

Lecture7: ATmega 2650 Datasheet and 16 bit timer

Vikram Iyer

9:30-11:30 OH (Deeksha) ECE 345 15 11:30-13:30 OH (Alex) ECE 345	13:00-15:00 OH (Zach) ECE 345 16	9:30-11:30 OH (Deeksha) ECE 345 17 12:00-13:00 Remote OH (Vikram) Zoom	10:00-12:00 OH (Zach) ECE 345 18 12:30-14:30 OH (Alex) ECE 345 23:59 Lab 1 due	12:30-14:20 Lecture MOR 230 19 <i>Lecture 7: ATmega 2560 Datasheet and Timers</i> 14:00-15:00 OH (Vikram) ECE 345
9:30-11:30 OH (Deeksha) ECE 345 22 11:30-13:30 OH (Alex) ECE 345	13:00-15:00 OH (Zach) ECE 345 23	9:30-11:30 OH (Deeksha) ECE 345 24 12:30-14:20 Lecture MOR 230 <i>Lecture 8: Reading Analog Data and Intro to Interrupts</i> 14:00-15:00 OH (Vikram) ECE 345	10:00-12:00 OH (Zach) ECE 345 25 12:30-14:30 OH (Alex) ECE 345 23:59 C Programming 2 due	12:30-14:20 Lecture MOR 230 26 <i>Lecture 9: Tasks, Threads, Scheduling I, Interrupts</i> 14:00-15:00 OH (Vikram) ECE 345
9:30-11:30 OH (Deeksha) ECE 345 29 11:30-13:30 OH (Alex) ECE 345	13:00-15:00 OH (Zach) ECE 345 30	9:30-11:30 OH (Deeksha) ECE 345 01 12:30-14:20 Lecture MOR 230 <i>Lecture 10: Scheduling II</i> 14:00-15:00 OH (Vikram) ECE 345	10:00-12:00 OH (Zach) ECE 345 02 12:30-14:30 OH (Alex) ECE 345 23:59 Lab 2 due	12:30-14:20 Lecture MOR 230 03 <i>Lecture 11: Scheduling III and Lab 3 Intro</i> 14:00-15:00 OH (Vikram) ECE 345

May

Monday	Tuesday	Wednesday	Thursday	Friday
9:30-11:30 OH (Deeksha) ECE 345 06 11:30-13:30 OH (Alex) ECE 345	13:00-15:00 OH (Zach) ECE 345 07	9:30-11:30 OH (Deeksha) ECE 345 08 12:30-14:20 Lecture MOR 230 <i>Lecture 12: Midterm Review</i> 14:00-15:00 OH (Vikram) ECE 345	10:00-12:00 OH (Zach) ECE 345 09 12:30-14:30 OH (Alex) ECE 345	12:30-14:00 Midterm 10

Assignments

- Programming assignment 2
 - Bit manipulation, more pointer practice, intro to structs
 - Tried to incorporate feedback from assignment 1 and fix bugs
 - Meant to be fairly straightforward, more practice in C

Structures (structs) in C

Structs are a way to group several related variables into one place. Each variable in the structure is known as a **member** of the structure. Unlike an array, a structure can contain many different data types (int, float, char, etc.).

```
struct MyStructure { // Structure declaration
    int myNum;       // Member (int variable)
    char myLetter;   // Member (char variable)
}; // End the structure with a semicolon
```

name (circled around the closing brace and semicolon)

```
s1.myNum = 13;
// Print values
printf("My number: %d\n", s1.myNum);
```

↳ (pointing to the first line)

Assignments

- Programming assignment 2
 - Bit manipulation, more pointer practice, intro to structs
 - Tried to incorporate feedback from assignment 1 and fix bugs
 - Meant to be fairly straightforward, more practice in C
- Lab 2
 - Longer than Lab 1, get started early
 - After today we'll have covered material for Part 1 and 2
 - Material will be in next two lectures



Last time

- MPU6050 demo
 - What is an IMU and how does it work?
 - Reading datasheets
 - Demo of reading and writing registers to control sensor

Plan for today

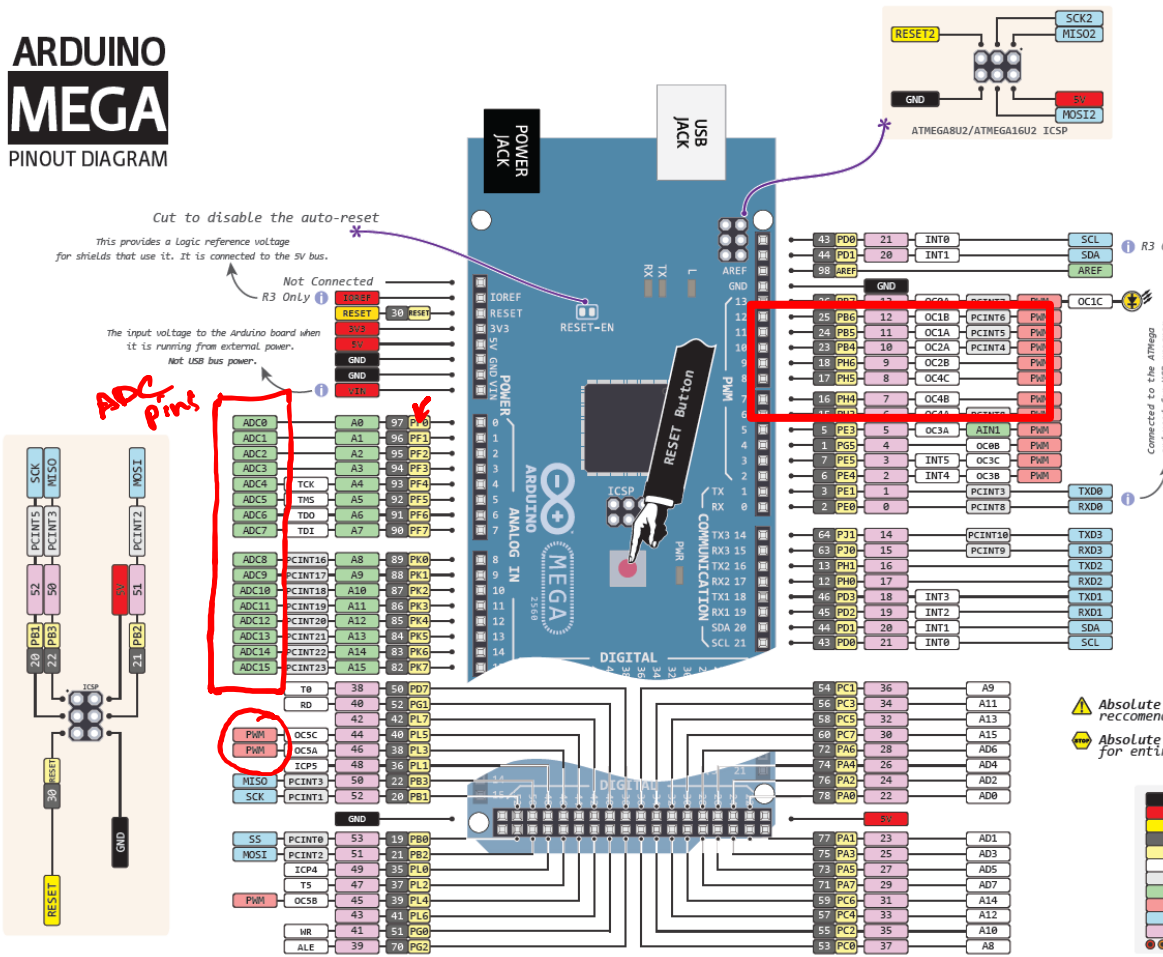
- Walk through ATmega datasheet
 - Point out where to look for documentation in lab 2
- Details of 16 bit timer counter

Plan for today

- Walk through ATmega datasheet
 - Point out where to look for documentation in lab 2
- Details of 16 bit timer counter

ATMega2650 Datasheet + Lab resources

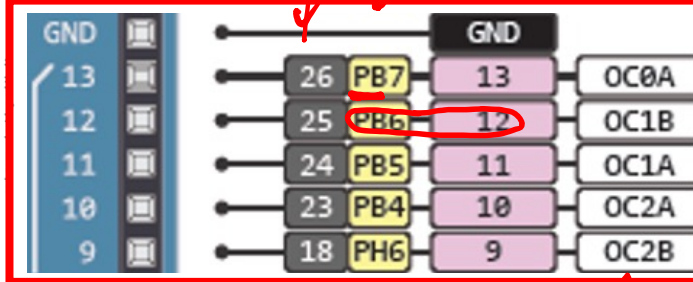
ARDUINO MEGA PINOUT DIAGRAM



Labels on arduino

pins on Atmega 2560

port B labels on arduino



⚠ Absolute max per pin 40mA recommended 20mA

⚡ Absolute max 200mA for entire package

Legend:

- Black: GND
- Red: Power
- Blue: Control
- Yellow: Physical Pin
- Light Blue: Port Pin
- Light Green: Pin Function
- Light Purple: Digital Pin
- Light Orange: Analog Related Pin
- Light Pink: PWM Pin
- Light Cyan: Serial Pin
- Light Blue-Grey: IDE
- Light Green-Cyan: Source Total 150mA

16-bit Timer/Counter

Block Diagram

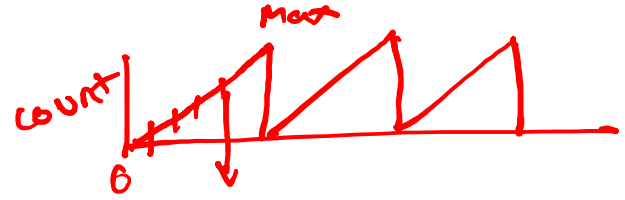
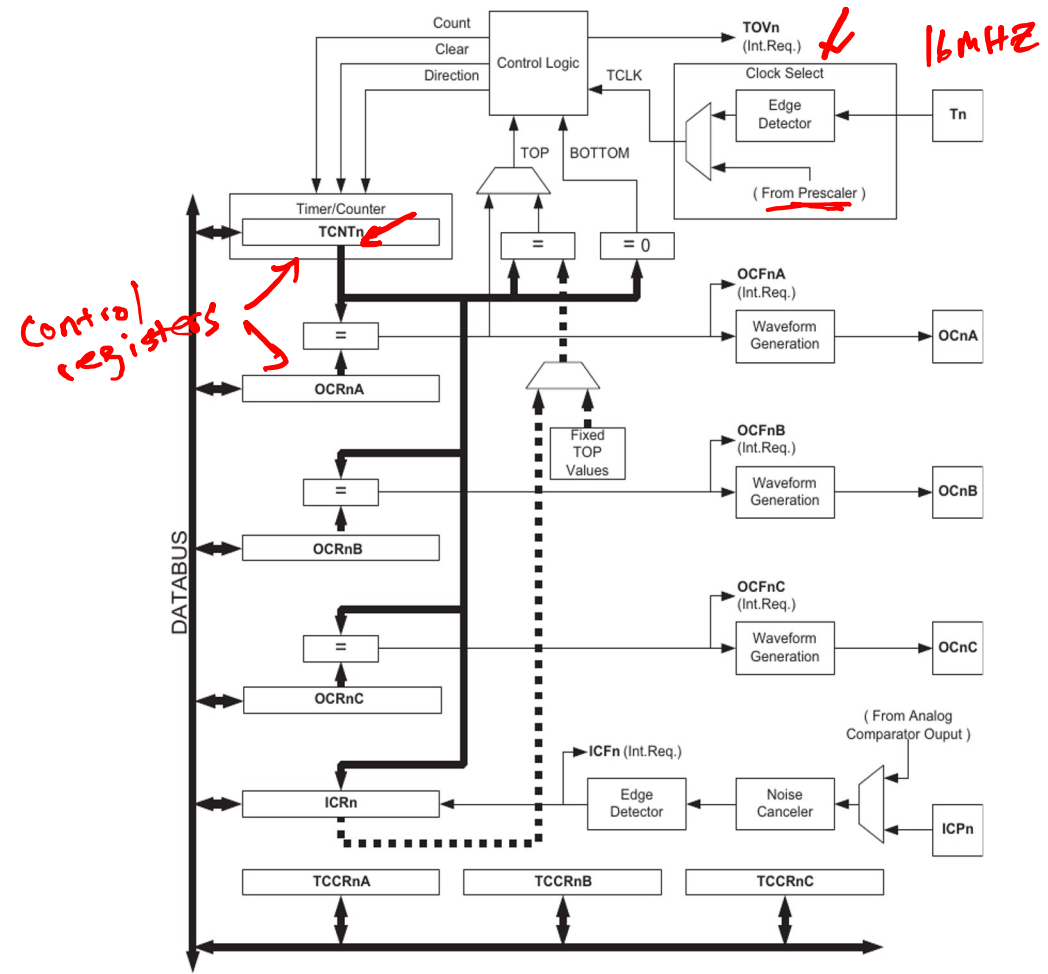
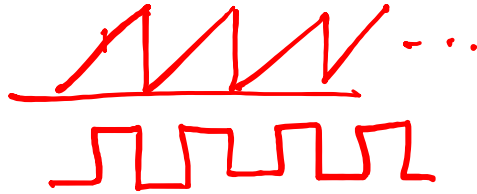


Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾



16-bit Timer/Counter

Block Diagram



Key terms:

“TOP” -- user programable

“MAX” -- 0xFFFF (65535)

“BOTTOM” -- 0x0000

Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾

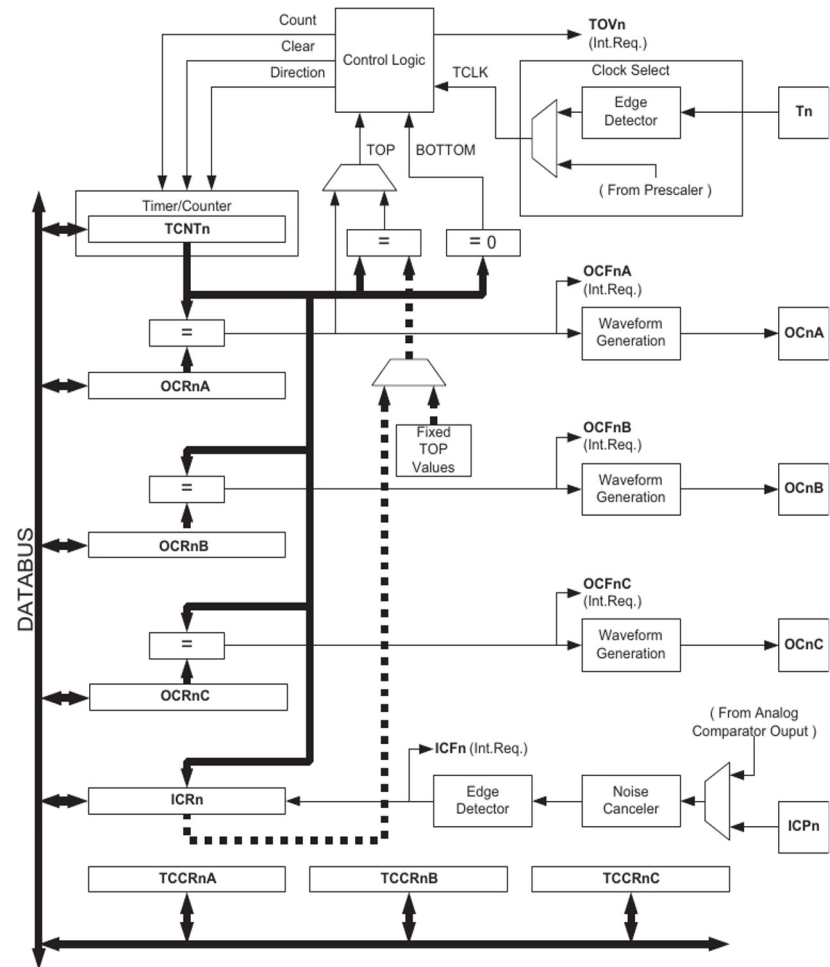


Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾

16-bit Timer/Counter

Control Registers

Configure all modes.

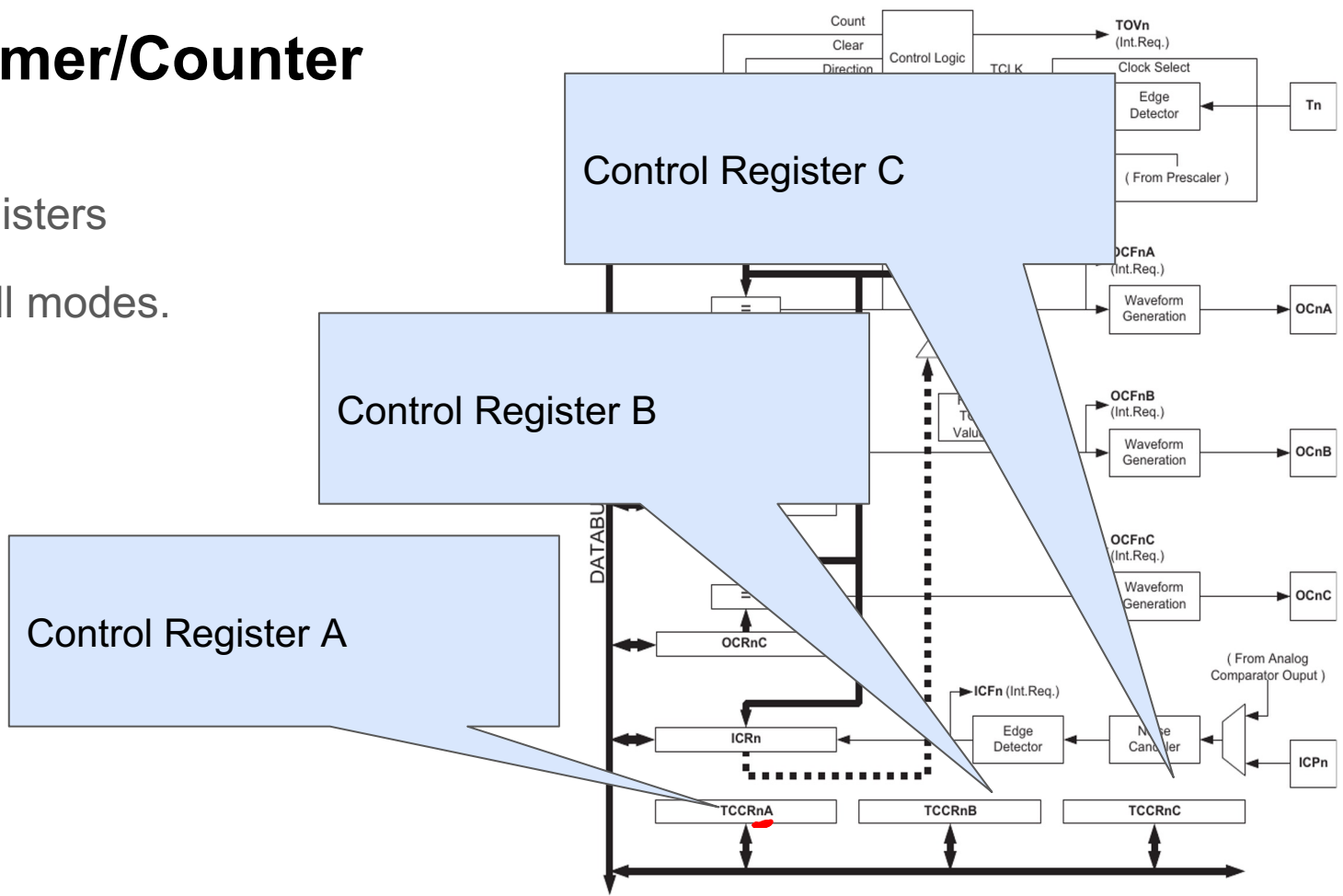
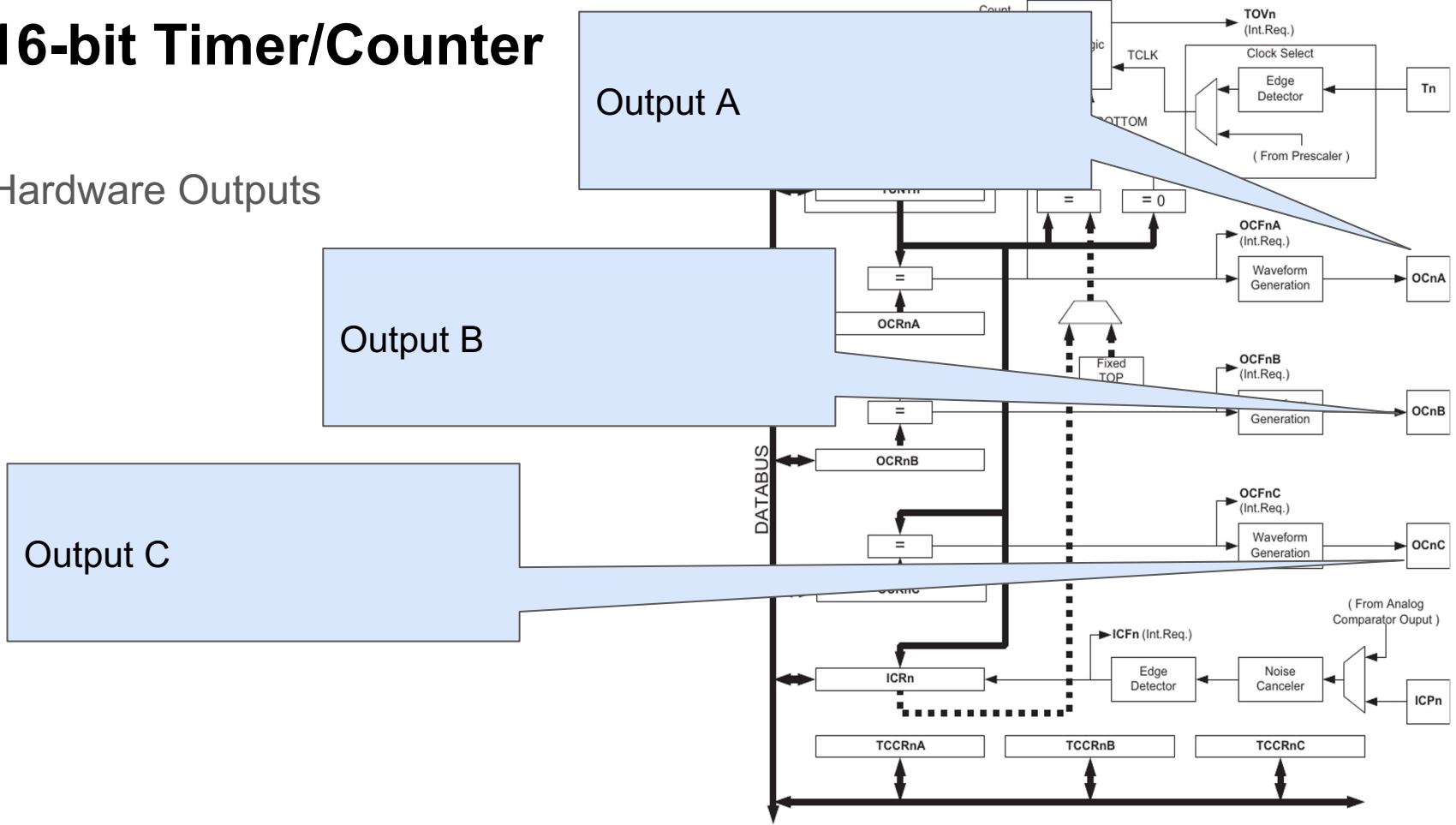


Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾

16-bit Timer/Counter

Hardware Outputs



16-bit Timer/Counter

Combinatorial circuit which sets or clears the output pin.

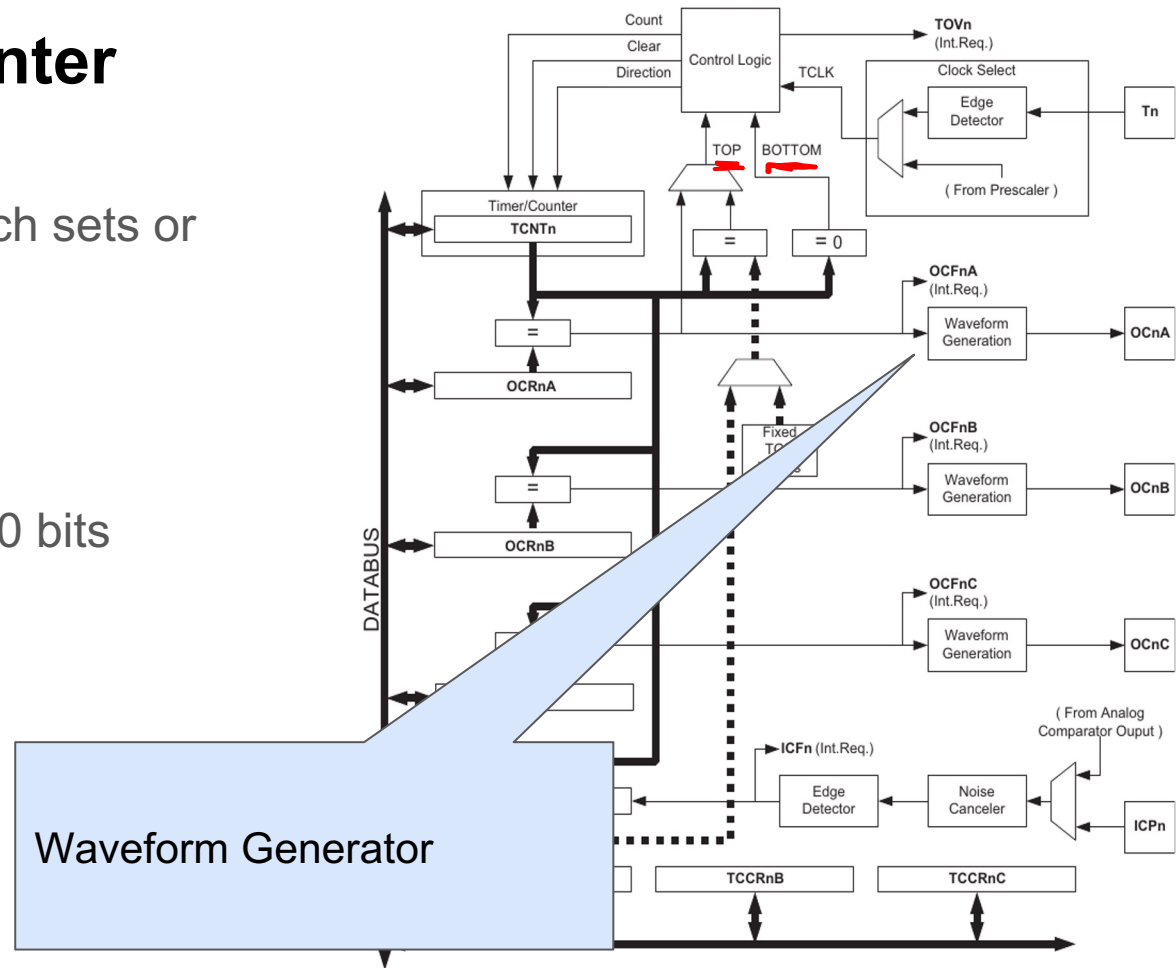
Inputs:

- Match
- WGMn3:0 COMnx1:0 bits
- TOP, BOTTOM

Actions:

- Set, Clear, Toggle

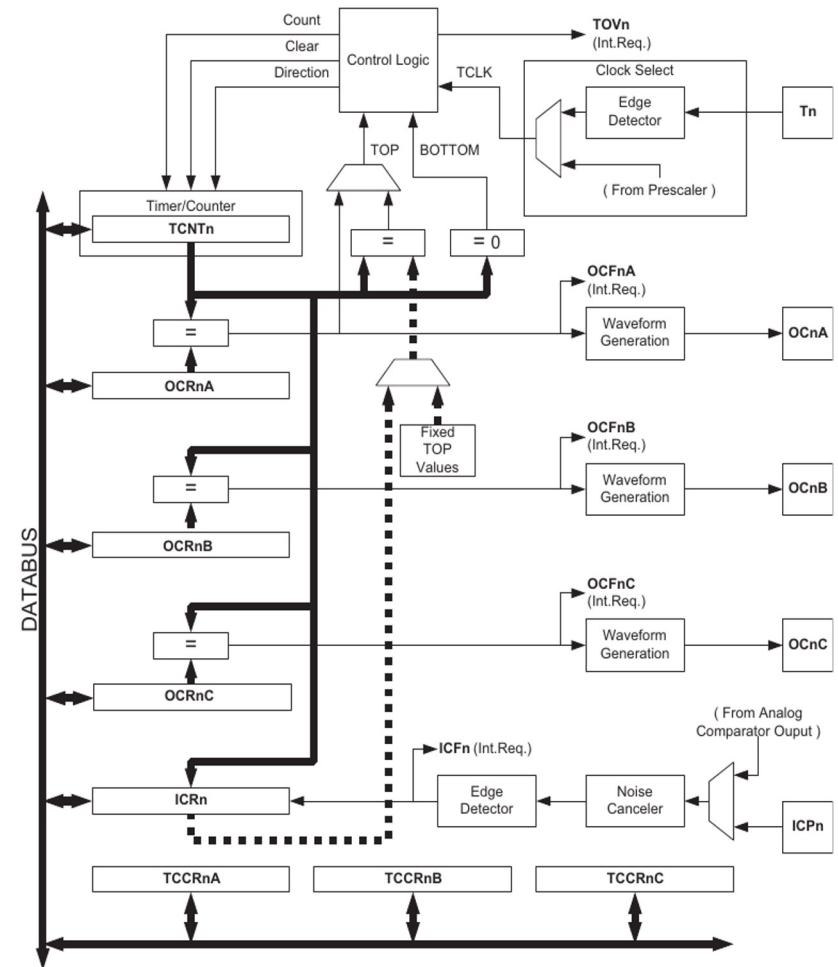
Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾



16-bit Timer/Counter

Block Diagram

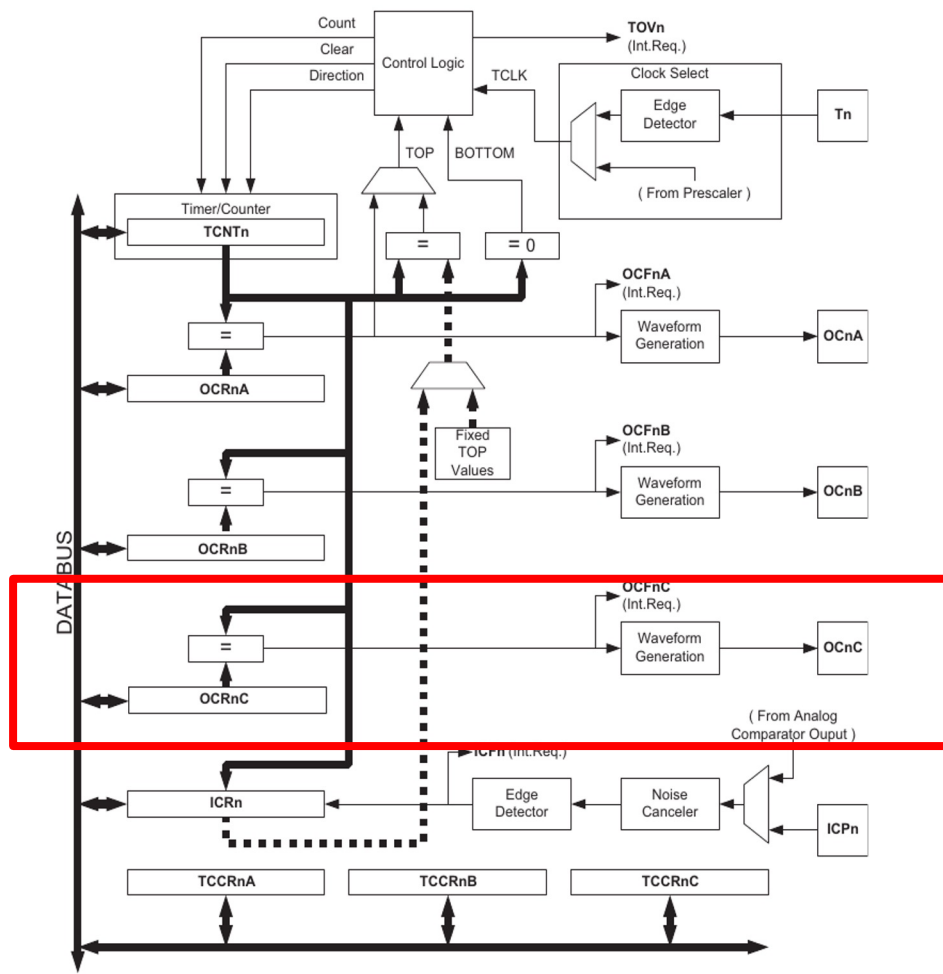
Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾



16-bit Timer/Counter

Block Diagram

Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾

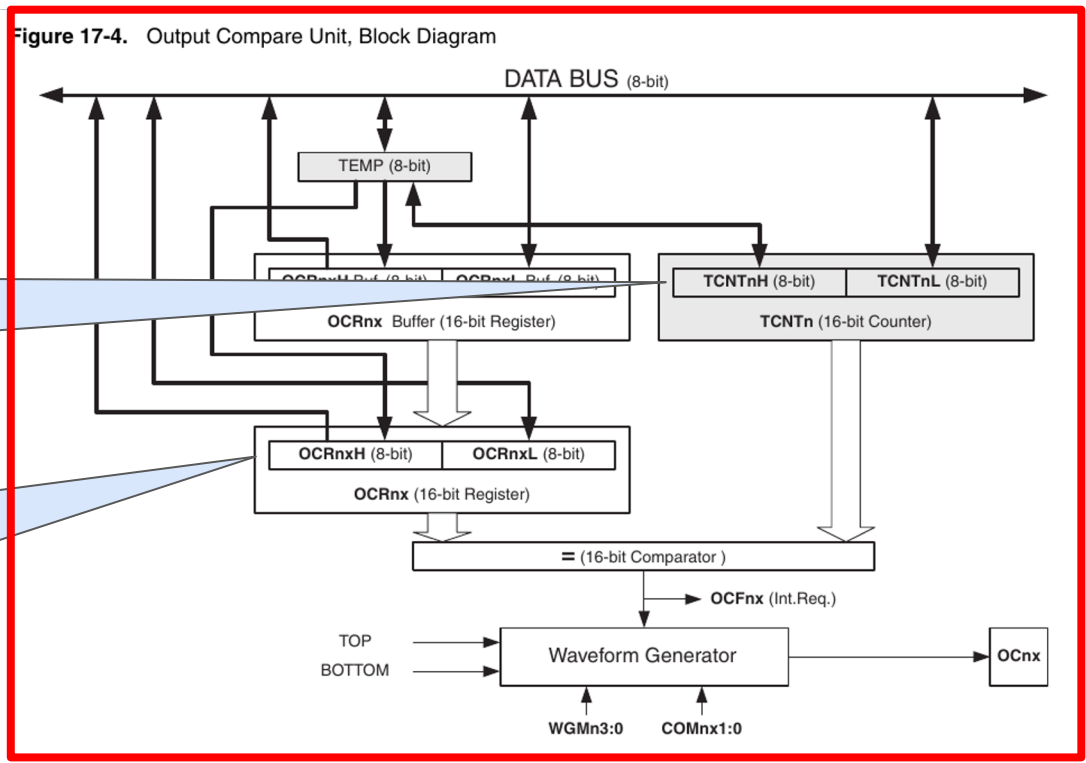


Output Compare Unit

Expanded Block Diagram

Current Counter Value

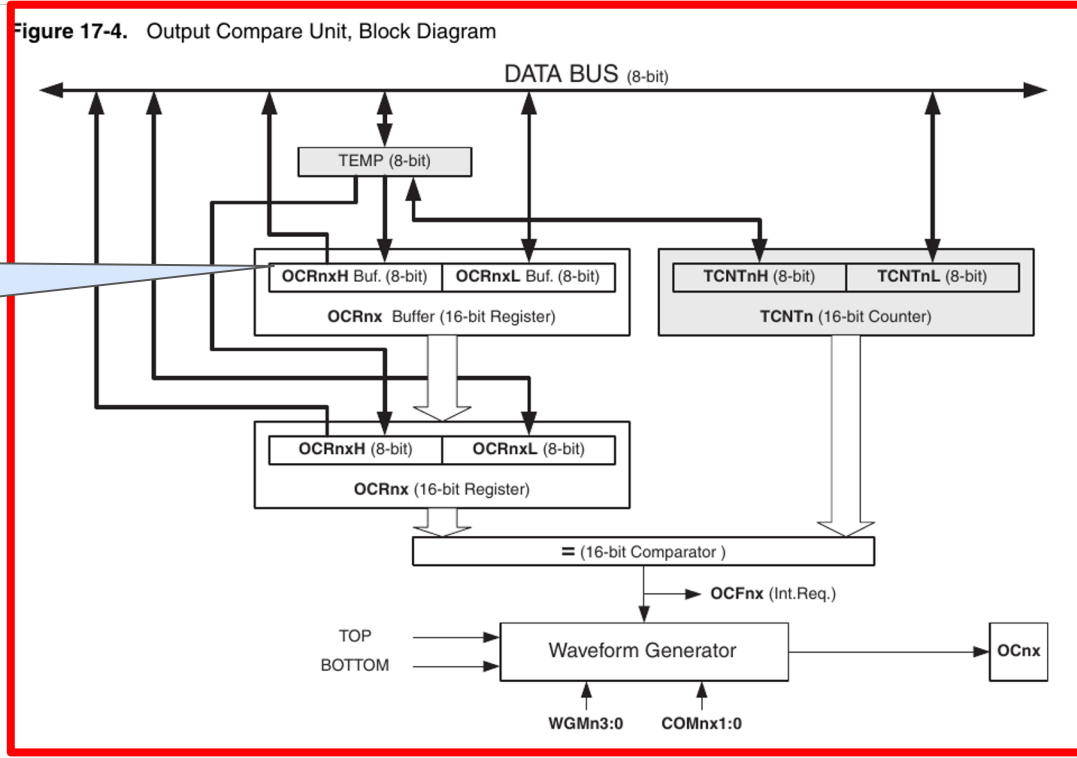
Compare Value



Output Compare Unit

Expanded Block Diagram

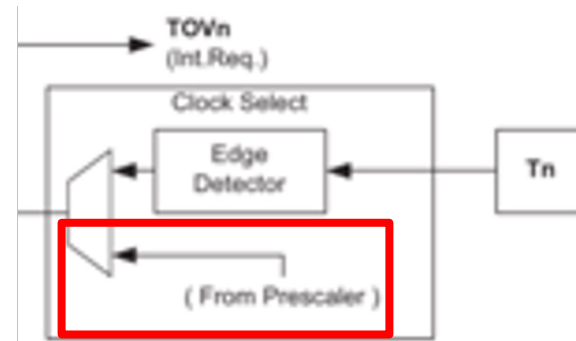
Local Counter Reset Value



Clock and Pre-scaler

Internal clock = 16MHz (Period = 62.5ns)

- Can also count external events
- Can divide the clock by 1, 8, 64, 256, or 1024



	A	B	C	D	E	F	G	H	I
1				16-bit			8-bit		
2		Freq (Hz)	Period (msec)	Max Time (sec)	Min Freq (Hz)	Max Freq (Hz)	Max Time (sec)	Min Freq (Hz)	Max Freq (Hz)
3	prescale	16,000,000	0.0000625	0.0041	488.3	8,000,000.00	0.00002	125000.0	8,000,000.00
4	8	2,000,000	0.0005	0.0328	61.0	1,000,000.00	0.00013	15625.0	1,000,000.00
5	64	250,000	0.004	0.2621	7.6	125,000.00	0.00102	1953.1	125,000.00
6	256	62,500	0.016	1.0486	1.9	31,250.00	0.00410	488.3	31,250.00
7	1024	15,625	0.064	4.1943	0.5	7,812.50	0.01638	122.1	7,812.50
8									

Assuming Toggle Mode

16-bit Timer MODES

Table 17-2. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnx at	TOVn 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	OCRnA	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	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

For detailed timing information refer to “Timer/Counter Timing Diagrams” on page 152.

16-bit Timer MODES

“Normal”

Count up to
0xFFFF (MAX)

Table 17-2. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnX at	TOVn 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	OCRnA	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	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

For detailed timing information refer to “Timer/Counter Timing Diagrams” on page 152.

16-bit Timer MODES

CTC (Clear Timer on Compare Match)

Table 17-2. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnx at	TOVn 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	OCRnA	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	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

For detailed timing information refer to [“Timer/Counter Timing Diagrams” on page 152.](#)

16-bit Timer MODES

CTC (Clear Timer on Compare Match)

Count up to
OCRnA (user
value)


1 0 
Set, Clear, or
Toggle Output on
match (COMnA1:0)

Table 17-2. Waveform Generation Mode Bit Description⁽¹⁾

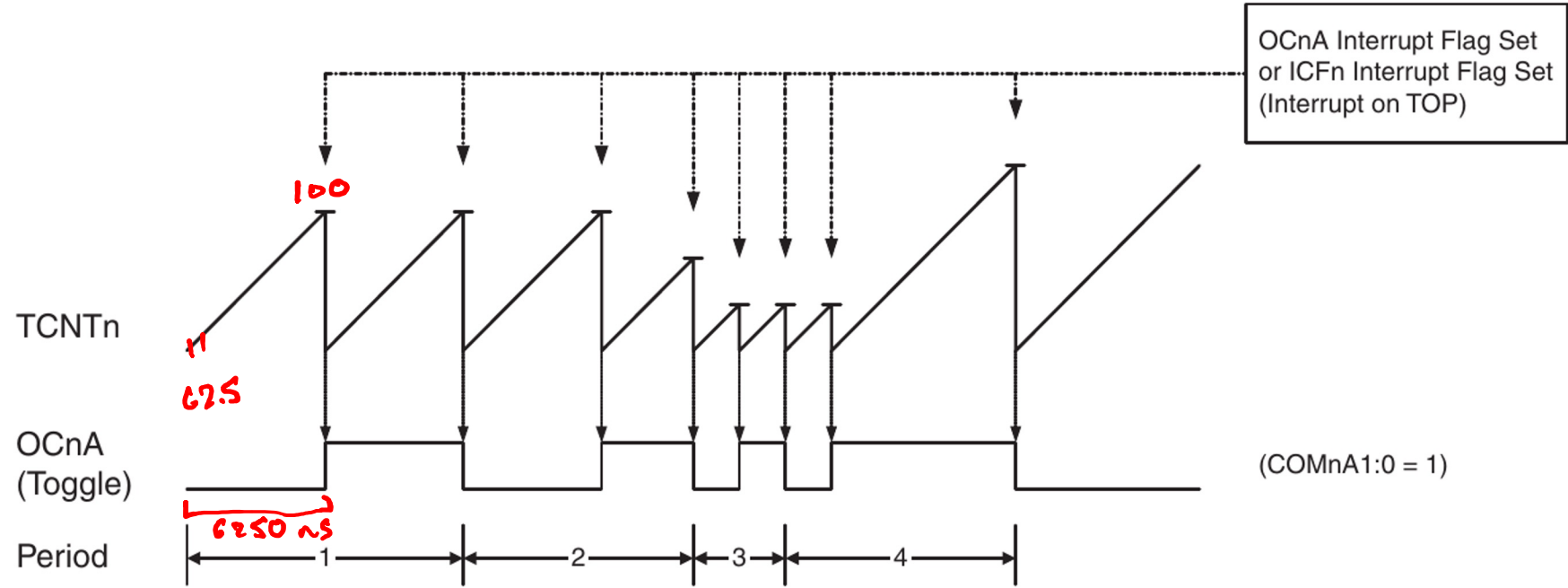
Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnx at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	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	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functional and location of these bits are compatible with previous versions of the timer.

For detailed timing information refer to “Timer/Counter Timing Diagrams” on page 152.

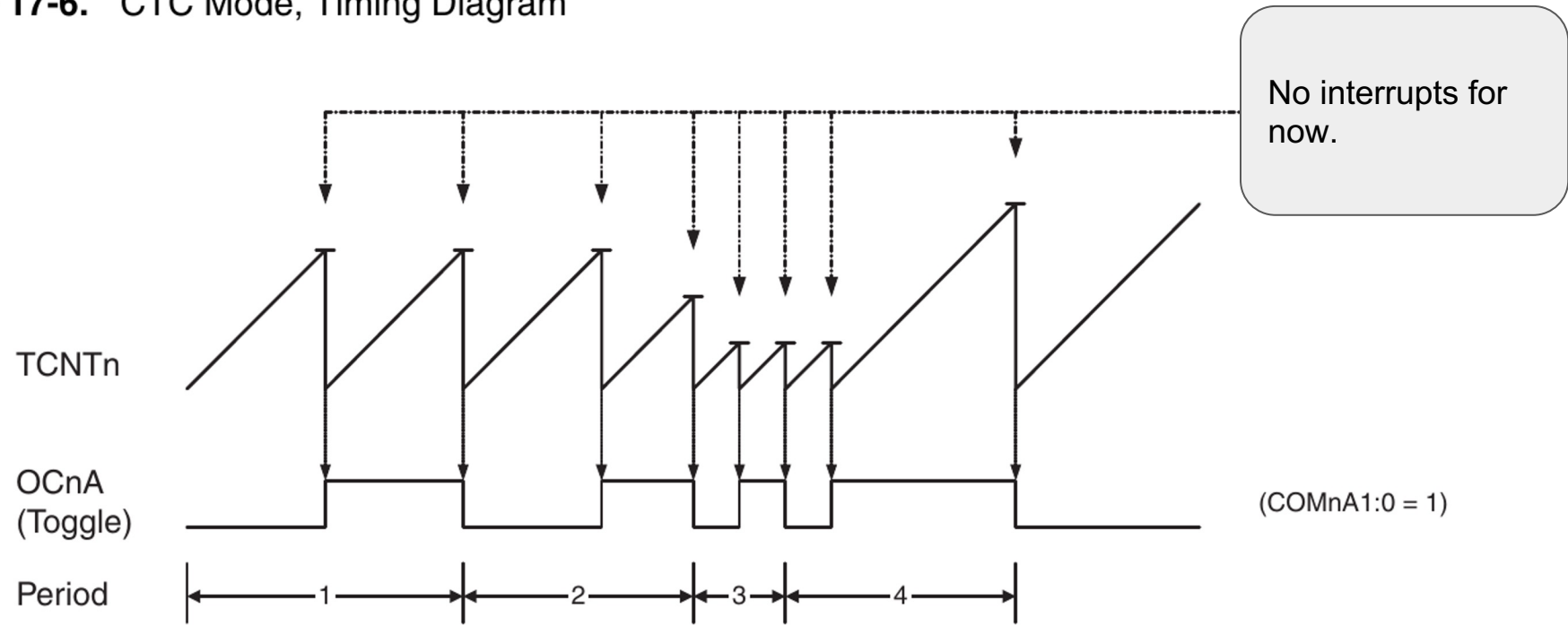
CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



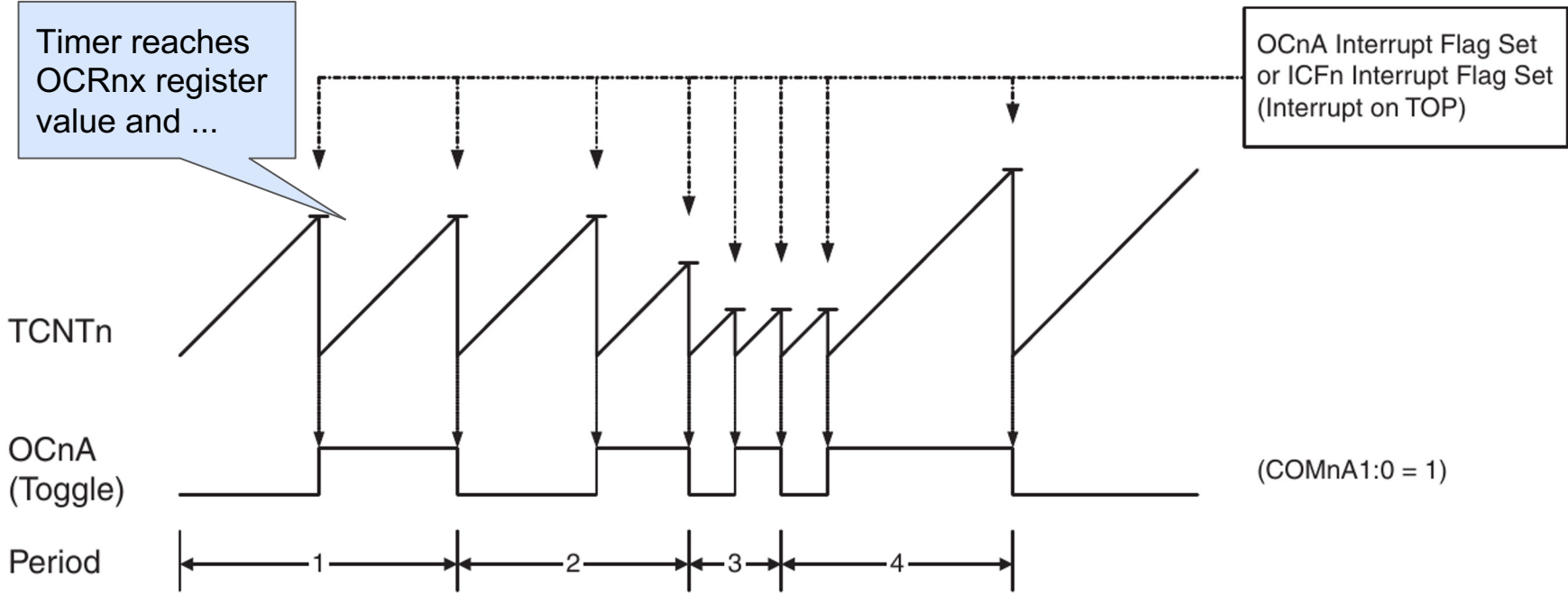
CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



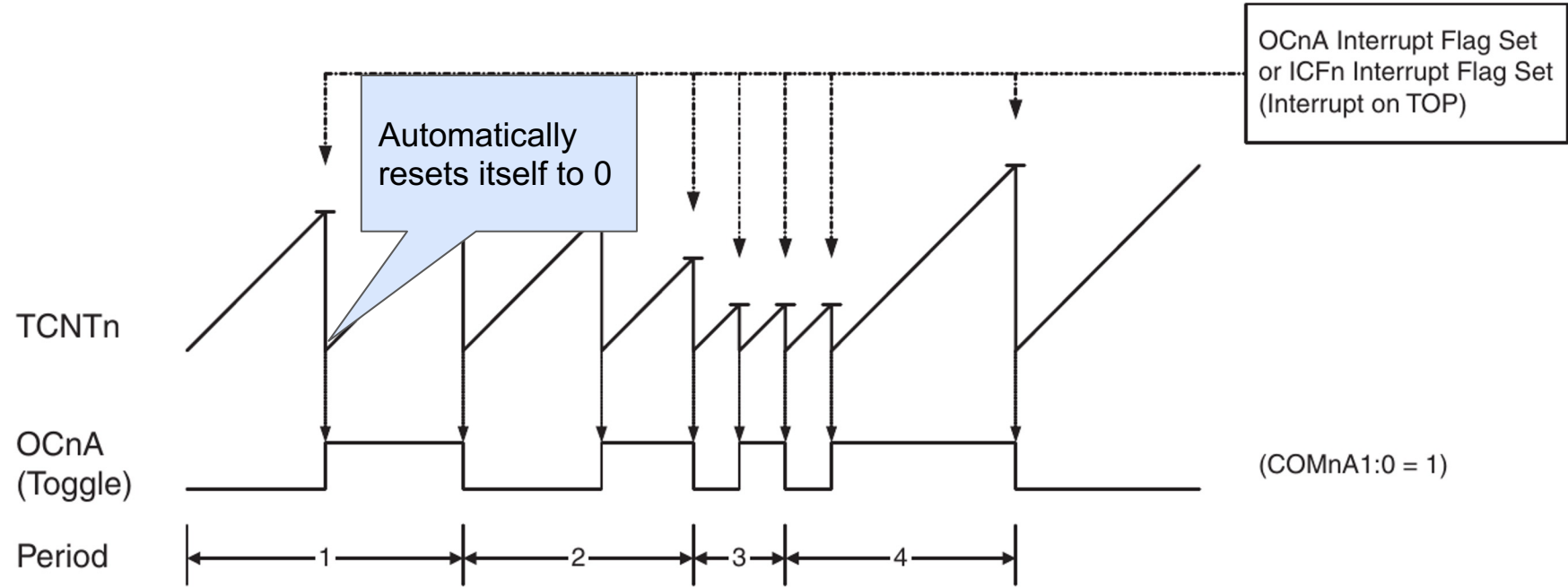
CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



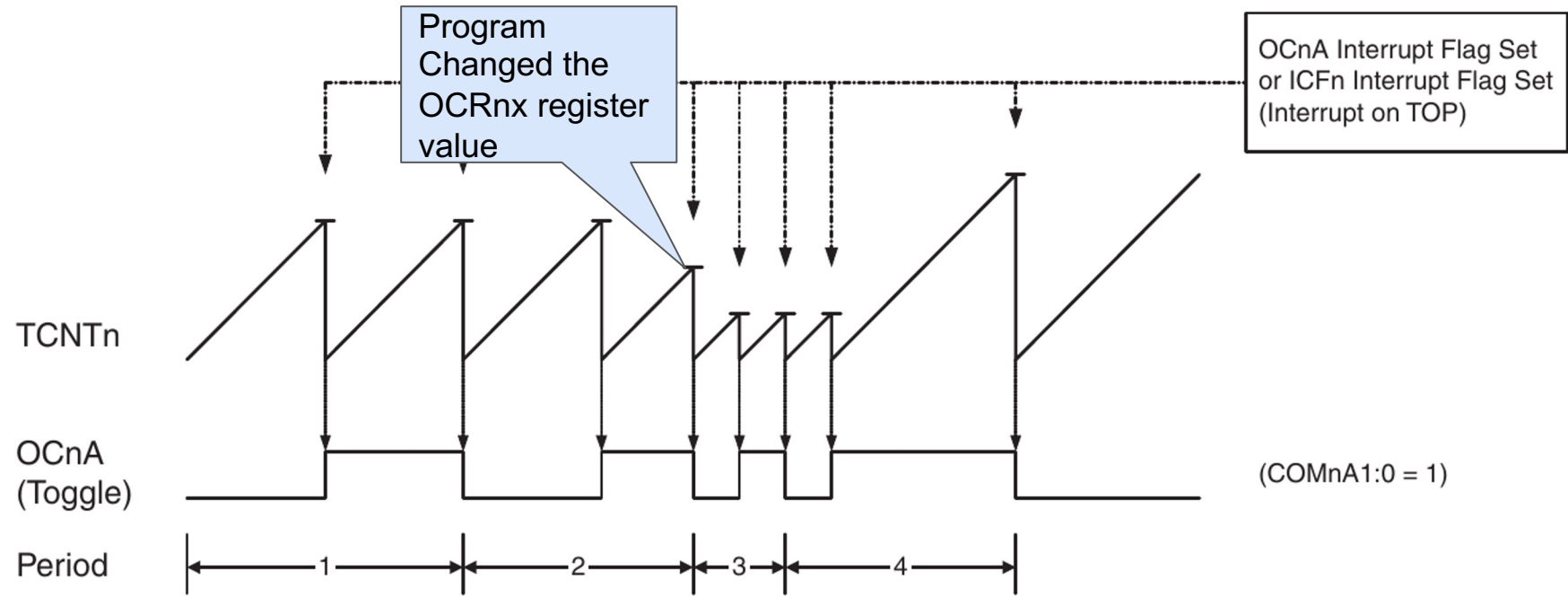
CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



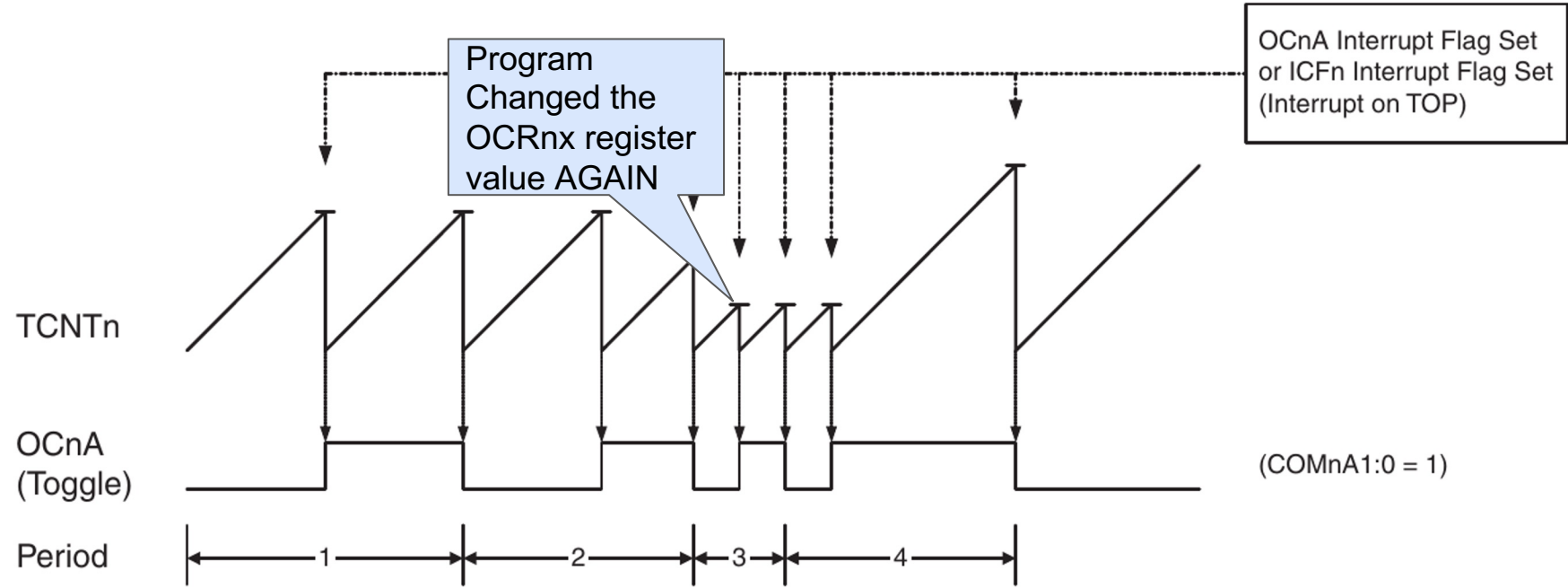
CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



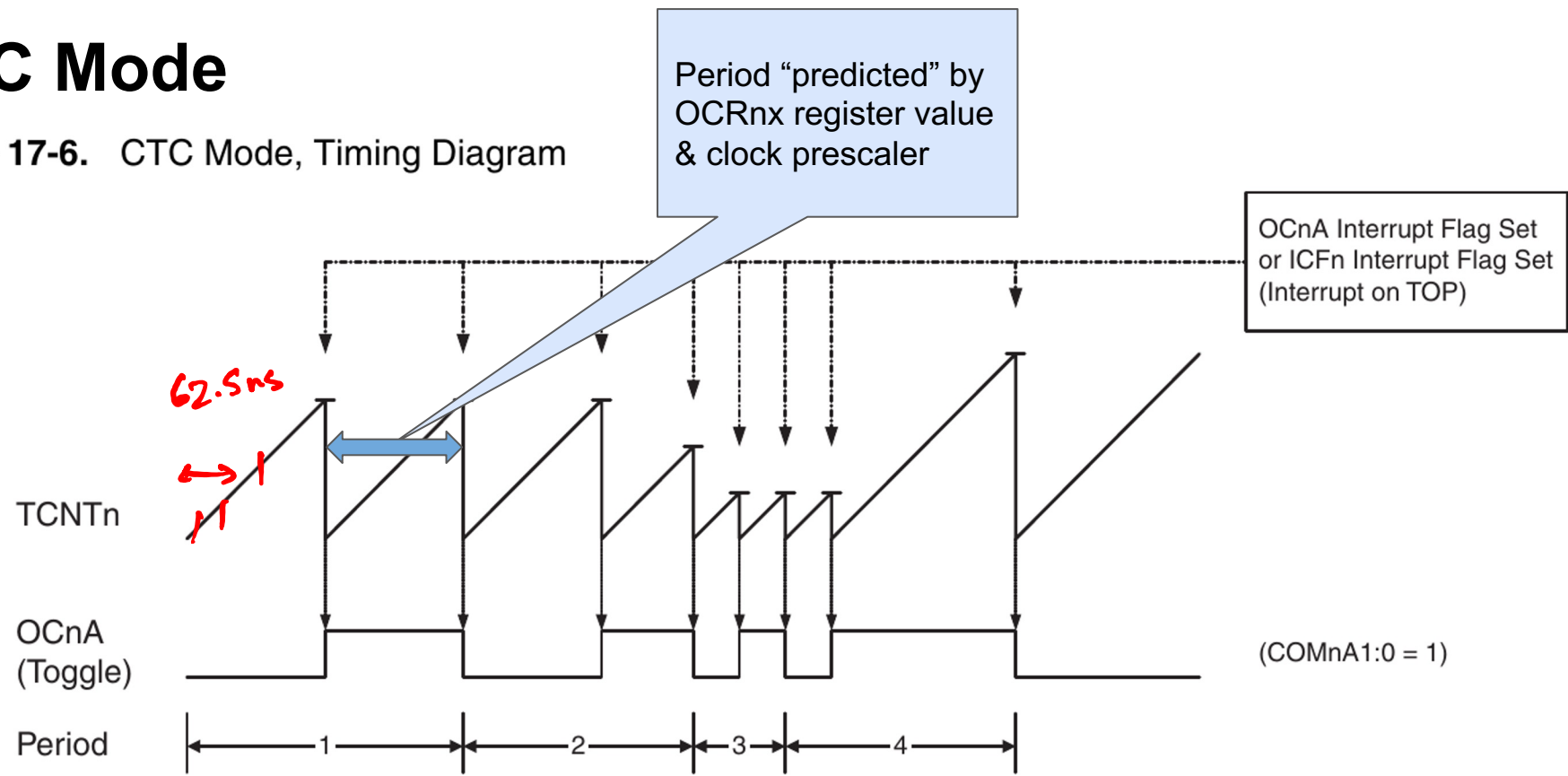
CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



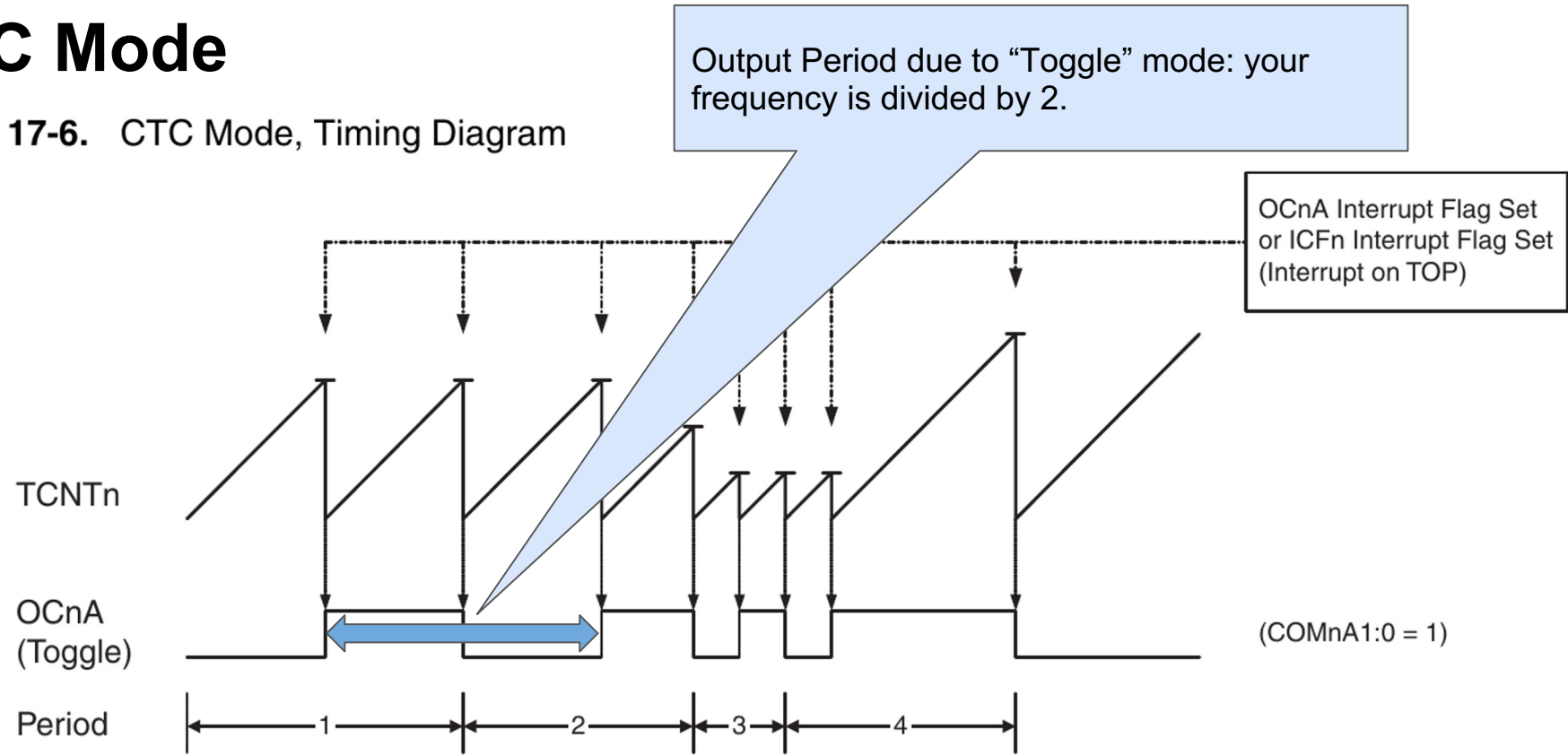
CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



CTC Mode

Figure 17-6. CTC Mode, Timing Diagram



16-bit Timer MODES

“Fast PWM”

Count up to ICRn/OCRnA (MAX), reset to 0x000

Clear output at compare match

Table 17-2. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnx at	TOVn 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	OCRnA	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	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

For detailed timing information refer to “Timer/Counter Timing Diagrams” on page 152.

Duty Cycle (PWM) Control

- PWM = Pulse Width Modulation
- Switch on and off “quickly”
- “Quickly” means faster than process can react.
- Effective output is time average of on-off signal.
- For this application replace “5V” with power for any device.

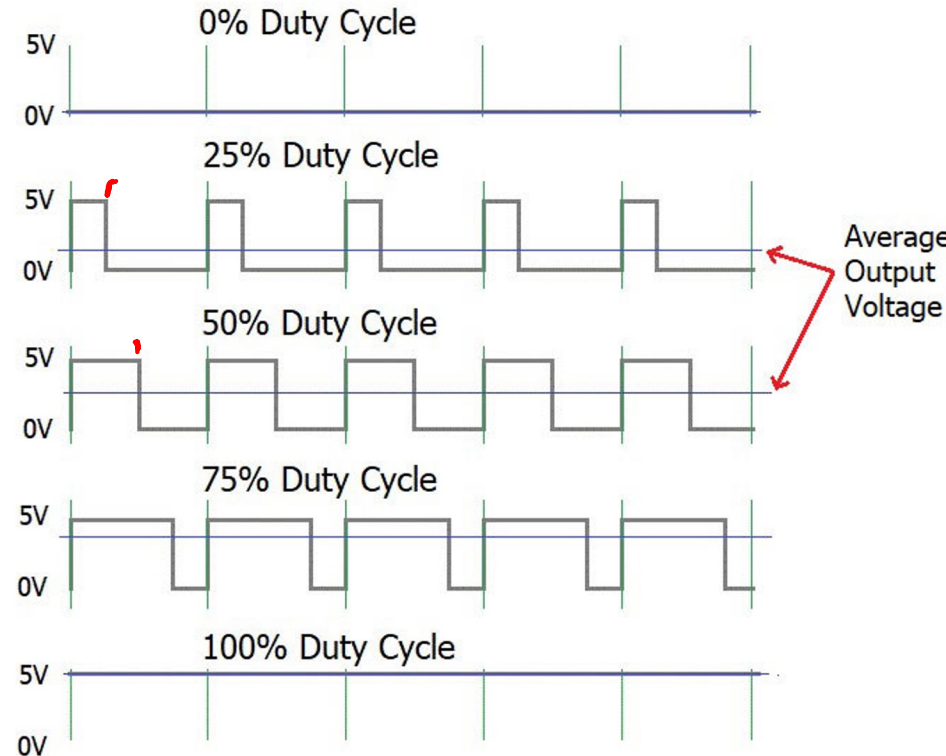
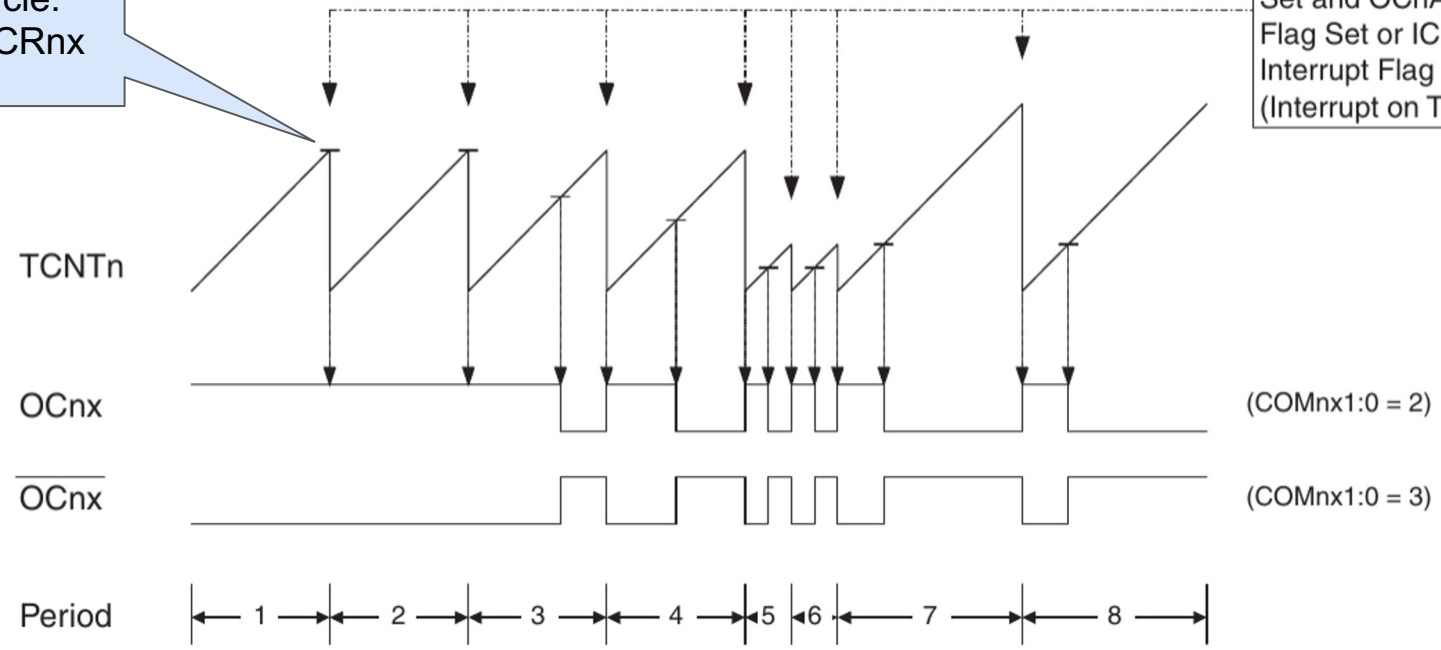


Figure 17-7. Fast PWM Mode, Timing Diagram

100% duty cycle:
TCNTn == OCRnx

OCRnx / TOP Update
and TOVn Interrupt Flag
Set and OCnA Interrupt
Flag Set or ICFn
Interrupt Flag Set
(Interrupt on TOP)



(COMnx1:0 = 2)

(COMnx1:0 = 3)

Figure 17-7. Fast PWM Mode, Timing Diagram

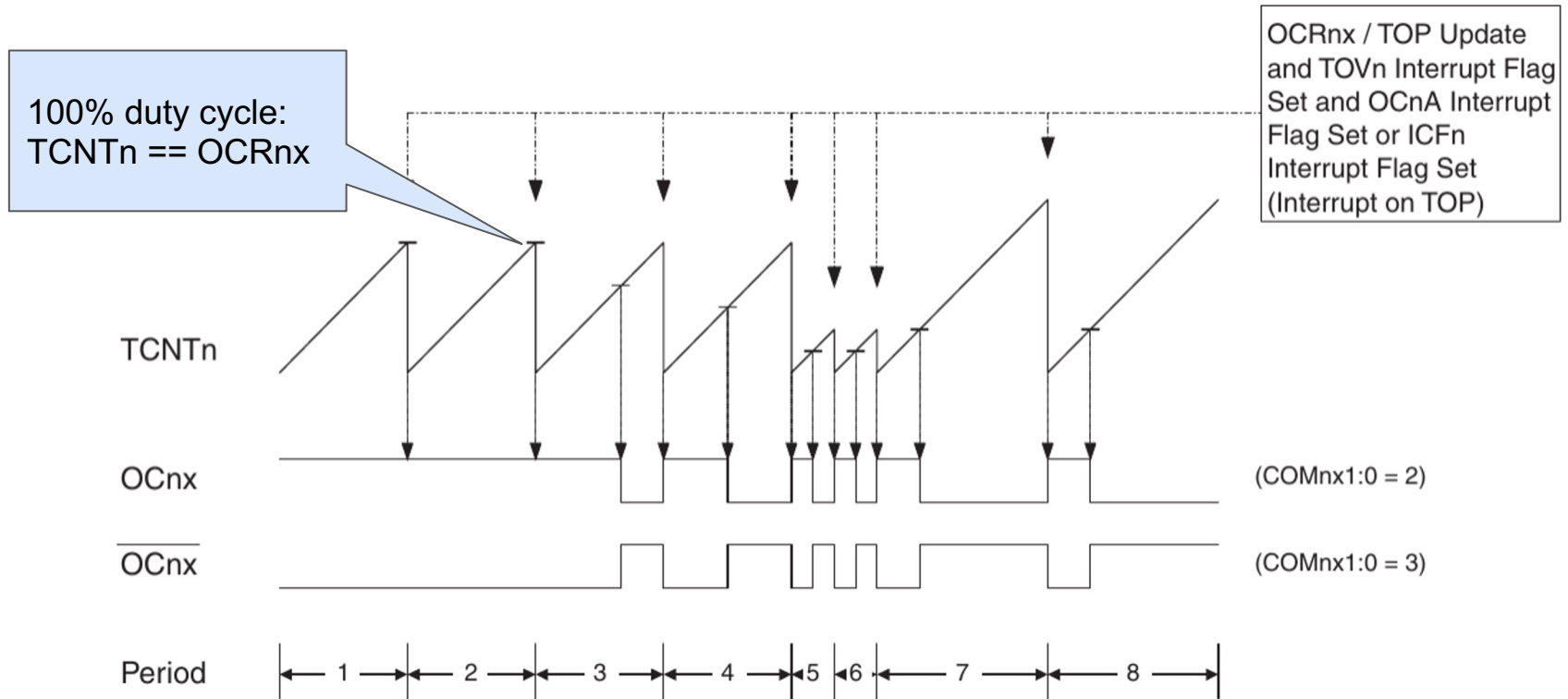


Figure 17-7. Fast PWM Mode, Timing Diagram

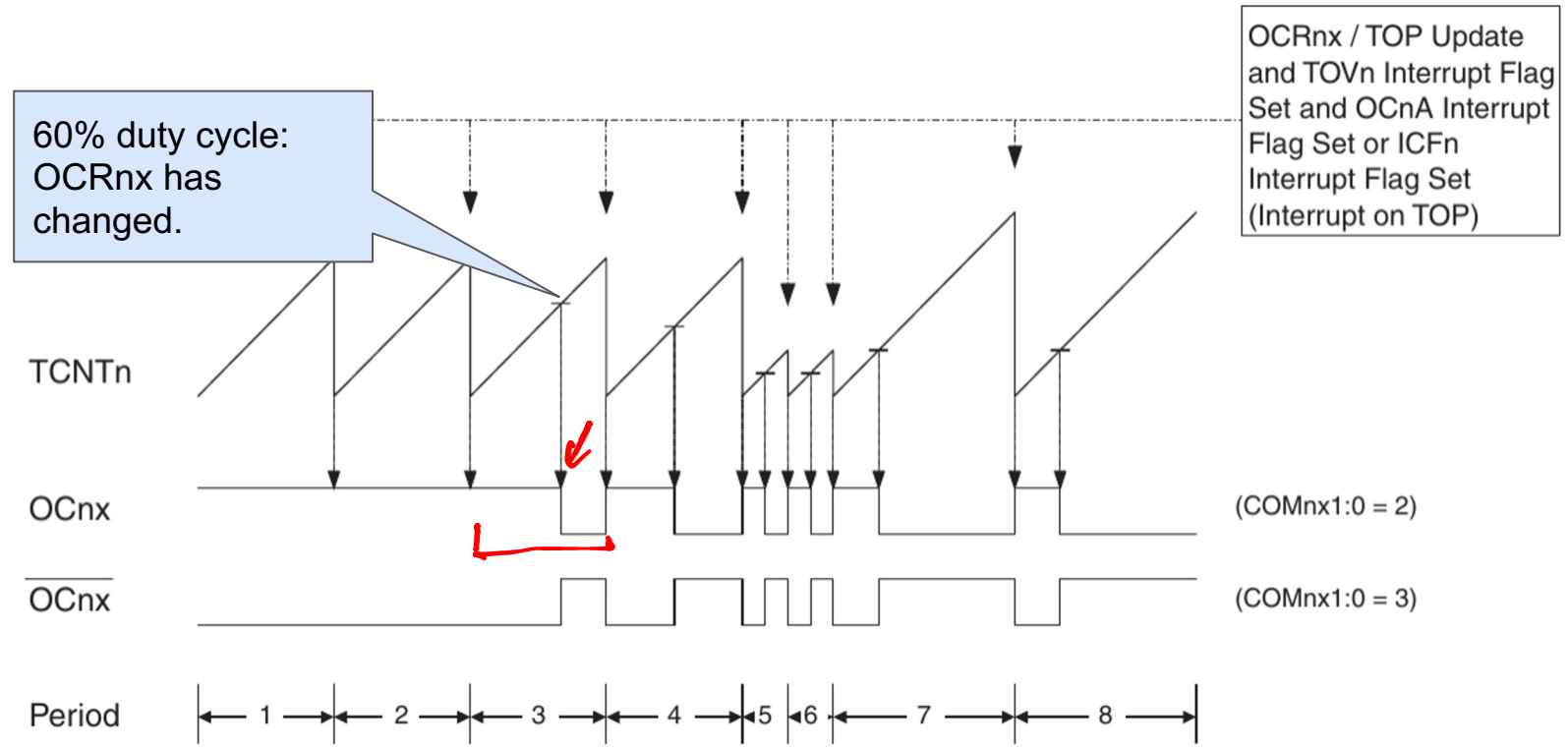


Figure 17-7. Fast PWM Mode, Timing Diagram

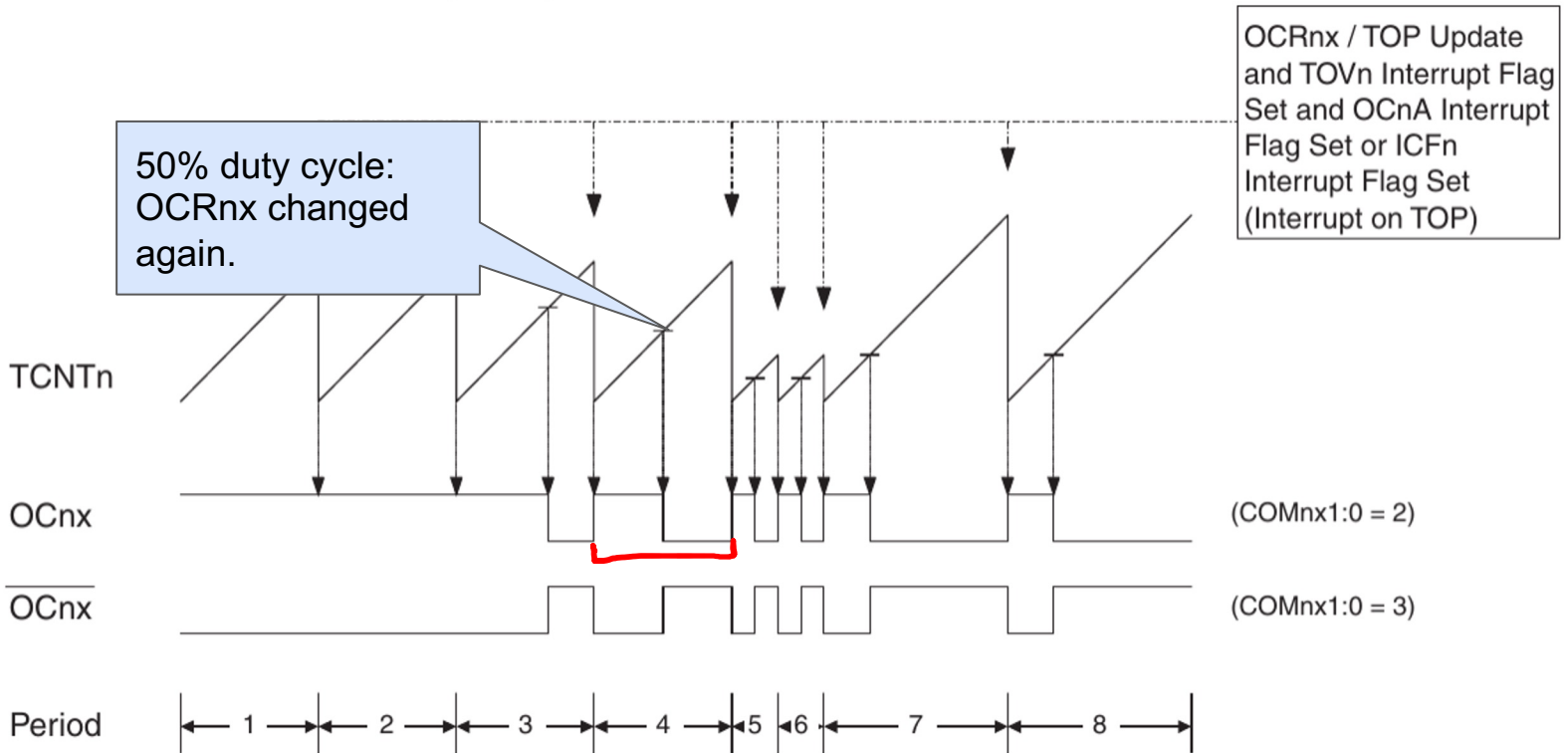
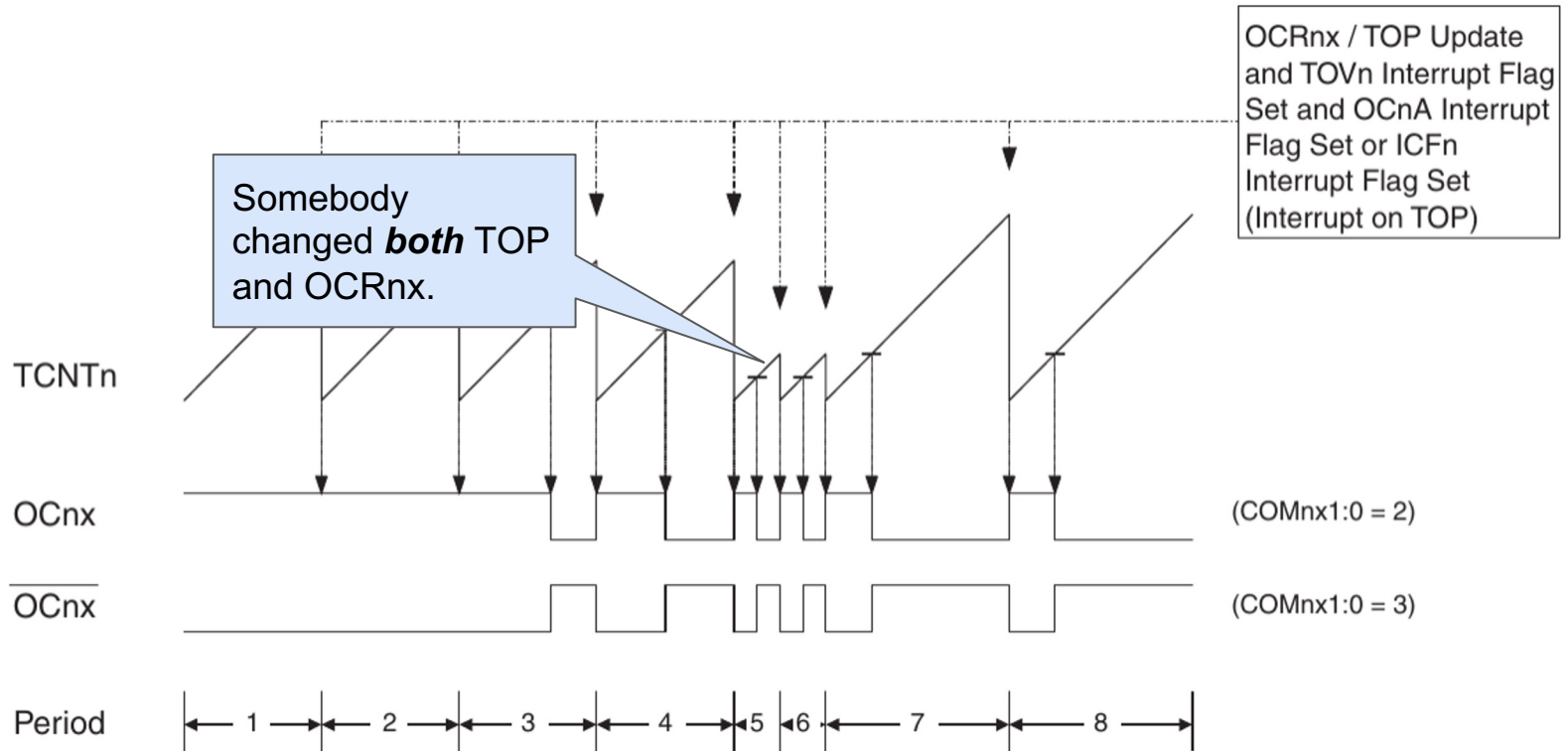


Figure 17-7. Fast PWM Mode, Timing Diagram



16-bit Timer/Counter

Programming Steps: Setup

- 1) Enable Timer (Arduino automatically does it already)

(write zero to a **Power Reduction Register** bit. "1" turns Timer/Counter off)
(p55,56)

- 1) Figure out what clock frequency (prescaler *setting*) you want to use.
- 2) Select a Mode using the 3 control registers (TCCRnA, TCCRnB, TCCRnC)
- 3) set/clear Waveform Generation Mode bits:

WGMn0 --- WGMn3 (two different registers)

- a) **TCCRnA:** Compare Mode and partial WG Modes
- b) **TCCRnB:** partial WG Modes and Clock/Prescaler select.
- c) "n" is your counter number.

- 4) Set your ISR and enable interrupts. (**LATER**)

Thinking about timers as software

Here's a way to think about timers for those more comfortable with software, it's basically a for loop counter. We also have a conditional to perform an action and break out of the loop. For example in CTC mode, counting up to `output_compare` triggers an action (e.g. toggling a pin). Note that this whole thing is wrapped in a big `while(1)` loop because the counter resets to zero and then starts over.

```
1 while(1){
2     int output_compare = 26; // some number we want to count up to
3     int timer_counter = 0;
4
5     // This assumes we're using an 8 bit timer
6     for(timer_counter = 0; timer_counter < 256; timer_counter++) {
7         if(timer_counter == output_compare){
8             // We counted up to the number we set, now we do something
9             . toggle_pin(); // or other action based on your registers
10            break; // Example: in CTC mode we stop counting now and start over
11        }
12    }
13 }
```

16-bit Timer/Counter

Programming Steps: Setup

- 5) Figure out the compare count you need and set it in **OCRnA**
- 6) Enable or disable timer hardware output with **DDRx** (data direction register for the output pin).