	
Junction Temperature	T <sub>J</sub>		-40	25	125	°C	
Operation Temperature	T <sub>A</sub>		0	25	70	°C	
Input Low Voltage	V <sub>IL</sub>		-0.3	-	0.8	V	
Input High Voltage	V <sub>IH</sub>		2.0	-	5.5	V	
Threshold point	V <sub>T</sub>		1.24	1.38	1.57	V	
Schmitt Trigger Low to High threshold point	V <sub>T+</sub>		1.69	-	-	V	
Schmitt Trigger High to Low threshold point	V <sub>T-</sub>		-	-	1.06	V	
Input leakage current	I <sub>I</sub>	V <sub>I</sub> = 3.3V or 0V	-	-	±1	uA	
Tri-state output leakage current	I <sub>OZ</sub>	V <sub>O</sub> = 3.3V or 0V	-	-	±1	uA	
Pull-up resistor	R <sub>PU</sub>		90	120	195	KΩ	
Pull-down resistor	R <sub>PD</sub>		60	130	176	KΩ	
Output Low Voltage	V <sub>OL</sub>	I <sub>OL</sub> = 4, 8mA	-	-	0.4	V	
Output High Voltage	V <sub>OH</sub>	I <sub>OH</sub> = 4, 8mA	2.4	-	-	V	
Oscillator Frequency	F <sub>OSC</sub>		0		33	MHz	
Crystal Frequency	F <sub>CRY</sub>		1		10	MHz	
Operating current	Normal mode	I <sub>DD</sub>	F <sub>MCLK</sub> = 24MHz	-	37.2	-	mA
			F <sub>MCLK</sub> = 12MHz	-	20.4	-	mA
			F <sub>MCLK</sub> = 6MHz	-	10.4	-	mA
	Sleep mode		F <sub>MCLK</sub> = 6KHz	-	1.6	-	mA
Stop mode current	I <sub>DD</sub>		-	250	-	uA	

$$V_{DD} = 3.3V \pm 0.3V$$

$$V_{EXT} = 5 \pm 0.25V$$

## 19.2 Voltage Regulator Electrical Characteristics

Table 19-2 Electrical Characteristics

<i>Parameter</i>	<i>Min</i>	<i>Typ</i>	<i>Max</i>	<i>Unit</i>
Supply Voltage	3.0	3.3	3.6	V
Voltage Output	2.1	2.4	2.75	V
Current Output	-	-	100	mA
Current Consumption	-	-	200	uA
Stop Current	-	-	2.0	uA
Setup Time	-	-	50	usec

## 19.3 POR ELECTRICAL CHARACTERISTICS

Table 19-3 POR Electrical Characteristics

<i>Parameter</i>	<i>Min</i>	<i>Typ</i>	<i>Max</i>	<i>Unit</i>
Start Voltage	-	1.8	-	V
Width of Reset	200	300	600	nsec
Current Consumption	-	5	10	uA

## 19.4 RC OSC ELECTRICAL CHARACTERISTICS

Table 19-4 RC OSC Electrical Characteristics

<i>Parameter</i>	<i>Min</i>	<i>Typ</i>	<i>Max</i>	<i>Unit</i>
Setup Time	-	-	1	usec
Current Consumption	-	-	0.3	mA
Static Current	-	1.0	-	uA

## 19.5 PLL ELECTRICAL CHARACTERISTICS

Table 19-5 PLL Electrical Characteristics

<i>Parameter</i>	<i>Conditions</i>	<i>Min</i>	<i>Typ</i>	<i>Max</i>	<i>Unit</i>
Input Frequency		2	-	33	MHz
Input Duty		45	50	55	-
Output Frequency		20	-	48	MHz
Output Duty		45	50	55	-
Clock Setup time		10	-	-	nsec
Lock Time		-	-	200	usec
Power Down Setup time		10	-	-	nsec
Output Jitter	Peak to peak	-	-	200	ps
Current Consumption	Fout = 48MHz	-	2.0	3.0	mA
Static Current		-	-	1	uA



## 19.6 ADC ELECTRICAL CHARACTERISTICS

Table 19-6 ADC Electrical Characteristics

<i>Parameter</i>	<i>Min</i>	<i>Typ</i>	<i>Max</i>	<i>Unit</i>
Input Range	-	-	0.99*VREF	V
Sampling Clock	-	-	500	KHz
ADC Clock Frequency (ADCCLK)	-	-	6.5	MHz
Data Latency	-	12	-	ADCCLK cycle
Integral Non-Linearity(INL)	-	-	±2.0	LSB
Differential Non-Linearity(DNL)	-	-	±1.0	LSB
Signal-to-Noise rate(SNR)	-	56	-	dB
Current Consumption	-	-	1	uA
Static Current	-	-	2	mA

## 19.7 DAC ELECTRICAL CHARACTERISTICS

Table 19-7 DAC Electrical Characteristics

<i>Parameter</i>	<i>Symbol</i>	<i>Conditions</i>	<i>Min</i>	<i>Typ</i>	<i>Max</i>	<i>Unit</i>
Resolution	Bit			12		Bits
Differential Linearity Error	DLE			±1.0		LSB
Integral Linearity Error	ILE			±4.0		LSB
Output Voltage	Vdac		0.127	-	2.792	V
LSB Size	Vlsb			0.651		mV
Data Settling Time				500		nsec
Current Consumption					8	mA
Static Current					10	uA

## 20 PACKAGE DIMENSION

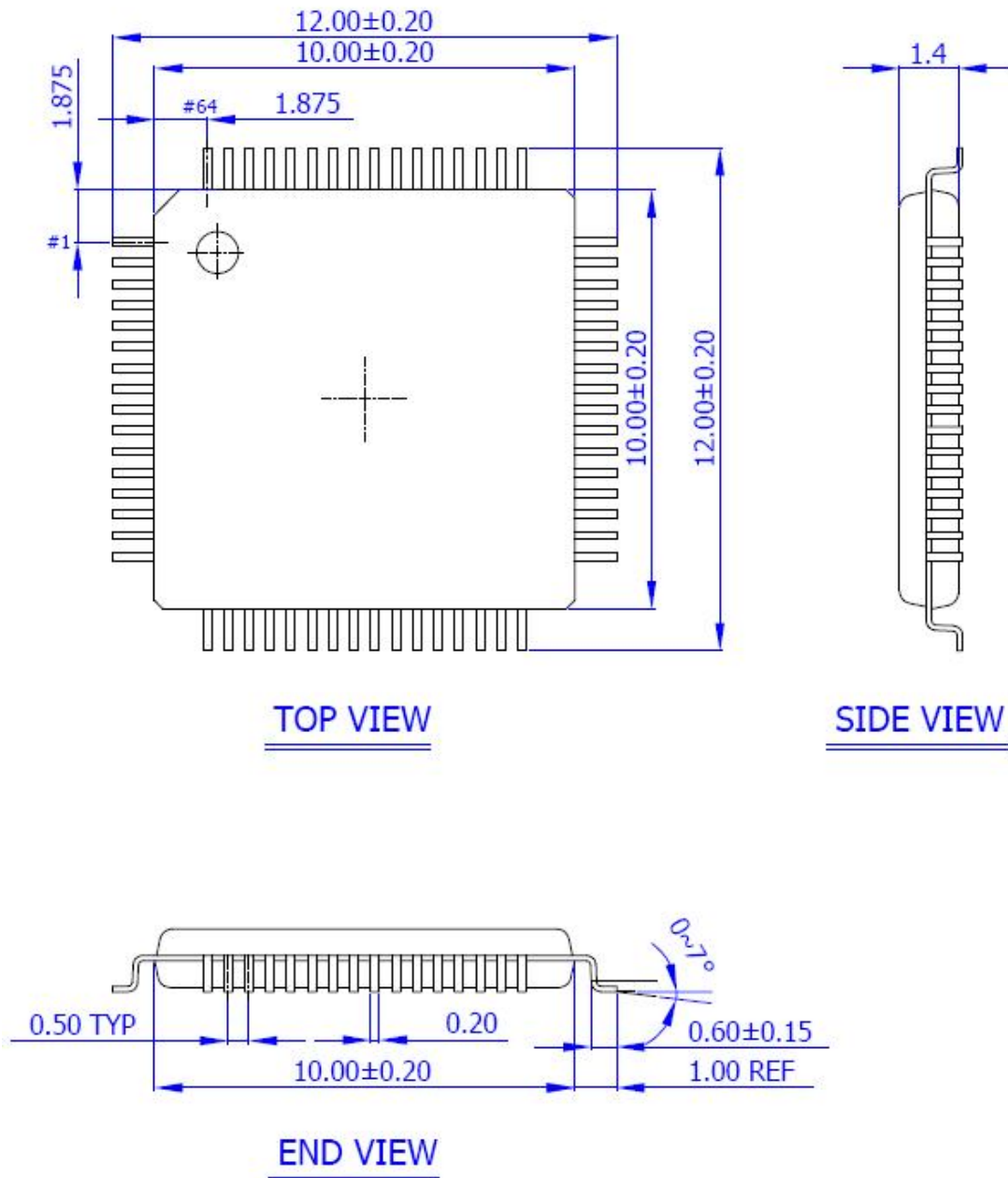


Figure 20-1 Package Dimension