2011년2학기
임베디드시스템 응용(#514118)

#3. Interrupt & Timer_1

한림대학교
전자공학과 이선우

한림대학교
HALLYM UNIVERSITY

순 서

- Ch.20. Interrupt
  - Types
  - Sources
  - Control registers
  - Example codes
- Timers
  - Overview
  - Clock generator
  - 8-bit timer/event counters 50/51
  - Control registers
  - Example codes

Interrupt types

- 2 types interrupts
  - Maskable interrupts
  - Software interrupts
  - NMI (Reset)
- Maskable interrupts
  - 사용자 프로그램에 의해 비활성화가 가능한(maskable) 인터럽트
  - Sources: external:9, internal:20
  - Priority 설정(2단계:H/L)에 의해 다단계 인터럽트 처리 (multiple interrupt service) 가능
- Software interrupt
  - BRK 명령어에 의해 발생
  - IE=0 in PSW 인 경우에도 수용되고 우선순위 없음.

Sources (1/2)

Table 20-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTLVI	Low-voltage detection[Note 3]	Internal	0004H	(A)
	1	INTP0	Pin input edge detection	External	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6	UART6 reception error generation	Internal	0012H	(A)
	8	INTSR6	End of UART6 reception		0014H	
	9	INTST6	End of UART6 transmission		0016H	
	10	INTCSI10/ INTST0	End of CSI10 communication/end of UART0 transmission		0018H	
	11	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)		001AH	
	12	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		001CH	
	13	INTTM50	Match between TM50 and CR50 (when compare register is specified)		001EH	
	14	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		0020H	
	15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	

- Default priority: 2개 Irq. req.가 동시 발생했을 경우 이 우선순위에 따라 처리함. 0이 가장 높고 28이 가장 낮음.

Sources (2/2)

	15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	
	16	INTAD	End of A/D conversion		0024H	
	17	INTSR0	End of UART0 reception or reception error generation		0026H	
	18	INTWTI	Watch timer reference time interval signal		0028H	
	19	INTTM51 Note 4	Match between TM51 and CR51 (when compare register is specified)		002AH	
	20	INTKR	Key interrupt detection	External	002CH	(C)
	21	INTWT	Watch timer overflow	Internal	002EH	(A)
	22	INTP6	Pin input edge detection	External	0030H	(B)
	23	INTP7			0032H	
Maskable	24	INTIIC0/ INTDMU	End of IIC0 communication/end of multiply/divide operation	Internal	0034H	(A)
	25	INTCSI11	End of CSI11 communication		0036H	
	26	INTTM001	Match between TM01 and CR001 (when compare register is specified), TI011 pin valid edge detection (when capture register is specified)		0038H	
	27	INTTM011	Match between TM01 and CR011 (when compare register is specified), TI001 pin valid edge detection (when capture register is specified)		003AH	
	28	INTACSI	End of CSIA0 communication		003CH	
Software	–	BRK	BRK instruction execution	–	003EH	(D)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on clear			
		LVI	Low-voltage detection Note 3			
		WDT	WDT overflow			

RESET의 경우 Non Maskable Irq로 간주함.

한림대학교 HALLYM UNIVERSITY

Control registers

- 6가지 종류의 레지스터가 제어
  - Int. request flag reg. (IF0L/H, IF1L/H)
    - 해당하는 소스가 인터럽트 요청하면 해당 bit=1
    - 요청이 수용되면(acknowledged) 자동으로 0이 되고 ISR 시작.
    - 8bit/1bit memory manipulation instruction으로 설정.
    - 16bit 명령어는 IF0, IF1 16bit reg. 간주하여 처리.
  - Int. mask flag reg. (MK0L/H, MK1L/H)
    - 해당 bit=1 이면 해당 Irq. request disable.
  - Priority specification flag reg. (PR0L/H, PR1L/H)
    - 해당 bit=0 일 때 high priority level
  - External int. edge enable reg. (EGP, EGN)
    - EGP: rising edge, EGN: falling edge
  - Program Status Word(PSW)

External interrupt control

- EGP/EGN 각 bit별 핀 설정

Table 20-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3
EGP4	EGN4	P33	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6
EGP7	EGN7	P141	INTP7

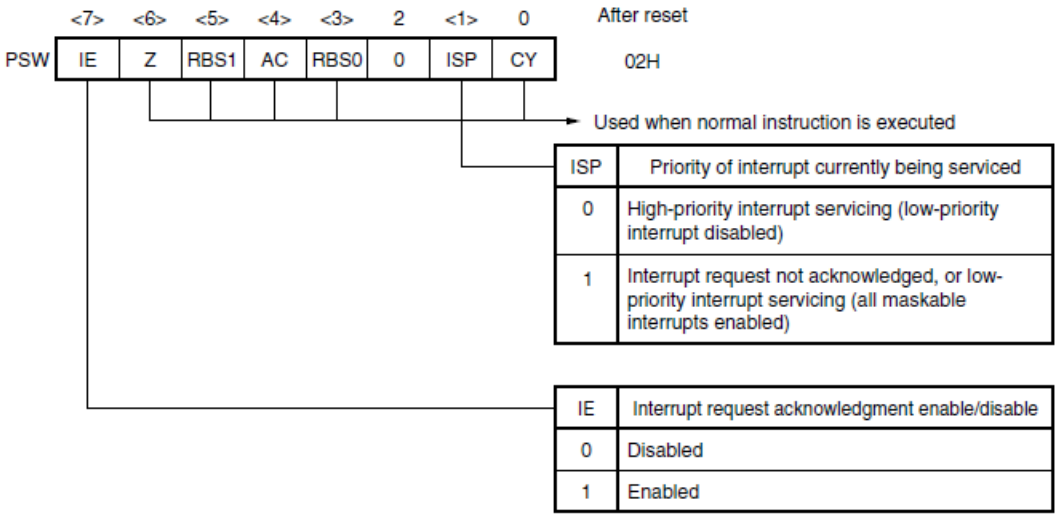
- 비트 설정 값 영향

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

PSW (program status word)

- PSW
  - CPU의 현재 상태를 나타내는 주요 flag를 모아 놓은 레지스터
  - IE: 전체 인터럽트 기능의 on/off (EI/DI 명령어 이용)
  - ISP:high-priority interrupt의 ISR을 수행 중일 때 0이 됨.
- Irq. 요청 수용되면 PSW는 자동으로 스택에 저장되고 IE=0 이 된 후 ISR로 이동(Irq. vector로 이동)

Figure 20-6. Format of Program Status Word



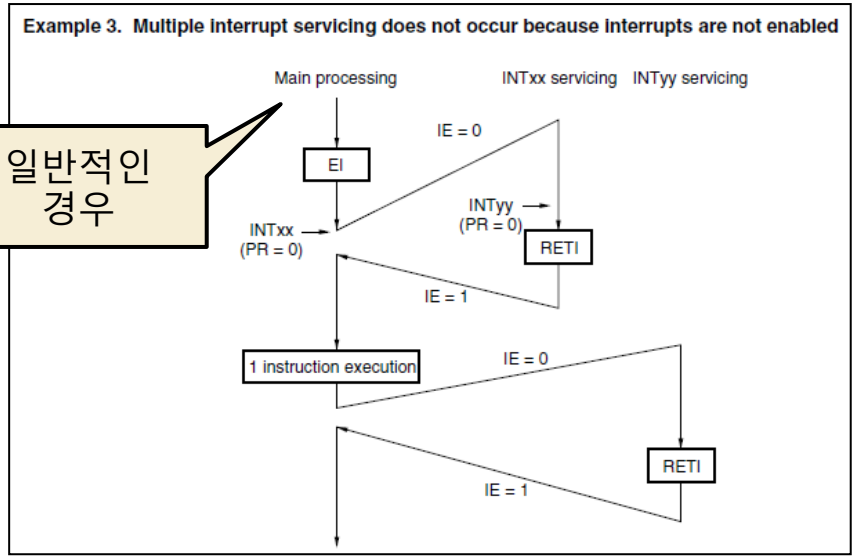
ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

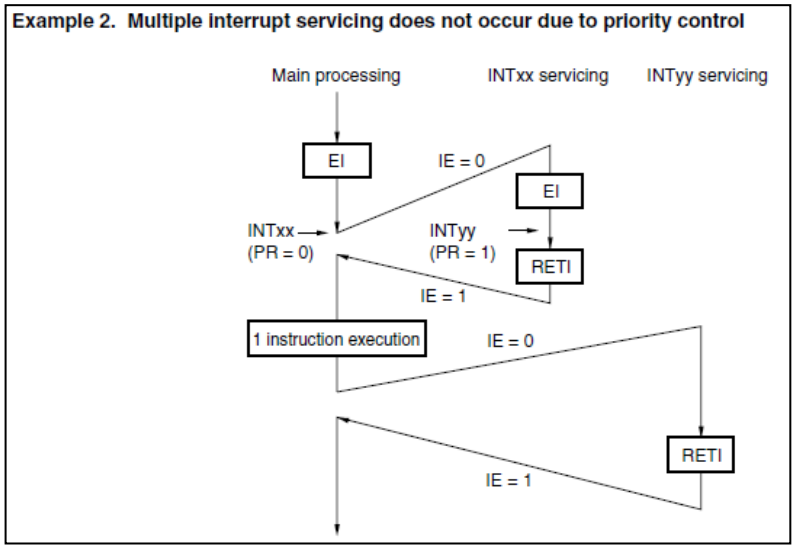
IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

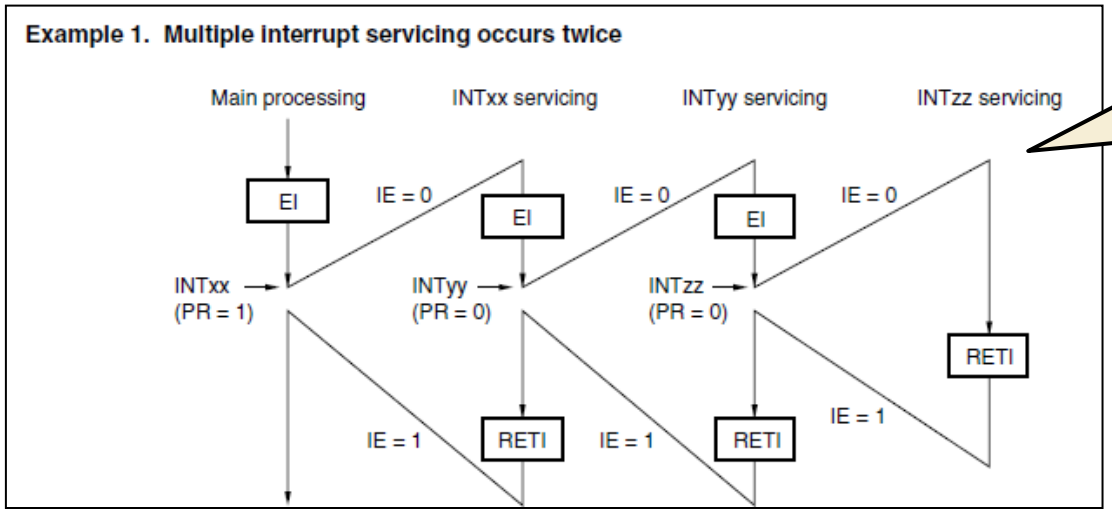
다단계 인터럽트 서비스

EI를 이용해 다른 요청 허용했으나 우선순위가 낮아 나중에 처리됨.

일반적인 경우







INTxx가 처리 중에 상위 순위의 INTyy/zz가 요청한 경우 처리 순서.

한림대학교 HALLYM UNIVERSITY

Example codes for external interrupt

```
#pragma DI
#pragma EI
#pragma vect INTP0 intp0_isr

void main(void)
{
    //포트 설정 (회로도 참고할 것!)
    PM12.0 = 1;  //make input
    PU12.0 = 1;  //enable pull-up
    //인터럽트 관련 설정
    EGN = 1; //falling edge 설정
    MK0L &= 0xfd; //bit1=0 → enable irq.

    //enable global irq. enable
    EI();  //or DI();

    while(1);
}
__interrupt void intp0_isr(void)
{
     LED0 ^= 1; //toggle LED0
}
```

#pragma directives

- **DI/EI:** c source에서 PSW.[IE] bit를 set/clear 할 수 있는 함수 EI()/DI()를 사용하도록 함.
- vect/interrupt:정의한 isr 함수의 주소를 irq. vector에 지정하는 명령.
- 사용 포맷:

#pragma vect(or interrupt) irq.req. name [space] function name [이 뒤에도 다른 설정이 가능하나 생략함. 자세한 사항은 CC780 manual 11.5(9) 내용을 참조함.]

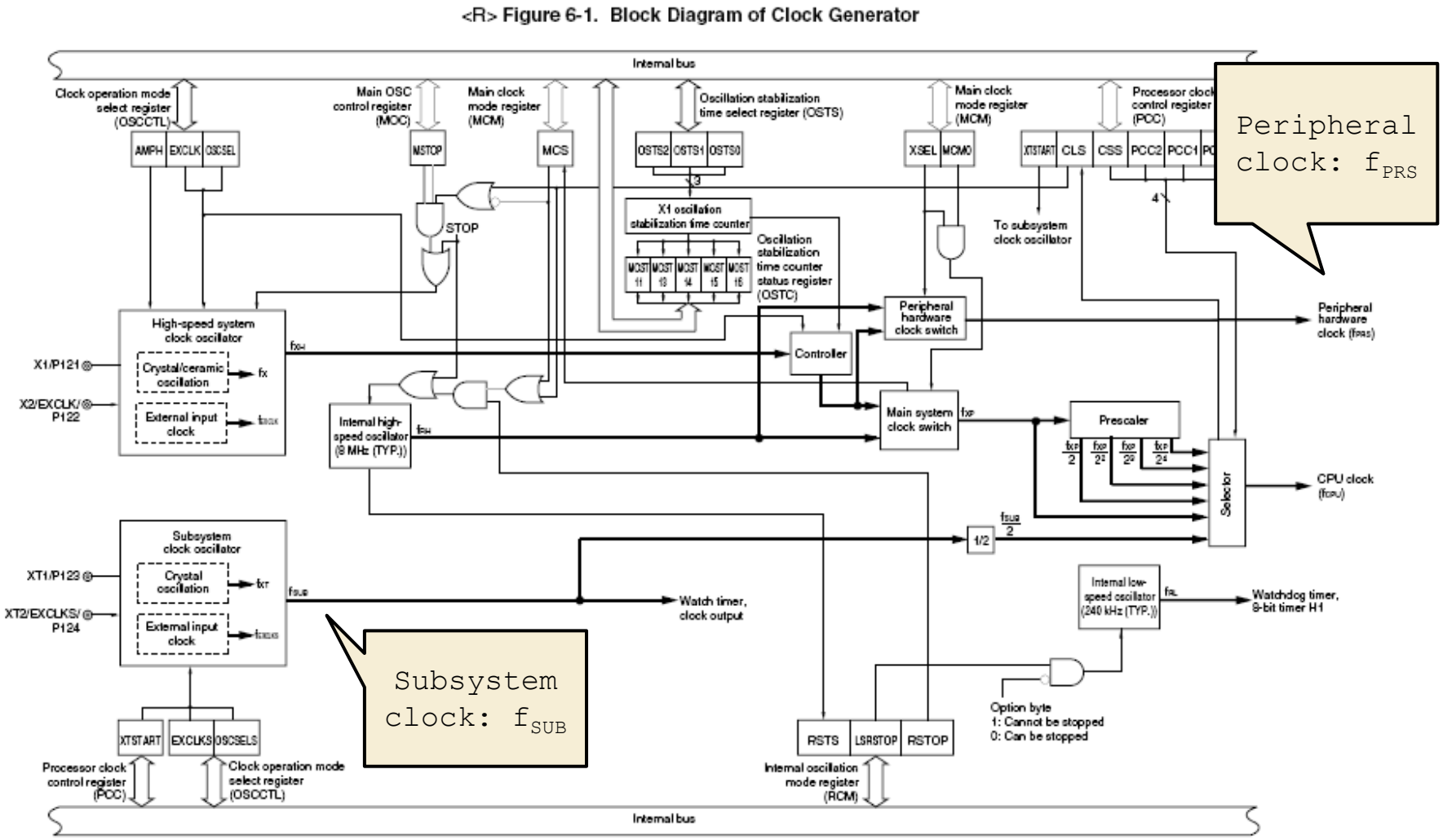
Ch.6 Clock generator

Ch.6 Clock generator

- CPU와 내부 주변 장치를 위한 클럭 발생 장치
  - 3가지 종류 발진기 회로 내장

1. Main system clock
   - X1 oscillator: X1, X2 pin에 연결된 발진기에 의해 1~20MHz 범위의 클럭 발생
   - Internal high-speed osc. (내부고속발진기):f_{RH}=8MHz의 내부 발진기 회로 내장. Reset후엔 이 클럭 이용하여 CPU동작함.
2. Subsystem clock
   - XT1, XT2 pin에 연결된 발진기 (보통 시계 발진기 연결 f_{XT} = 32768Hz) 이용 클럭 발생
3. Internal low-speed osc. Clock (for watchdog timer)
   - fRL=240KHz 주파수의 클럭을 발생시키는 회로 내장.
   - Watchdog timer, TMH1(8bit counter)에 공급

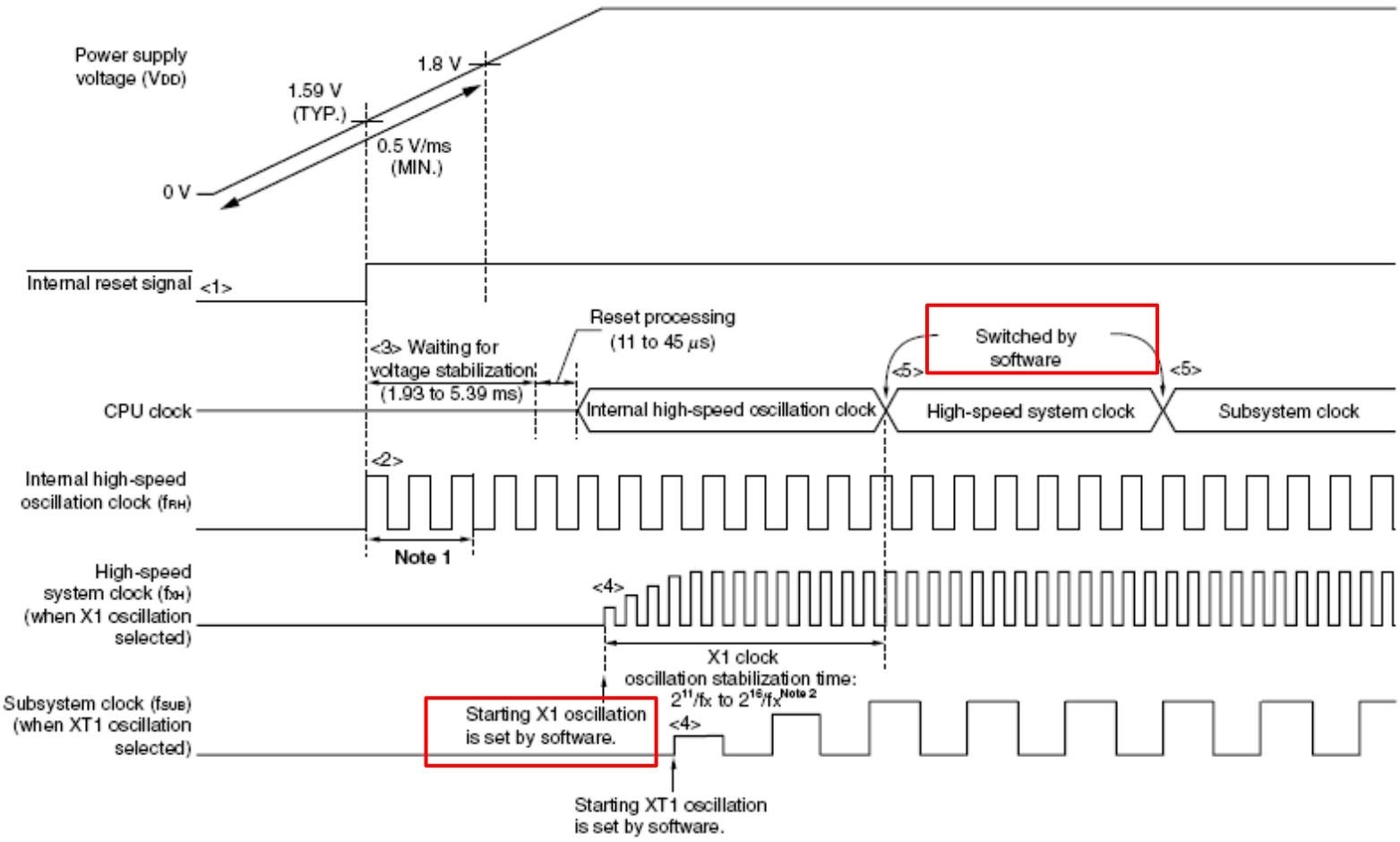
Block diagram of clock generator



한림대학교 HALLYM UNIVERSITY

전원 인가 후 MCU 동작 순서(클럭 발생기 관련)

Figure 6-12. Clock Generator Operation When Power Supply Voltage Is Turned On
(When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0))



한림대학교 HALLYM UNIVERSITY

Control registers

- OSCCTL (clock operation mode select reg.)
  동작 모드 결정, 발진기 이득 결정
- PCC (processor clock control reg.)
  CPU clock 소스 및 주파수 결정, 서브시스템클럭 동작 설정
- RCM(internal osc. mode reg.)
  내부발진기(high/low-speed) 동작 제어 및 상태 표시
- MOC(main OSC control reg.)
  X1(EXCLK pin) 동작 제어
- MCM (main clock mode reg.)
  CPU 및 주변장치 클럭 종류 선택
- OSTS (osc. stabilization time select reg.)/OSTC(osc. stabilization time counter status reg.)
  X1 clock 발진기 안정화 시간 결정, 상태 표시

OSCCTL, PCC

Figure 6-3. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 01H R/W[Note 1]

Symbol	7	6	<5>	<4>	3	2	1	0
PCC	0	XTSTART[Note2]	CLS	CSS	0	PCC2	PCC1	PCC0

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

CSS	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
0	0	0	0	f_{XP}
	0	0	1	$f_{XP}/2$ (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
1	0	0	0	$f_{SUB}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

Table 6-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: 2/f_{CPU}			
	Main System Clock			Subsystem Clock
	High-Speed System Clock[Note]		Internal High-Speed Oscillation Clock[Note]	
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{XP}	0.2 μs	0.1 μs	0.25 μs (TYP.)	–
$f_{XP}/2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	–
$f_{XP}/2^2$	0.8 μs	0.4 μs	1.0 μs (TYP.)	–
$f_{XP}/2^3$	1.6 μs	0.8 μs	2.0 μs (TYP.)	–
$f_{XP}/2^4$	3.2 μs	1.6 μs	4.0 μs (TYP.)	–
$f_{SUB}/2$	–		–	122.1 μs

Figure 6-2. Format of Clock Operation Mode Select Register (OSCCTL)

Address: FF9FH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>
OSCCTL	EXCLK	OSCSEL	EXCLKS[Note]	OSCSELS[Note]	0	0	0	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	P121/X1 pin	P122/X2/EXCLK pin
0	0	I/O port mode	I/O port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	I/O port mode	I/O port	
1	1	External clock input mode	I/O port	External clock input

AMPH	Operating frequency control
0	1 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

Table 6-3. Setting of Operation Mode for Subsystem Clock Pin

PCC	OSCCTL		Subsystem Clock Pin Operation Mode	P123/XT1 Pin	P124/XT2/EXCLKS Pin
Bit 6	Bit 5	Bit 4			
XTSTART	EXCLKS	OSCSELS			
0	0	0	I/O port mode	I/O port	
0	0	1	XT1 oscillation mode	Crystal resonator connection	
0	1	0	I/O port mode	I/O port	
0	1	1	External clock input mode	I/O port	External clock input
1	×	×	XT1 oscillation mode	Crystal resonator connection	

RCM, MOC, MCM

Figure 6-4. Format of Internal Oscillation Mode Register (RCM)

Address: FFA0H After reset: 80H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	Status of internal high-speed oscillator
0	Waiting for accuracy stabilization of internal high-speed oscillator
1	Stability operating of internal high-speed oscillator

LSRSTOP	Internal low-speed oscillator oscillating/stopped
0	Internal low-speed oscillator oscillating
1	Internal low-speed oscillator stopped

RSTOP	Internal high-speed oscillator oscillating/stopped
0	Internal high-speed oscillator oscillating
1	Internal high-speed oscillator stopped

Figure 6-5. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 80H R/W

Symbol	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	Control of high-speed system clock operation	
	X1 oscillation mode	External clock input mode
0	X1 oscillator operating	External clock from EXCLK pin is enabled
1	X1 oscillator stopped	External clock from EXCLK pin is disabled

Figure 6-6. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	<2>	<1>	<0>
MCM	0	0	0	0	0	XSEL	MCS	MCM0

XSEL	MCM0	Selection of clock supplied to main system clock and peripheral hardware	
		Main system clock (f_{XP})	Peripheral hardware clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})
1	1	High-speed system clock (f_{XH})	

MCS	Main system clock status
0	Operates with internal high-speed oscillation clock
1	Operates with high-speed system clock

OSTS, OSTC

Figure 6-8. Format of Oscillation Stabilization Time Select Register (OSTS)

Osc. Stabilization Time **Select** Reg.

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fx = 10 MHz	fx = 20 MHz
0	0	1	2^{11}/fx	204.8 μs	102.4 μs
0	1	0	2^{13}/fx	819.2 μs	409.6 μs
0	1	1	2^{14}/fx	1.64 ms	819.2 μs
1	0	0	2^{15}/fx	3.27 ms	1.64 ms
1	0	1	2^{16}/fx	6.55 ms	3.27 ms
Other than above			Setting prohibited		

Figure 6-7. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Osc. Stabilization Time **Status** Reg.

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status		
						fx = 10 MHz	fx = 20 MHz
1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	0	0	2^{14}/fx min.	1.64 ms min.	819.2 μs min.
1	1	1	1	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	2^{16}/fx min.	6.55 ms min.	3.27 ms min.

Controlling clock

- High-speed system clock 제어
  - X1 클럭(crystal/resonator 연결), 외부 클럭 중 하나 선택 가능
  - X1/P121, X2/EXCLK/P122 pin은 리셋 후 I/O port로 설정됨
- X1 clock 설정 방법
  1. OSCCTL 설정 (AMPH는 사용 발진자의 주파수에 따라 결정, EXCLK=0, OSCSEL=1)
  2. MSTOP=0이 되면 발진회로 시작
  3. 안정화될 때까지 기다림 (OSTS 값으로 결정)
- External clock 사용하는 법
  - OSCCTL 설정 (EXCLK=1, OSCSEL=1) → X1/P121:IO port, X2/EXCLK pin만 사용(외부클럭)

Controlling clock

- CPU와 주변장치에서 고속시스템 클럭 사용법
  1. High-speed system clock osc. 설정하기
     안정적으로 동작한다고 가정.
  2. MCM 설정
     XSEL=1 ,MCM0=1 → f_{XP} =f_{XH}, f_{PRS}=f_{XH}
  3. PCC 설정 (CSS=0)
- 고속시스템 클럭 정지하기 위한 설정 방법
  - 2가지 방법 : STOP 명령어 실행, MSTOP=1 in MOC (주의사항: 이 명령 실행 전에 CPU 구동 클럭을 내부고속클럭이나 서브시스템 클럭으로 변경해야 함.)
- 내부 고속 발진기 클럭 제어 방법
  - RSTOP=0되면 내부발진기 동작 시작, RSTS=1(안정)될 때까지 기다린 후 사용
  - 정지 방법: STOP 명령어 사용, RSTOP=1 사용

Example code for clock controlling

```
void hdwinit(void)
{
          IMS = 0b11001100;
          IXS = 0b00000000;

  /* X1/XT1발진,10M<fxh<=20M,X1발진동작 */
          AMPH = 1;          /* OSCCTL.0 */
          EXCLK = 0;         /* OSCCTL.7 */
          OSCSEL = 1;        /* OSCCTL.6 */
  /* X1발진 회로 동작 */
          MSTOP = 0;         /* MOC.7 */
  /* 발진 안정 시간 경과 대기 */
          do{
              NOP();
          }while(OSTC.0 == 0);
  /* 메인 시스템=fXH,주변H/W=fXH */
          XSEL = 1;          /* MCM.2 */
          MCM0 = 1;          /* MCM.0 */
  /* CPU Clock =fxp, XTSTART=1 */
          PCC = 0b01000000;
  /* 고속 시스템 CLK이 동작 하기까지 Wait */
          do{
             NOP();
          }while(MCS == 0);
  /* 내부 고속/저속 발진회로의 정지 */
          RCM = 0x03;
}
```

- Project에 추가되어 있는 "sysinitRH.c" file에 정의된 hdwinit() 함수에서 설정한다.
- 지금까지는 internal high-speed osc. (8MHz) 사용했음.

새로운 hdwinit() 함수 내용

- X1/X2에 연결된 발진기(주파수>10MHz) 사용하여 CPU, 주변장치 클럭으로 사용.
- 서브시스템 발진기 사용 (XTSTART=1) (XT1/XT2에 연결된 발진기 사용)
- 내부 발진기(고속/저속) 모두 turn OFF

78K0/Kx2 Timers

Timer overview

- Total 8 channels
  1. 16-bit timer/event counter :2 (00/01)
  2. 8-bit timer/event counter: 2 (50/51)
  3. 8-bit timer: 2 (H0/H1)
  4. Watch timer: 1
  5. Watchdog timer: 1
- 공통 기능
  - Interval timer (all), square-wave output, PWM output (4,5제외), external event counter (1,2), pulse-width measurement, one-shot pulse output (1 only)
- 8-bit timer의 차이점
  - 50/51은 외부 신호 입력 가능 → event counter란 이름 사용

Ch.10 Watch timer

- 기능
  - 시계용 타이머/인터벌 타이머 2개 동시 기능. 즉 2개 별도 인터벌마다 별도 IRQ. (INTWT, INTWTI) 발생.
  - Watch timer: $2^4 \sim 2^{14}/f_w$
  - Interval timer: $2^4 \sim 2^{11}/f_w$

Watch timer

Interrupt Time	When Operated at f_{SUB} = 32.768 kHz	When Operated at f_{PRS} = 2 MHz
$2^4/fw$	488 μs	1.02 ms
$2^5/fw$	977 μs	2.05 ms
$2^{13}/fw$	0.25 s	0.52 s
$2^{14}/fw$	0.5 s	1.05 s

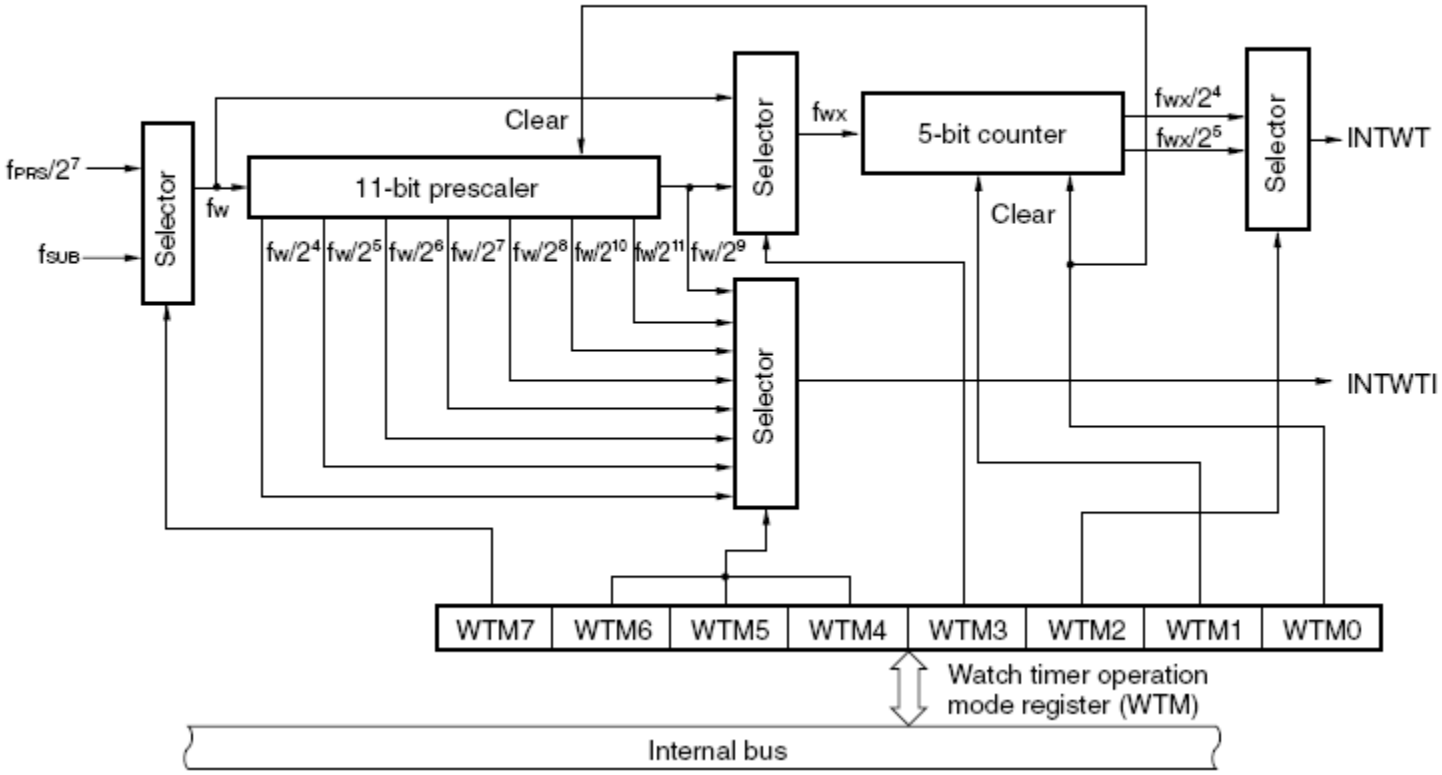
Remark f_{PRS}: Peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency
fw: Watch timer clock frequency ($f_{PRS}/2^7$ or f_{SUB}

Interval timer

Interval Time	When Operated at f_{SUB} = 32.768 kHz	When Operated at f_{PRS} = 2 MHz
$2^4/fw$	488 μs	1.02 ms
$2^5/fw$	977 μs	2.05 ms
$2^6/fw$	1.95 ms	4.10 ms
$2^7/fw$	3.91 ms	8.20 ms
$2^8/fw$	7.81 ms	16.4 ms
$2^9/fw$	15.6 ms	32.8 ms
$2^{10}/fw$	31.3 ms	65.5 ms
$2^{11}/fw$	62.5 ms	131.1 ms

Block diagram of watch timer

Figure 10-1. Block Diagram of Watch Timer



Control register

- WTM (watch timer operation mode reg.)
  - 7bit: 클럭 소스 선택
  - 6~4bit: INTWTI 인터벌 결정
  - 3,2bit: INTWT 인터벌 결정
  - 1bit: 5bit counter start/stop
  - 0bit: 타이머 전체 동작 on/off

Figure 10-2. Format of Watch Timer Operation Mode Register (WTM)

Address: FF6FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	Watch timer count clock selection (fw)					
		f_{SUB} = 32.768 kHz	f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	$f_{PRS}/2^7$	–	15.625 kHz	39.062 kHz	78.125 kHz	156.25 kHz
1	f_{SUB}	32.768 kHz	–			

WTM6	WTM5	WTM4	Prescaler interval time selection
0	0	0	2^4/fw
0	0	1	2^5/fw
0	1	0	2^6/fw
0	1	1	2^7/fw
1	0	0	2^8/fw
1	0	1	2^9/fw
1	1	0	2^{10}/fw
1	1	1	2^{11}/fw

WTM3	WTM2	Selection of watch timer interrupt time
0	0	2^{14}/fw
0	1	2^{13}/fw
1	0	2^5/fw
1	1	2^4/fw

WTM1	5-bit counter operation control
0	Clear after operation stop
1	Start

WTM0	Watch timer operation enable
0	Operation stop (clear both prescaler and 5-bit counter)
1	Operation enable

Example code for watch timer

```
#pragma DI
#pragma EI
#pragma vect INTWT intwt_isr

void main(void)
{
    //포트 설정 for LED0
    PM12.0 = 1;  //make input

    //watch timer 0.5s interval 설정
    WTM = 0b10000000;
    //enable INTWT
    WTMK = 0; //MK1L.[5]=0
    //enable global irq. enable
    EI();  //or DI();

    while(1);
}
__interrupt void intwt_isr(void)
{
     LED0 ^= 1; //toggle LED0
}
```

- Watch timer의 IRQ source : INTWT & INTWTI
- 관련 bit: IF1L, MK1L, PR1L의 2번, 5번 bit
- CC78K0의 경우 비트 이름으로도 접근 가능 (ex. WTMK=0)