2011년2학기
임베디드시스템 응용(#514118)

#4. Timer_2

한림대학교
전자공학과 이선우

한림대학교
HALLYM UNIVERSITY

순 서

- Character LCD 사용법
- Clock/Buzzer output controller
  - Overview
  - Codes
- 16bit timers 00/01
  - Overview
  - Control registers
  - Codes

EVM에 장착된 character LCD

- 문자 LCD
  - 정해진 문자 (사용자 기호 입력 가능)를 표시하는 LCD
  - 품명: AX07001 by Apex display lt.
  - 특징: 16x2 character LCD
    - 표시가능 문자 set: 기호, 숫자, 영어 알파벳 대소문자, 일어 가타가나 등
- MUC와의 연결
  - LCD 제어를 위해 7개의 digital output 사용
  - 4bit data transfer 방법 이용: 1byte data를 4bit씩 나누어 2번에 전송 ➔ DB4~DB7 ←→ P0.0~P0.3
  - Control bus : RS=P6.4, RW=P6.5, E=P6.6

Code to use 16x2 char. LCD

```
#include "lcd2.h"

void main(void)
{
    //LCD init.
    wLCDinit();  //make input

    //전체 LCD를 클리어
    wLCDclr();
    //둘째 줄, 두번째 칸에 커서 표시
    wLCDcsron(1,1,1);
    wLCDcsron(1,1,0); //커서 지우기
    //문자 1개를 첫번째 4번째 칸에 A표시
   wLCDputc(0,3,'A');
    wLCDblink(0,3,1); //위 문자 blinking
    //(1,3)위치부터 문자열 표시
   wLCDputs(1,3,"Hello");
}
```

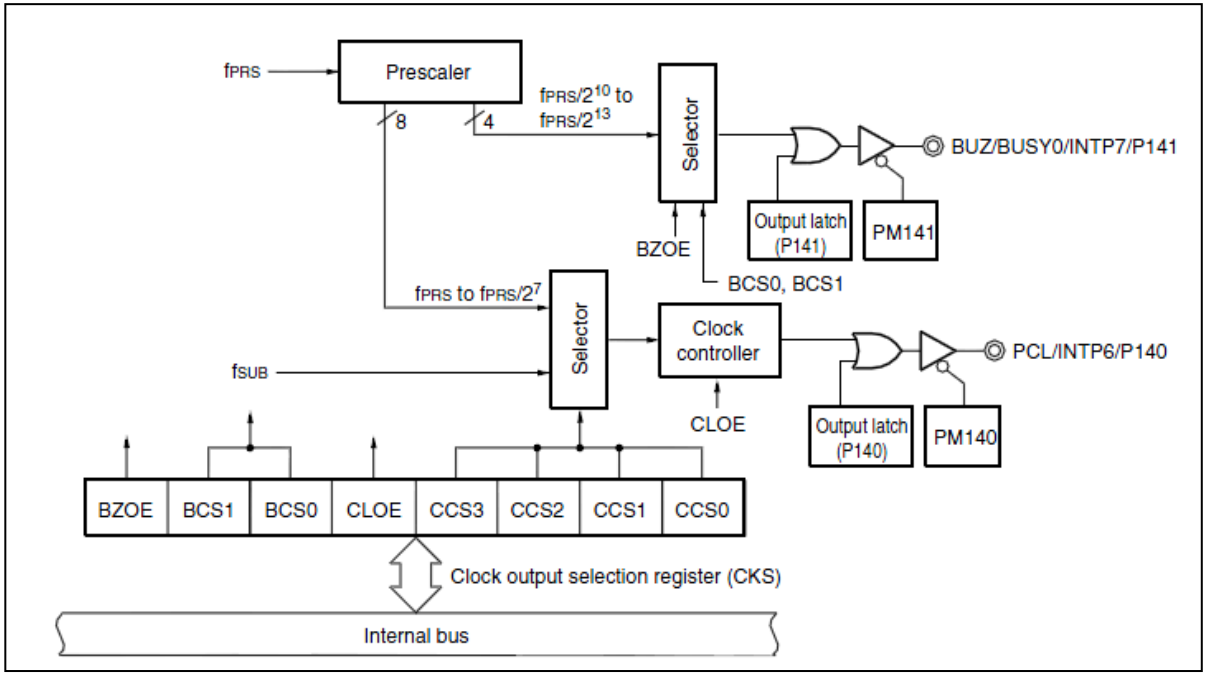
'lcd2.c' 파일에 정의된 함수들의 prototype이 작성되어 있는 헤더 파일

LCD사용법

- lcd2.c source file을 프로젝트에 추가
- include header file
- 모두 6개의 함수 제공
- 모든 함수의 인자(argument)는 unsigned char 형으로 해야 함.
- 16x2 칸의 위치 지정은 0~15까지로 표시.

Ch. 12: Clock/Buzzer output controller

- 기능
  - 정해진 주파수의 클럭 신호를 생성하여 출력하는 장치
  - BUZ/P141-버저용 클럭 출력용 핀
  - PCL/INTP6/P140-클럭 신호 출력용 핀



한림대학교 HALLYM UNIVERSITY

Control registers

- Clock output selection reg. (CKS)
  - ON/OFF 제어:BZOE, CLOE (1 →on, 0→off)
  - BUZ 출력 클럭 주파수 결정 ; [BCS1:BCS0]
    - 0→ $f_{PRS}/2^{10}$
    - 1→ $f_{PRS}/2^{11}$
    - 2→ $f_{PRS}/2^{12}$
    - 3→ $f_{PRS}/2^{13}$
  - PCL 출력 클럭 주파수 결정; [CCS3~CCS0]
    - $f_{PRS}/2$ ~$f_{PRS}/2^{7}$, f_{SUB}

Figure 12-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	<7>	6	5	<4>	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Clock division circuit operation stopped. BUZ fixed to low level.
1	Clock division circuit operation enabled. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection		
			f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	$f_{PRS}/2^{10}$	9.77 kHz	19.54 kHz
0	1	$f_{PRS}/2^{11}$	4.88 kHz	9.77 kHz
1	0	$f_{PRS}/2^{12}$	2.44 kHz	4.88 kHz
1	1	$f_{PRS}/2^{13}$	1.22 kHz	2.44 kHz

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

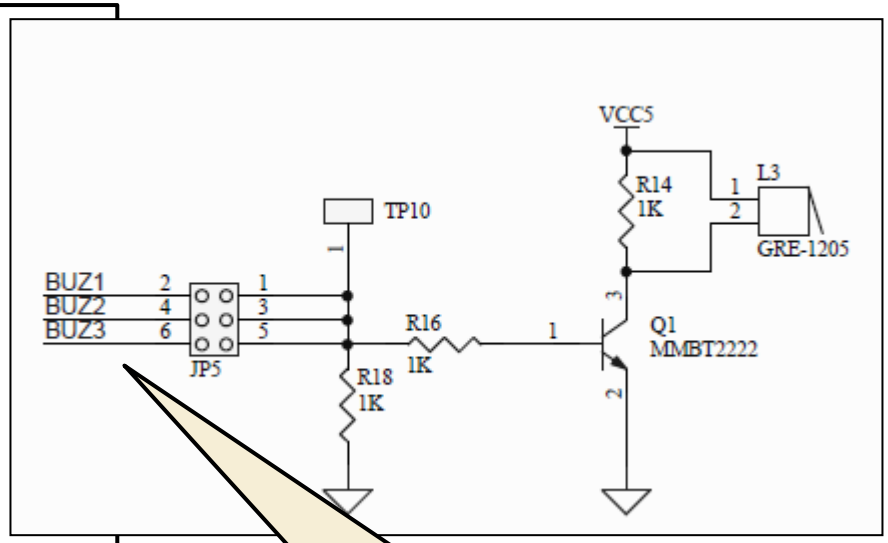
CCS3	CCS2	CCS1	CCS0	PCL output clock selection			
					f_{SUB} = 32.768 kHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS} Note 1	–	10 MHz	Setting prohibited Note 2
0	0	0	1	$f_{PRS}/2$		5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$		2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$		1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$		625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$		312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$		156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$		78.125 kHz	156.25 kHz
1	0	0	0	f_{SUB}	32.768 kHz	–	
Other than above				Setting prohibited			

EVM buzzer 회로 & Code

```
void main(void)
{
    //port setupt
    PM14.1 = 0;  //output
    PM2.4 = 1; //input for slide

    //주파수 결정
    CKS = 0b01100000;

    while(1) {
      if(P2.4)
        BZOE = 1;
      else
        BZOE = 0;
    }
}
```



- BUZ1 = BUZ/P141
- BUZ2 = P144
- BUZ3 = P120

16-bit Timer/Event counter overview

- 2개 내장: 00/01
- Functions
  - Interval timer
  - Square-wave output
  - External event counter
  - One-shop pulse output
  - PPG output (PWM like)
  - Pulse width measurement

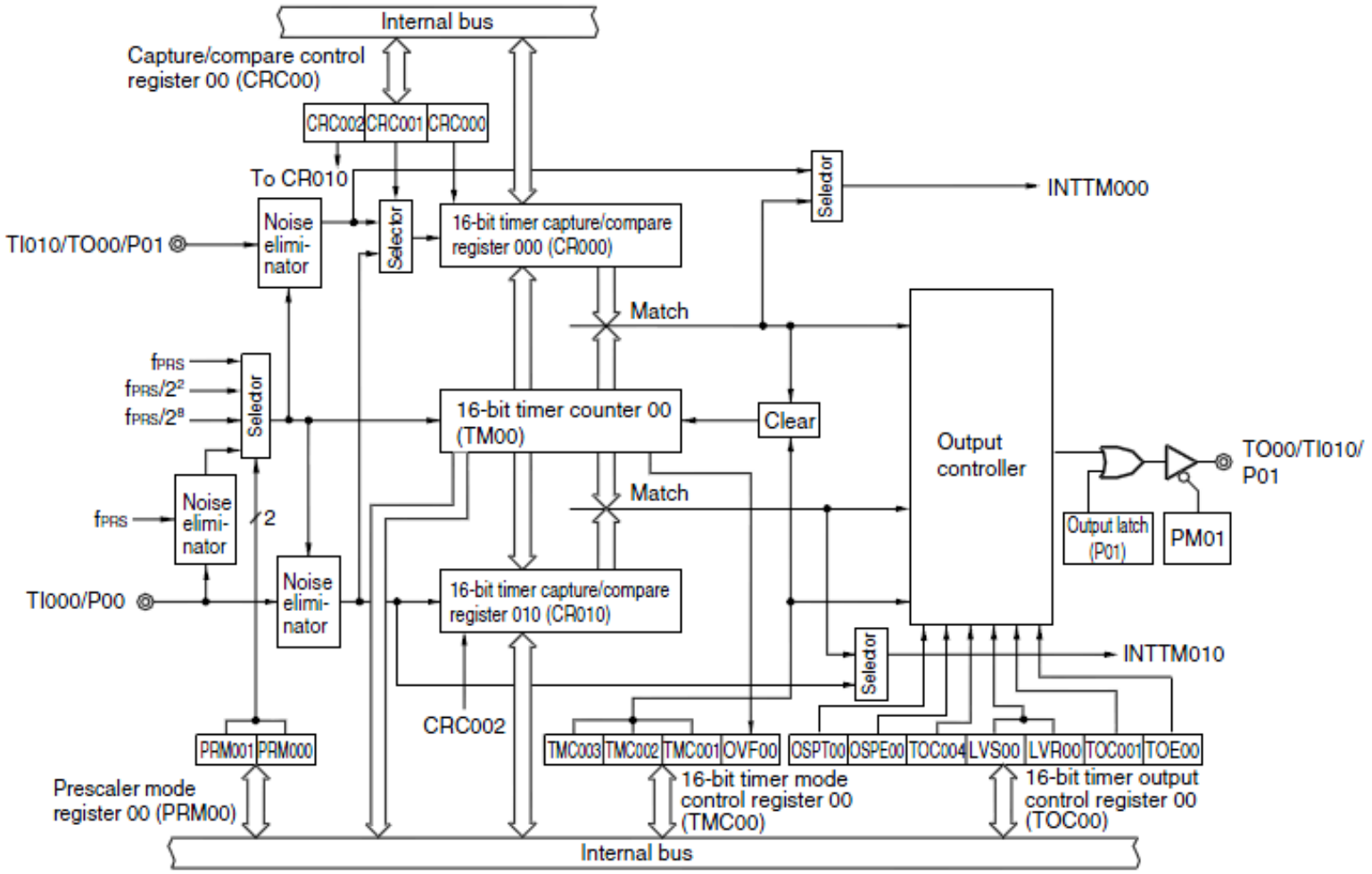
Table 7-1. Configuration of 16-Bit Timer/Event Counters 00 and 01

Item	Configuration
Time/counter	16-bit timer counter 0n (TM0n)
Register	16-bit timer capture/compare registers 00n, 01n (CR00n, CR01n)
Timer input	TI00n, TI01n pins
Timer output	TO0n pin, output controller
Control registers	16-bit timer mode control register 0n (TMC0n) 16-bit timer capture/compare control register 0n (CRC0n) 16-bit timer output control register 0n (TOC0n) Prescaler mode register 0n (PRM0n) Port mode register 0 (PM0) Port register 0 (P0)

Remark n = 0, 1

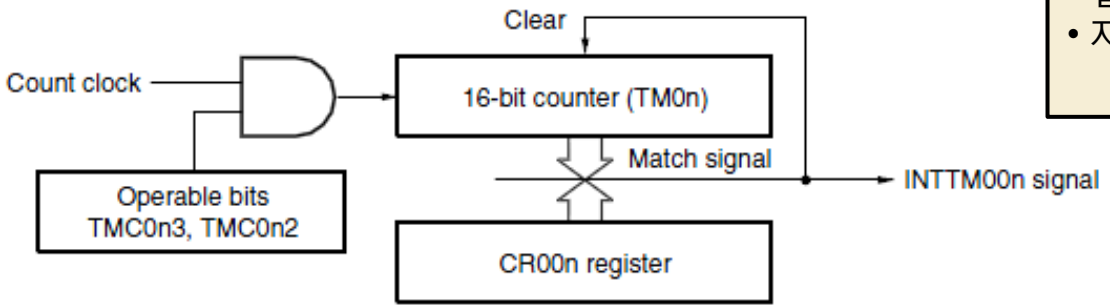
Block diagram

Figure 7-1. Block Diagram of 16-Bit Timer/Event Counter 00



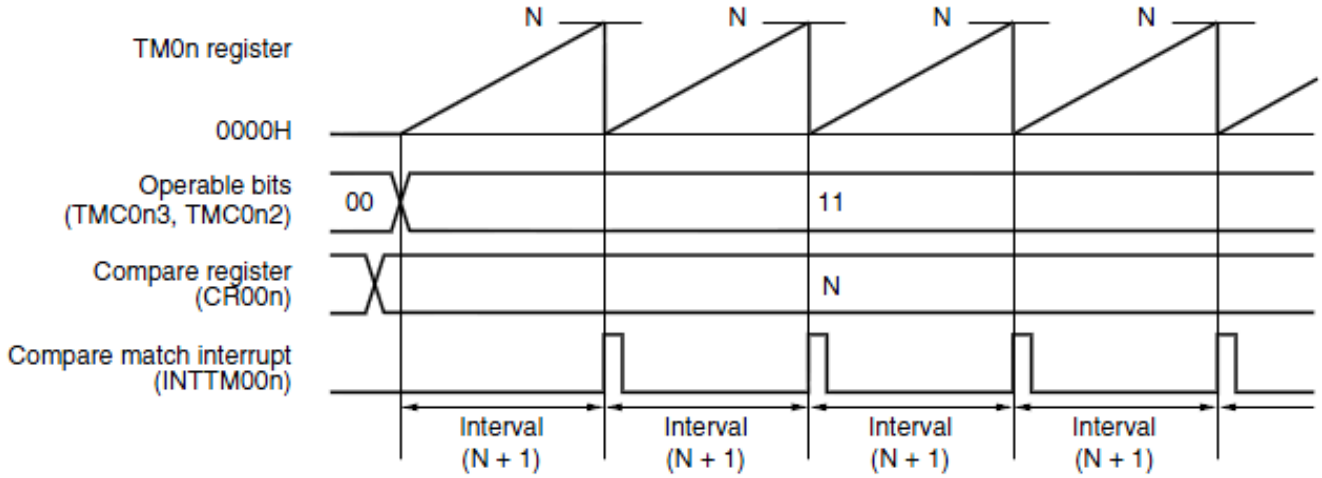
Interval timer operation

Figure 7-16. Block Diagram of Interval Timer Operation



- 일정시간 마다 인터럽트 발생
- 일정시간 설정은 CRoon reg.
- 자동으로 TMon clear됨.

Figure 7-17. Basic Timing Example of Interval Timer Operation



Square-wave output operation

Figure 7-20. Block Diagram of Square-Wave Output Operation

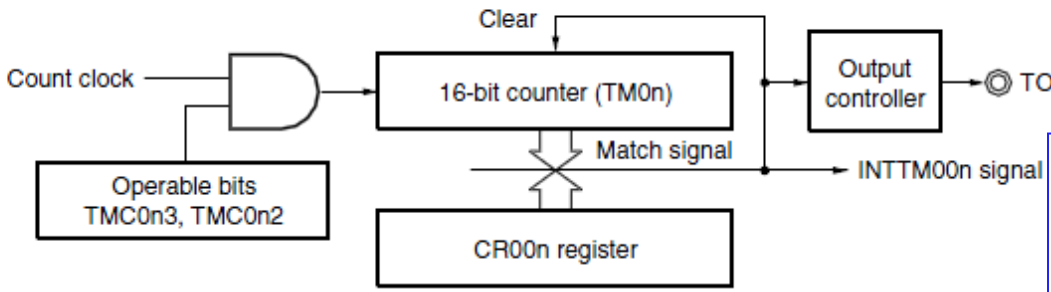
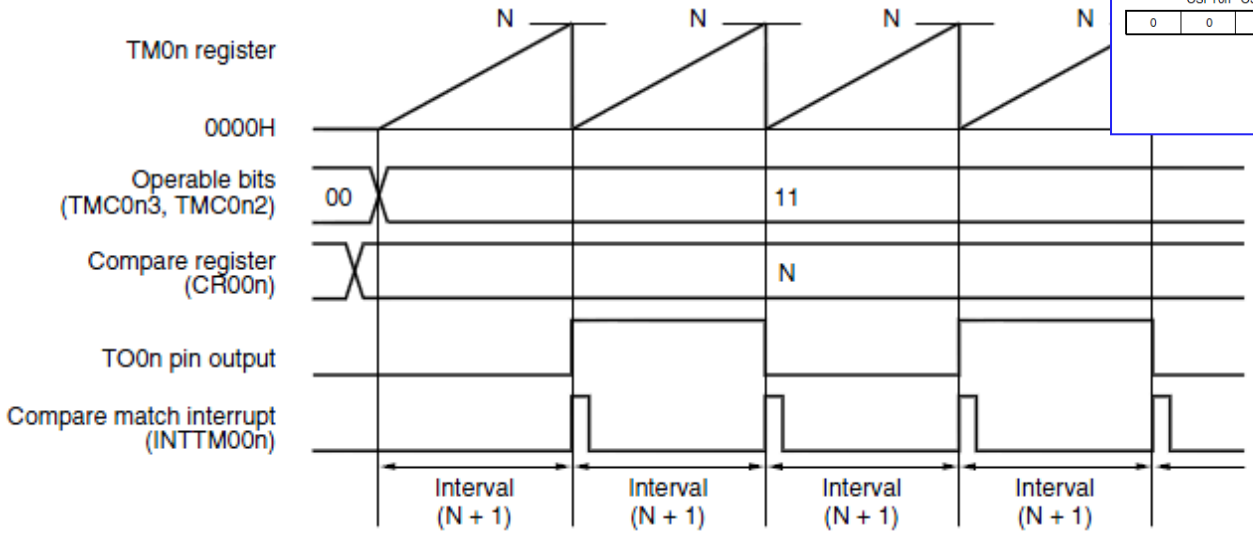


Figure 7-21. Basic Timing Example of Square-Wave Output Operation



(a) 16-bit timer mode control register 0n (TMC0n)

				TMC0n3	TMC0n2	TMC0n1	OVF0n
0	0	0	0	1	1	0	0

Clears and starts on match between TM0n and CR00n.

(b) Capture/compare control register 0n (CRC0n)

					CRC0n2	CRC0n1	CRC0n0
0	0	0	0	0	0	0	0

CR00n used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

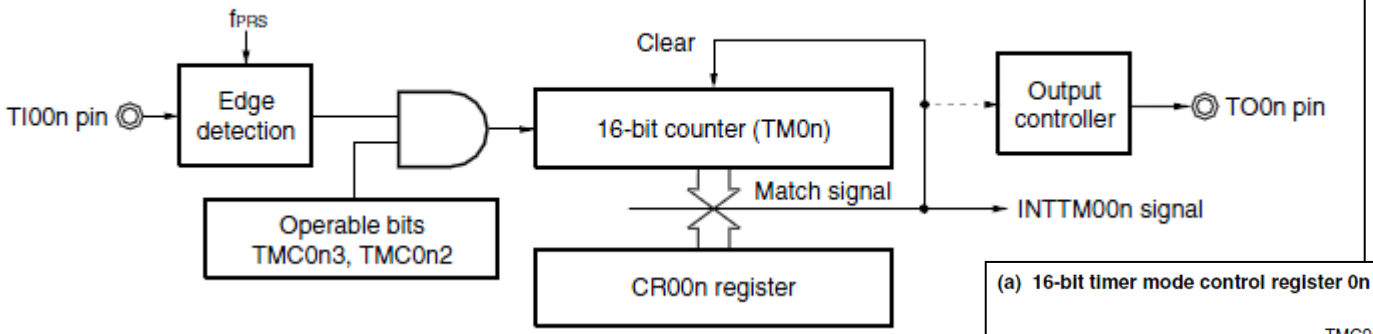
	OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n
0	0	0	0	0/1	0/1	1	1

Enables TO0n pin output.
Inverts TO0n pin output on match between TM0n and CR00n.
Specifies initial value of TO0n output F/F

한림대학교 HALLYM UNIVERSITY

External event counter

Figure 7-24. Block Diagram of External Event Counter Operation



(a) 16-bit timer mode control register 0n (TMC0n)

				TMC0n3	TMC0n2	TMC0n1	OVF0n
0	0	0	0	1	1	0	0

Clears and starts on match between TM0n and CR00n.

(b) Capture/compare control register 0n (CRC0n)

			CRC0n2	CRC0n1	CRC0n0
0	0	0	0	0	0

CR00n used as compare register

(d) Prescaler mode register 0n (PRM0n)

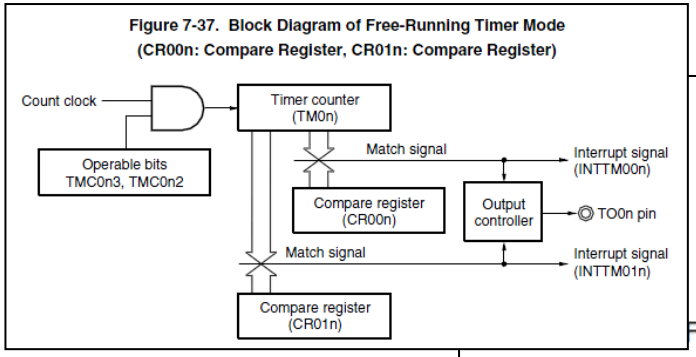
ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
0	0	0/1	0/1	0	0	1	1

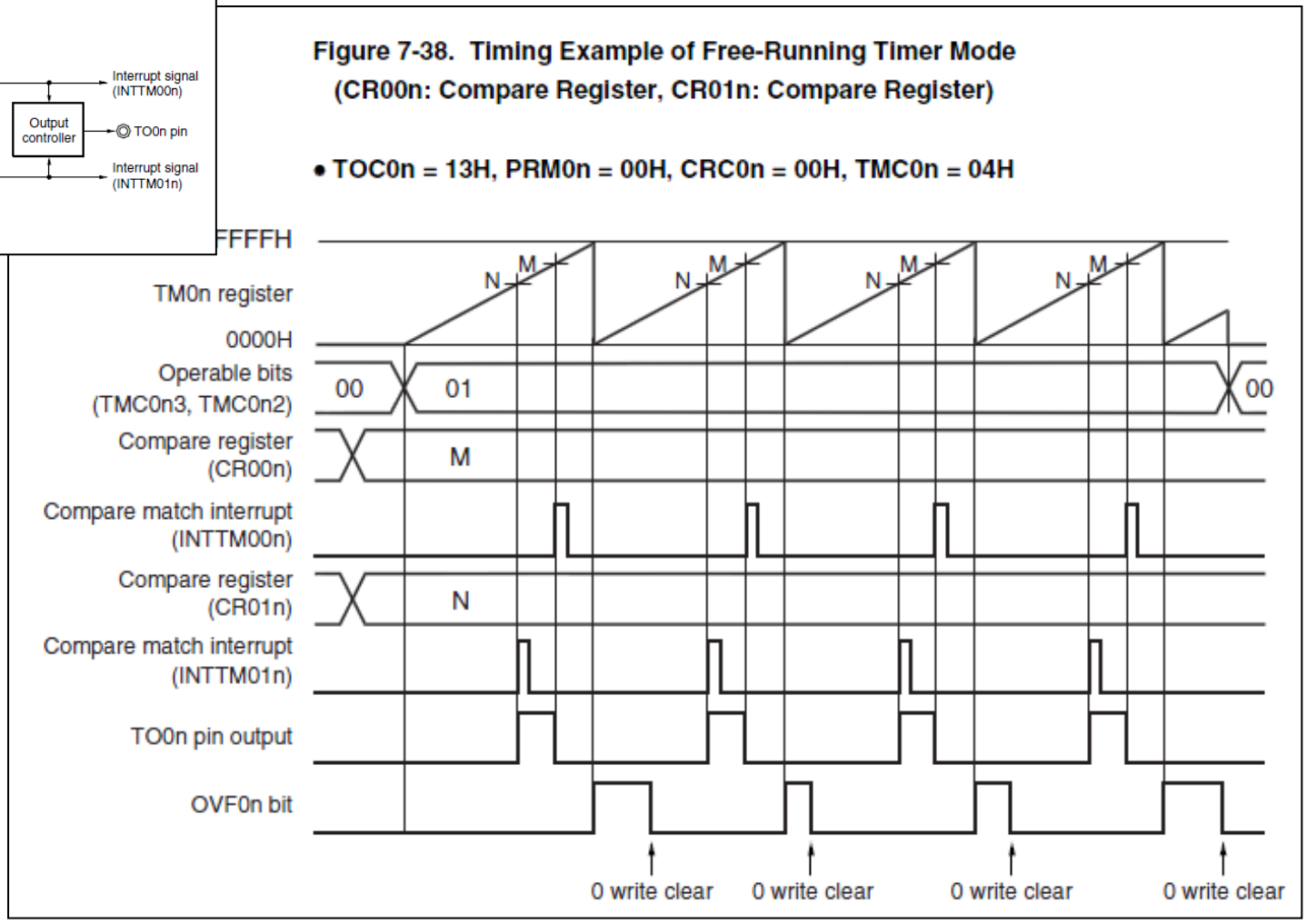
Selects count clock (specifies valid edge of TI00n).

00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection

Free-running timer operation 1

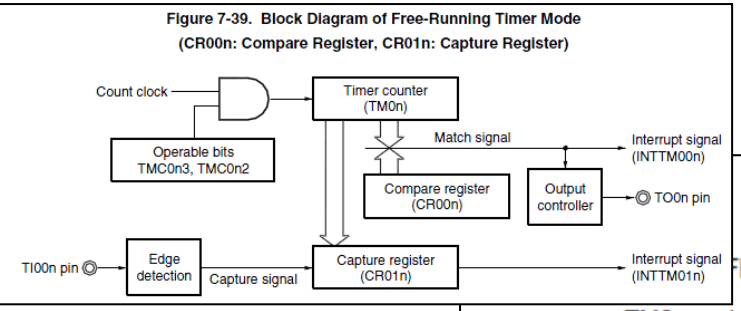
CR00n & CR01n → compare reg

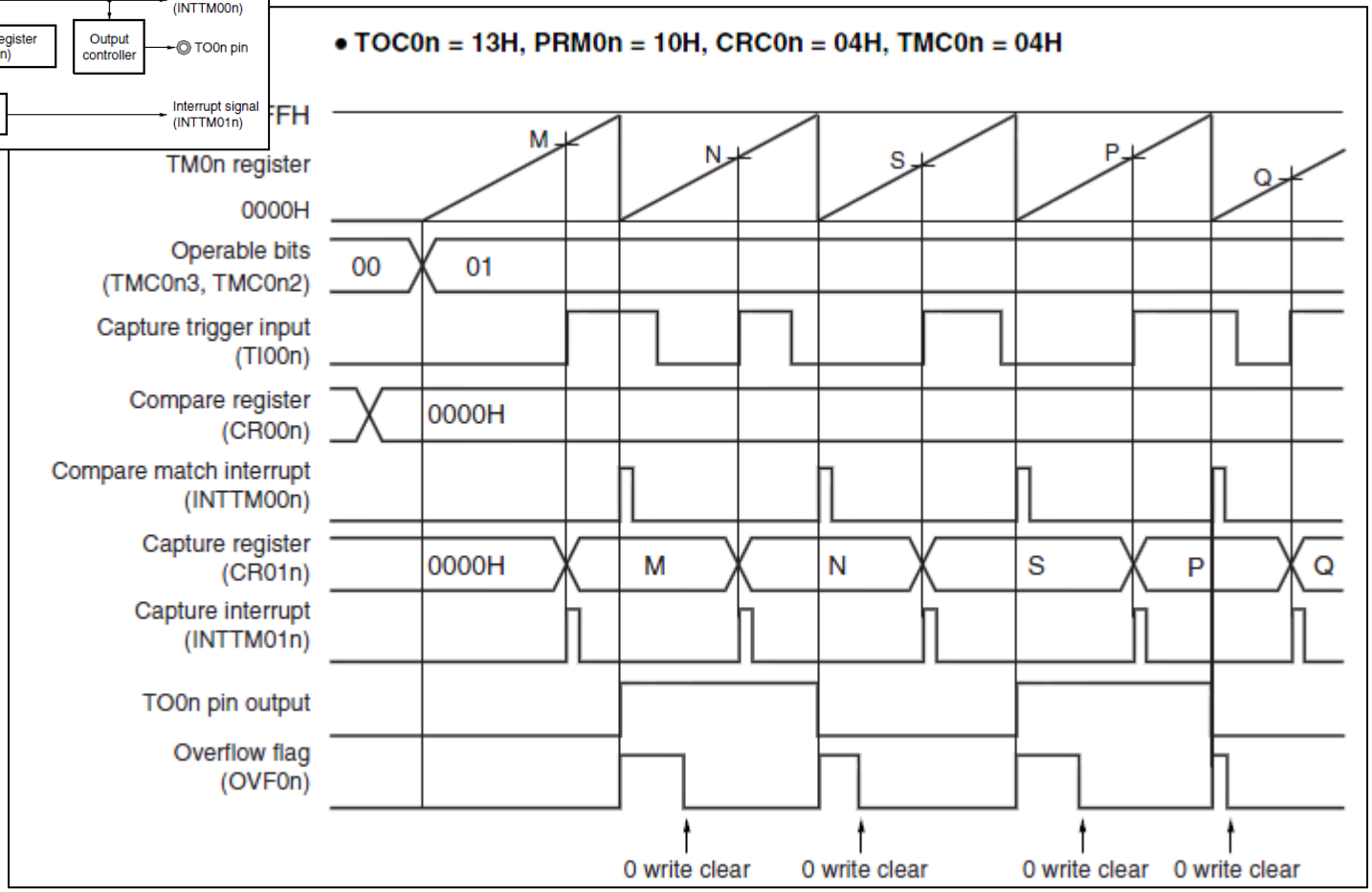




Free-running timer operation 2

CR00n: compare, CR01n : capture





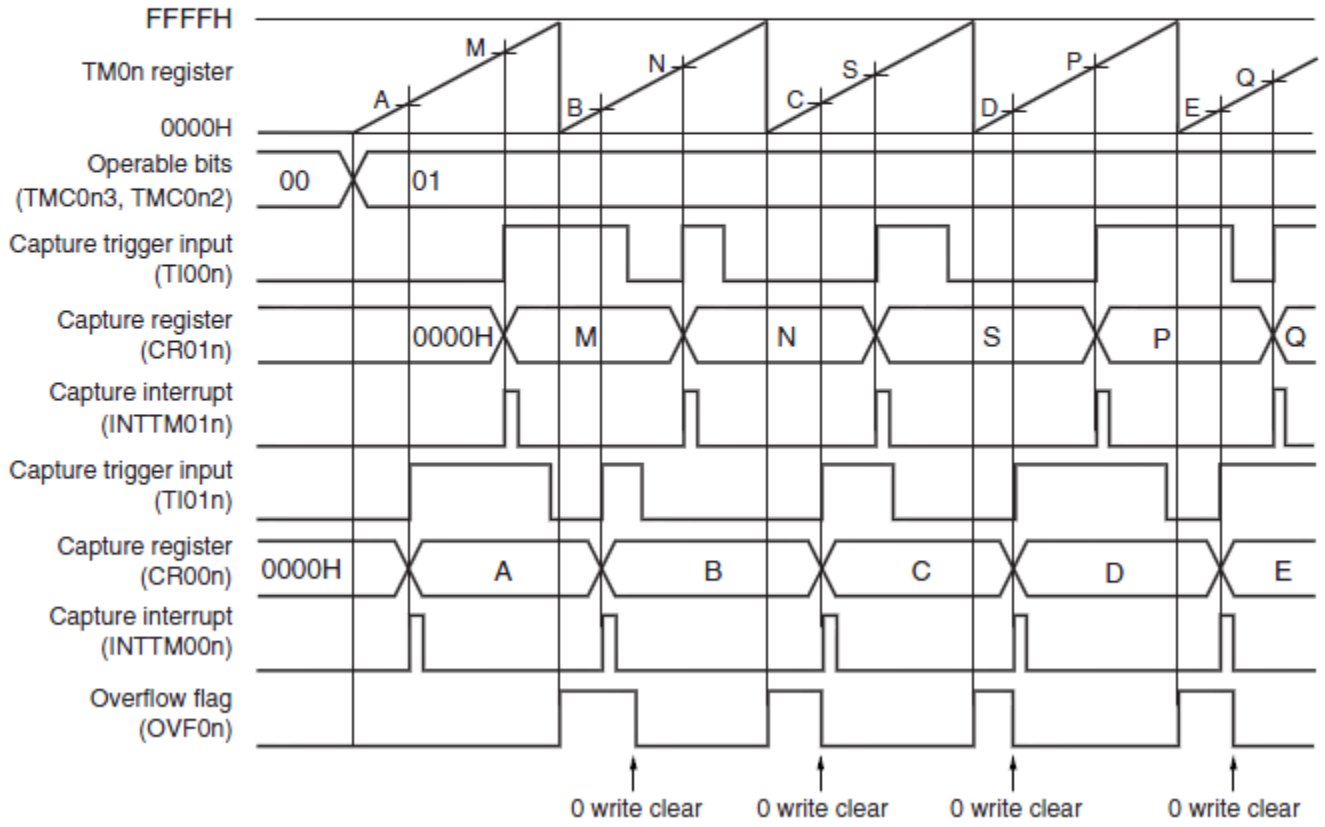
한림대학교 HALLYM UNIVERSITY

Free-running timer operation 3

CR00n: capture, CR01n : capture

Figure 7-42. Timing Example of Free-Running Timer Mode (CR00n: Capture Register, CR01n: Capture Register) (1/2)

(a) TOC0n = 13H, PRM0n = 50H, CRC0n = 05H, TMC0n = 04H

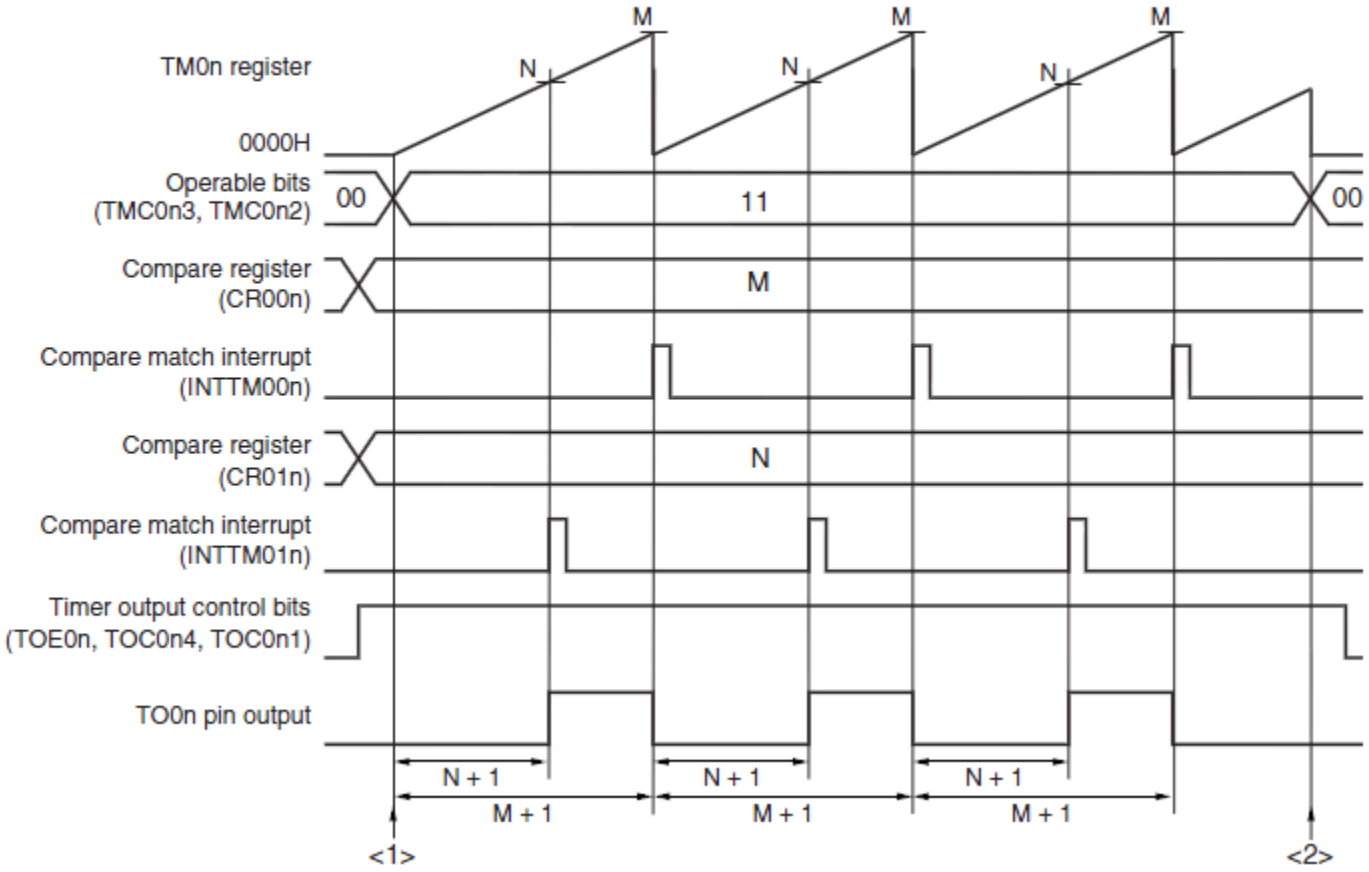


한림대학교 HALLYM UNIVERSITY

PPG output operation

- PPG: Programmable Pulse Generation

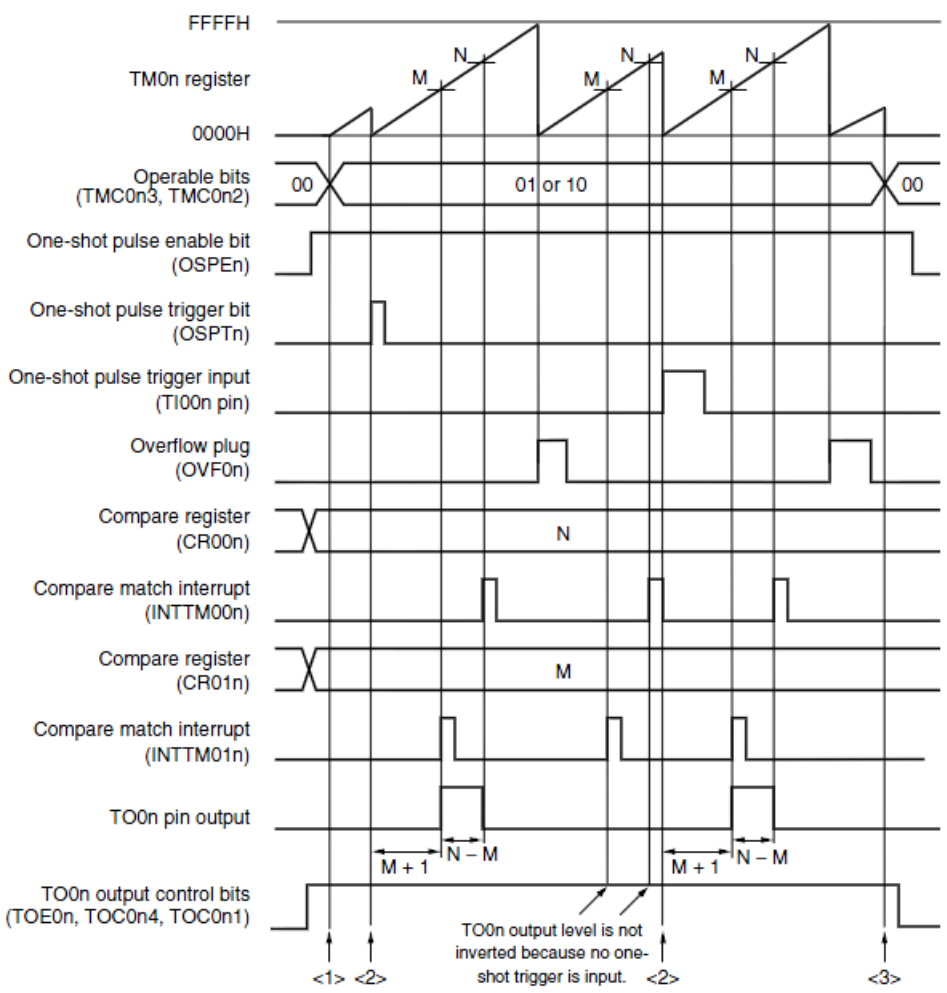
Figure 7-47. Example of Software Processing for PPG Output Operation



한림대학교 HALLYM UNIVERSITY

One-shot pulse output operation

Figure 7-50. Example of Software Processing for One-Shot Pulse Output Operation (1/2)



(a) 16-bit timer mode control register 0n (TMC0n)

0	0	0	0	TMC0n3 0/1	TMC0n2 0/1	TMC0n1 0	OVF0n 0

01: Free running timer mode
10: Clear and start mode by valid edge of TI00n pin.

(b) Capture/compare control register 0n (CRC0n)

0	0	0	0	0	CRC0n2 0	CRC0n1 0	CRC0n0 0

CR00n used as compare register
CR01n used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

0	OSPT0n 0/1	OSPE0n 1	TOC0n4 1	LVS0n 0/1	LVR0n 0/1	TOC0n1 1	TOE0n 1

Enables TO0n pin output
Specifies initial value of TO0n pin output
Inverts TO0n output on match between TM0n and CR00n/CR01n.
Enables one-shot pulse output
Software trigger is generated by writing 1 to this bit (operation is not affected even if 0 is written to it).

한림대학교 HALLYM UNIVERSITY

Code for sounding using Tmr01

```
#pragma vect INTTM001 inttm001_isr
void main(void)
{
    //포트 설정 for TO01
    PM6.0 = 0;   //make output

    //TM01 설정
    PRM01 = 0x02; //setup freq.=fPRS/2^6 ==> 312.5KHz
    CRC01 = 0;
    TOC01 = 0x03; // enable TO01 pin, invert match
    CR001 = 3125; //make 200Hz clock
    TMC01 = 0x0c;
    //interrupt setup
    TMMK001 = 0;
    EI();   //or DI();
    while(1);
}
__interrupt void inttm001_isr(void)
{
     LED0 ^= 1; //toggle LED0
}
```

TMR01의 출력(TO01/P0.6)이 간단한 OP amp.를 거쳐 스피커에 연결되어 있다.
이를 이용하여 간단하게 도/레/미를 연주한다.

- TMR 설정 순서
- Prescaler mode reg., Capture control reg., Output control reg., Timer control mode reg. (starting timer..)