CPE 323
Intro to Embedded Computer Systems
Timers

Aleksandar Milenkovic

milenka@uah.edu

Admin

1. Practice quiz for interrupts (closes Sunday)
2. Quiz_05 Monday or Tuesday next week

Midterm Review (1)

Question 1 12 pts

Consider a processor (iChargeputer20) with 16 20-bit registers, R0-R15. The register R0 acts as the program counter and a byte is the smallest addressable unit. Answer the following questions.

Note: 2^8=256; 2^9=512; 2^10=1,024; 2^15=32,768; 2^16=65,536; 2^19 =524,288; 2^20=1,048,576; 2^23=8,388,608; 2^24=16,777,216.

a. What is the size of the iChargeputer20 address space?

Size: [ ] bytes; (Enter the number of bytes) 1048576

b. What is range of unsigned integers stored in an iChargeputer20 register?

Unsigned int range: Min= 0 , Max= 1048575 (enter minimum and maximum numbers in the decimal number system)

c. What is the range of 2's complement signed integers stored in an iChargeputer20 registers?

Signed int range: Min= -524,288 , Max= 524287 (enter minimum and maximum numbers in the decimal number system)

d. How many bits do we need in an iCrageputer20 instruction to encode one general-purpose register?

Register field size: [4] bits

$\log_2 N_{reg}$

19 ... 0

$0 \div 2^{20} - 1$

2^{20}

24

$n = 24$

$0 \div 2^{24} - 1$

$-2^{23} \div 2^{23} - 1$

$[-2^{n-1} \div 2^{n-1} - 1]$

$-2^{19} \div 2^{19} - 1$

opcode	S-reg		D-reg

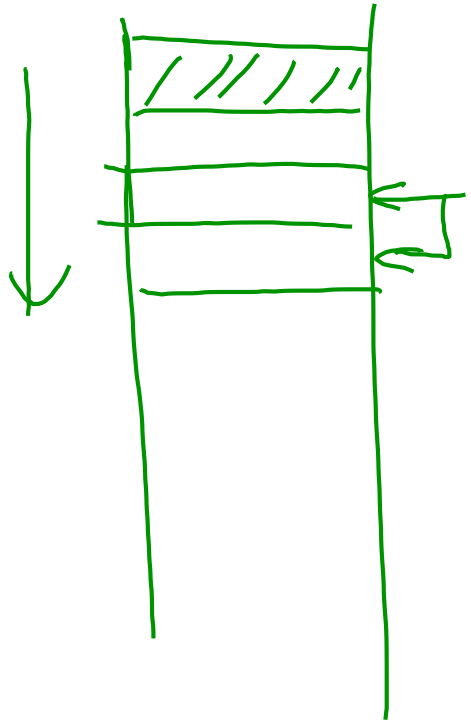
Midterm Review (2, 3)

Question 2 2 pts

MSP430 PUSH onto the stack [Select] decreases the stack pointer and POP from the stack [Select] increases the stack pointer.

Question 3 2 pts

Select correct options in the following statement. The baseline MSP430 architecture (not extended) includes [Select] 16 [Select] 16 -bit registers. Register R1 is [Select] stack pointer, register R2 is [Select] status register.



Midterm Review (4)

0400 + 0FFE = 13FE

Question 4 4 pts

Consider an MSP430 processor with the baseline architecture (address space is 2^16 bytes). It has 32 KB flash memory placed at the upper half of address space and 4 KB of RAM memory that starts at the address 0x0400. First 512 bytes of address space (starting at the address 0x0000) is reserved for I/O peripherals. Fill in the following entries by specifying the first and last word addresses. Note: for all addresses enter hex values starting with 0x.

Flash memory: Start address: 0x8000 End address: 0xFFFE

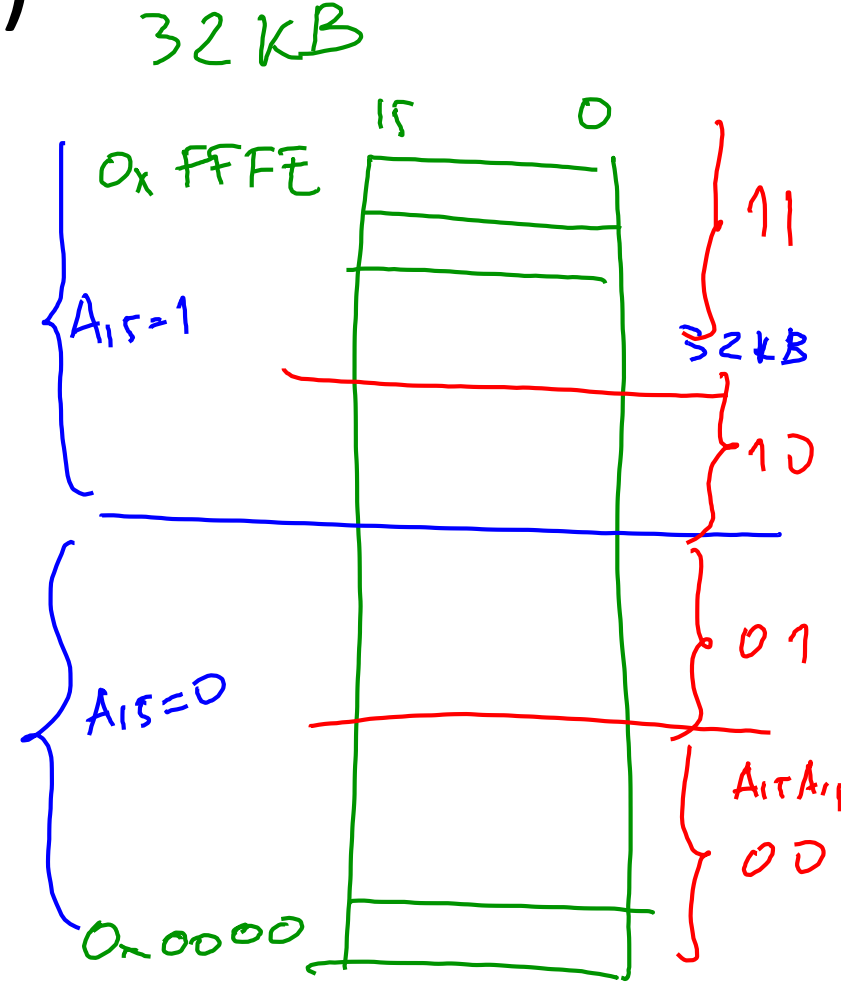
RAM memory: Start address: 0x0400 End address: 0x13FE

I/O space: Start address: 0x0000 End address: 0x01FE

29 1.1111.1110 1 F E

0FFE 1111... 12

4 KB = $2^2 \cdot 2^{10} = 2^{12}$



A15 A14 A13 - - - A0
1 0 0 - - - 0

Midterm Review (5)

Question 5 20 pts

Consider the following .data section starting at the address 0x1000. Assume R5=0x2000.

```
.data
lws: .word 1, 2, -4
```

Finish statements below by selecting correct options from dropdown menus for the following two instructions (A and B).

Instruction A:

```
MOV.W  lws, 0(R5);
```

Source addressing mode is [Select] ;

Destination addressing mode is [Select] ;

Source operand address is [Select] ;

Destination operand address is [Select] ;

Instruction size is [Select] bytes.

Instruction B:

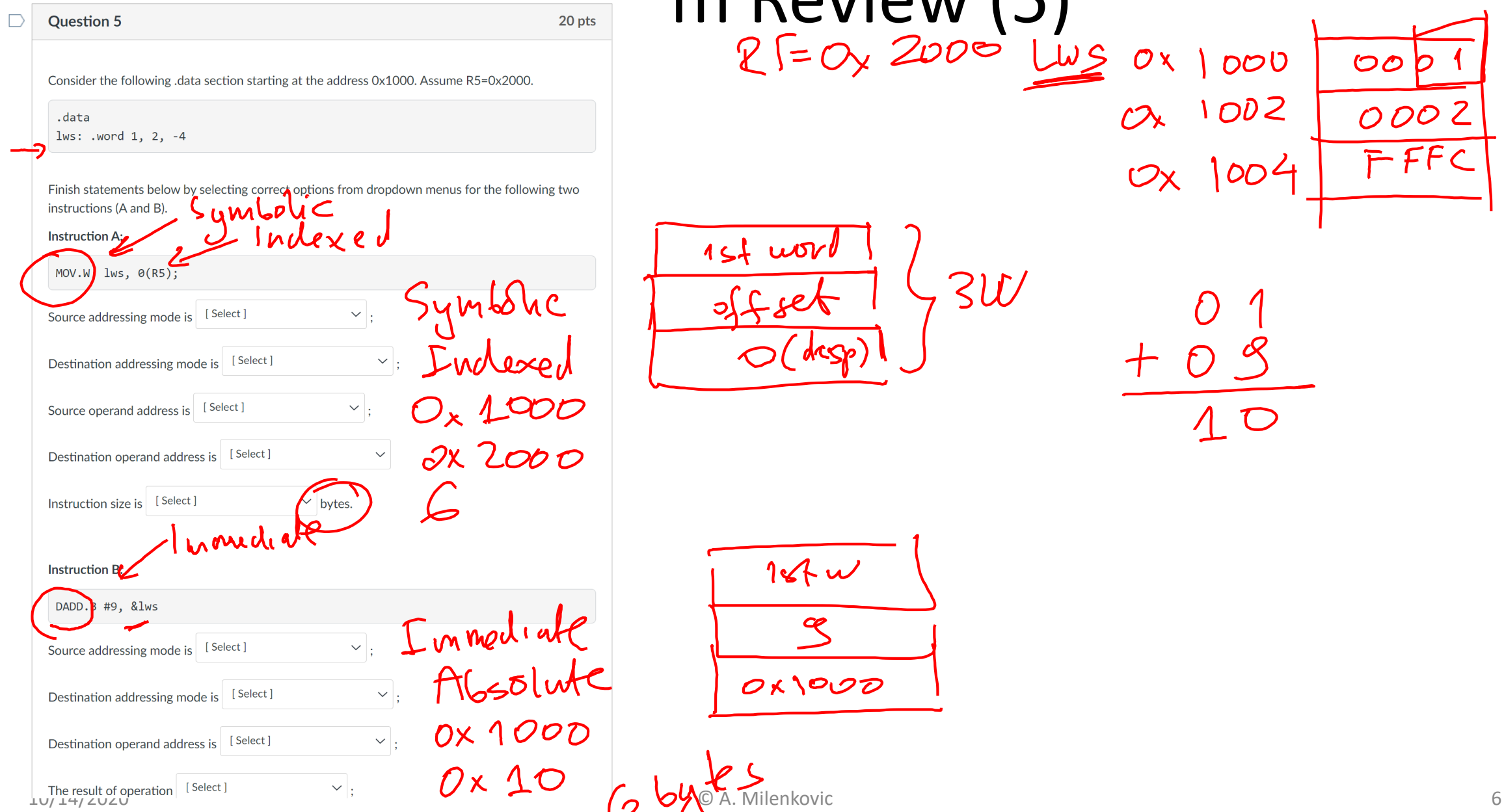
```
DADD.B #9, &lws
```

Source addressing mode is [Select] ;

Destination addressing mode is [Select] ;

Destination operand address is [Select] ;

The result of operation [Select] ;



Watchdog Timers, Timers A/B

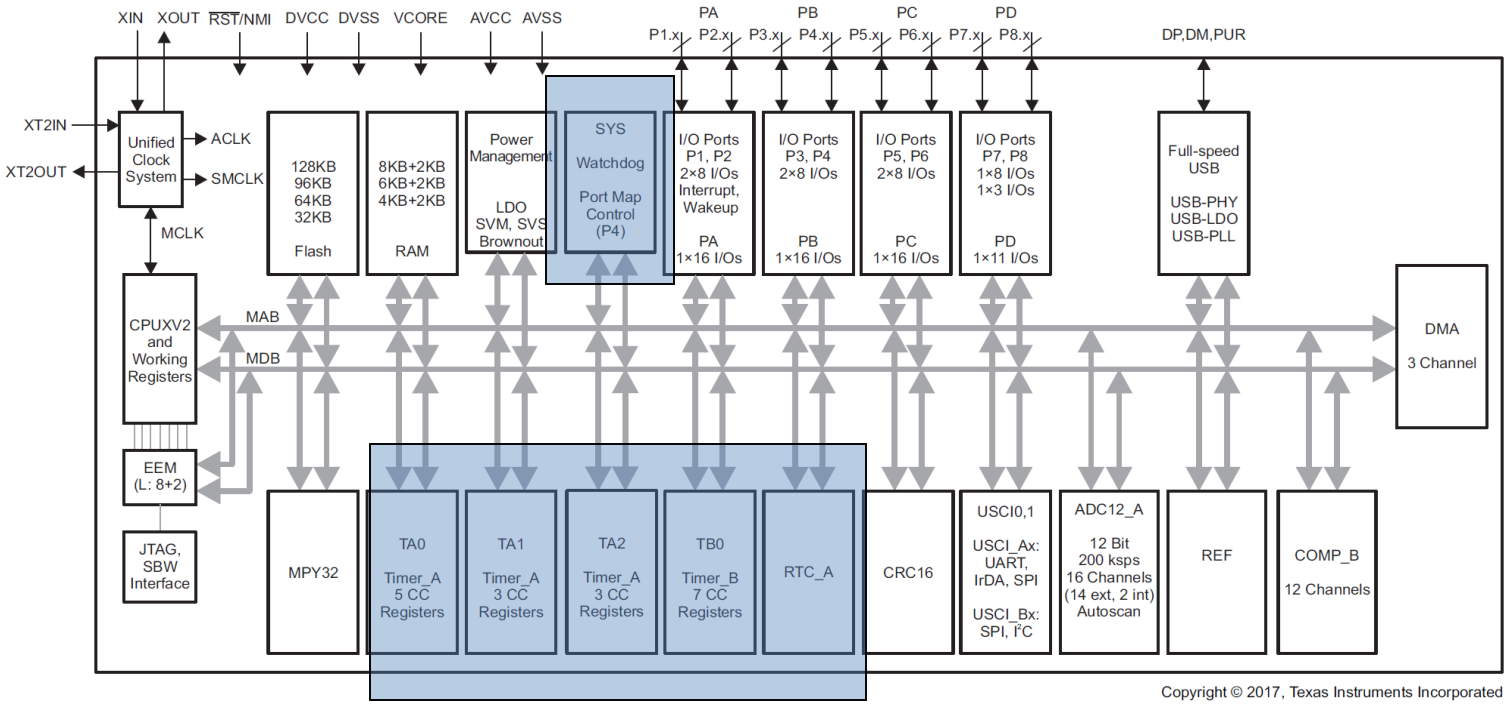
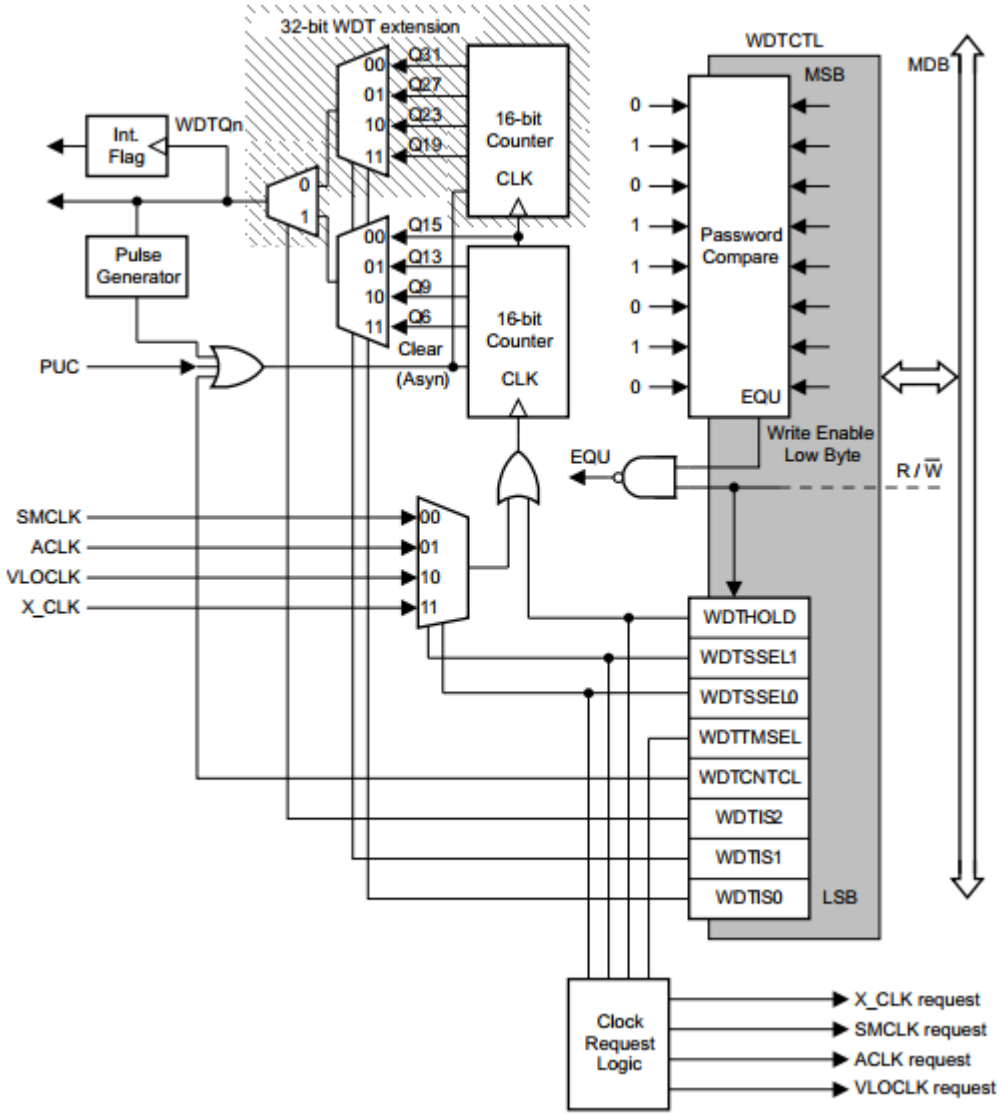


Figure 1-1. Functional Block Diagram – MSP430F5529IPN, MSP430F5527IPN, MSP430F5525IPN, MSP430F5521IPN

Watchdog Timer



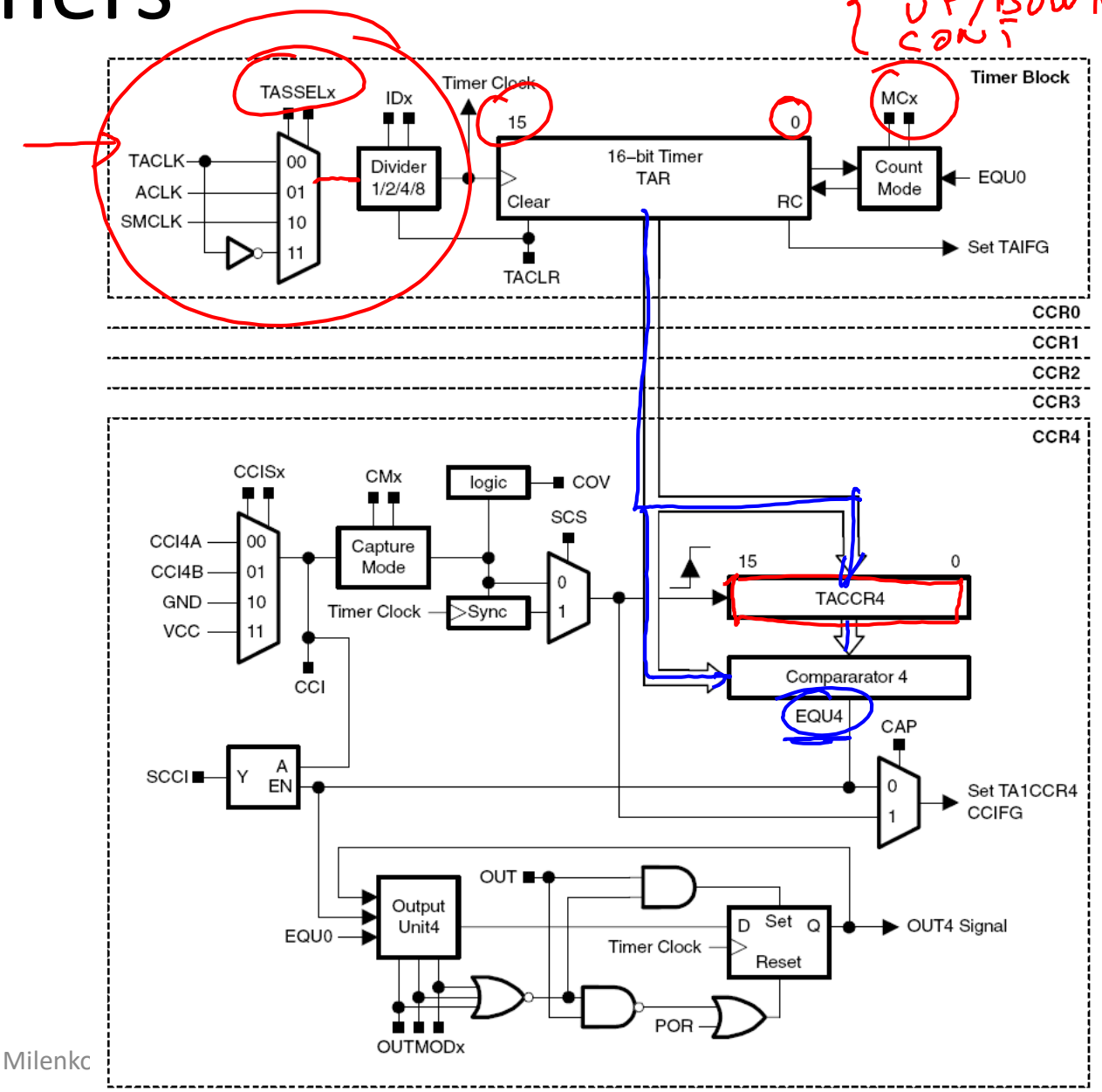
Watchdog Timer Registers

15	14	13	12	11	10	9	8
WDTPW							
7	6	5	4	3	2	1	0
WDTHOLD	WDTSSEL		WDTTMSEL	WDTCNTCL	WDTIS		
rw-0	rw-0	rw-0	rw-0	r0(w)	rw-1	rw-0	rw-0

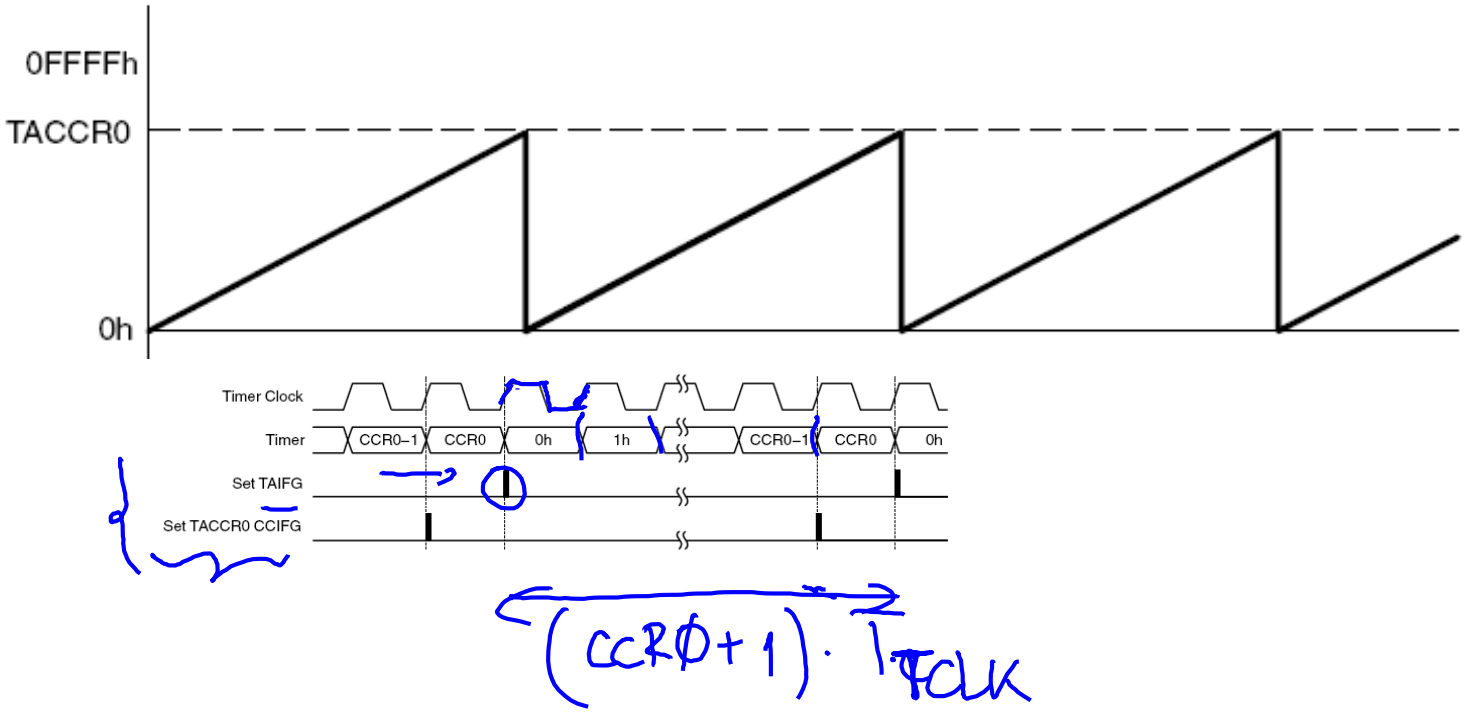
Bit	Field	Type	Reset	Description
15-8	WDTPW	RW	69h	Watchdog timer password. Always read as 069h. Must be written as 5Ah; if any other value is written, a PUC is generated.
7	WDTHOLD	RW	0h	Watchdog timer hold. This bit stops the watchdog timer. Setting WDTHOLD = 1 when the WDT is not in use conserves power. 0b = Watchdog timer is not stopped. 1b = Watchdog timer is stopped.
6-5	WDTSSEL	RW	0h	Watchdog timer clock source select 00b = SMCLK 01b = ACLK 10b = VLOCLK 11b = X_CLK; VLOCLK in devices that do not support X_CLK
4	WDTTMSEL	RW	0h	Watchdog timer mode select 0b = Watchdog mode 1b = Interval timer mode
3	WDTCNTCL	RW	0h	Watchdog timer counter clear. Setting WDTCNTCL = 1 clears the count value to 0000h. WDTCNTCL is automatically reset. 0b = No action 1b = WDTCNT = 0000h
2-0	WDTIS	RW	4h	Watchdog timer interval select. These bits select the watchdog timer interval to set the WDTIFG flag and/or generate a PUC. 000b = Watchdog clock source /(2^{31}) (18h:12m:16s at 32.768 kHz) 001b = Watchdog clock source /(2^{27}) (01h:08m:16s at 32.768 kHz) 010b = Watchdog clock source /(2^{23}) (00h:04m:16s at 32.768 kHz) 011b = Watchdog clock source /(2^{19}) (00h:00m:16s at 32.768 kHz) 100b = Watchdog clock source /(2^{15}) (1 s at 32.768 kHz) 101b = Watchdog clock source /(2^{13}) (250 ms at 32.768 kHz) 110b = Watchdog clock source /(2^{9}) (15.625 ms at 32.768 kHz) 111b = Watchdog clock source /(2^{6}) (1.95 ms at 32.768 kHz)

Timers

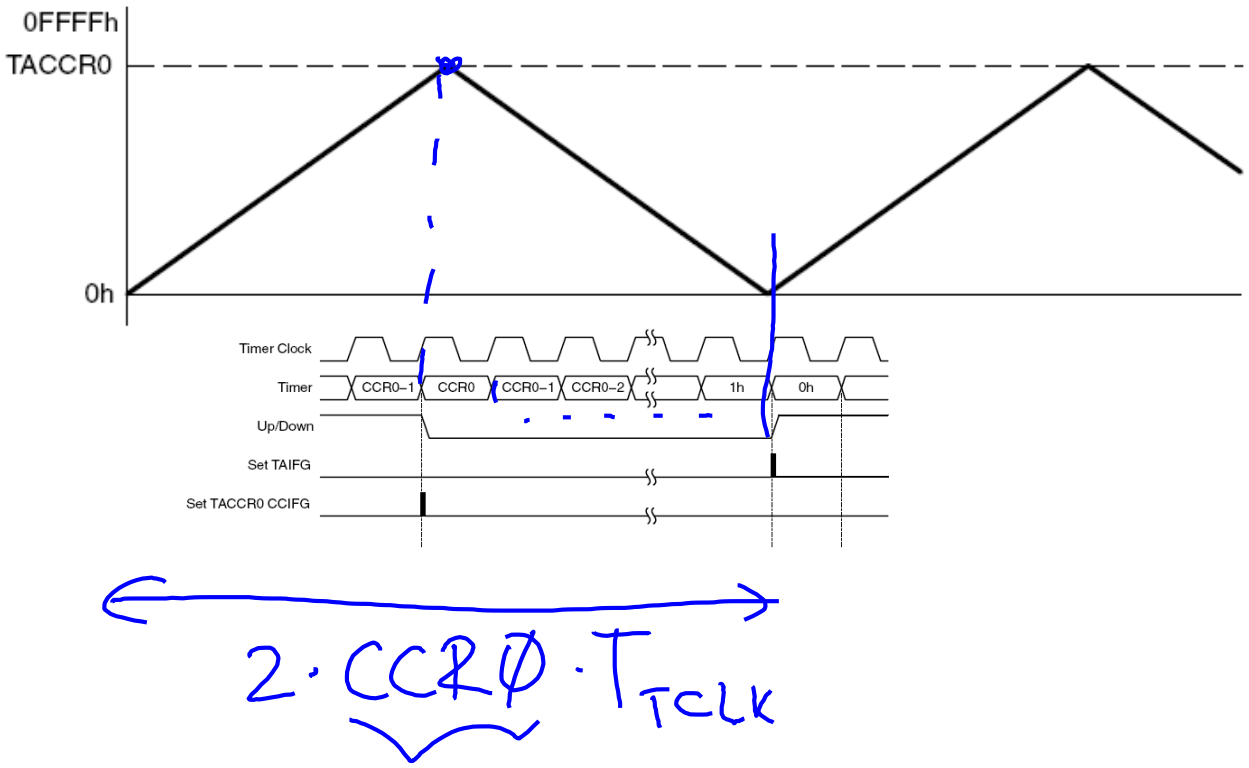
- Timer Block (counter)
- Capture/Compare Blocks



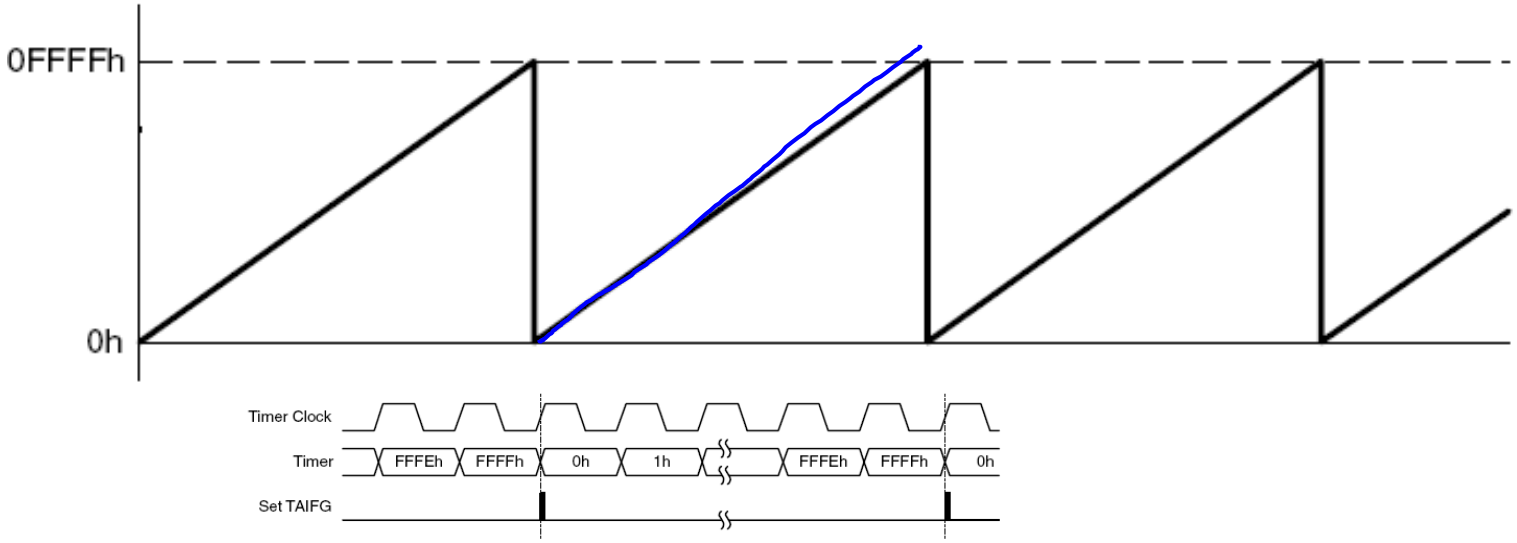
UP Mode



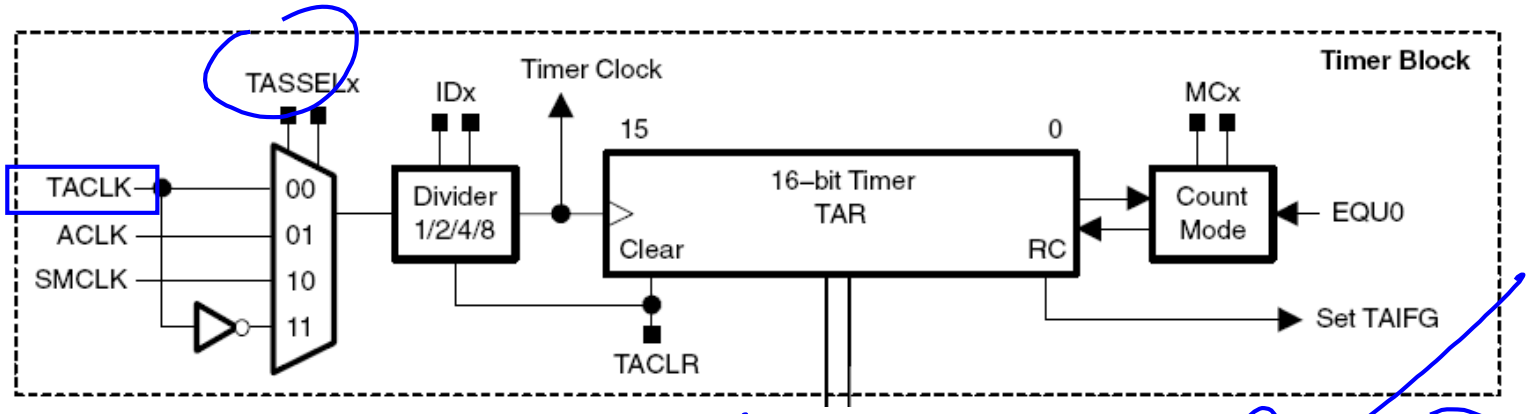
UP/Down Mode



Continuous Mode



Counter



TACTL

160h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
unused						Input Select		Input Divider		Mode Control		un-used	CLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	(w)-(0)	rw-(0)	rw-(0)

MC1	MC0	
0	0	Stop Mode
0	1	Up Mode
1	0	Continuous Mode
1	1	Up/Down Mode

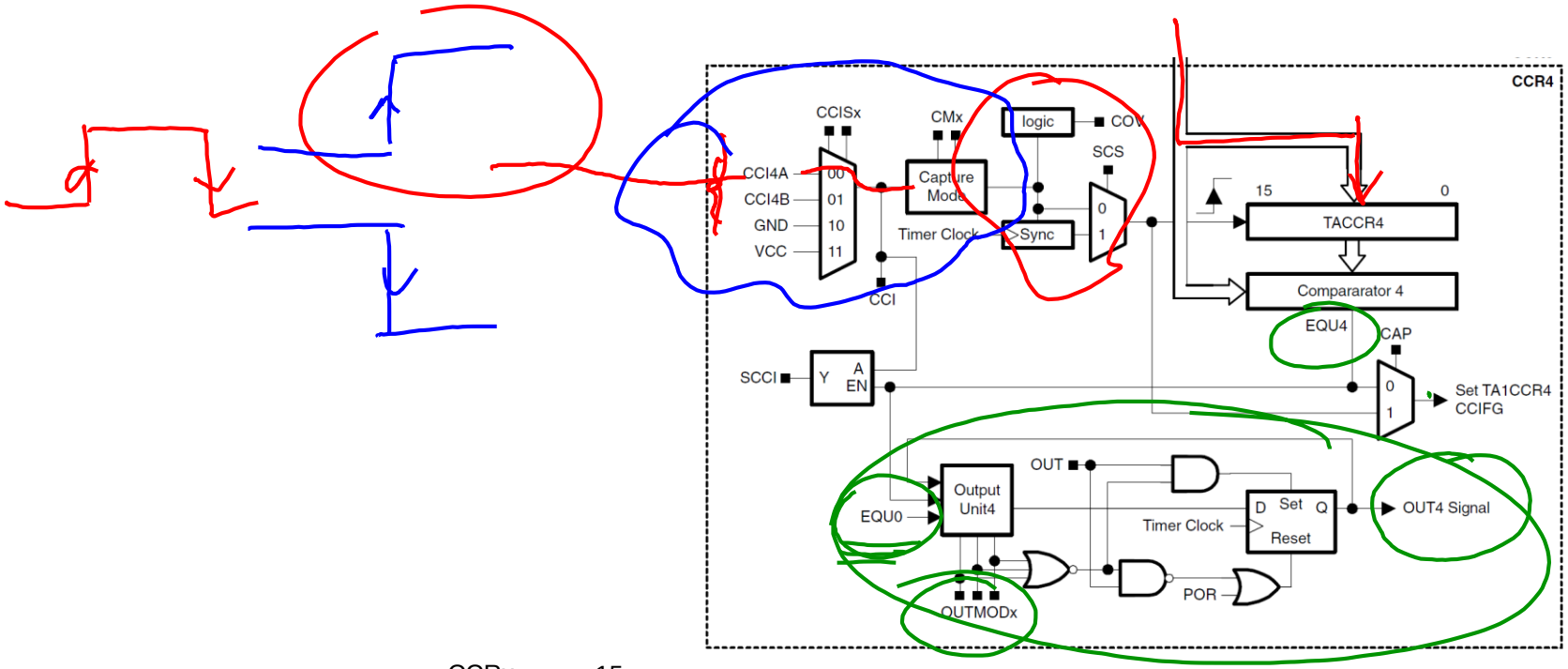
ID1	ID0	
0	0	1/1, Pass
0	1	1/2
1	0	1/4
1	1	1/8

SSEL1	SSEL0	
0	0	TACLK
0	1	ACLK
1	0	MCLK
1	1	INCLK (often = #TACLK)

TACLR – clears TAR and resets the direction of counting (it clears automatically itself)

TAIFG – set when the timer counts to 0; a maskable interrupt is requested if TAIE bit is set

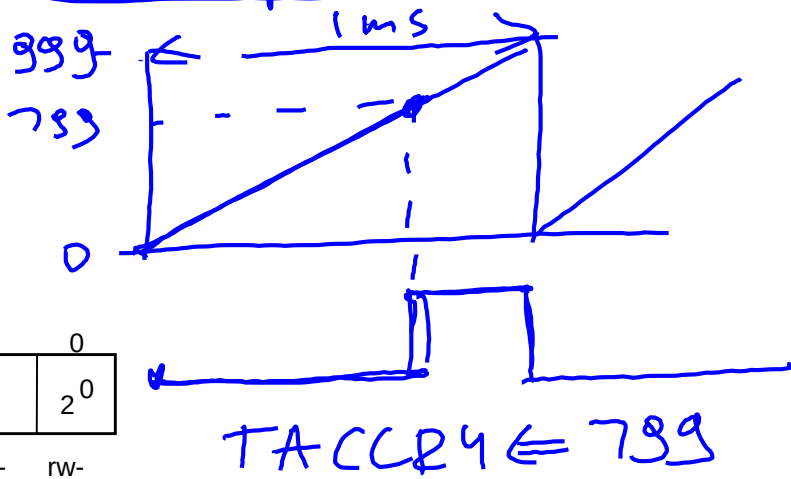
Capture & Compare Block



Capture

TACCR4 ⇐ TA

Compare (1 μs)



CCRx
0172h
to
017Eh

15															0
2^{15}															2^0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

CCTLx
162h
to
16Eh

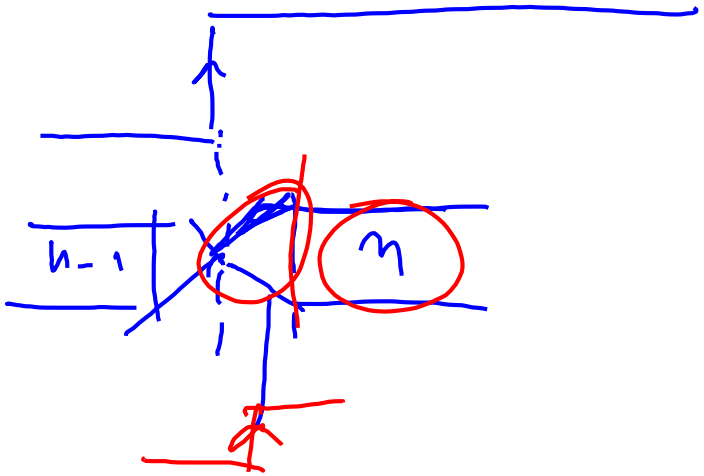
15															0
CAPTURE MODE		INPUT SELECT		SCS	SCCI	un-used	CAP	OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

TACCTLn: Capture Control

- **CMx** (Capture Mode)
  - 00 – disabled
  - 01 – positive edge
  - 10 – negative edge
  - 11 – both edges
- **CCISx** (Capture Input Select)
  - 00 – CCInA (outside timer)
  - 01 – CCInB (outside timer)
  - 10 – Gnd (pointless, but allows captures from SW)
  - 11 – Vdd (pointless, but allows captures from SW)
  - (for SW-triggered captures: use CMx=11, set CCIS1=1, and toggle CCIS0)
- **SCS** – synchronizer bit ensures synchronization with the timer clock (SHOULD always be set)
  - Race conditions: the selected input changes at the same time as the timer clock
- **CCI** – the state of the selected input can be read at any time from SW
- **OUT** – For output mode 0, this bit directly controls the state of the output

15	14	13	12	11	10	9	8
CMx		CCISx		SCS	SCCI	Unused	CAP
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	r	r0	rw–(0)

7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw–(0)	rw–(0)	rw–(0)	rw–(0)	r	rw–(0)	rw–(0)	rw–(0)



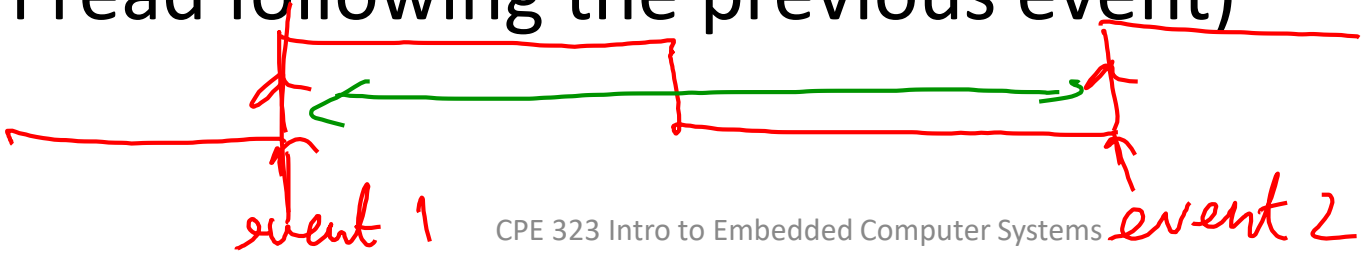
TACCTLn: Capture Control

- CAP: Capture mode
  - 0 - Compare mode
  - 1 - Capture mode

15	14	13	12	11	10	9	8
CMx		CCISx		SCS	SCCI	Unused	CAP
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	r	r0	rw–(0)

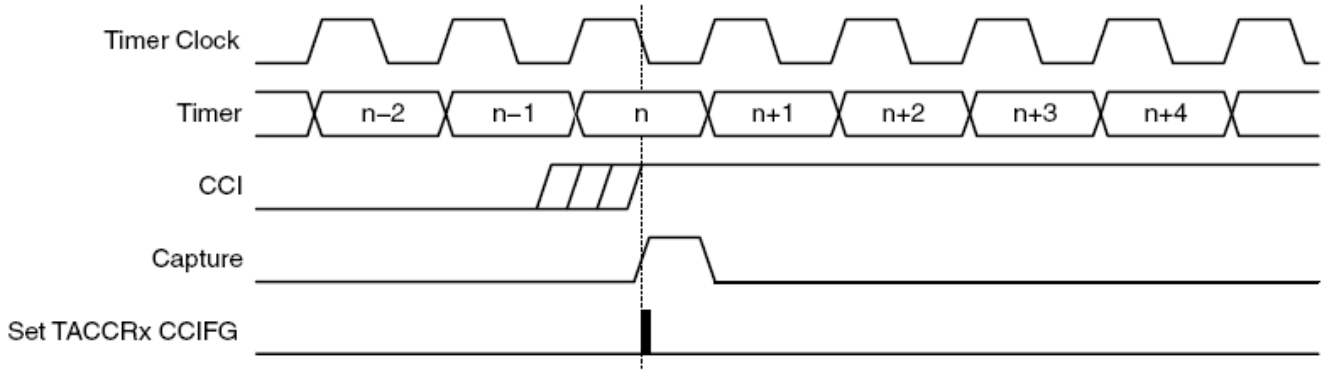
7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw–(0)	rw–(0)	rw–(0)	rw–(0)	r	rw–(0)	rw–(0)	rw–(0)

- Capture: TAR is copied into TACCRn, the channel flag CCIFGn is set, and a maskable interrupt is requested if bit CCIE in TACCTLx is set
- COV: Capture Overflow (next capture occurs before the TACCRn has been read following the previous event)

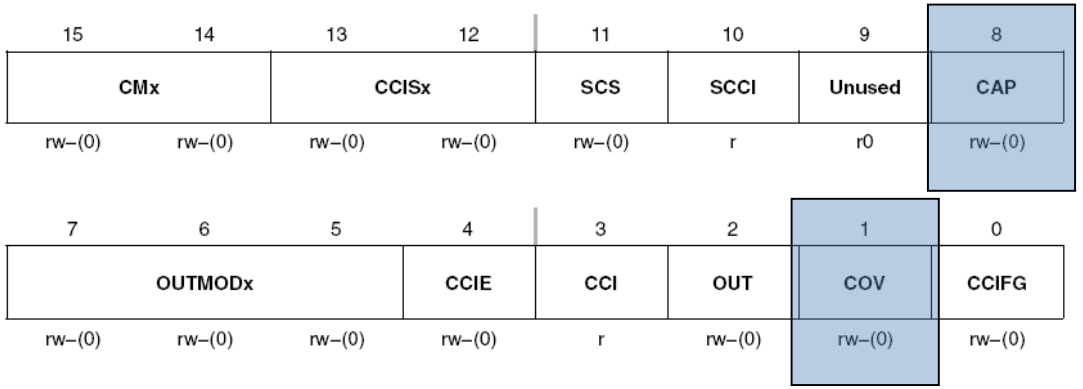


Capture Signal Synchronization

- The capture signal can be asynchronous to the timer clock and cause a race condition
- => Setting the SCS bit synchronizes the capture with the next timer clock



TACCTLn: Compare Mode

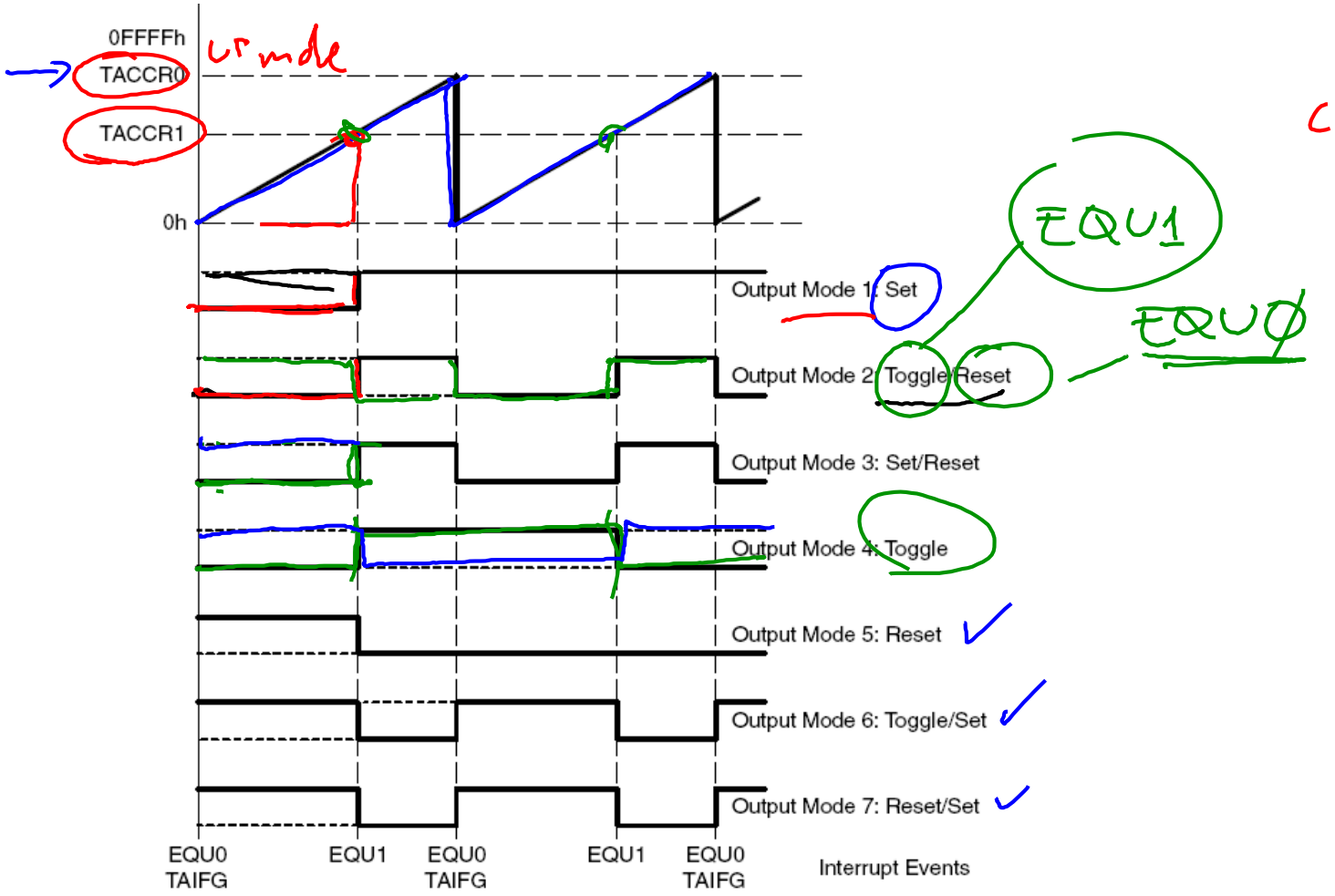


15	14	13	12	11	10	9	8
CMx		CCISx		SCS	SCCI	Unused	CAP
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	r	r0	rw–(0)

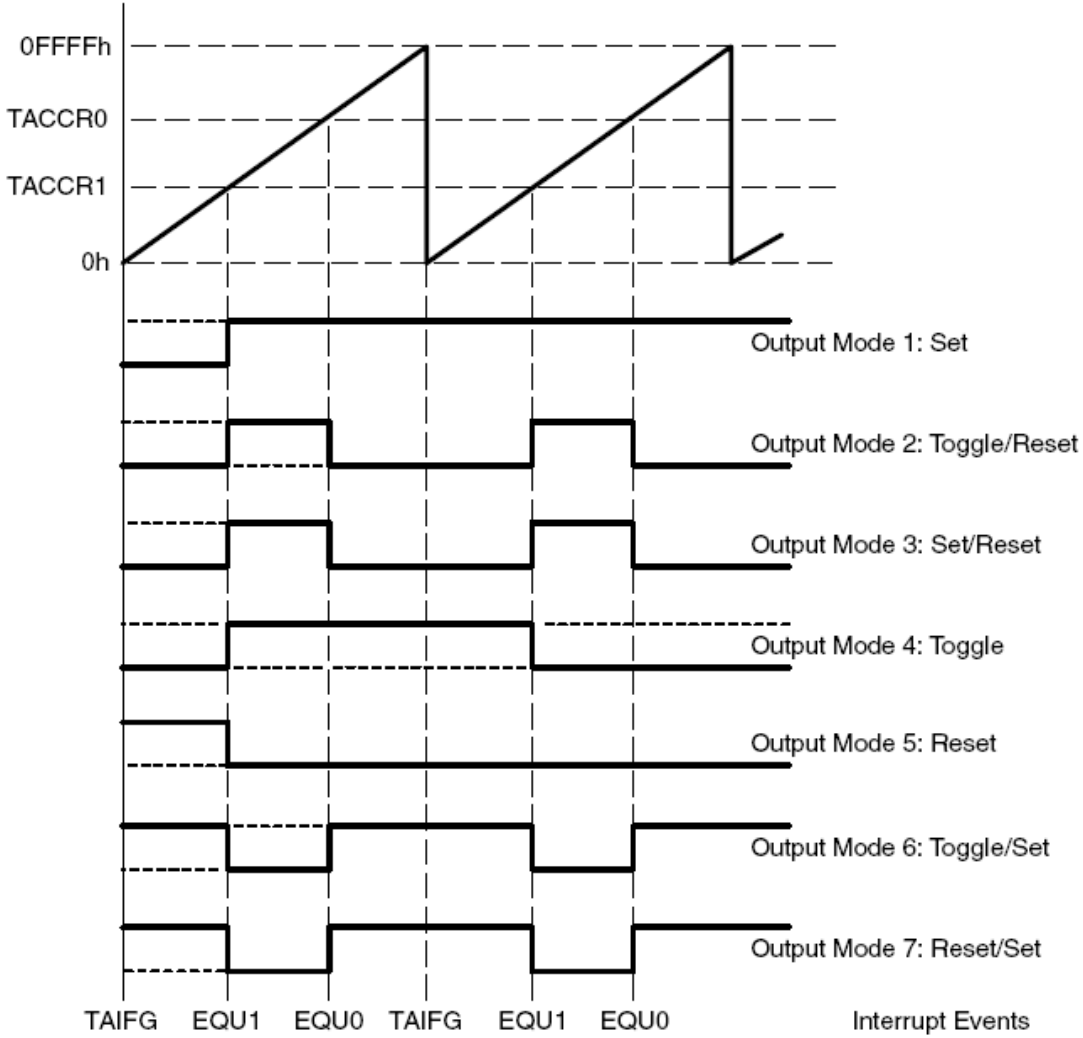
7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw–(0)	rw–(0)	rw–(0)	rw–(0)	r	rw–(0)	rw–(0)	rw–(0)

- Compare mode: produces an output anc at the time stored in TACCRn
- Actions when TAR reaches value in TACCRn
  - Internal EQU is set
  - CCIFGn flag is set and an interrupt is requested if enabled
  - Output OUTn is changed according to the mode set in OUTMODx bits in TACCTLn
  - Input signal to the capture HW, CCI, is latched into the SCCI bit
- Use compare mode to trigger periodic events on other peripherals (e.g., DAC, ADC)

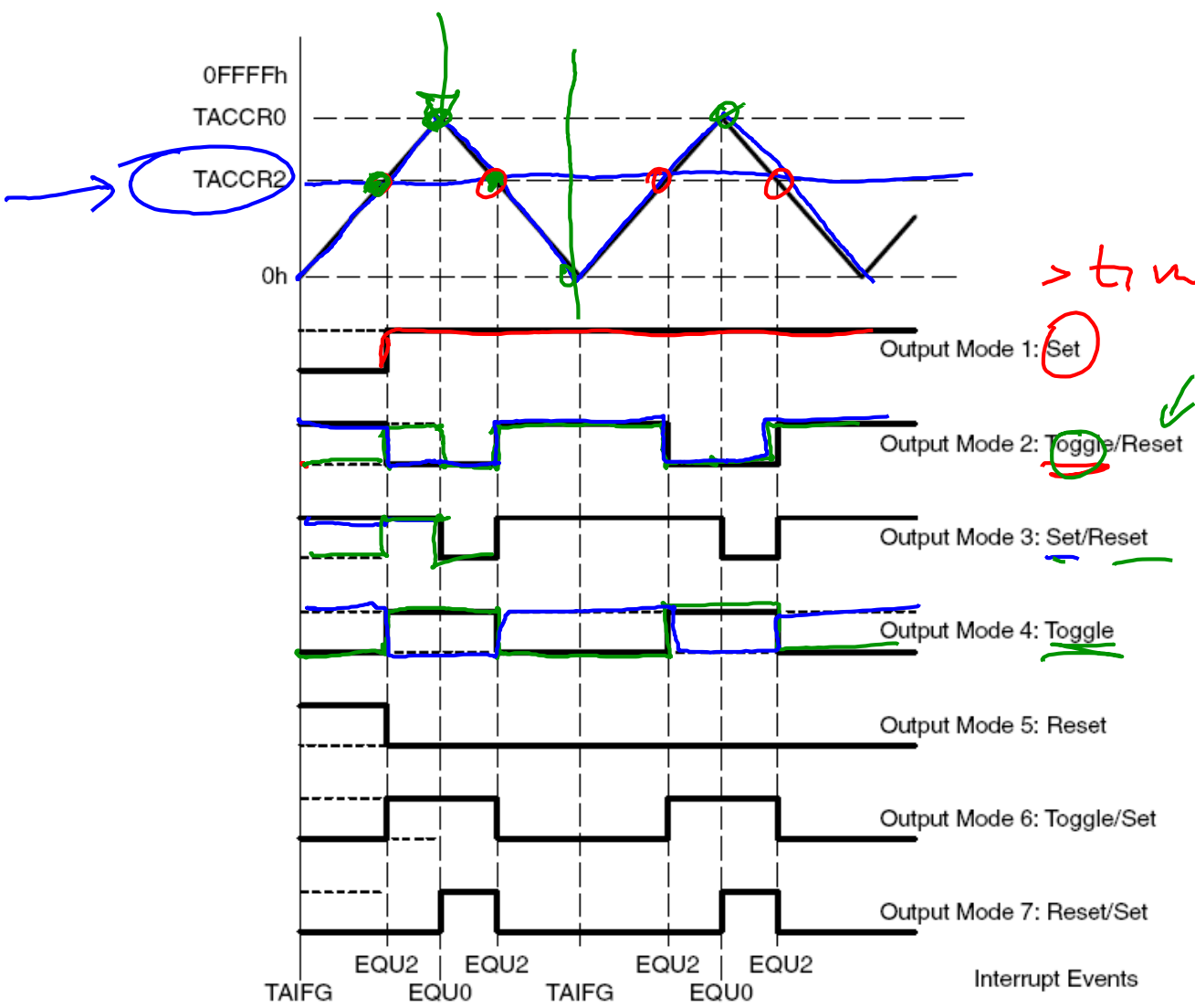
Output Modes (UP)

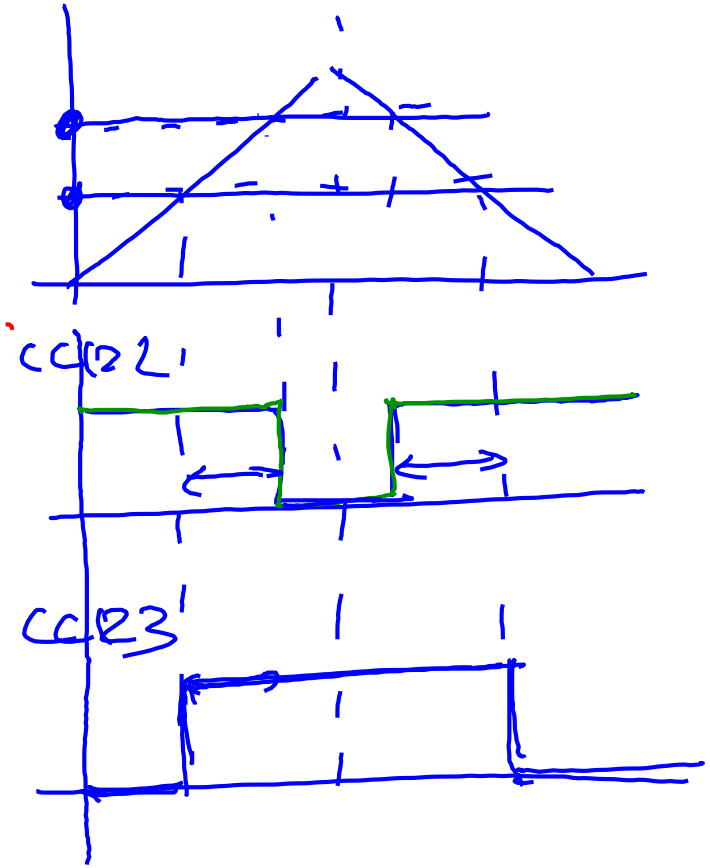


Output Modes (CONT)



Output Modes (UP/DOWN)





Toggle Pin 2.4 Using Timer A (A2)

```
/*-------------------------------------------------------------------------
 * File:        Lab7_D4.c (CPE 325 Lab7 Demo code)
 *
 * Function:    Toggling signal using Timer_A2 in continuous mode (MPS430F5529)
 *
 * Description: In this C program, Timer_A2 is configured for continuous mode. In
 *              this mode, the timer TA2 counts from 0 up to 0xFFFF (default 2^16).
 *              So, the counter period is 65,536*1/2^20 = 62.5ms when SMCLK is
 *              selected. The TA2.1 output signal is configured to toggle every
 *              time the counter reaches the maximum value, which corresponds to
 *              62.5ms. TA2.1 is multiplexed with the P2.4, and there is a extension
 *              header from this pin.
 *
 *              Thus the output frequency on P2.4 will be f = SMCLK/(2*65536) ~ 8 Hz.
 *              Please note that once configured, the Timer_A toggles the signal
 *              in pin P2.4 automatically even when the CPU is in sleep mode.
 *              Please use oscillator to see this.
 *
 *              Using the Grove Boosterpack, you can hook-up the Buzzer to the
 *              J14 header. This connects the Signal Pin of buzzer to P2.4.
 *              The buzzer produces sound when the signal value is high
 *              and vice versa.
 *
```

Toggle Pin 2.4 Using Timer A (A2)

```
 *
 * Clocks:      ACLK = LFXT1 = 32768Hz, MCLK = SMCLK = DCO = default (2^20 Hz)
 *              An external watch crystal between XIN & XOUT is required for ACLK
 *
 *                          MSP430F5529
 *                      -----------------
 *                  /|\|              XIN|-
 *                   | |                 | 32kHz
 *                   --|RST          XOUT|-
 *                     |                 |
 *                     |       P2.4/TA2.1|-->Buzzer
 *                     |                 |
 * Input:       None
 * Output:      Toggle output at P2.4 at 8Hz frequency using hardware TA2
 * Author:      Aleksandar Milenkovic, milenkovic@computer.org
 *              Prawar Poudel
 *-------------------------------------------------------------------------------*/

#include <msp430F5529.h>

void main(void) {
    WDTCTL = WDTPW +WDTHOLD;  // Stop WDT

    P2DIR |= BIT4;            // P2.4 output (TA2.1)
    P2SEL |= BIT4;            // P2.4 special function (TA2.1 output)

    TA2CCTL1 = OUTMOD_4;      // TA2.1 output is in toggle mode
    TA2CTL = TASSEL_2 + MC_2; // SMCLK is clock source, Continuous mode

    _BIS_SR(LPM0_bits + GIE); // Enter Low Power Mode 0
}
```