ECE3411 – Fall 2016

Lecture 4c.

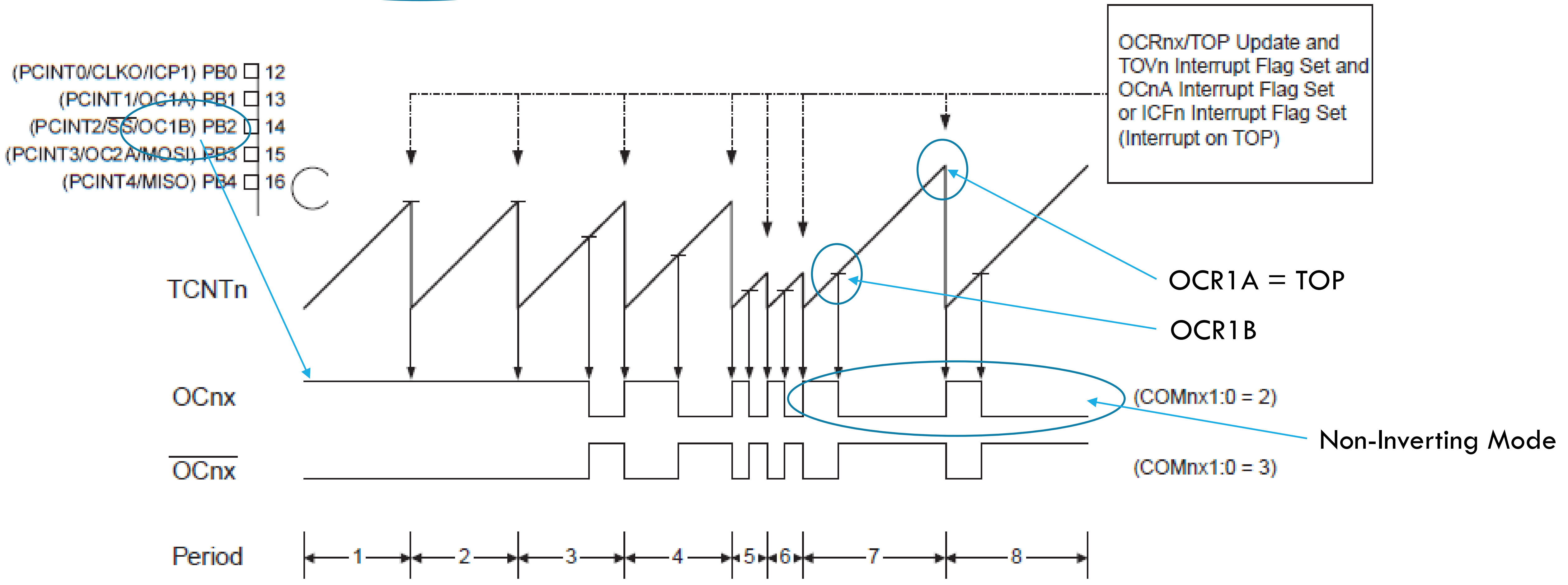
PWM: Pulse Width Modulation

Marten van Dijk
Department of Electrical & Computer Engineering
University of Connecticut
Email: marten.van_dijk@uconn.edu

Copied from Lecture 4c, ECE3411 – Fall 2015, by
Marten van Dijk and Syed Kamran Haider

Pulse Width Modulation using Timer 1

Figure 15-7. Fast PWM Mode, Timing Diagram



Frequency & Duty Cycle

- $F_{\{OC1B\}} = \frac{F_{\{CPU\}}}{prescalar * (1+OCR1A)}$

- $DutyCycle = \frac{1+OCR1B}{1+OCR1A}$

For example, frequency OCB1 at 523 Hz (c-note) with F_CPU = 16 MHz and no prescalar gives OCR1A = 30592

Waveform Generation Mode

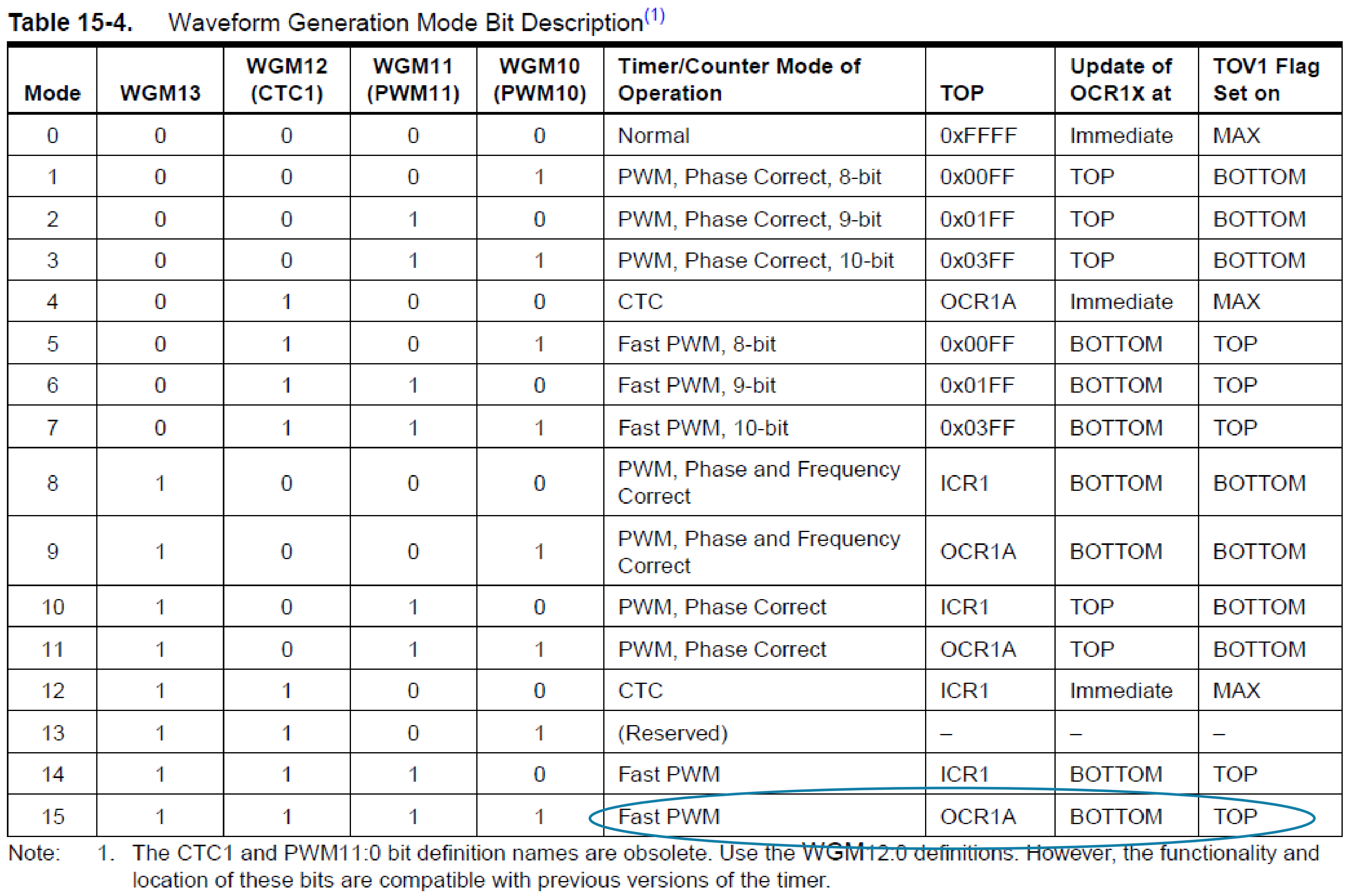


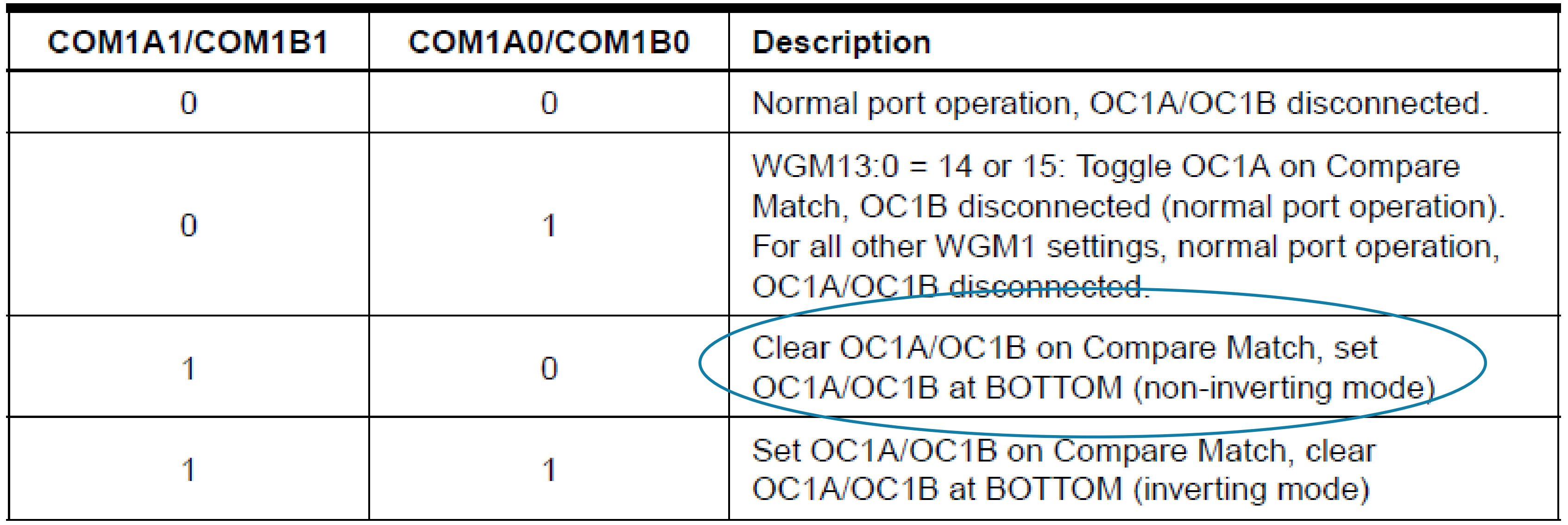
Table 15-4. Waveform Generation Mode Bit Description[1]

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1X at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

Compare Output Mode

Table 15-2. Compare Output Mode, Fast PWM[1]



COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

Prescalar

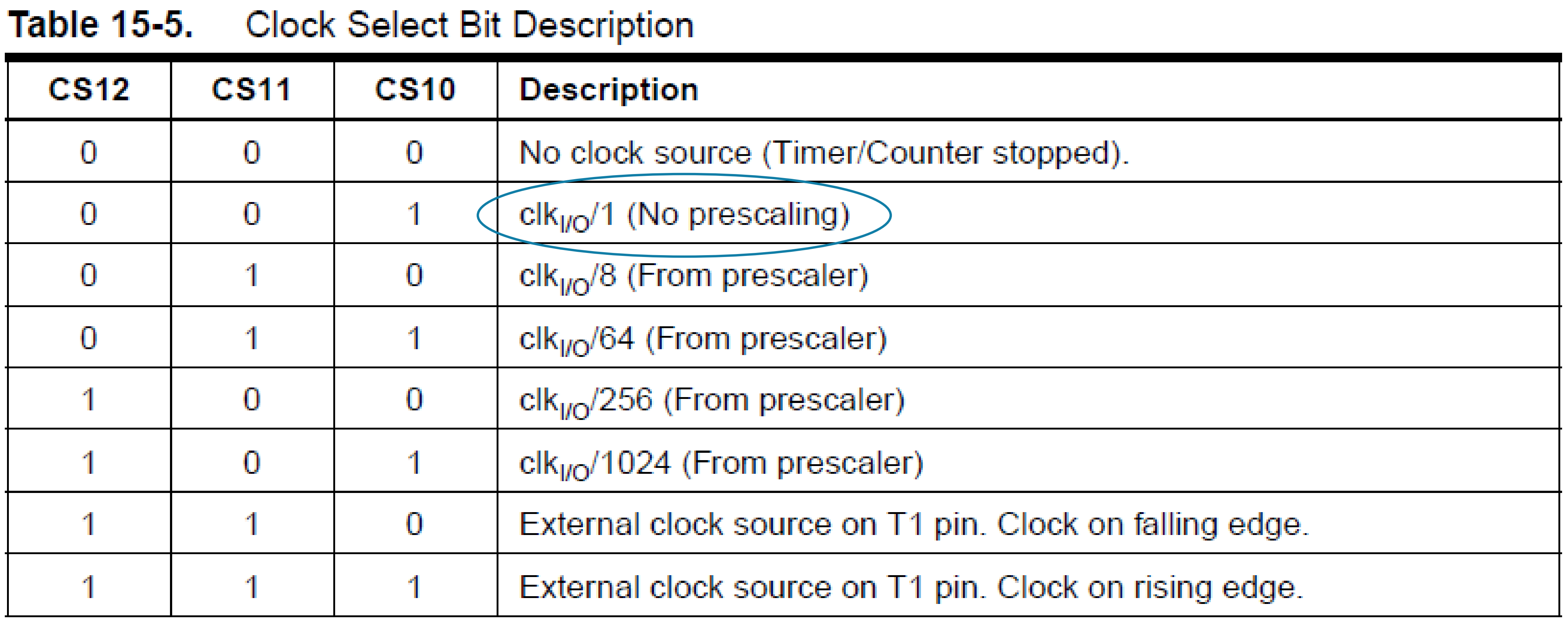


Table 15-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}/1$ (No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Control Registers

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
			1	0			1	1	

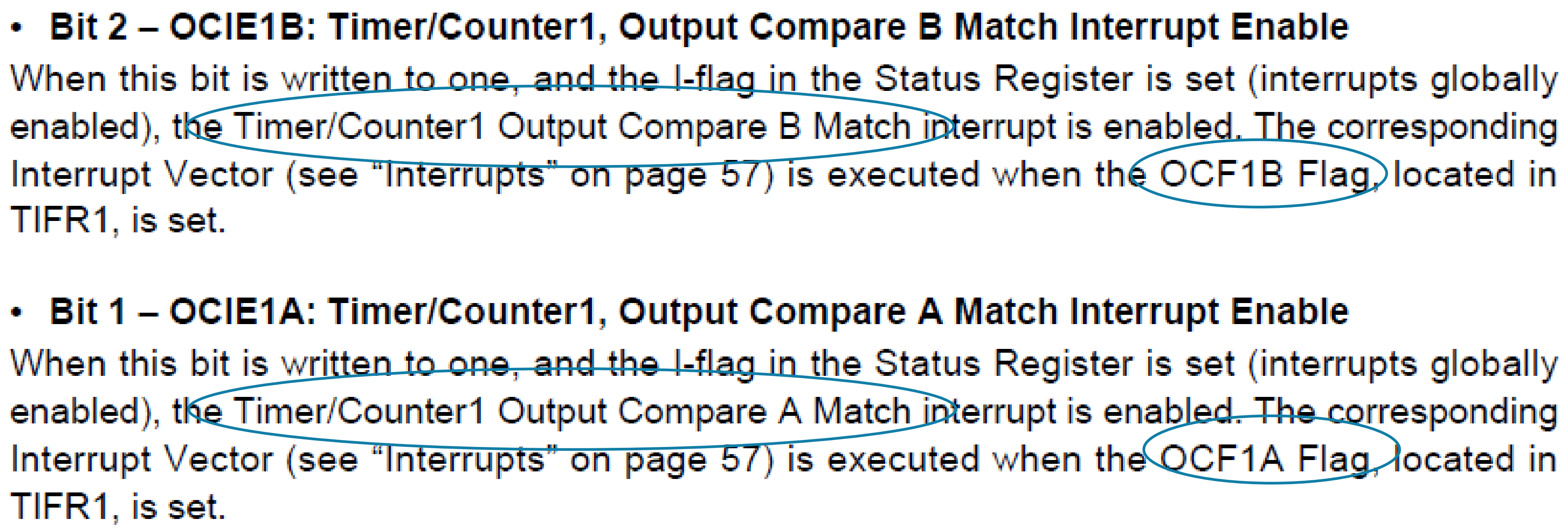
Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
				1	1	0	0	1	

Interrupts

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	**TIMSK1**
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	

	1	1
Allows one to play music!	Modulate duty cycle in ISR (change OCR1B)	Modulate frequency in ISR (change OCR1A)

Interrupt Vectors



- **Bit 2 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare B Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 57) is executed when the OCF1B Flag, located in TIFR1, is set.

- **Bit 1 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare A Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 57) is executed when the OCF1A Flag, located in TIFR1, is set.

Compare Match Flags

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	

- **Bit 2 – OCF1B: Timer/Counter1, Output Compare B Match Flag**

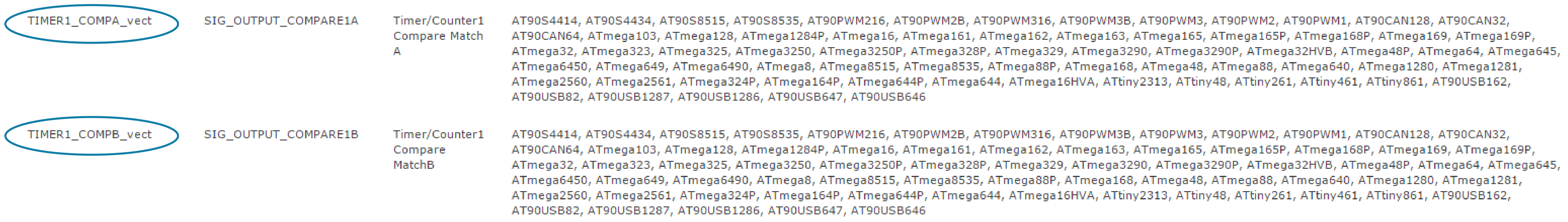
This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register B (OCR1B).

- **Bit 1 – OCF1A: Timer/Counter1, Output Compare A Match Flag**

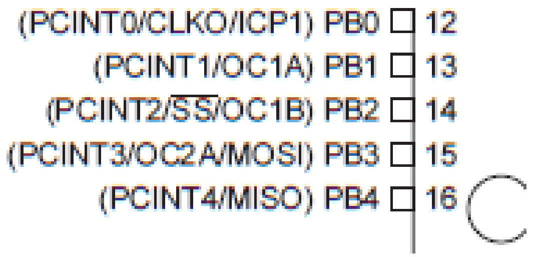
This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register A (OCR1A).

ISRs

http://www.atmel.com/webdoc/AVRLibcReferenceManual/group__avr__interrupts.html



TIMER1_COMPA_vect	SIG_OUTPUT_COMPARE1A	Timer/Counter1 Compare Match A	AT90S4414, AT90S4434, AT90S8515, AT90S8535, AT90PWM216, AT90PWM2B, AT90PWM316, AT90PWM3B, AT90PWM3, AT90PWM2, AT90PWM1, AT90CAN128, AT90CAN32, AT90CAN64, ATmega103, ATmega128, ATmega1284P, ATmega16, ATmega161, ATmega162, ATmega163, ATmega165, ATmega165P, ATmega168P, ATmega169, ATmega169P, ATmega32, ATmega323, ATmega325, ATmega3250, ATmega3250P, ATmega328P, ATmega329, ATmega3290, ATmega3290P, ATmega32HVB, ATmega48P, ATmega64, ATmega645, ATmega6450, ATmega649, ATmega6490, ATmega8, ATmega8515, ATmega8535, ATmega88P, ATmega168, ATmega48, ATmega88, ATmega640, ATmega1280, ATmega1281, ATmega2560, ATmega2561, ATmega324P, ATmega164P, ATmega644P, ATmega644, ATmega16HVA, ATtiny2313, ATtiny48, ATtiny261, ATtiny461, ATtiny861, AT90USB162, AT90USB82, AT90USB1287, AT90USB1286, AT90USB647, AT90USB646
TIMER1_COMPB_vect	SIG_OUTPUT_COMPARE1B	Timer/Counter1 Compare MatchB	AT90S4414, AT90S4434, AT90S8515, AT90S8535, AT90PWM216, AT90PWM2B, AT90PWM316, AT90PWM3B, AT90PWM3, AT90PWM2, AT90PWM1, AT90CAN128, AT90CAN32, AT90CAN64, ATmega103, ATmega128, ATmega1284P, ATmega16, ATmega161, ATmega162, ATmega163, ATmega165, ATmega165P, ATmega168P, ATmega169, ATmega169P, ATmega32, ATmega323, ATmega325, ATmega3250, ATmega3250P, ATmega328P, ATmega329, ATmega3290, ATmega3290P, ATmega32HVB, ATmega48P, ATmega64, ATmega645, ATmega6450, ATmega649, ATmega6490, ATmega8, ATmega8515, ATmega8535, ATmega88P, ATmega168, ATmega48, ATmega88, ATmega640, ATmega1280, ATmega1281, ATmega2560, ATmega2561, ATmega324P, ATmega164P, ATmega644P, ATmega644, ATmega16HVA, ATtiny2313, ATtiny48, ATtiny261, ATtiny461, ATtiny861, AT90USB162, AT90USB82, AT90USB1287, AT90USB1286, AT90USB647, AT90USB646

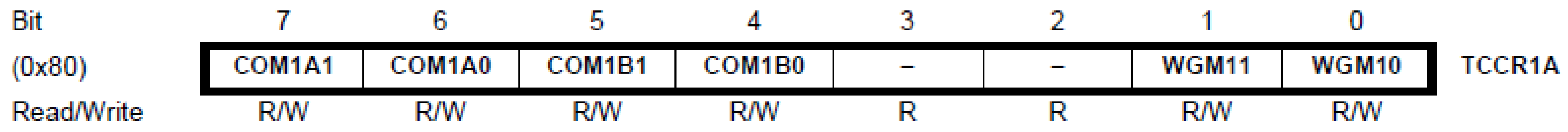


Suppose your buzzer is not connected to PB2, but to some other pin say PB4:

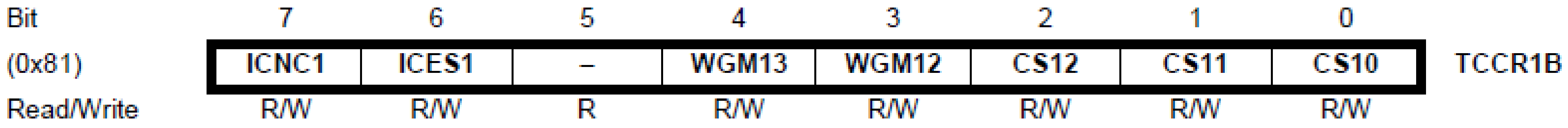
- Toggle PB4 in ISR(TIMER1_COMPA_vect)
- Toggle PB4 in ISR(TIMER1_COMPB_vect)

In addition update frequency and duty cycle in the ISRs by updating OCR1A and OCR1B

Easier Solution for 50% Duty Cycle



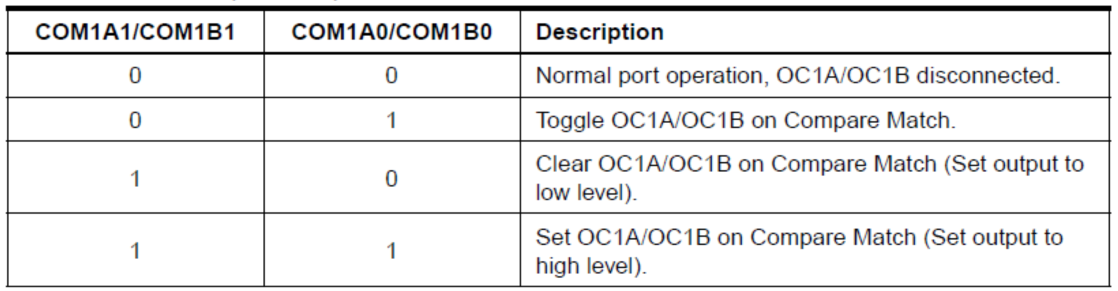
Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
	0	**1**					**0**	**0**	



Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
				0	**1**	**0**	**0**	**1**	

- Mode 4 in Table 15.4: CTC Mode for Waveform Generation
- No prescalar
- Toggle OC1A on Compare Match (see previous lecture on generating waveform using Timer 2)

Table 15-1. Compare Output Mode, non-PWM



COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on Compare Match.
1	0	Clear OC1A/OC1B on Compare Match (Set output to low level).
1	1	Set OC1A/OC1B on Compare Match (Set output to high level).

Updating Frequency

- When changing frequencies in ISR(TIMER1_COMPA_vect) by modifying OCR1A:
- Be careful not to modify OCR1A to a value <TCNT1
- Otherwise, TCNT1 cycles through to $2^{16} - 1$
- This may create a glitch!

Example Calculations

- Assume a clock frequency of F_CPU=20 MHz.
- In non-inverting fast PWM mode, configure a PWM output signal on pin OC1B so that it is active high for 1/3 of a 24ms period:
- Which prescalar for the timer do you use (pick one of 8, 64, or 256; it does not matter which one)? Given the prescalar of your choice, at what values should you set OCRnA and OCRnB?

Example Calculations

- Solution: No prescaling does not work.

- For a prescalar of 8 we obtain 8/20MHz = 0.4 micro seconds / TCNT1 tick. So, 24ms = 24ms/0.4us TCNT1 ticks, i.e., 60,000 TCNT1 ticks. By setting OCR1A = 59999 we get a PWM of period 24 ms. By setting OCR1B = 19999 we get a duty cycle of (OCR1B+1)/(OCR1A+1) = 1/3.

- For a prescalar of 64 we obtain 64/20MHz = 3.2 micro seconds / TCNT1 tick. So, 24ms = 24ms/3.2us TCNT1 ticks, i.e., 7,500 TCNT1 ticks. By setting OCR1A = 7499 we get a PWM of period 24 ms. By setting OCR1B = 2499 we get a duty cycle of (OCR1B+1)/(OCR1A+1) = 1/3.

- For a prescalar of 256 we obtain 256/20MHz = 12.8 micro seconds / TCNT1 tick. So, 24ms = 24ms/12.8us TCNT1 ticks, i.e., 1,875 TCNT1 ticks. By setting OCR1A = 1874 we get a PWM of period 24 ms. By setting OCR1B = 624 we get a duty cycle of (OCR1B+1)/(OCR1A+1) = 1/3.

- For a prescalar of 1024 we obtain 256/20MHz = 51.2 micro seconds / TCNT1 tick. So, 24ms = 24ms/51.2us TCNT1 ticks, i.e., 468.75 TCNT1 ticks. By setting OCR1A = 468 we get a PWM of period ~24 ms (but not exact). By setting OCR1B = 155 we get a duty cycle of (OCR1B+1)/(OCR1A+1) approximately 1/3.

- So, only the prescalars 8, 64, or 256 give exact results.

Example Calculations

- Besides registers OCRnA and OCRnB which other registers *need* to be set in order to have pin 18 output the PWM signal (do not add more register names than necessary)? Do not worry about the values for these registers, you only need to write down the names of the registers.

- Solution (with values, just to be complete):
  - DDRB = 0x04; // pin PB2 = OCB1 is an output
  - TCCR1A = (1<<COM1B1)|(1<<WGM11)|(1<<WGM10);
    // non-inverting, fast PWM with TOP = OCR1A
  - TCCR1B = (1<<WGM13)|(1<<WGM12)|(1<<CS11);
    // fast PWM with TOP = OCR1A, prescalar = 8