Timers & Counters.

Note Title 2020-04-19

1- Why do we need timers?

* Count something!
* Keep track of time!

2- CPU can do this?

yes!

3- delay()?.... How does it work?

↳

```
for(int i=0 ; i < some number ; i++)
    {
      #asm("nop")
    }
```

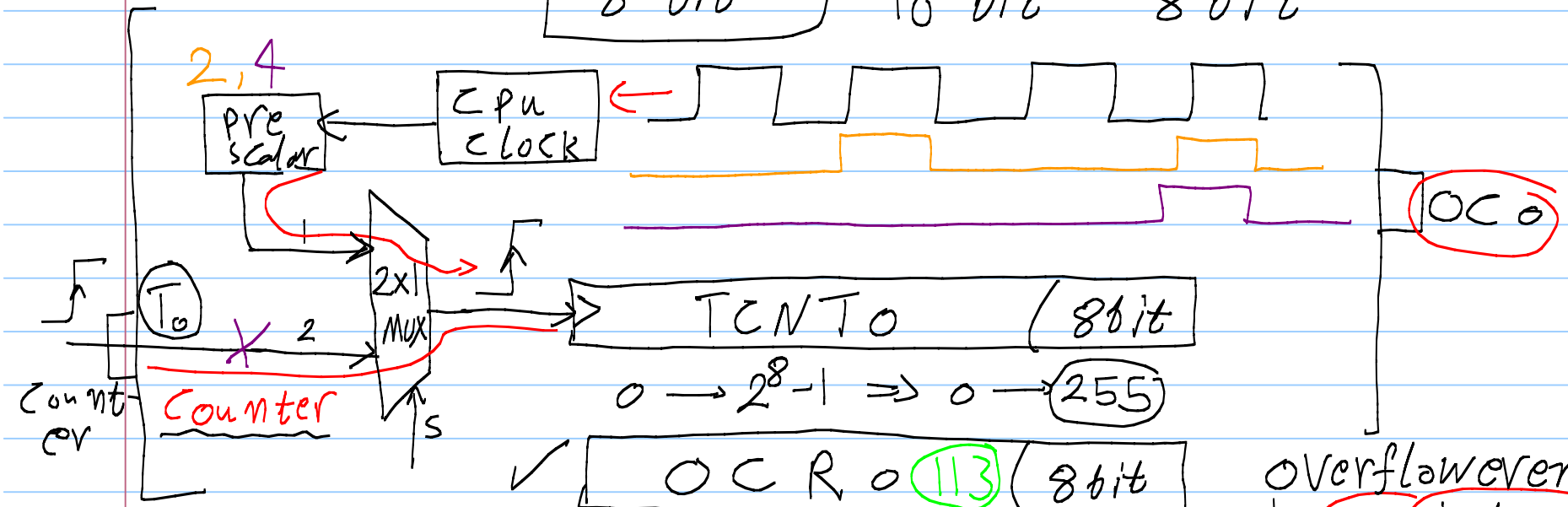
if cpu clock is 1MHz.

What is the time for 1K instruction?

= 1K instruction X 1μsec/instruction = 1msec

Atmega 16: Timer0, Timer1, Timer2

8 bit 16 bit 8 bit



COM01 : COM00

0	0	Discom.
0	1	Toggle
1	0	clear
1	1	set

TCCR0 (8bit

CS2 : CS0 ← prescalor + input source

Compare match event!

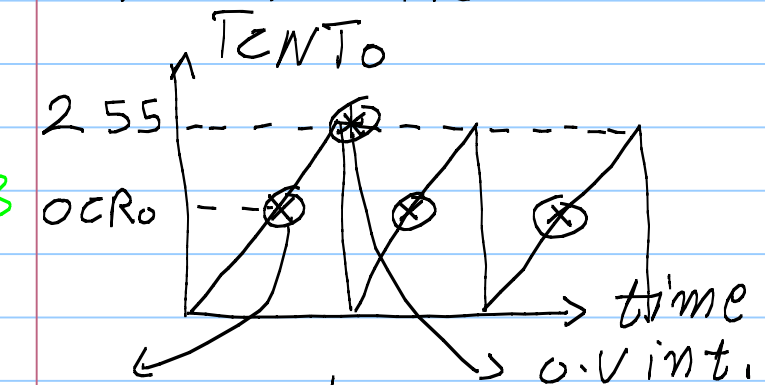
↳ C.M interrupt

↳ output pin changes

WGM01 : WGM00

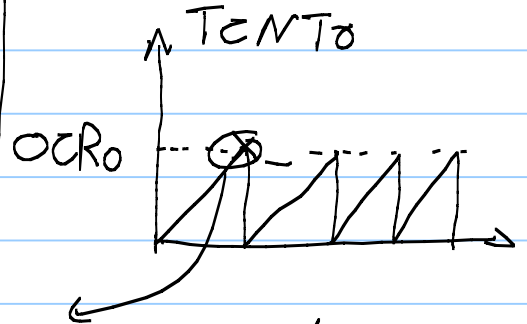
- 0 0 Normal mode ✓
- 0 1 PWM, phase correct ✗
- 1 0 CTC ✓
- 1 1 PWM, fast ✗

Normal mode



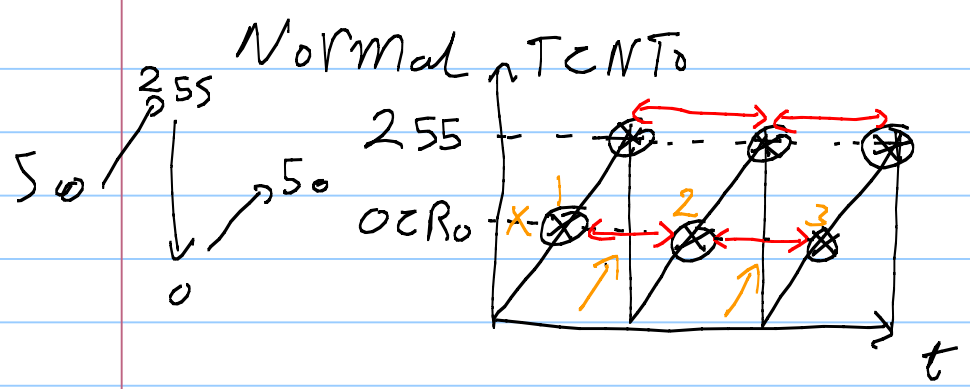
113

CTC ; clear Timer in Compare match



Interrupt time Calculations

Normal



→ O.V interrupt time:

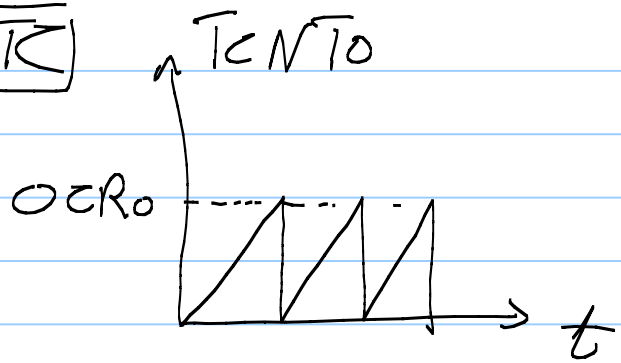
$$= \underbrace{2^{8}}_{\text{عدد الاعداد}} \times \left(\frac{1}{f_{cpu}} \times \text{prescalar}\right)$$

(8 ← عدد الاعداد ; 1 usec × 2)

→ C.M interrupt time:

$$= 2^{8} \times \left(\frac{1}{f_{cpu}} \times \text{prescalar}\right)$$

CTC



O.V interrupt time: ?

N.A !!!

→ C.M interrupt

$$= \underbrace{(OCR0+1)}_{\text{عدد الاعداد}} \times \left(\frac{1}{f_{cpu}} \times \text{prescalar}\right)$$

EX: Use timer0 to give an interrupt every 1sec. (1msec)

$\Rightarrow$ cpu = 8 MHz, Normal mode.

o.v int. time $= 2^8 \times \left(\frac{1}{f_{cpu}} \times \text{prescalar}\right)$

prescalar	o.v int. time	(1/o.v int time) # interrupts in 1sec
→ 1	32 μsec	31250
8	256 μsec	3906.25
64	2.048 msec	488.28125
256	8.192 msec	122.0703
1024	32.768 msec	30.5175

prescalor:	1024
# int/sec:	31
# 8 MHz	

int in 1msec

31 (25)

3.9

Norm. O.V interrupt: $2^8 \times \left(\frac{1}{f_{cpu}} \times pre\right)$: 5 options (pre: 5)

CTC: C.M interrupt: $(OCR_0 + 1) \times \left(\frac{1}{f_{cpu}} \times pre\right)$: 256 × 5 option

(OCR_0: 256, pre: 5)

prescalar	O.V int. time
→ 1	→ 32 μsec
→ 8	→ 256 μsec
→ 64	2.048 msec
256	8.192 msec
→ 1024	32.768 msec

CTC Mode
- pre = 64
- OCR_0 = 124
- → C.M int = 1 msec.

we want to get 1msec:
- → without error
- → in one interrupt

$(OCR_0 + 1) \times \left(\frac{1}{8MHz} \times 64\right) = 1msec$

$OCR_0 = 124$

$124 \times \left(\frac{1}{8M} \times 64\right) = 1msec$ ✓

When ?

Normal + o.V : Large times > 100 msec

CTC + C.M : small times <32 msec.

codevision + proteus

interrupt every 1sec

↳ use : normal mode

pre : 64

#int : 488