# Lecture7: ATMega 2650 Datasheet and 16 bit timer

Vikram Iyer

Adapted from material by Blake Hannaford

# Administrative

- Lab 1 due tonight
  - Everyone should have a partner, lab kit, and a speaker
  - Turn in 1 lab report with both partners names on Canvas
  - Reports for this lab can be brief, the goal is to get in the habit of documenting your code and process for future labs
- Programming assignment 2 is up and due 4/24
  - 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
```

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

# Administrative

- Lab 1 due tonight
  - Everyone should have a partner, lab kit, and a speaker
  - Turn in 1 lab report with both partners names on Canvas
  - Reports for this lab can be brief, the goal is to get in the habit of documenting your code and process for future labs
- Programming assignment 2 is up and due 4/24
  - 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 will be posted tonight/tomorrow
  - Longer than Lab 1, get started early
  - After today we'll have covered material for Part 1 and 2
  - Part 3 material will be in next couple of 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

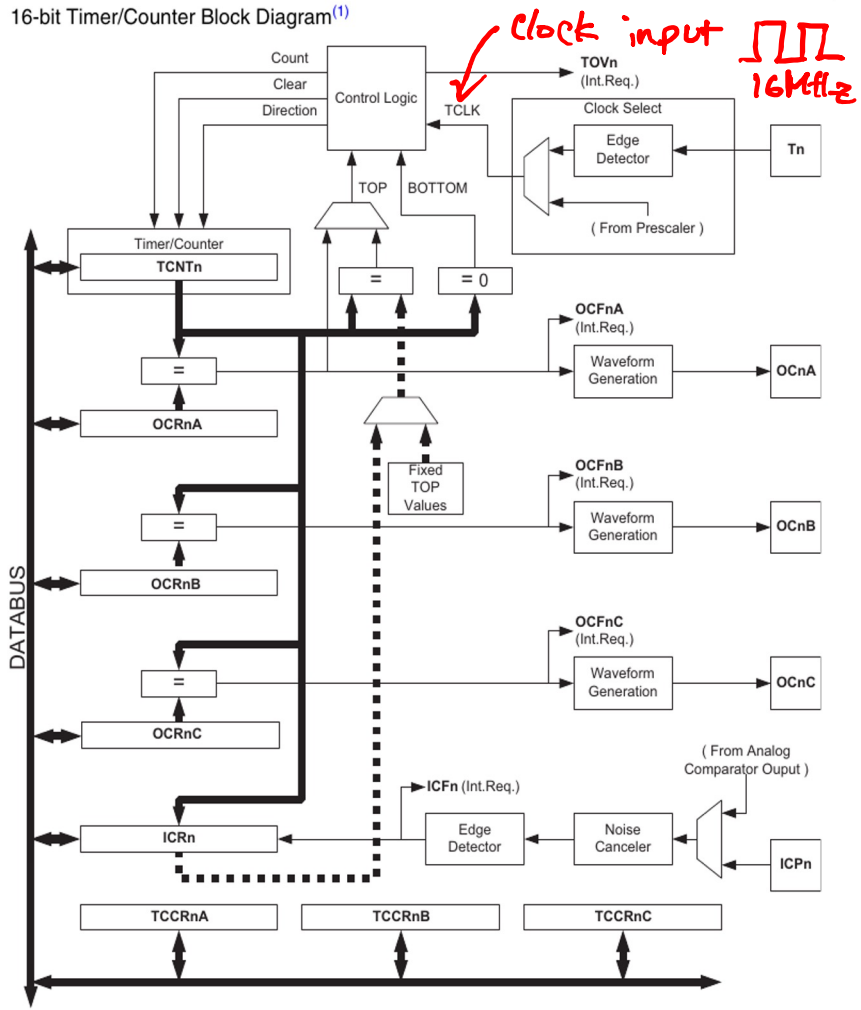
- Walk through ATMega datasheet
  - Point out where to look for documentation in lab 2
- Details of 16 bit timer counter

# ATMega2650 Datasheet

# 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

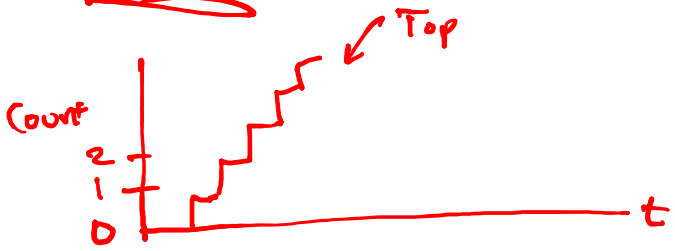
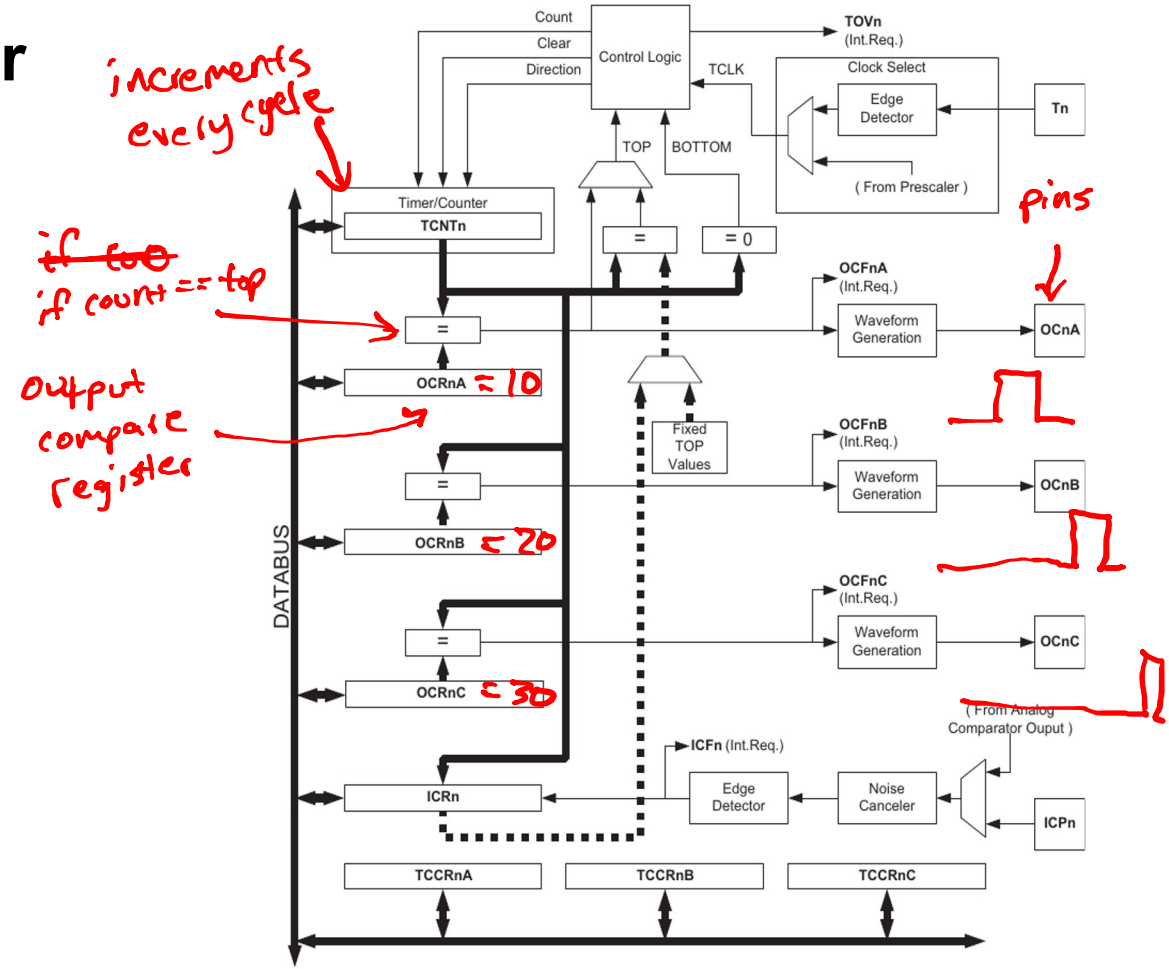


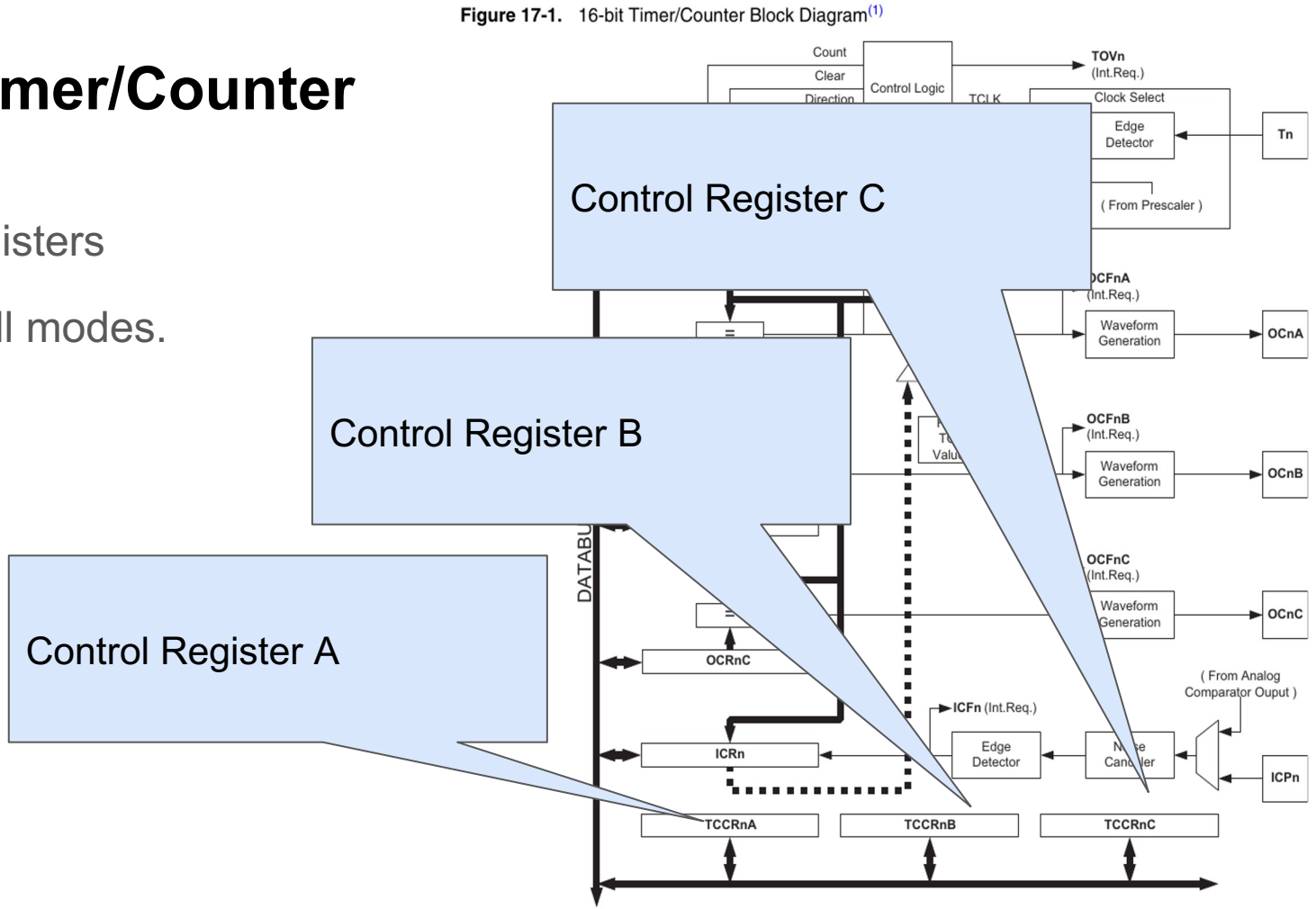

**Figure 17-1.** 16-bit Timer/Counter Block Diagram[1]



# 16-bit Timer/Counter

Control Registers

Configure all modes.



# 16-bit Timer/Counter

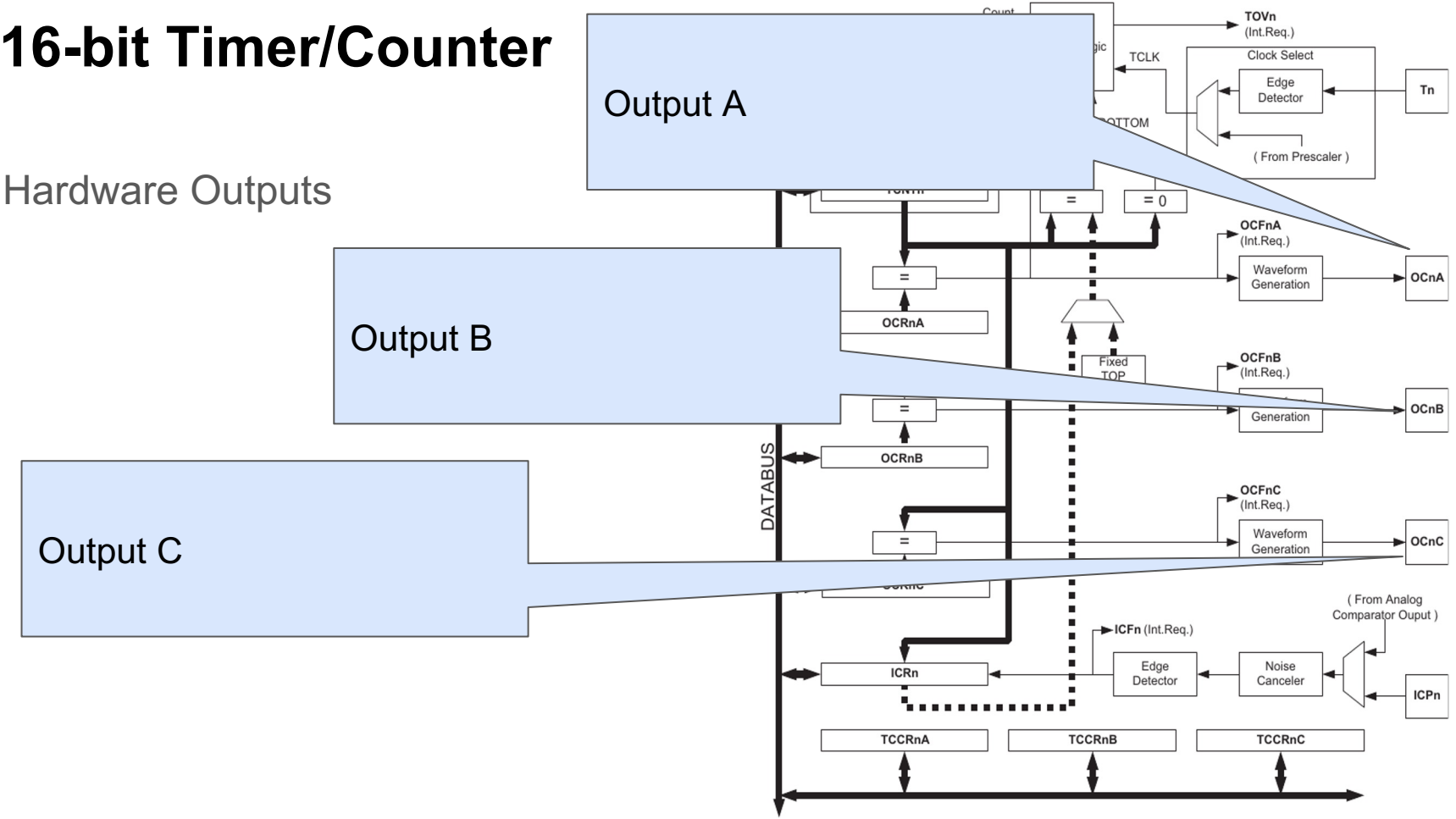
Hardware Outputs

Output A

Output B

Output C

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



# 16-bit Timer/Counter

Combinatorial circuit which sets or clears the output pin.

Inputs:

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

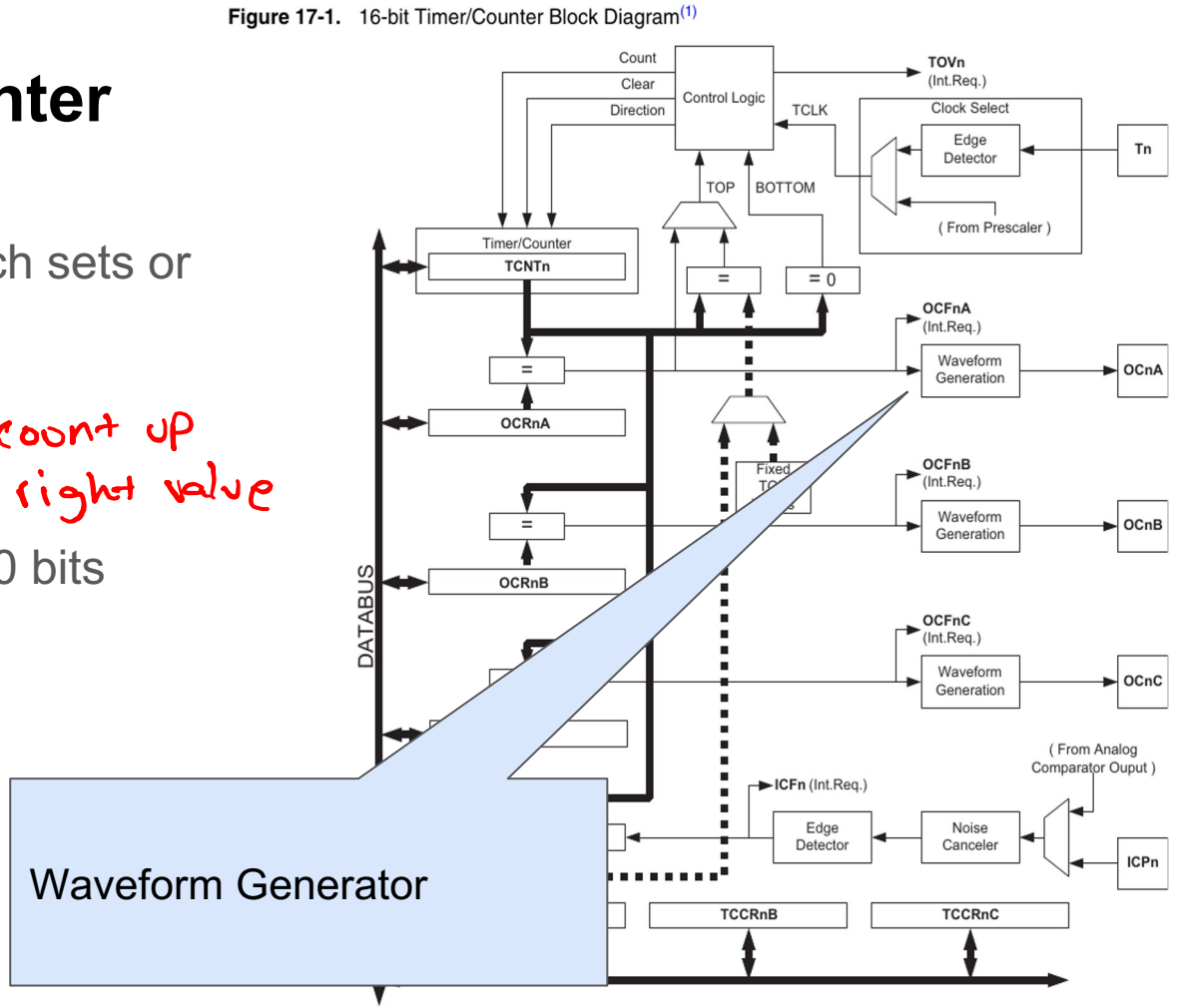
did we count up to the right value

Actions:

- Set, Clear, Toggle

=1 =0 XOR


**Figure 17-1.** 16-bit Timer/Counter Block Diagram[1]

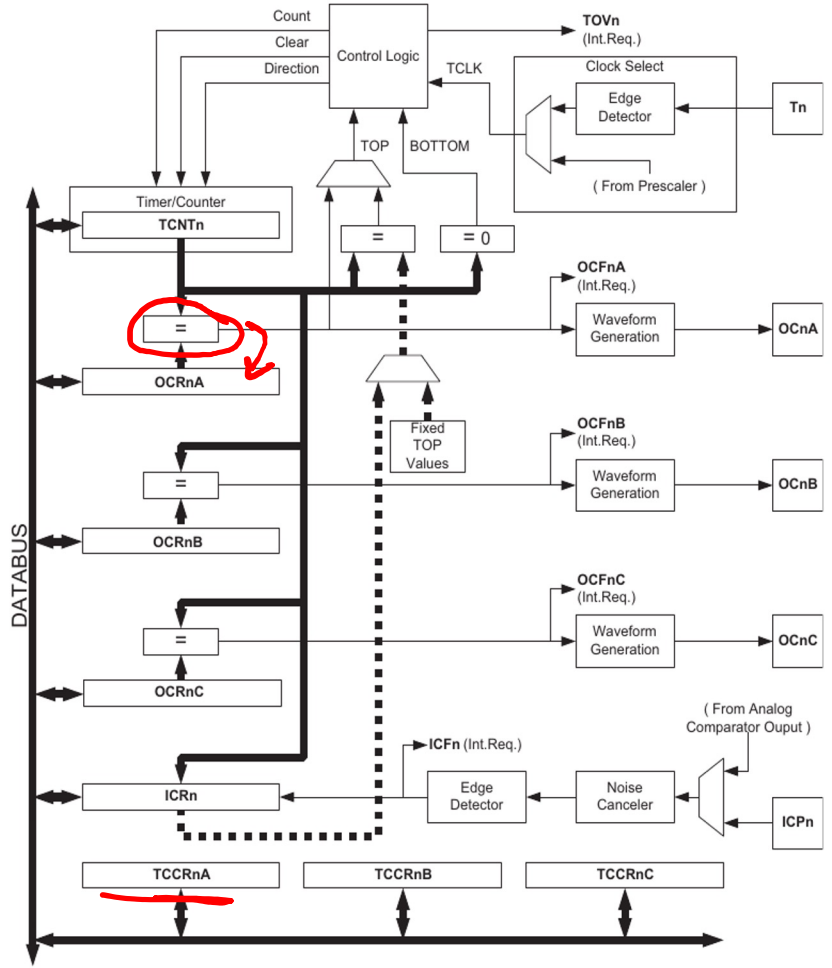


# 16-bit Timer/Counter

## Block Diagram

write TOP

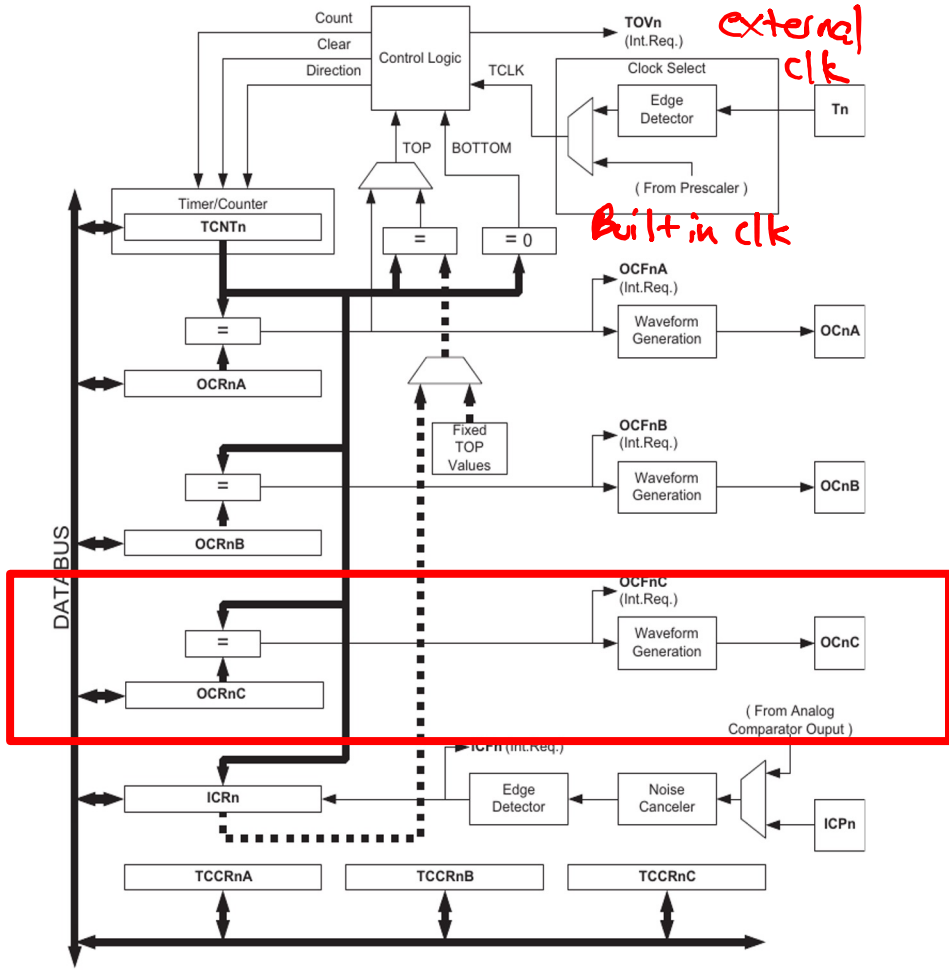
**Figure 17-1.** 16-bit Timer/Counter Block Diagram[1]



# 16-bit Timer/Counter

Block Diagram

**Figure 17-1.** 16-bit Timer/Counter Block Diagram[1]

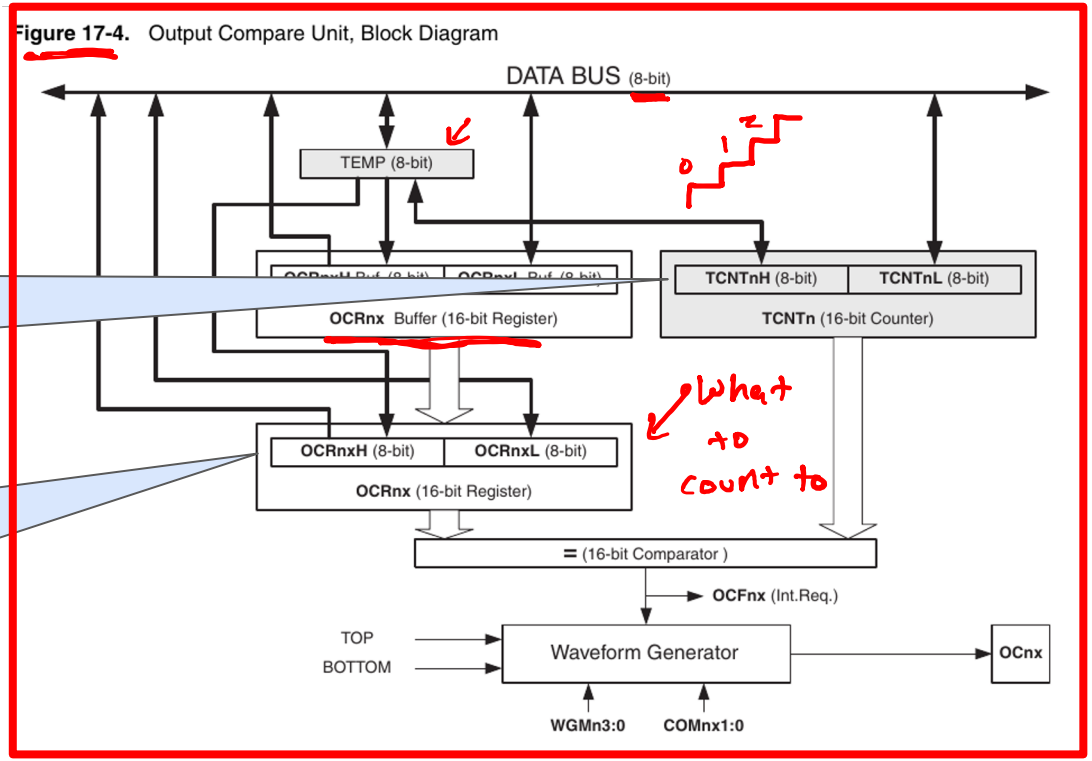


# Output Compare Unit

## Expanded Block Diagram

Current Counter Value

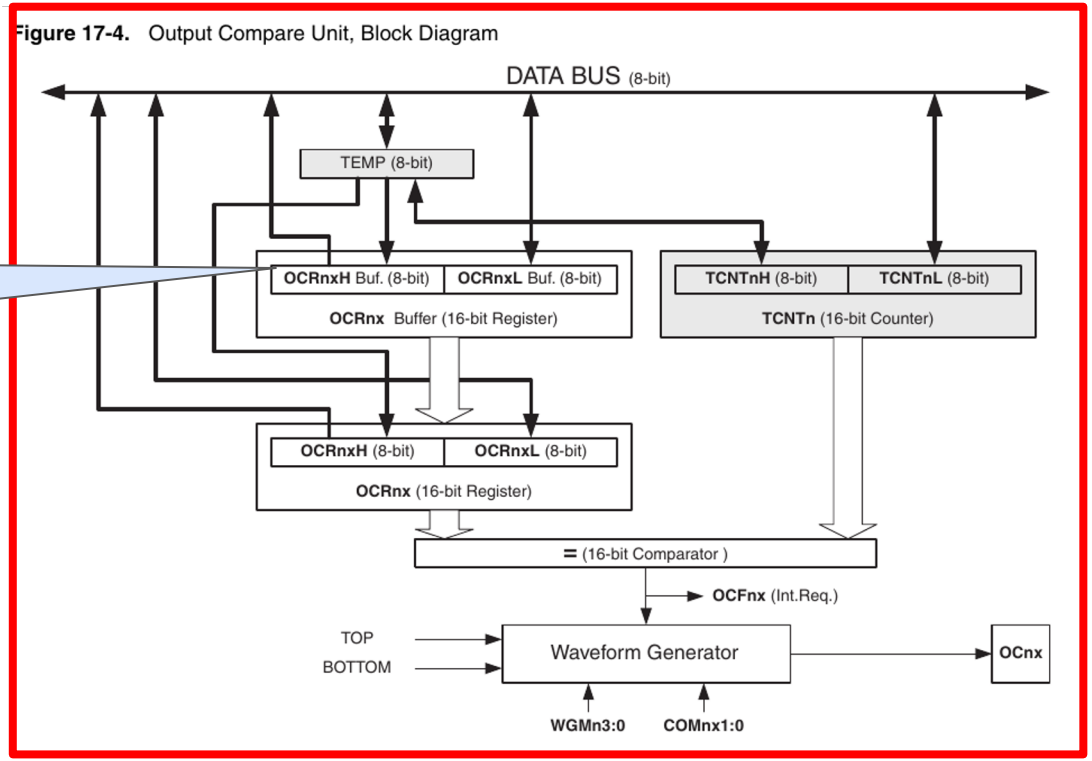
Compare Value



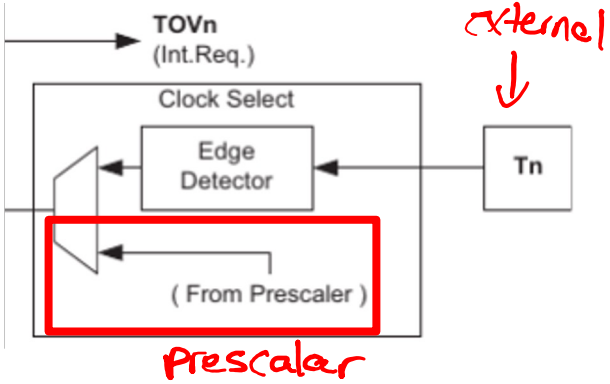
# Output Compare Unit

## 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 1 | 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)

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

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

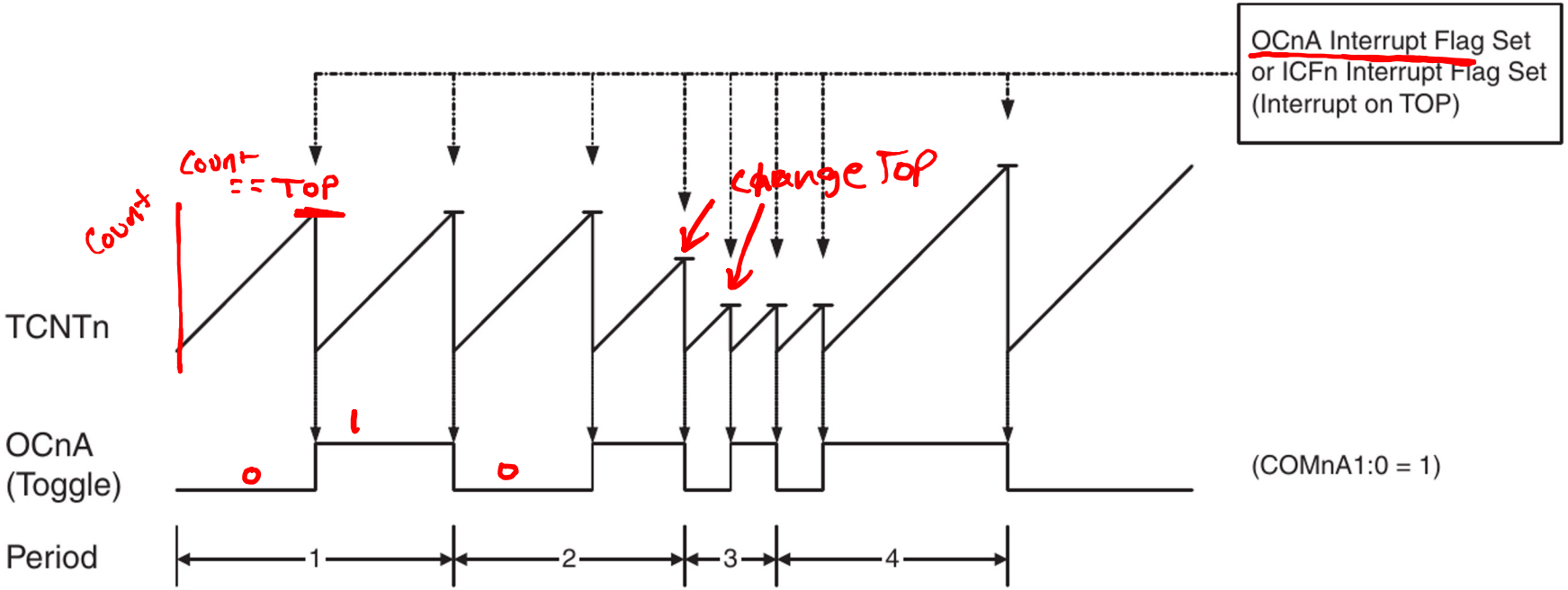
| Mode | WGMn3 | WGMn2 (CTCn) | WGMn1 (PWMn1) | WGMn0 (PWMn0) | Timer/Counter Mode of Operation | TOP | Update of OCRnx at | TOVn Flag Set on |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Normal | 0xFFFF | Immediate | MAX |
| 1 | 0 | 0 | 0 | 1 | PWM, Phase Correct, 8-bit | 0x00FF | TOP | 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 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.

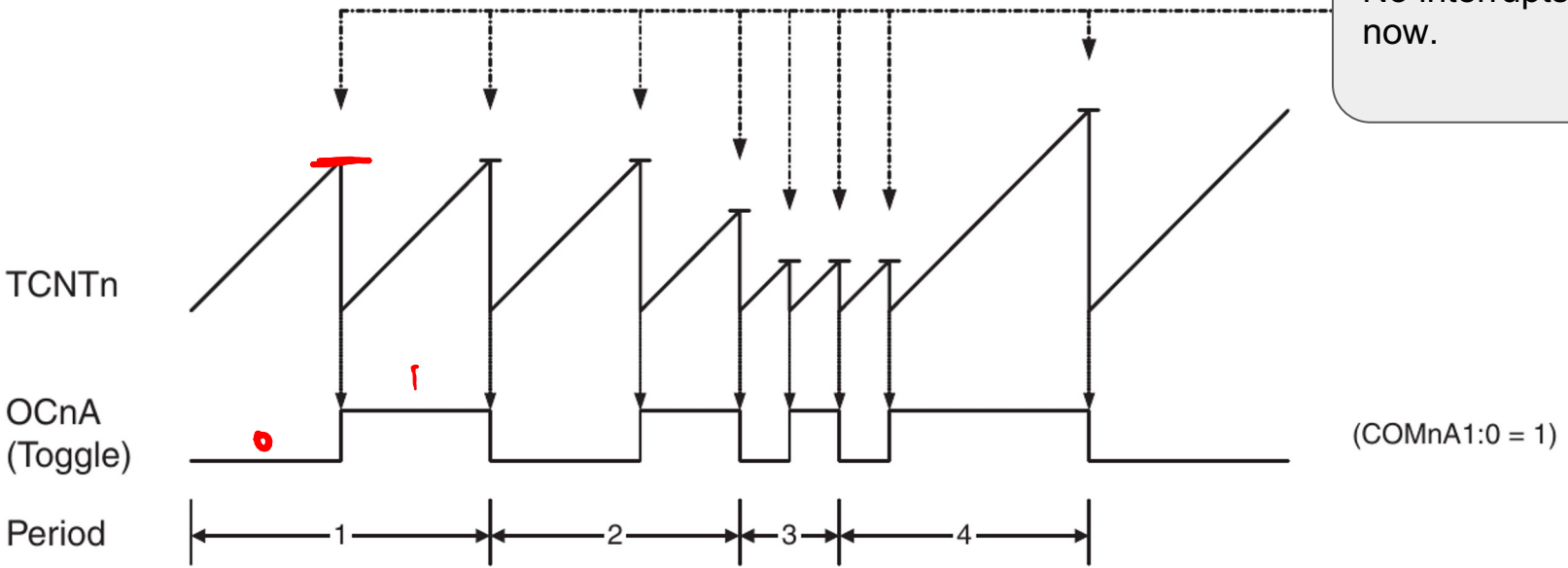
# CTC Mode

**Figure 17-6.** CTC Mode, Timing Diagram



# CTC Mode

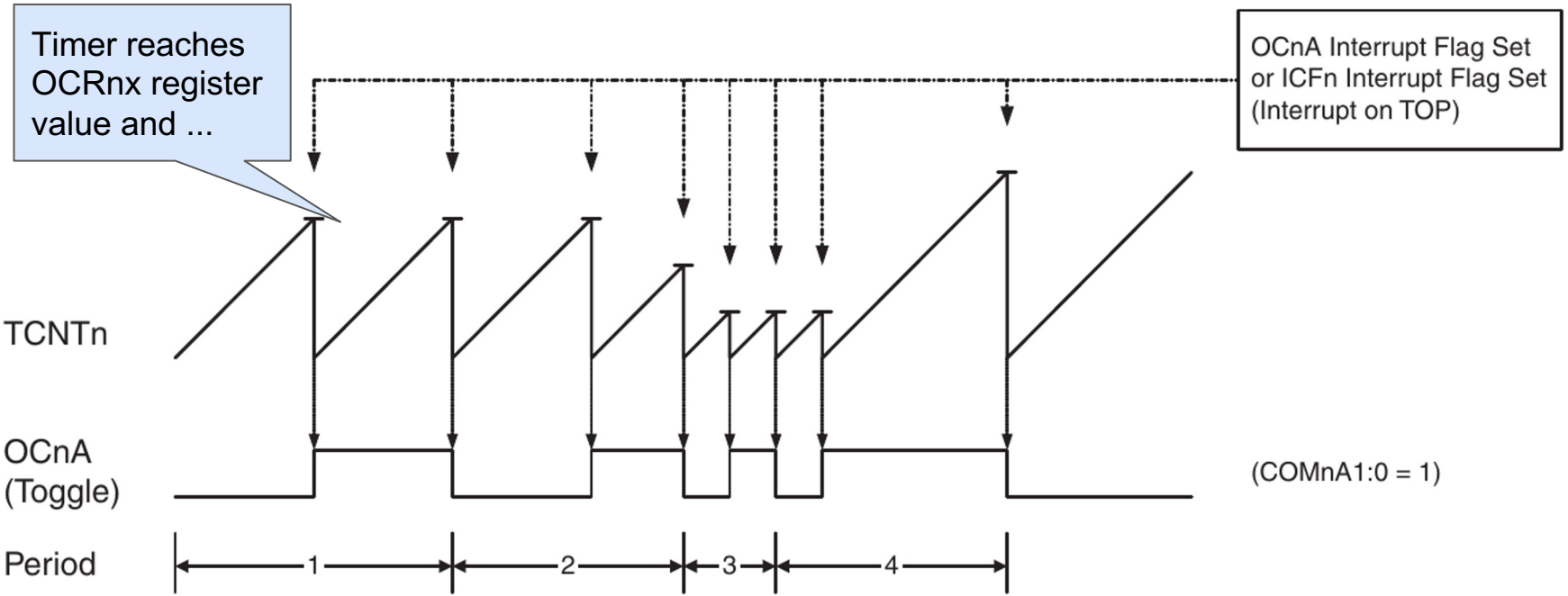
**Figure 17-6.** CTC Mode, Timing Diagram




No interrupts for now.

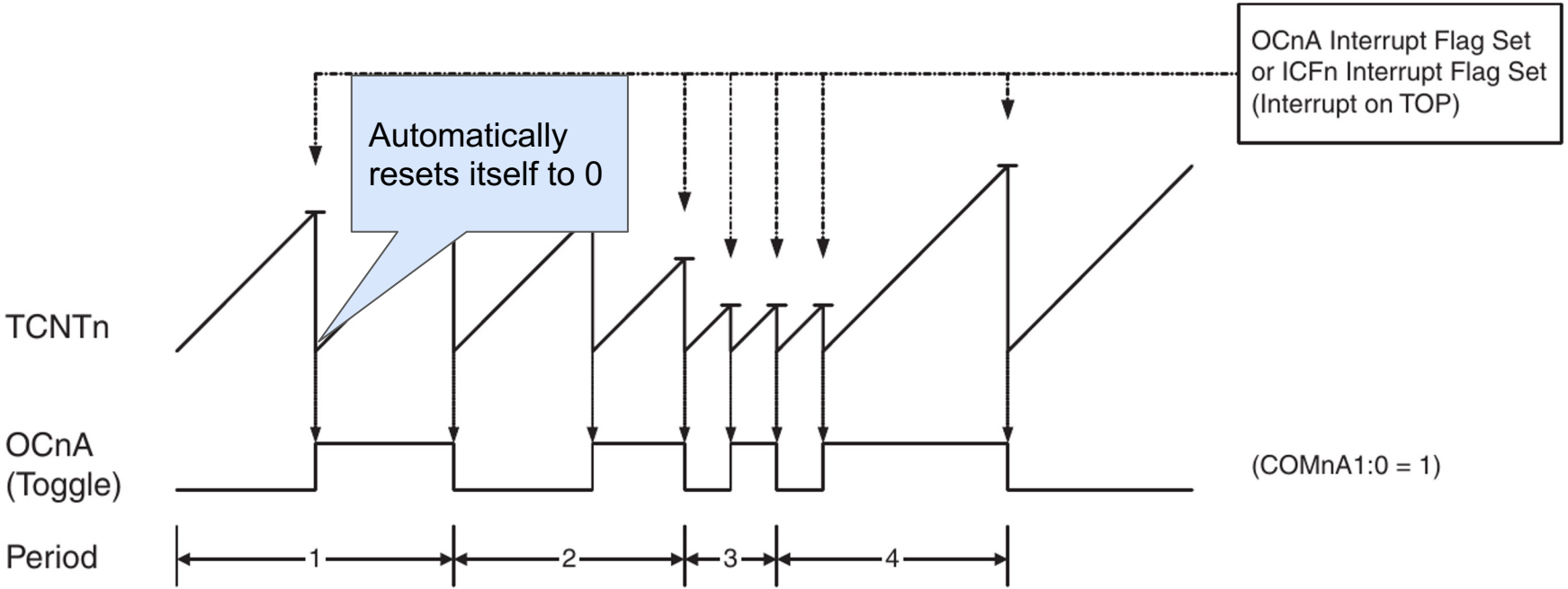
# CTC Mode

**Figure 17-6.** CTC Mode, Timing Diagram



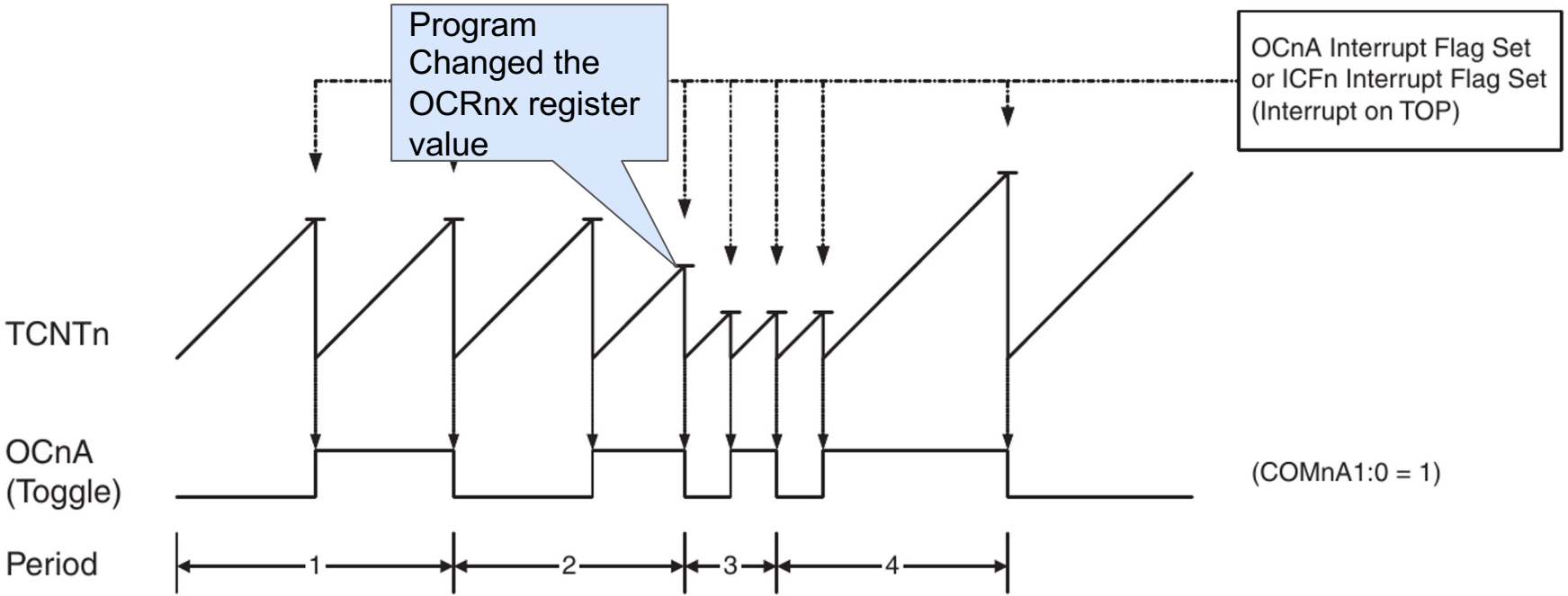
# CTC Mode

**Figure 17-6.** CTC Mode, Timing Diagram



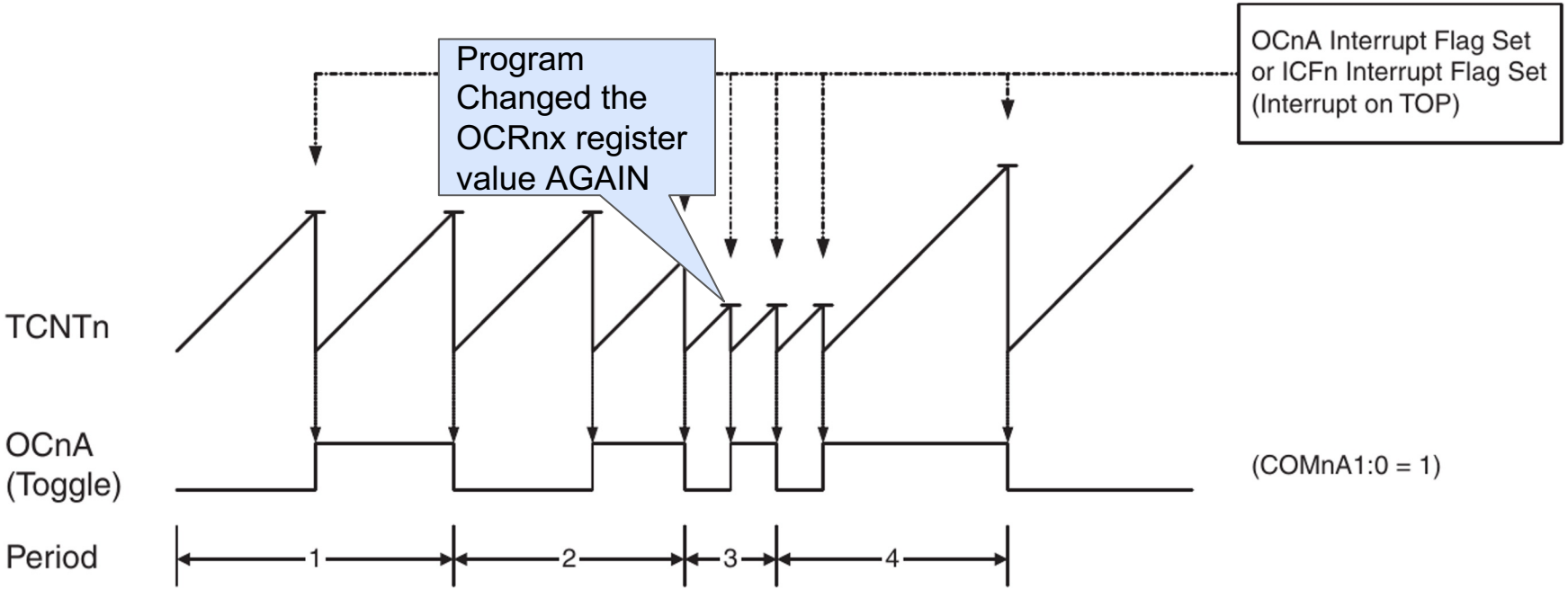
# CTC Mode

**Figure 17-6.** CTC Mode, Timing Diagram



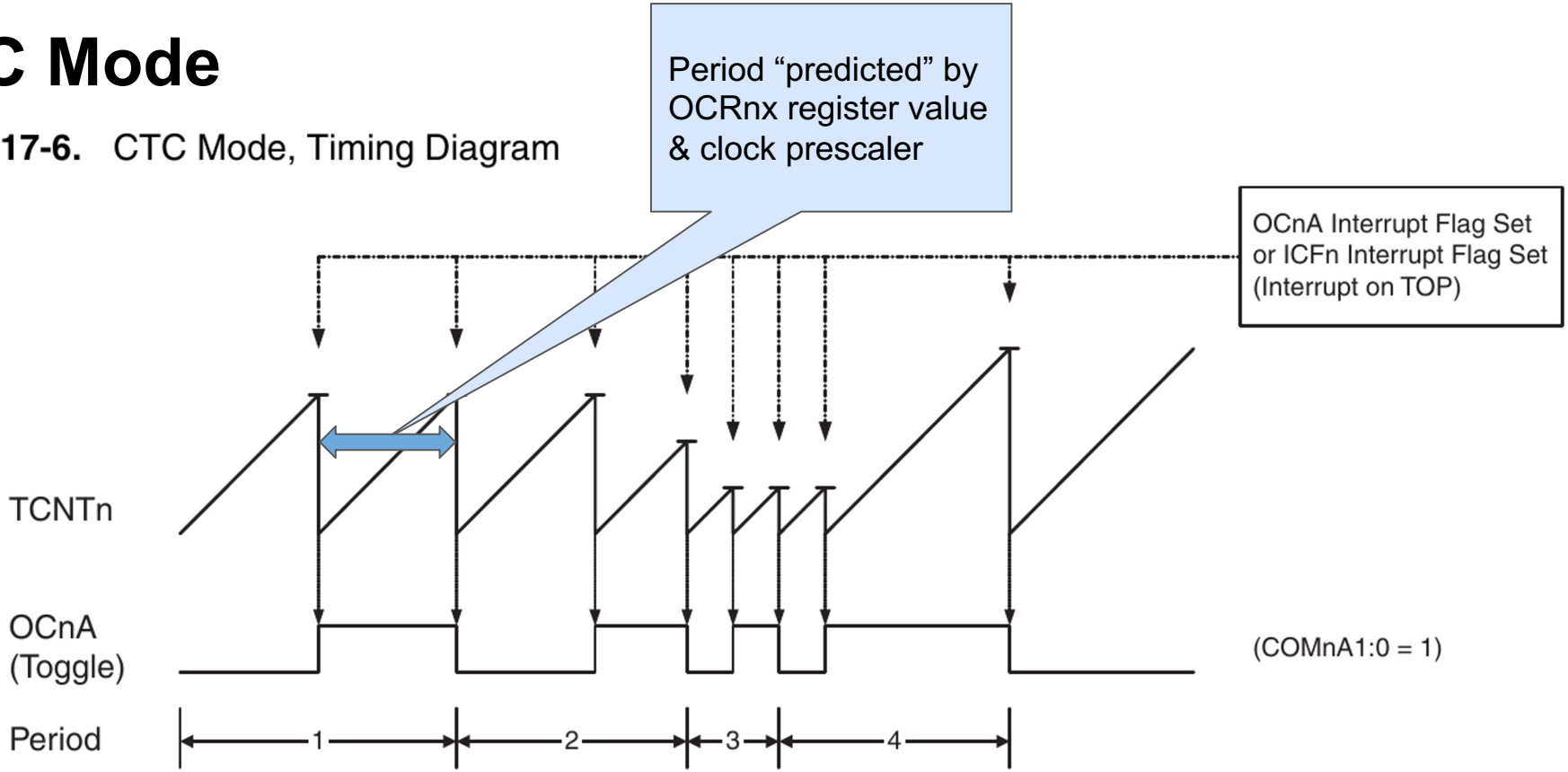
# CTC Mode

**Figure 17-6.** CTC Mode, Timing Diagram



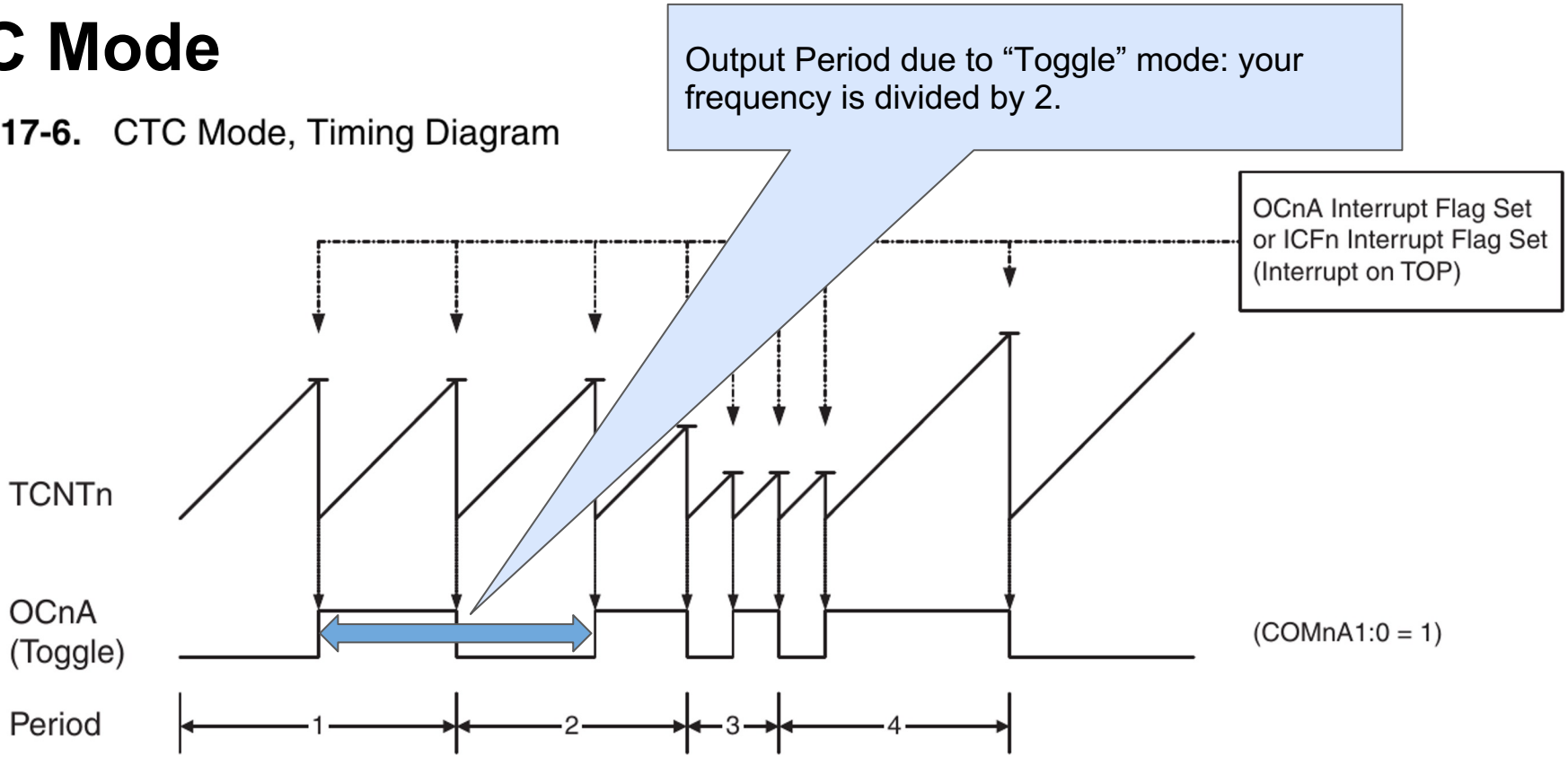
# CTC Mode

**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.

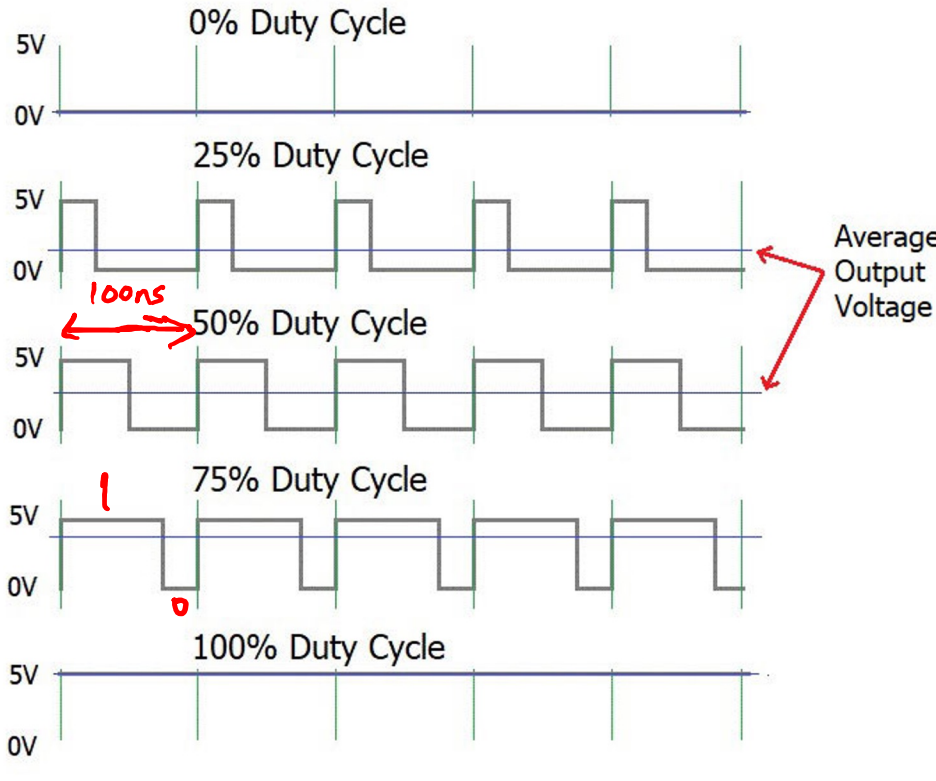


**Figure 17-7.** Fast PWM Mode, Timing Diagram

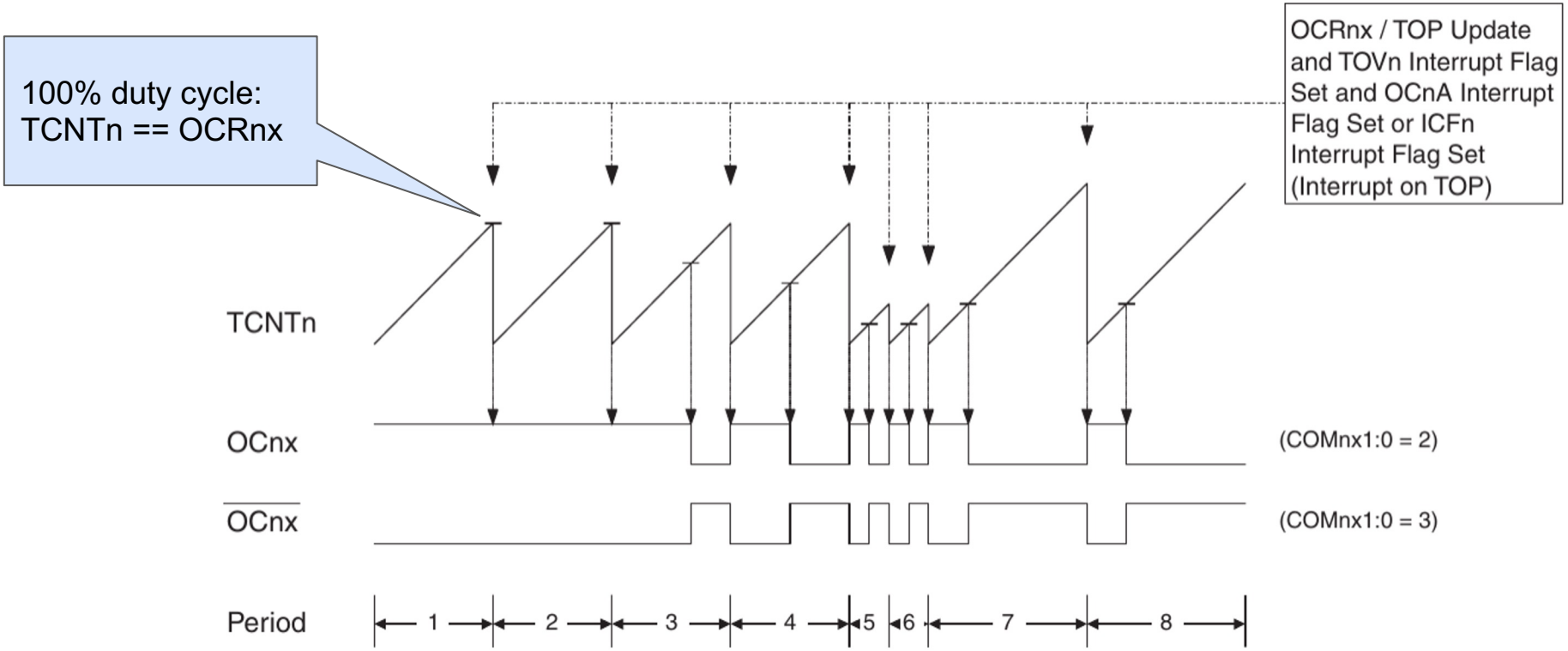


**Figure 17-7.** Fast PWM Mode, Timing Diagram

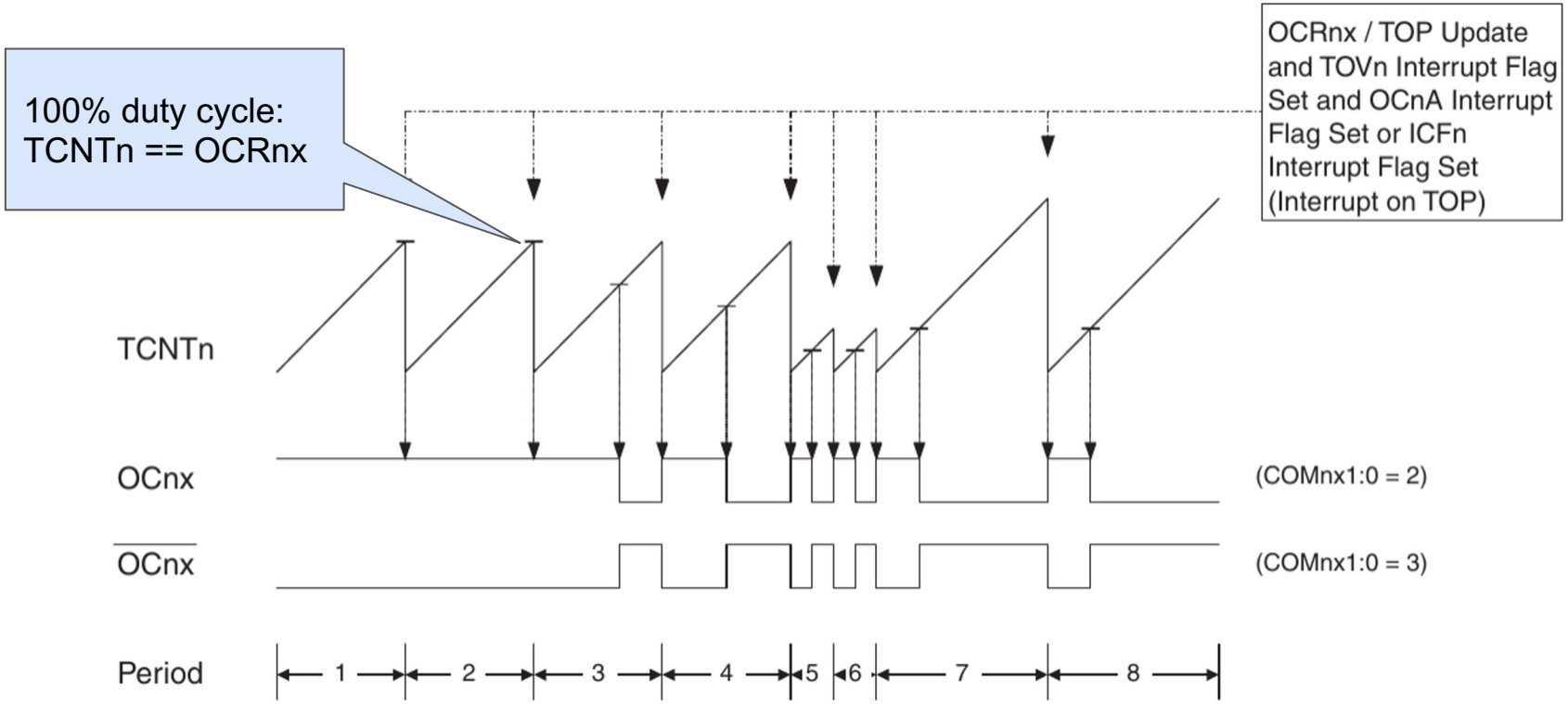


**Figure 17-7.** Fast PWM Mode, Timing Diagram

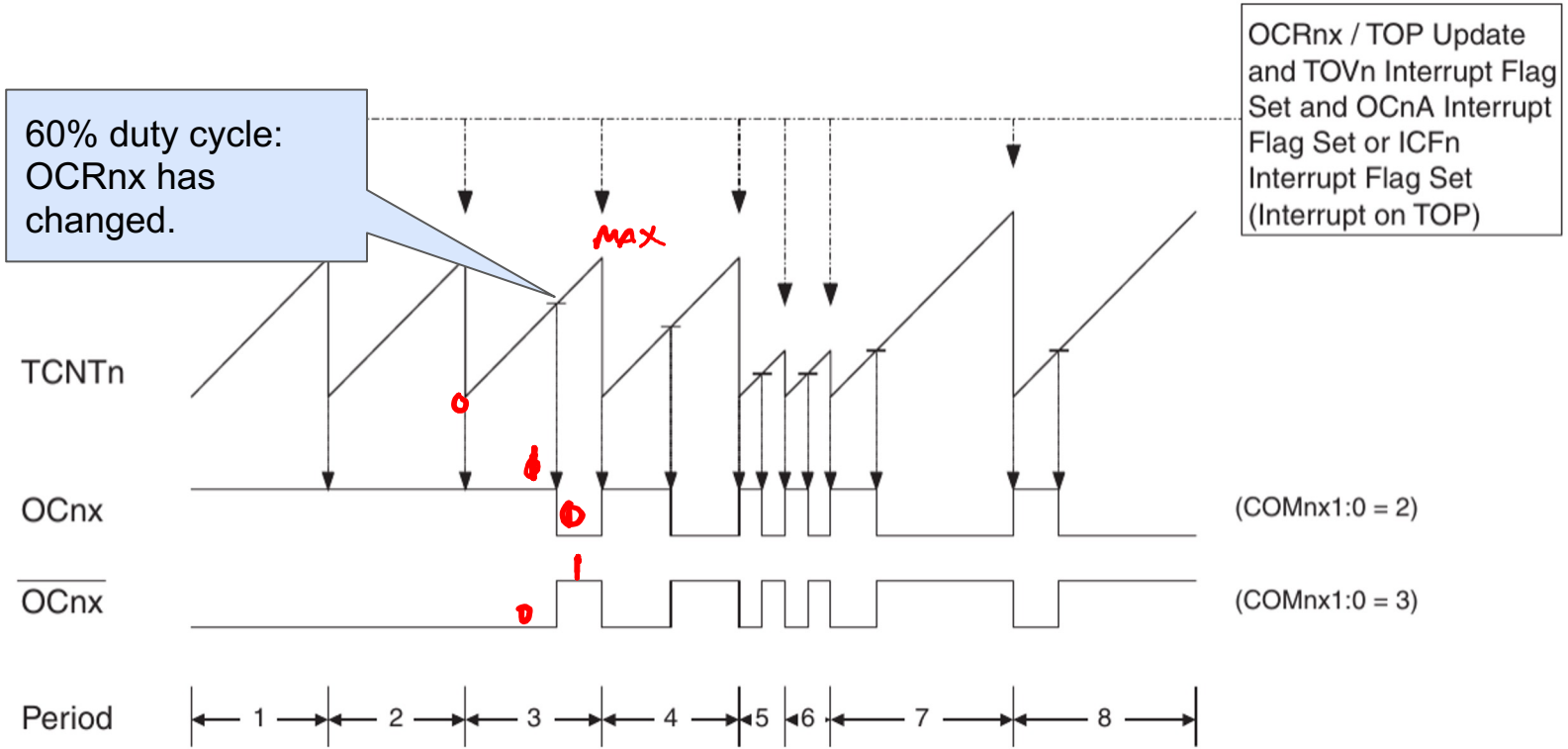


**Figure 17-7.** Fast PWM Mode, Timing Diagram

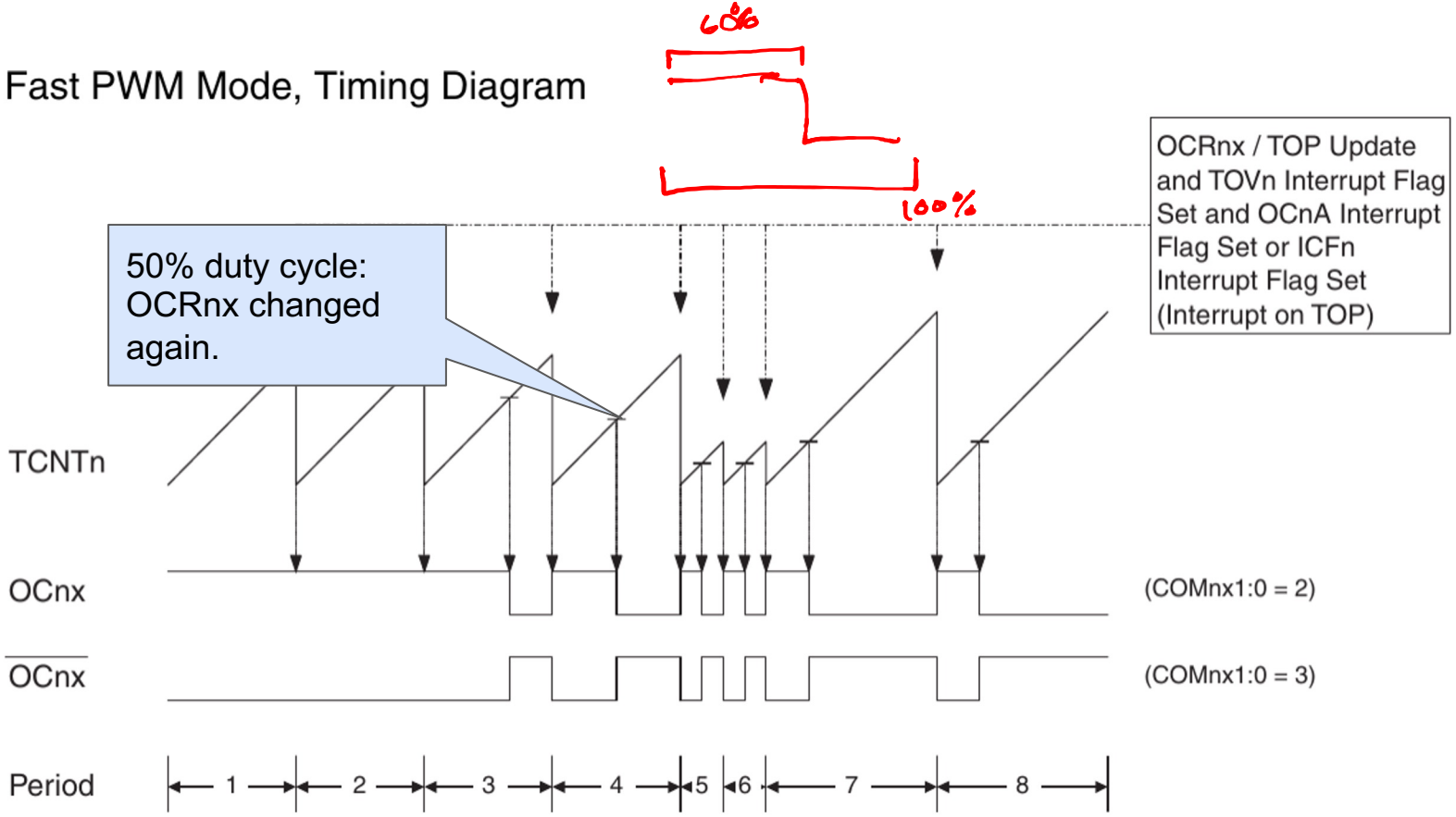
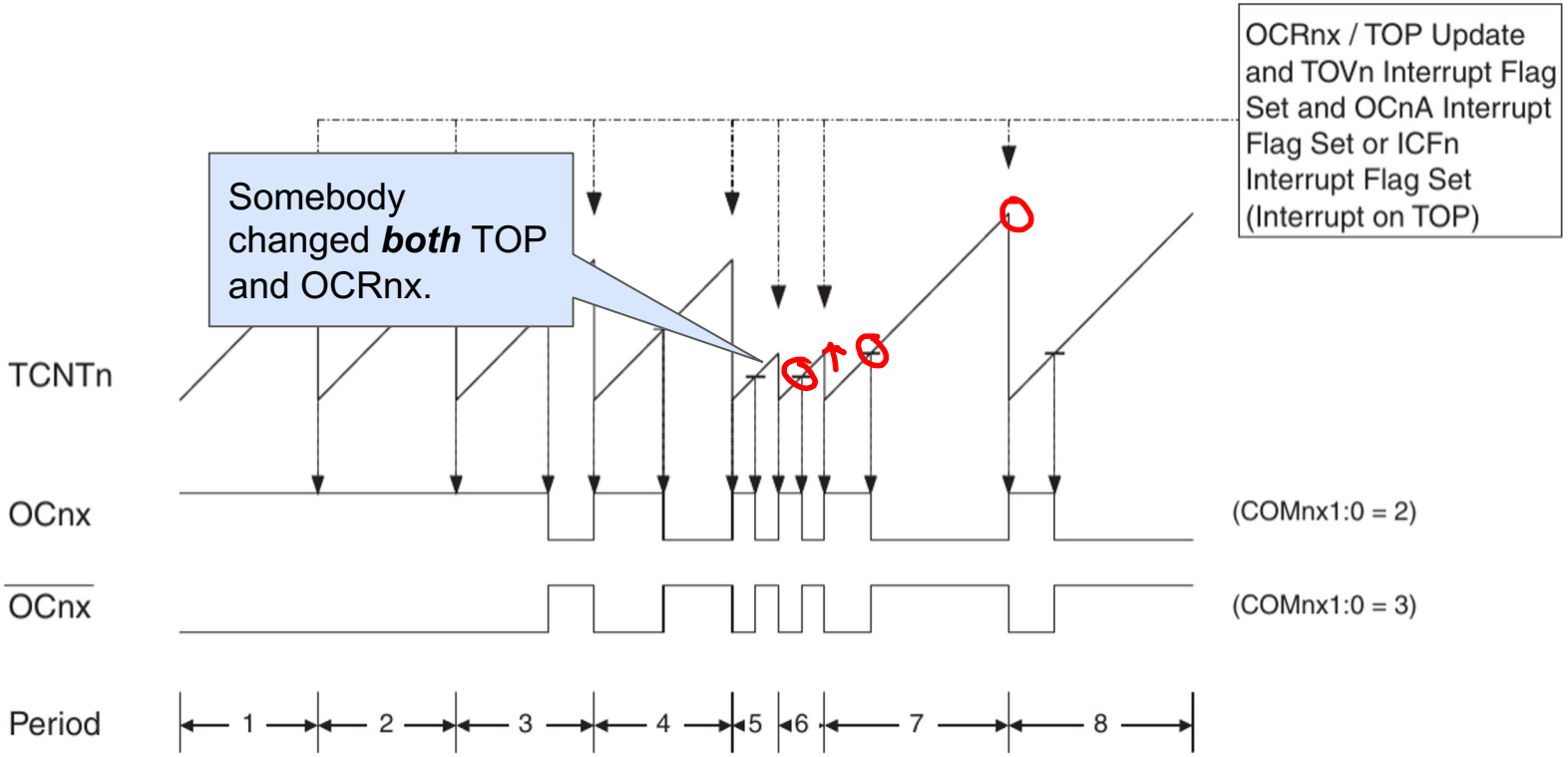


**Figure 17-7.** Fast PWM Mode, Timing Diagram



# 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**)

# 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).