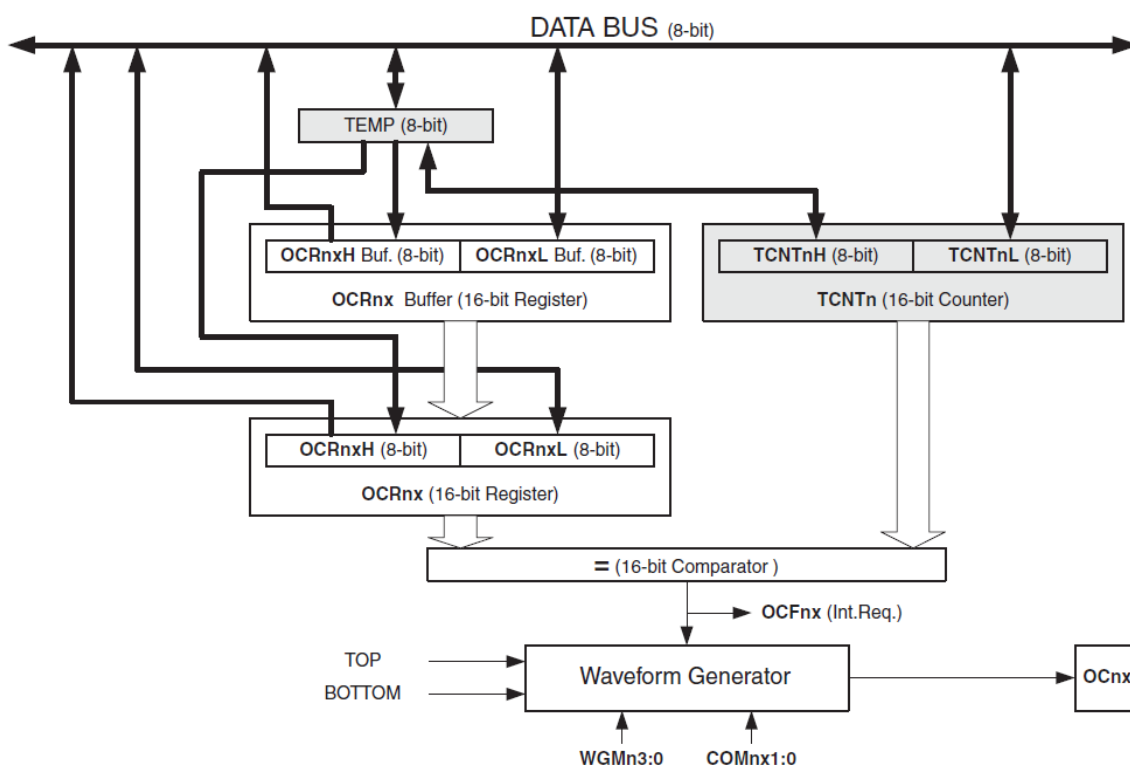


Počítadlá / časovače (T/C), časť 2.

Výstupný komparačný blok - Output Compare Units

16-bitový komparátor priebežne porovnáva obsah **TCNT1** s obsahom *Output Compare Register* (**OCR1x**). Ak sa obsah **TCNT1** = **OCR1x** komparátor ohlásí zhodu. Pri zhode sa v ďalšom **SC** nastaví *Output Compare Flag* (**OCF1x**). Ak je prerušenie povolené (**OCIE1x** =1), **Output Compare Flag** generuje prerušenie. Flag **OCF1x** je nulovaný po vyvolaní prerušenia. *Waveform Generator* využíva signál zhody pri generovaní výstupu v zhode s nastaveným módom: *Waveform Generation mode* (**WGM13:0**) bity and *Compare Output mode* (**COM1x1:0**) bity.



Pre tento obrázok $n = 1$, a x buď **A**, resp. **B**

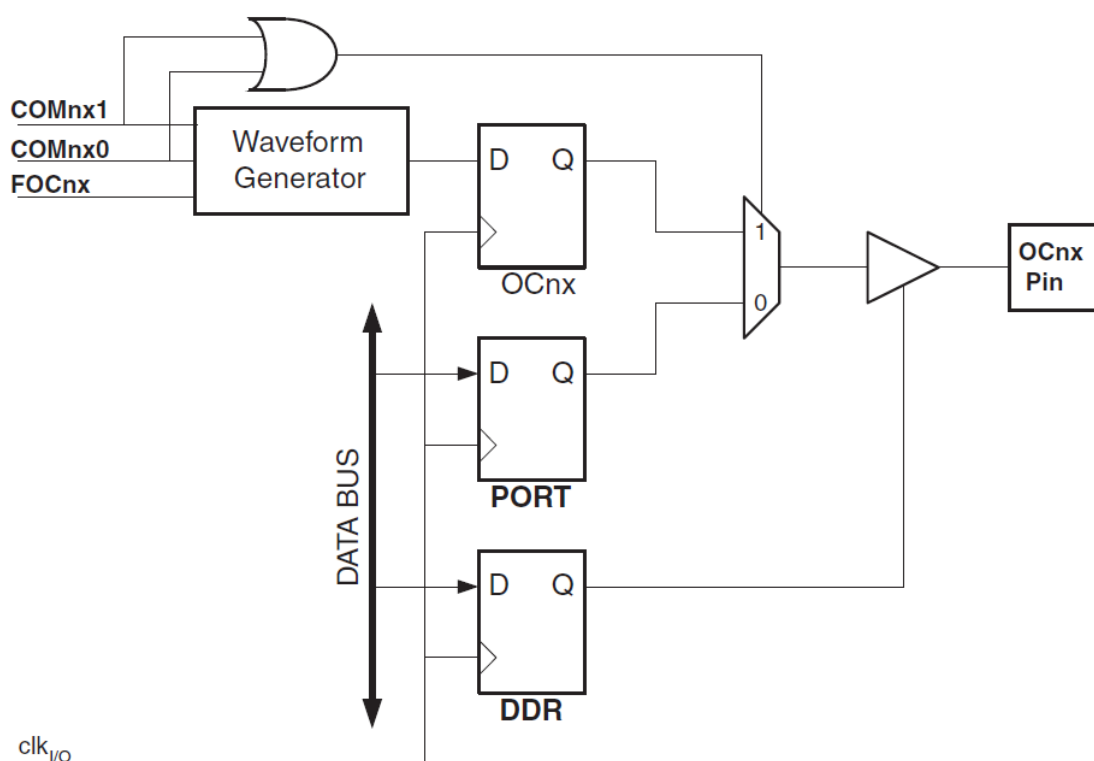
Prístup k registrom **OCR1x** je relatívne zložitý. Niektoré PWM módy umožňujú použitie vyrovnávacej pamäte. CPU prístupuje k **OCR1x** cez buffer register. Ak je vypnuté, CPU prístupuje k registru **OCR1x** priamo.

Využitie výstupnej komparačnej jednotky

Opäť niekoľko zložitejších vecí vynecháme. Poznamenajme, že bity **COM1x1:0** nie sú dvojnásobne ukladané do vyrovnávacej pamäte zmena na týchto bitoch pôsobí okamžite.

Compare Output mode (COM1x1:0) bity majú dve funkcie. *Waveform Generator* používa bity **COM1x1:0** na definovanie stavu výstupu *Output Compare (OC1x)* pri nasledovnej zhode. Po druhé bity **COM1x1:0** riadia zdroj pre pin **OC1x**.

Ak nastavíme režim práce (**COM1x1:0 = 3**) tak, že výstup komparačného registra generátora signálov (*Waveform Generator*) je „smerovaný“ na pin *Output Compare (OC1x)* procesora, potom treba nastaviť pin v **DDR** registri ako výstupný.

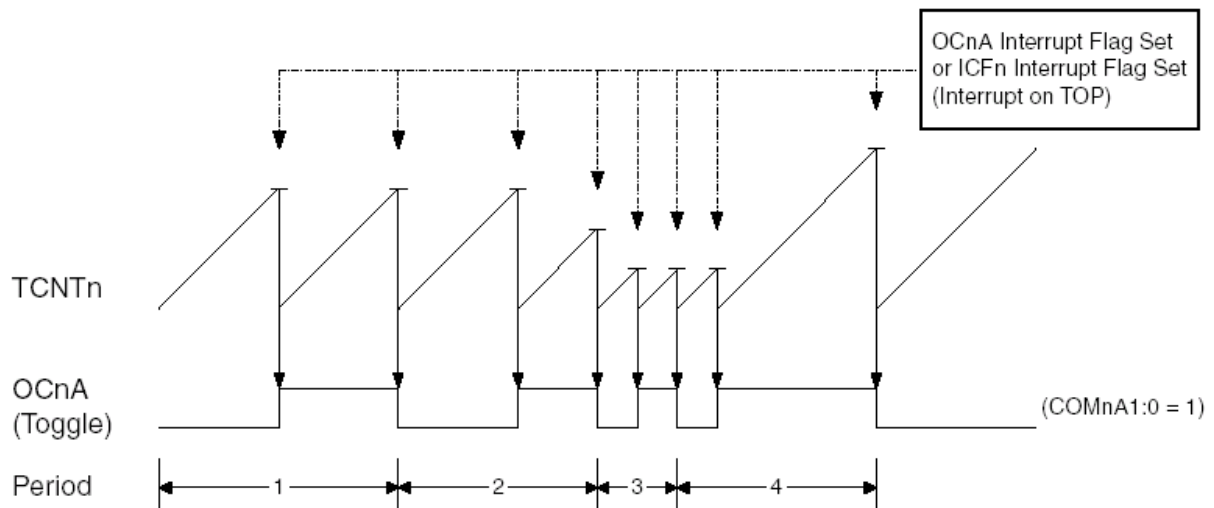


Normal Mode (= 0)

Normal mode (WGM13:0 = 0). V tomto móde sa vždy počíta smerom hore. Nenuluje sa. Jednoducho sa pretočí z **MAX** ($0xFFFF$) na **BOTTOM** ($0x0000$). V tomto móde sa bit *Timer/Counter Overflow Flag (TOV1)* nastaví v tom istom **SC** ako sa **TCNT1** vynuluje.

Clear Timer on Compare Match (CTC) Mode (= 4, 12)

V móde *Clear Timer on Compare* alebo **CTC** móde (**WGM13:0 = 4** or **12**), sa využívajú registre **OCR1A** alebo **ICR1** na nastavenie rozlíšenia. V móde **CTC** je čítač nulovaný keď nastane zhoda obsahu čítača (**TCNT1**) buď s registrom **OCR1A** (**WGM13:0 = 4**) alebo s registrom **ICR1** (**WGM13:0 = 12**). Registre **OCR1A** alebo **ICR1** určujú vrch – top hodnotu pre čítač.



Obdobný obr. platí aj pre ICR1.

Frekvencia opakovania je daná vzťahom

$$f_{OCnA} = \frac{f_{clk_IO}}{2 \cdot N \cdot (1 + OCRnA)}$$

Kde $N = 1, 8, 64, 256, 1024$.

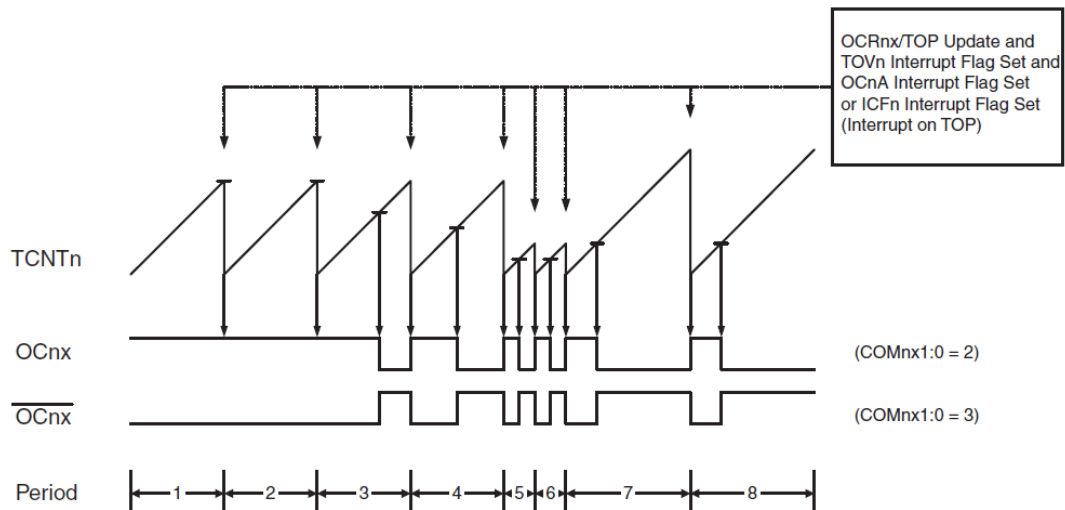
Fast PWM Mode

Rýchly PWM mód (**WGM13:0 = 5,6,7,14, alebo 15**) poskytuje PWM s vysokou frekvenciou. Tento mód sa líši od iných PWM módov jednoduchou pilou.

Počítadlo počíta od **BOTTOM** do **TOP** a opäť sa reštartne na **BOTTOM**. V móde *non-inverting Compare Output*, výstup Output Compare (**OC1x**) sa nuluje pri zhode medzi **TCNT1** a **OCR1x**, a nastaví na hodnote **BOTTOM**. Pri inverznom režime je to naopak. Tieto módy sa používajú hlavne v taských aplikáciách ako sú: usmerňovače, D/A prevodníky, ...

Rozlíšenie pre f-PWM mód je 8-,9-, 10-bitov alebo sa dá nastaviť pomocou **ICR1** resp. **OCR1A**. PWM rozlíšenie v bitoch je dané vzťahom:

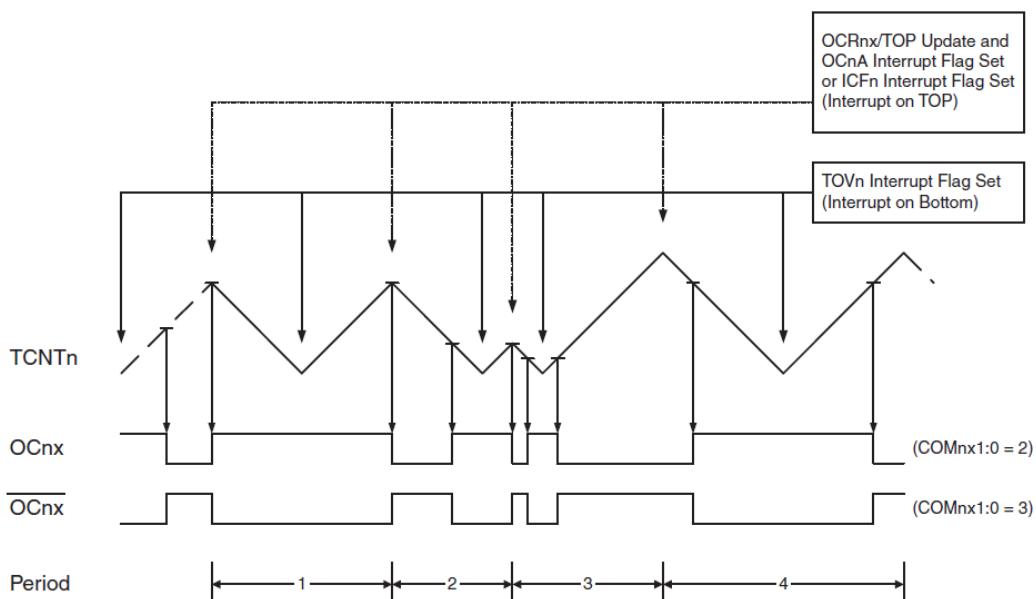
$$R_{FPWM} = \frac{\log(TOP + 1)}{\log(2)}$$



Phase Correct PWM Mode

Tento mód podobne ako **Phase and Frequency Correct PWM Mode** je vytvorený pomocou duálneho sklonu. Počítadlo počíta smerom hore (od **BOTTOM** =0x0000 do **TOP**) a potom smerom dole (od **TOP** do **BOTTOM**). V neinvertujúcom režime sa výstup (**OC1x**) nuluje pri zhode **TCNT1** a **OCR1x** a súčasne ak počítadlo počíta smerom hore. Nastaví sa ak počíta smerom dole a nastane zhoda.

Rozlíšenie pre tento mód je 8-,9-, 10-bitov alebo sa dá nastaviť pomocou **ICR1** resp. **OCR1A**.



Tab. Možné módy generovania signálu na výstupe.

Mode	WGM13	WGM12	WGM11	WGM10	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

Príklad 1.:

Nastavme T/C1 tak, aby generoval

- časové značky 50 ms bez prerušenia
- časové značky 50 ms s prerušením

Prvá vec ktorá nás zaujíma, je frekvencia oscilátora: $f_{OSC} = 8\,000\,000$ Hz.

Jeden SC trvá 0,125us. Počítadlo v normálnom móde počíta smerom nahor. Nato aby sme napočítali 50ms oneskorenie potrebujeme

$$0,05\text{ s} / 0,000\,000\,125\text{ s} = 400\,000\text{ SC.}$$

To je viac ako sme schopný načítať 16 bitovým počítadlom.

To znamená, treba použiť *Preddelič*.

Ak budeme načítavať $\text{clk}_{IO}/8$ treba načítať 50 000 impulzov. Počiatočná hodnota počítadla je 0x3CB0.

Toto číslo sme získali takto:

$$0x10000 - 50\,000_{10} = 0x3CB0.$$

Pre úlohu s prerušením by to teoreticky postačovalo. Príznak by testoval hardware.

Treba si položiť otázku: Ako presne chceme toto oneskorenie generovať? Alebo ináč: Čo sme schopný programovo vykonať za 8SC? Presne toľko máme na odchytenie príznaku pretečenia, znovu prednastavenie na hodnotu 0x3CB0. Takýto postup by viedol zrejme ku kumulujúcej sa chybe.

Ak budeme načítavať $\text{clk}_{IO}/64$ treba načítať 6250 impulzov. A na samotnú „obsľuhu“ prednastavenia, ... budeme mať 64 SC. Možno to postačí.

Otázka: Prečo sme nenavrhlí deleno 256? (Odpoveď: $400\,000/256$ nie je celé číslo.) Načo sme ešte pozabudli. (Preddelič je voľne bežiacie počítadlo => nie je zosynchronizované.)

Vašou úlohou nájsť medzi módmi T/C1 taký, ktorý bude využívať prerušenie a bude sa prednastavovať pri pretečení. Napíšte nastavenie registrov.

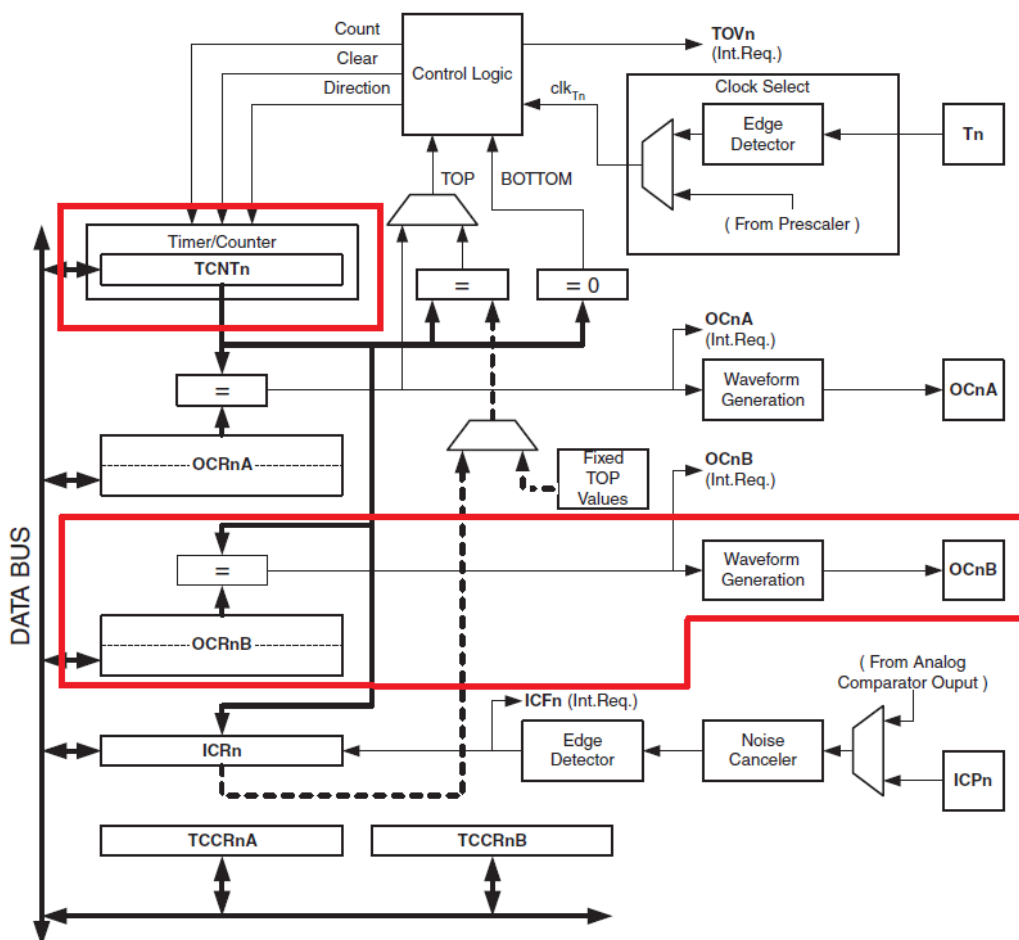
Úloha: Navrhnite iný spôsob riešenia tejto úlohy.

Príklad 2.:

nastaviť frekvenciu opakovania PWM signálu tak, aby $T_{OP} = 2 [ms]$, Tejto hodnote odpovedá frekvencia opakovania $f_{OP} = 500 Hz$.

!!! Nesmieme zabudnúť, že toto celé je spriahnuté s f_{OSC} !!!

Na obrázku je bloková schéma T/C1



Fast PWM Mode

Rýchly PWM mód má 8-bitové rozlíšenie (**WGM13:0 = 5**). Poskytuje PWM s vysokou frekvenciou. Tento mód sa líši od iných PWM módov jednoduchou pílou.

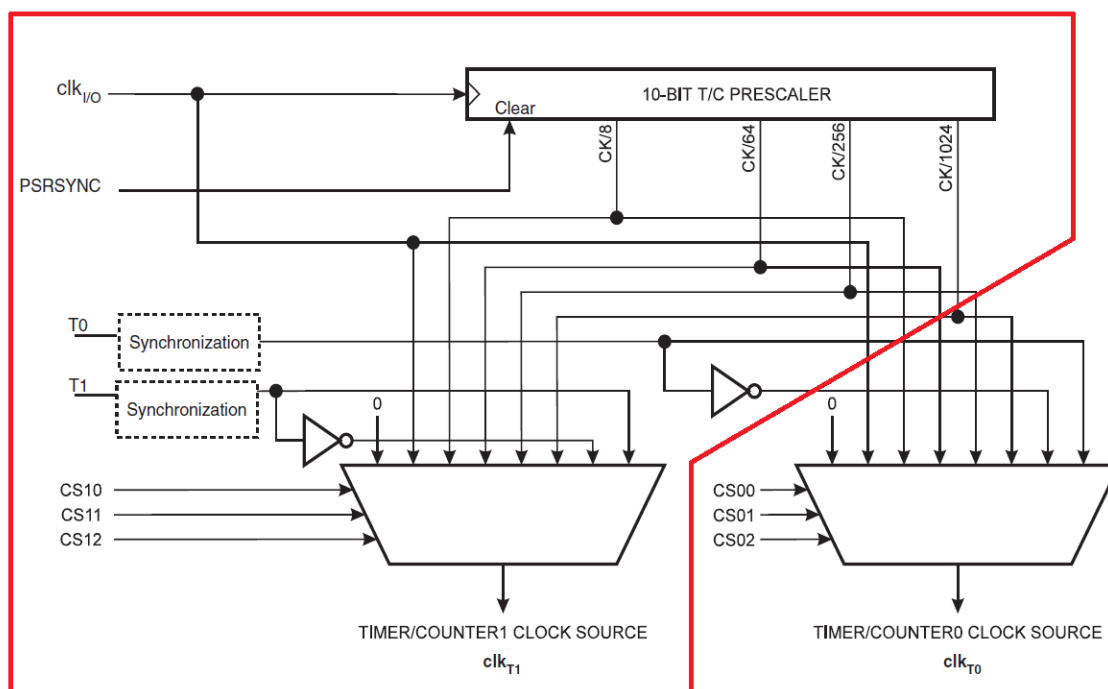
Počítadlo počíta od **BOTTOM** do **TOP** (**TOP = 0xFF**) Vstave **TOP+1=0x100** sa jednotka v ôsmom bite použije na generovanie prerušenia a **TOP= BOTTOM**,

To znamená, že počítadlo sa opäť reštartuje na **BOTTOM**. V móde *non-inverting Compare Output*, výstup Output Compare (**OC1B**) sa nastaví pri zhode medzi **TCNT1** a **OCR1B**, a vynuluje na hodnote **TOP**.

Tab. Možné módy generovania signálu na výstupe.

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

Nastavenie preddeľiča pre T/C1



Každé počítadlo ma 8 rôznych vstupov.

Na vstup počítadla je privedený signál odvodený od pracovnej frekvencie procesora $f_{CLK_{I/O}}$ alebo jeden z vývodov preddeľiča $f_{CLK_{I/O}}/8, f_{CLK_{I/O}}/64, f_{CLK_{I/O}}/256, f_{CLK_{I/O}}/1024$.

F-PWM počítadlo inkrementuje pri nábežnej hrane.

Predelič je voľne bežiaci. Jeho prednastavenie je *nezávislé* na pripojenom počítadle. Ak počítadlo pripojíme k preddeliču, môže načítať 1 až N+1 systémových hodinových signálov, kde N je deliaci pomer preddeliča (8, 64, 256, or 1024), než čítač prvý krát inkrementuje svoj obsah.

Bit	7	6	5	4	3	2	1	0	
	TSM	-	-	-	-	-	PSRASY	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Zápis 1 do **PSRSYNC** vynuluje preddelič pre **T/C1** aj **T/C0**. Tento byt sa vynuluje po vykonaní operácie.

Prednastavenie preddeliča pre **T/C1** sa vykoná nastavením bitov **CS12:0** v registri **TCCR1B**.

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
				0	1	0	1	1	
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Nasledujúca tabuľka nám udáva hodnoty približné T_{OP} pre $f_{osc} = 8\text{MHz}$ a $f_{osc} = 18,432\text{ MHz}$. Predpokladáme mód PWM „klasický“ 8-bitový (256)

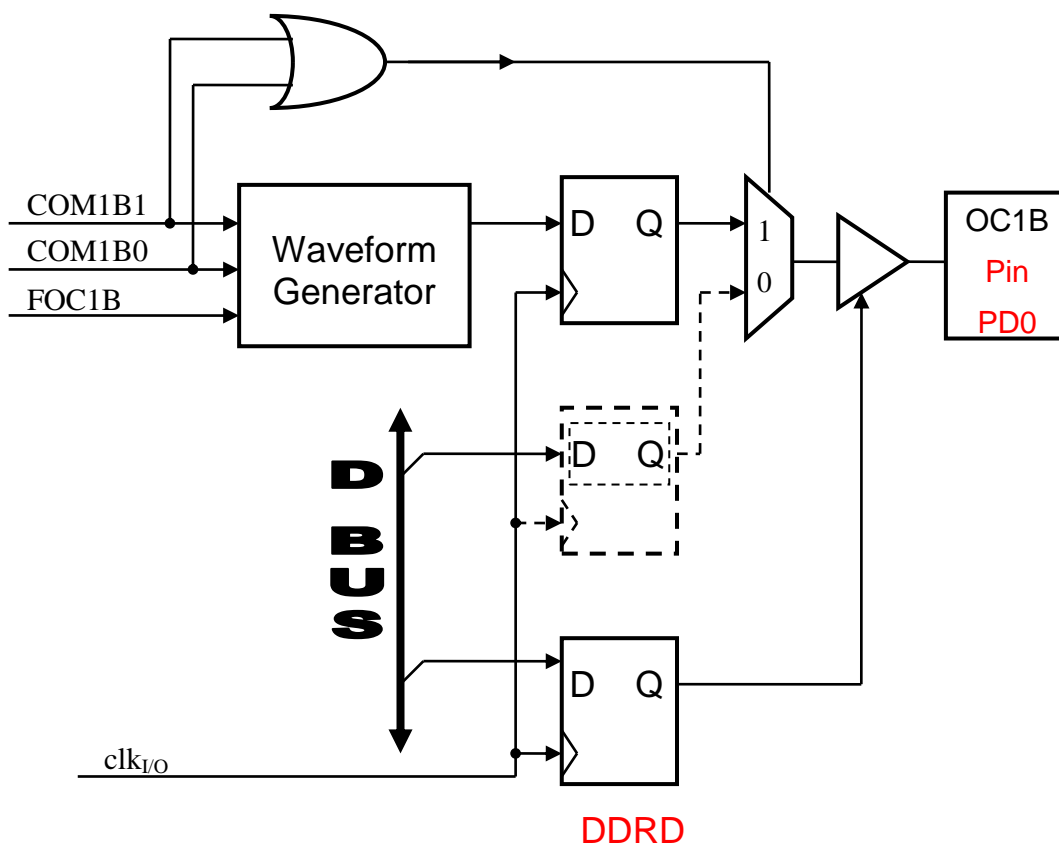
Nastavenie preddeliča pre **T/C1**.

CS12	CS11	CS10		$f_{osc} = 8\text{MHz}$	$f_{osc} = 18,432\text{ MHz}$
				$T_{OP} =$	$T_{OP} =$
0	0	1	$\text{clk}_{i/o}$ (No prescaler)	32 μs	14 μs
0	1	0	$\text{clk}_{i/o}/8$ (From prescaler)	256 μs	111 μs
0	1	1	$\text{clk}_{i/o}/64$ (From prescaler)	2 ms	0,9 ms
1	0	0	$\text{clk}_{i/o}/256$ (From prescaler)	16 ms	7.1 ms
1	0	1	$\text{clk}_{i/o}/1024$ (From prescaler)		

TCCR1A – počítadlo/časovač riadiaci register,

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0			WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Ak nastavíme režim práce (**COM1B1:0 = 3**) tak, že výstup komparačného registra generátora signálov (*Waveform Generator*) je „smerovaný“ na pin *Output Compare (OC1B)* procesora, potom treba nastaviť pin v **DDR** registri ako výstupný.



Port D Data Direction

Bit	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

LITERATÚRA: ATmega 328P