

Gingl Zoltán, 2020, Szeged

Mikrovezérlők Alkalmazástechnikája

Működést támogató perifériák és használatuk (C8051F410)

Oscillátor

Oscillátor

- ▶ A processzornak ütemjel (órajel) szükséges
- ▶ Számos periféria órajelét is adja
- ▶ Rendkívül sokféle beállítás és frekvencia lehet
- ▶ DC-100MHz, pl. 32768Hz (órakvarc)
- ▶ Fogyasztás/sebesség hangolható

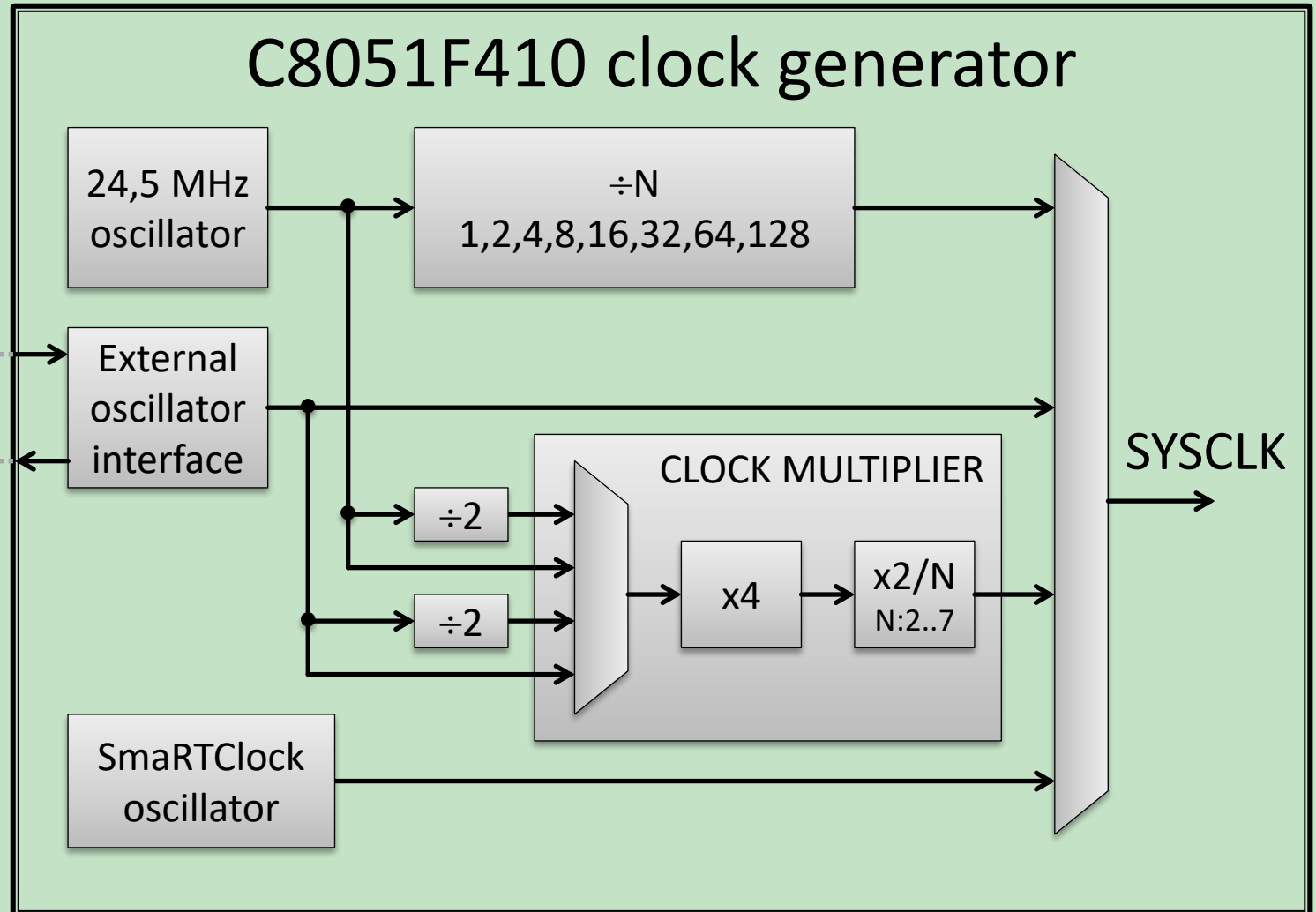
C8051F410 mikrovezérlő

Órajel	Sebesség	Áramfelvétel	Üzemidő (CR2032, 200mAh)
32768 Hz	32768 IPS	17 uA	490 nap
1 MHz	1 MIPS	430 uA	19 nap
50 MHz	50 MIPS	13,5 mA	Az elem áramkorlátja felett

Oscillátor ▶ Alapok

- ▶ Az oszcillátor analóg periféria, érzékeny
- ▶ Ha külső elemeket (pl. kvarckristály) használ, a kivezetéseket analóg módba kell konfigurálni
- ▶ Az oszcillátor kimenete digitális
 - ▶ Ez a mikrovezérlőn belüli rész, de az áramkörből opcionálisan kivezethető az órajel

Oszillátor ▶ Oszillátor blokkvázlat

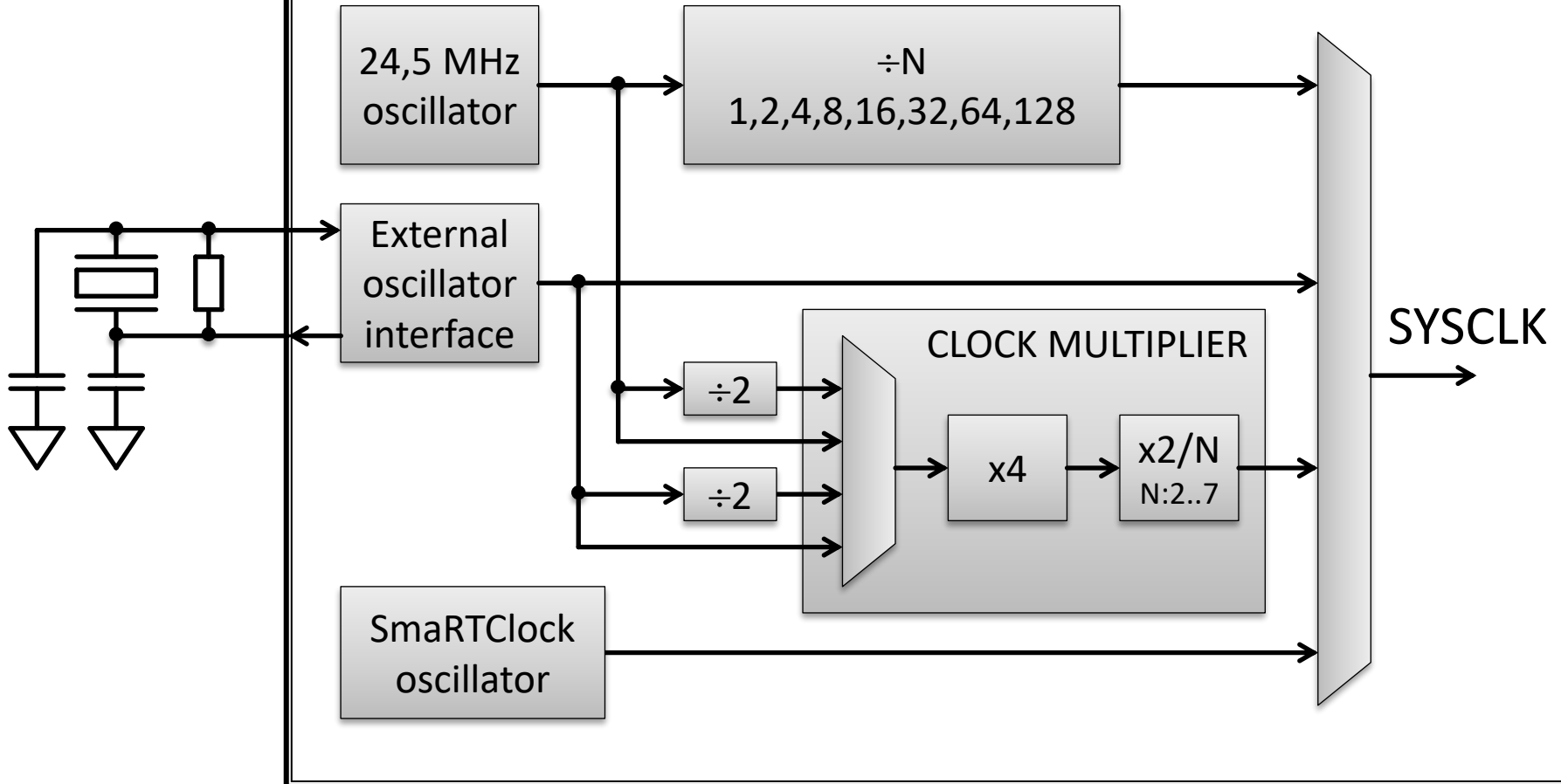


Oscillátor ▶ belső

- ▶ Nem szükséges külső alkatrész
- ▶ Egy ismert frekvenciájú oszcillátor van beépítve
- ▶ Pontosság: $24,5 \pm 0,5 \text{MHz}$ (2%)
- ▶ Programozható leosztással előállítható a rendszerórajel: frekvencia: $24,5 \text{MHz} / 2^i$, $i=0..7$
- ▶ A finomhangolás programozható (kalibrálható)
- ▶ Gyors feléledésű

Oscillátor ▶ külső ▶ kvarckristály

C8051F410 clock generator



Oscillátor ▶ külső ▶ kvarckristály

- ▶ Legpontosabb (20ppm-100ppm)
 - ▶ Pontos időzítésekhez (UART, mintavételezés)
 - ▶ Pontos időmérésekhez
 - ▶ RTC – real time clock alkalmazásokhoz
- ▶ **Stabilizálódási idő (~1ms)**
 - ▶ ha használjuk, várni kell a feléledésre!
- ▶ Gondos tervezést igényel
 - ▶ Rövid vezetékek
 - ▶ „loading capacitor” értékek, CL
 - ▶ Érzékeny a nemideális áramköri részekre, zavarokra

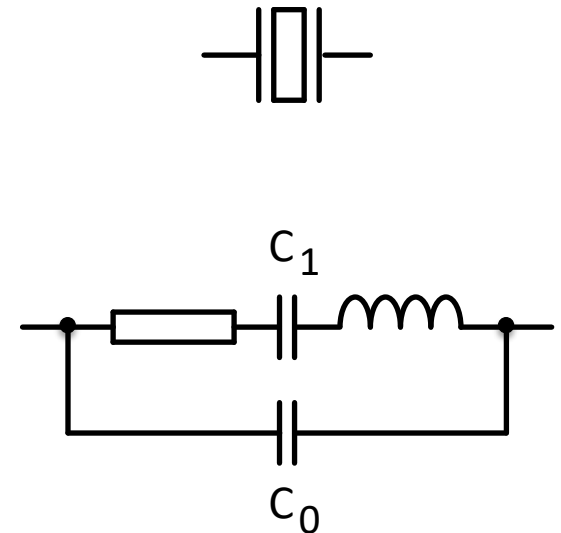
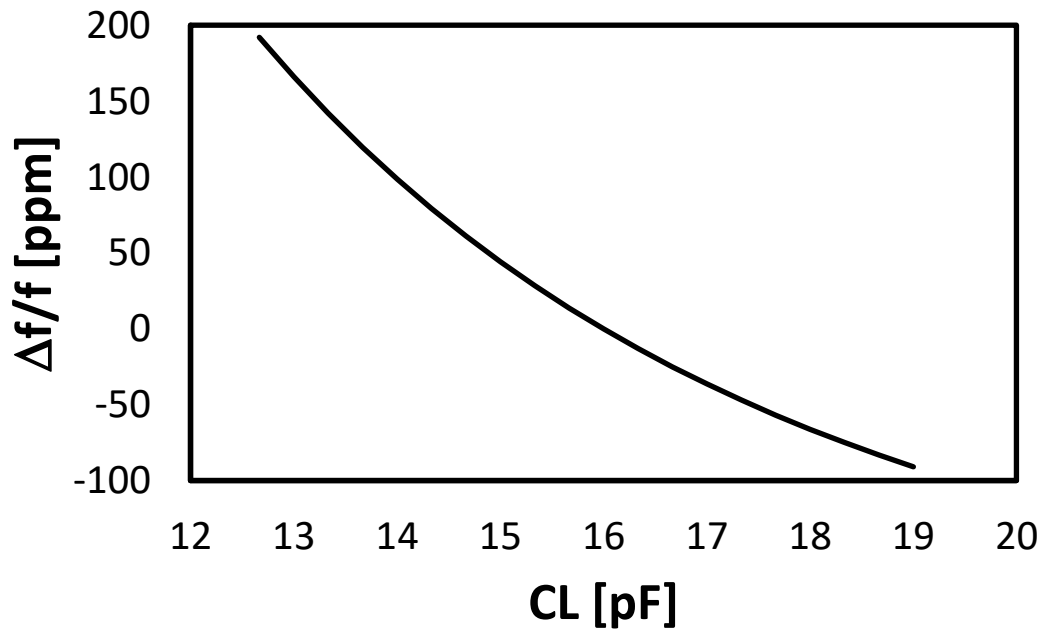
Oscillátor ▶ külső ▶ kvarckristály

$$\frac{\Delta f}{f} = -\frac{C_1}{(C_0 + C_L)^2} \Delta C_L$$

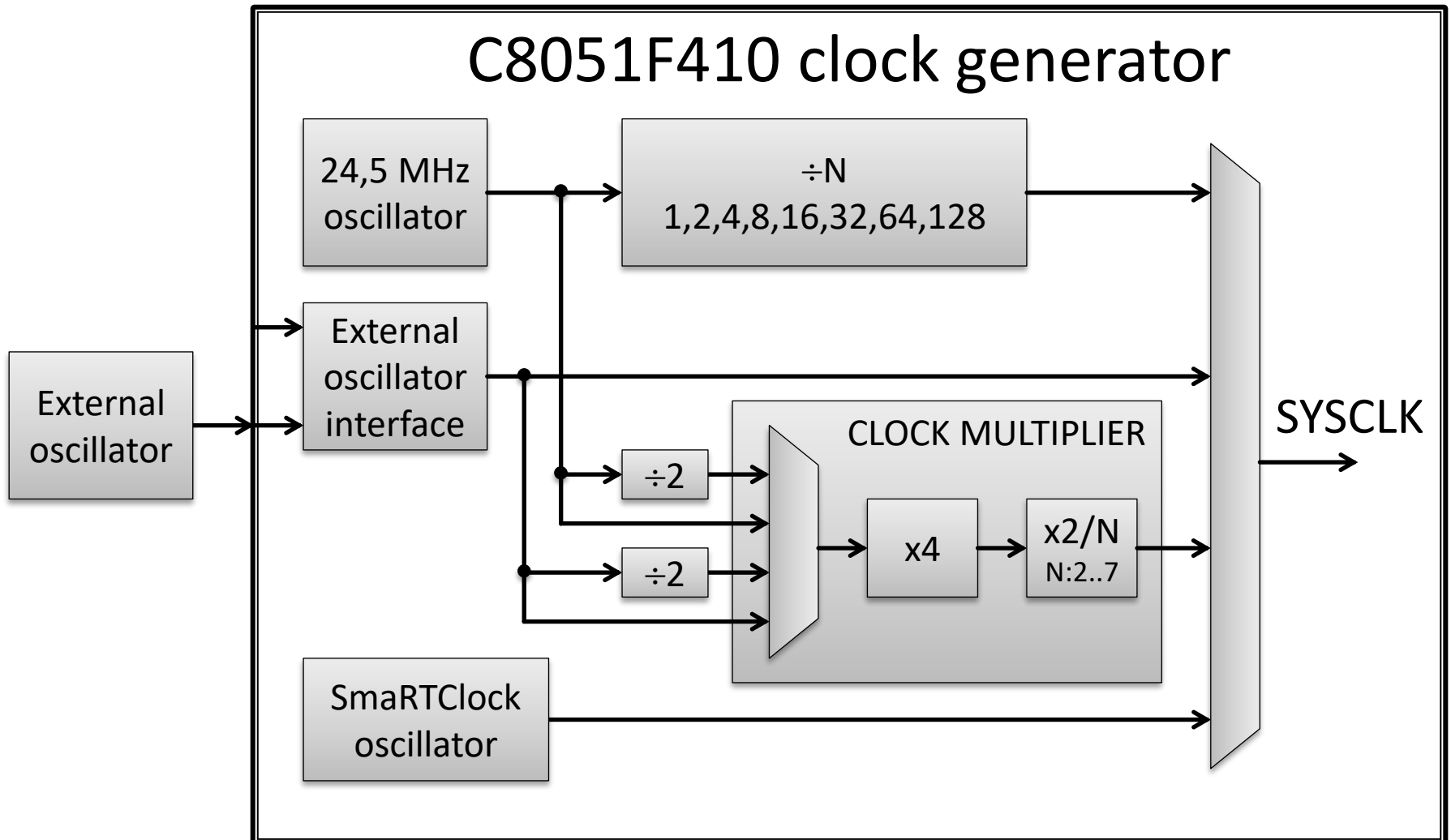
$$C_0 = 4 \text{ pF}$$

$$C_1 = 16 \text{ fF}$$

$$C_L = 16 \text{ pF}$$



Oszillátor ▶ külső ▶ külső oszcillátor áramkör



Oscillátor ▶ külső ▶ külső oszcillátor áramkör

- ▶ Komplettsz oszcillátor IC
- ▶ Logikai kimenőjel (TTL/CMOS kompatibilis)
- ▶ Kvarc vagy félvezető alapú
- ▶ Pontos, gyári specifikálás
- ▶ Kész megoldás, megbízható
- ▶ Több processzor vagy periféria azonos órajelet használhat

Oscillátor ▶ Megjegyzések

- ▶ A belső oszcillátor gyorsan feléled, RESET után alapértelmezett
- ▶ Kvarc és PLL ms körüli feléledési idővel
- ▶ Ha az oszcillátorok működnek, bármelyikre válthatunk futás közben
- ▶ Lehet rövid időre átkapcsolni a gyors működésre
- ▶ Vigyázni kell, hogy a váltás ne befolyásoljon más perifériát (pl. UART)
- ▶ **Missing clock detector** RESET-et adhat

Oscillátor ▶ Programozás ▶ Belső

- ▶ Belső (24,5MHz±0,5MHz)
- ▶ f:24,5MHz/N, N=1,2,4,8,16,32,64,128
- ▶ RESET érték: N=128, f=191406Hz

```
// example  
void Oscillator_Init()  
{  
    OSCICN    = 0x82;  
}
```

$$f = \frac{24,5\text{MHz}}{N}$$

$$N = \frac{128}{2^{\text{OSCICN}\&7}}$$

$$f = \frac{24,5\text{MHz}}{128} \cdot 2^{\text{OSCICN}\&7}$$

Fogyasztásmenedzsment

Fogyasztáscsökkentési módok

Mód	leírás	kilépés	áramfelvétel (@ 1MHz)
NORMAL	a processzor fut	-	300uA
IDLE	PCON.0 beállításával CPU nem fut perifériák működnek, oszillátor fut memória, regiszterek állapota	RESET WDT engedélyezett IRQ	150uA
STOP	PCON.1 beállításával CPU, perifériák, belső oszillátor mind áll Analóg perifériákat a programozó állíthatja le	csak RESET	0,15uA
SUSPEND	belső oszillátor megállítása	Port 0 or Port 1 Match Comparator0/1 is logic 0. smaRTClock Oscillator Fail smaRTClock Alarm	0,15uA

Az alacsony fogyasztású módok használata

- ▶ Egy utasítással kapcsolhatjuk be
- ▶ SFR regiszterbe kell írni:
 - ▶ Idle mód: PCON 0. bitjébe 1
 - ▶ Stop mód: PCON 1. bitjébe 1
 - ▶ Suspend mód: OSCICN.5 bitjébe 1
- ▶ A processzor a következő utasítást már nem hajtja végre!
- ▶ Csak az előzőkben felsorolt események válthatják ki a folytatást, semmilyen kód nem fut addig

Fogyasztáscsökkentési módok – példa

- ▶ Várakozás adott ideig – idle módban
- ▶ Az engedélyezett megszakítás felkelti a processzort
- ▶ Más megszakítás megzavarhatja! Hogy kerülhetjük el?

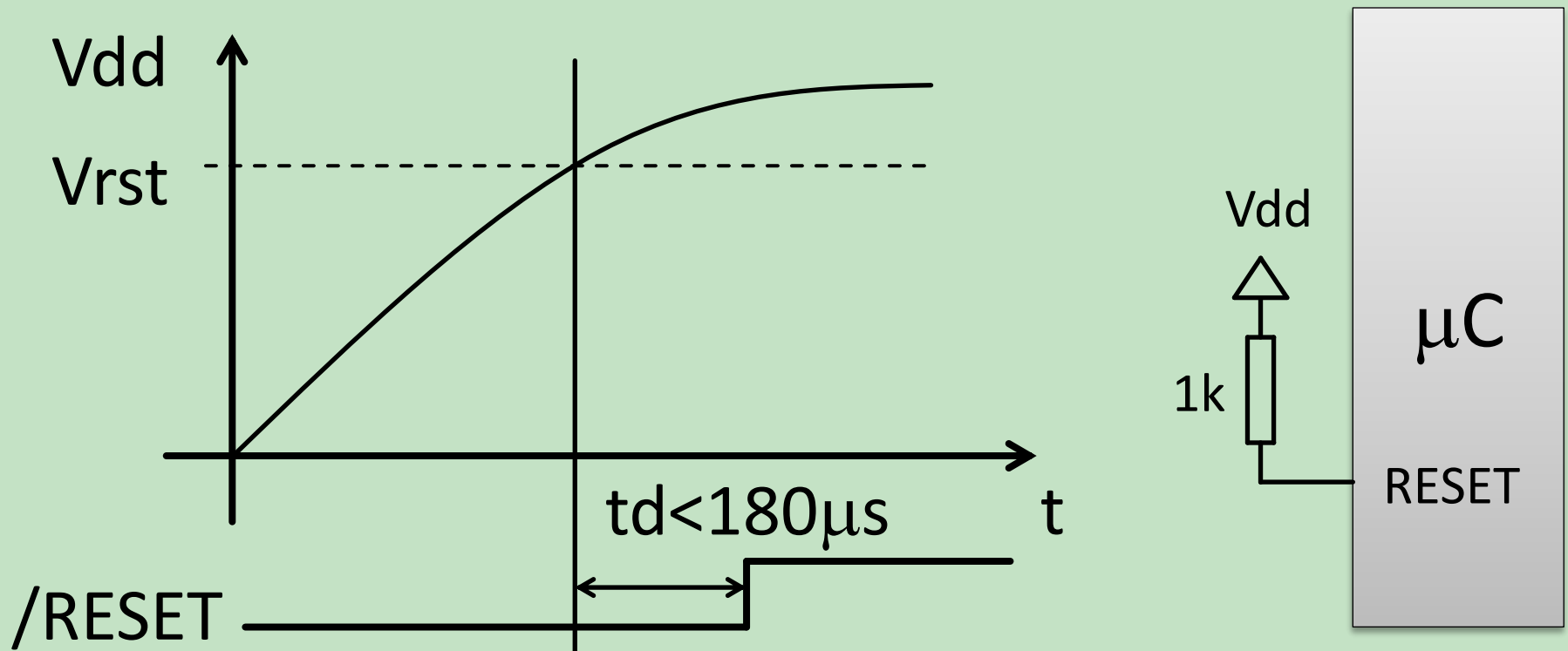
```
TMR2H=-Steps >> 8; // 65536-Steps
TMR2L=-Steps;
TF2H=0; // clear timer 2 flag
EA=0xA0; // enable interrupts
TR2=1; // run timer 2
PCON|=1; // set idle mode
```

Power-on reset (POR)

Tápfeszültség monitor

Power-on reset generátor

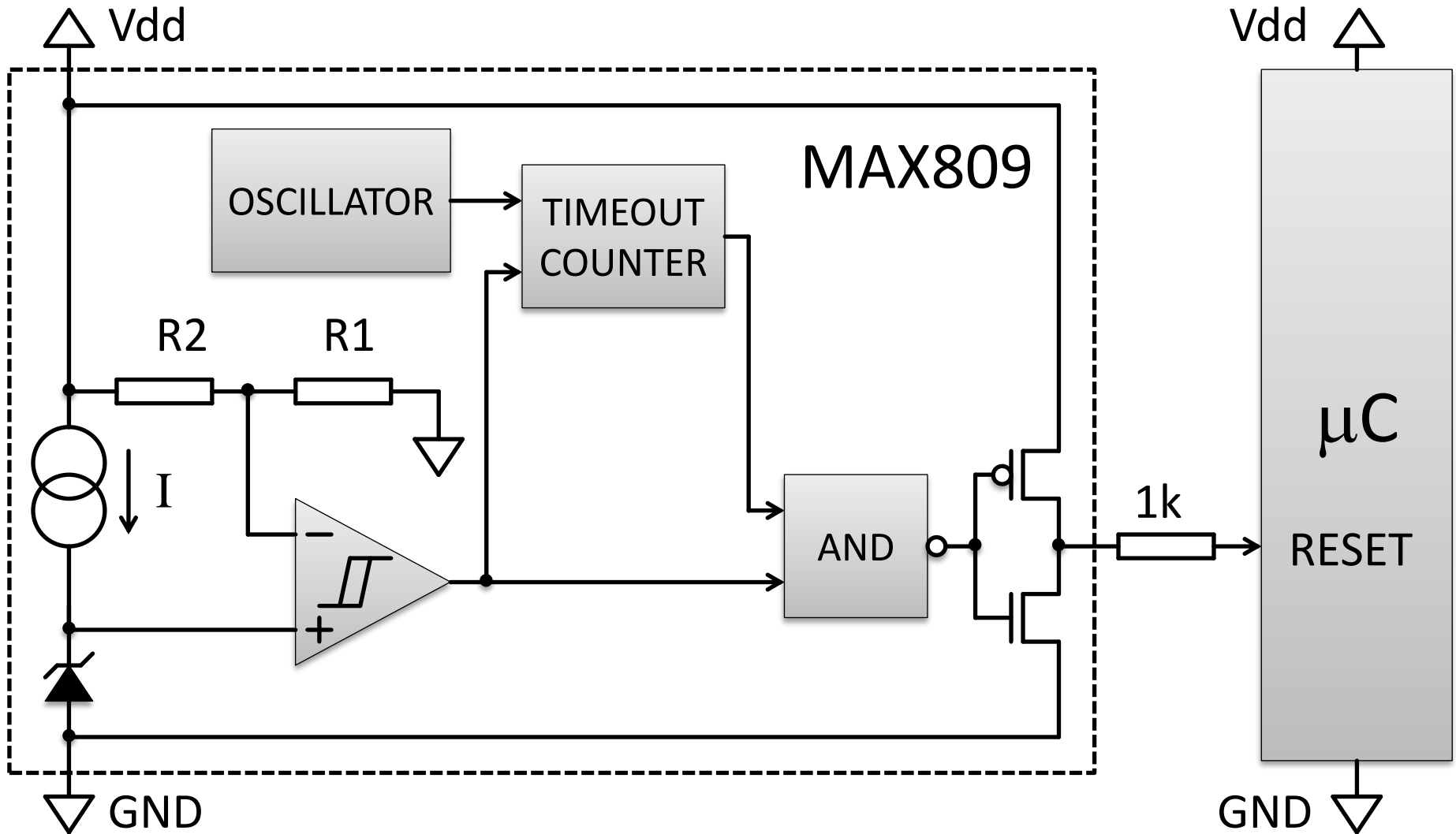
- ▶ A táp felfutása adott ideig tart
- ▶ A megfelelő V_{rst} szint elérésekor egy rövid (t_d) még RESET állapotban tartja a processzort



Külső reset generátor

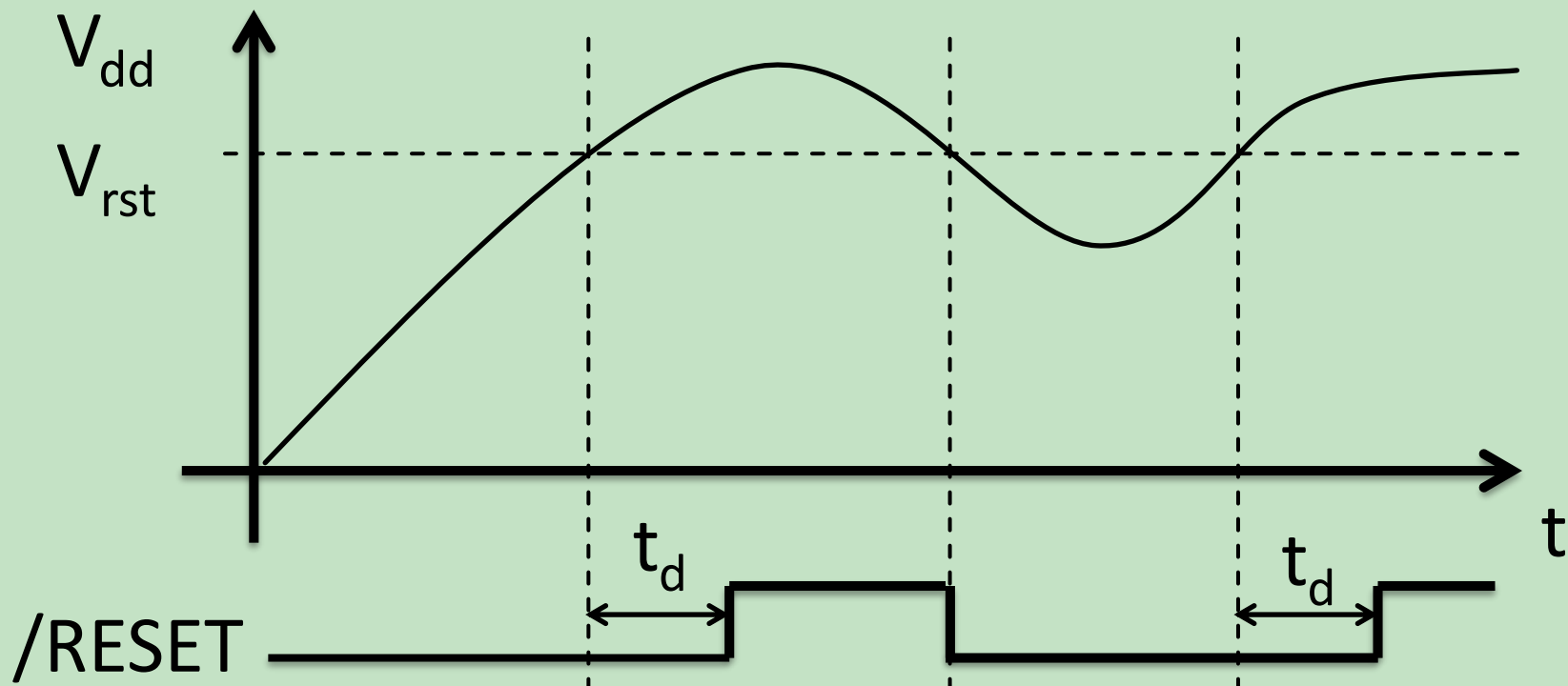
- ▶ Ha a táp felfutási ideje $>1\text{ms}$, akkor
 - ▶ külső RESET áramkör szükséges
- ▶ A legbiztonságosabb megoldás a külső áramkör
 - ▶ kritikus alkalmazásokban mindig célszerű
- ▶ Választék:
 - ▶ aktív alacsony, aktív magas
 - ▶ küszöbfeszültség számos tápfeszültséghez
 - ▶ reset időtartam (1ms..200ms)
- ▶ Egy ellenálláson keresztül kell bekötni (a processzor is vezérelheti a RESET jelet)
- ▶ Minimum 20us ideig aktív legyen

Külső resetgenerátor ▶ példa



Tápfeszültségmonitor

- ▶ Power supply monitor
- ▶ más néven: brown-out detector



Tápfeszültségmonitor

- ▶ Hasonló a működés a power-on resethez
- ▶ Ha a tápfeszültség egy szint alá esik, resetet generál az áramkör

Feszültségstabilizátor

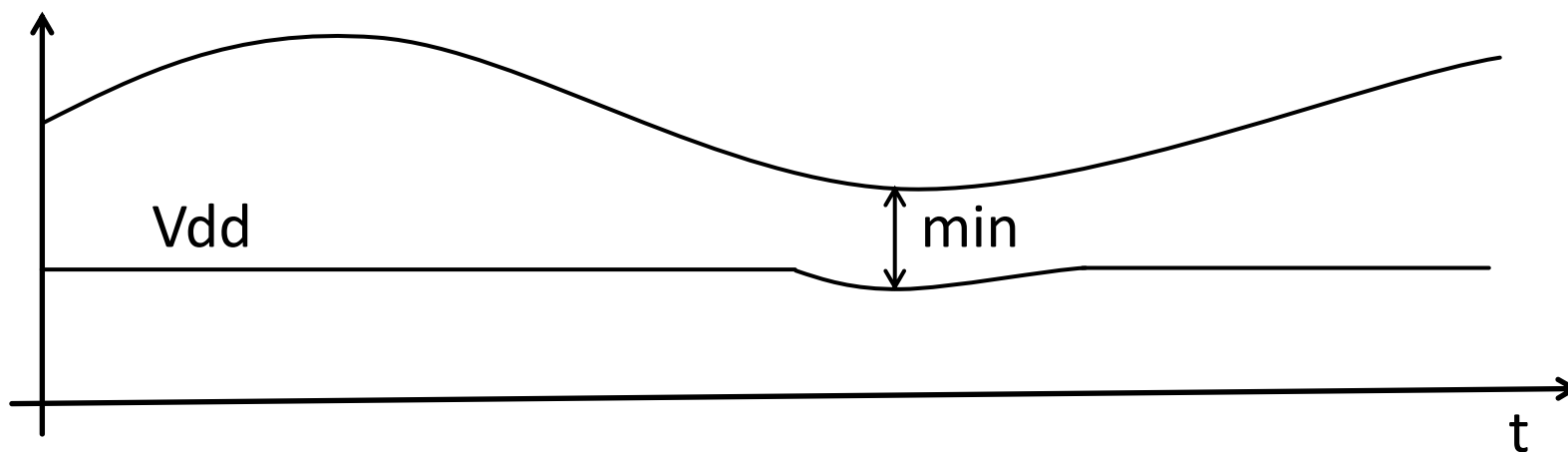
Feszültségstabilizátor

- ▶ A mag gyakran kis tápfeszültséget igényel
- ▶ A külső perifériák tápfeszültsége lehet más
- ▶ Ne kelljen külső stabilizátor a kisebb tápfeszültséghez
- ▶ A stabilizálást is javíthatjuk, de nem ez a fő cél
- ▶ Low-dropout regulator – kis feszültségesés

Feszültségstabilizátor ▶ Adatlap

parameter	conditions	min	typ	max	units
Input Voltage Range		*	-	5,25	V
Load Current		-	-	50	mA
Load Regulation		-	7	15	mV/mA
Output Voltage (VDD)	REG0MD=0	2,0	2,1	2,25	V
Output Voltage (VDD)	REG0MD=1	2,35	2,5	2,55	V

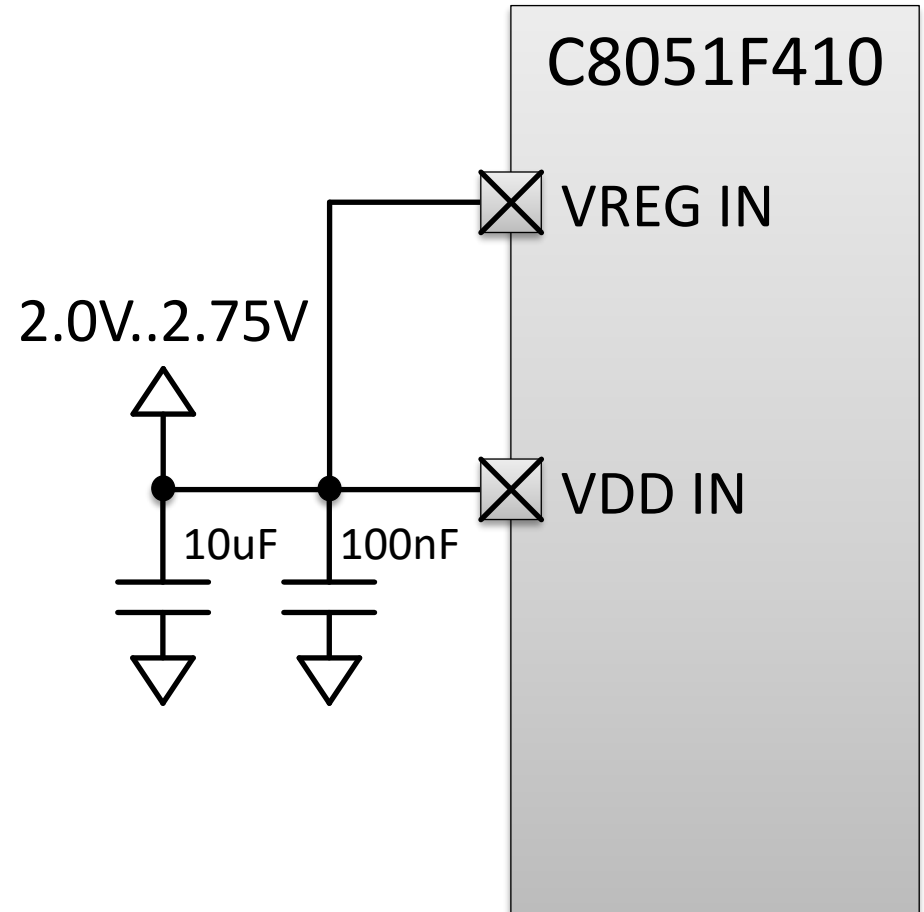
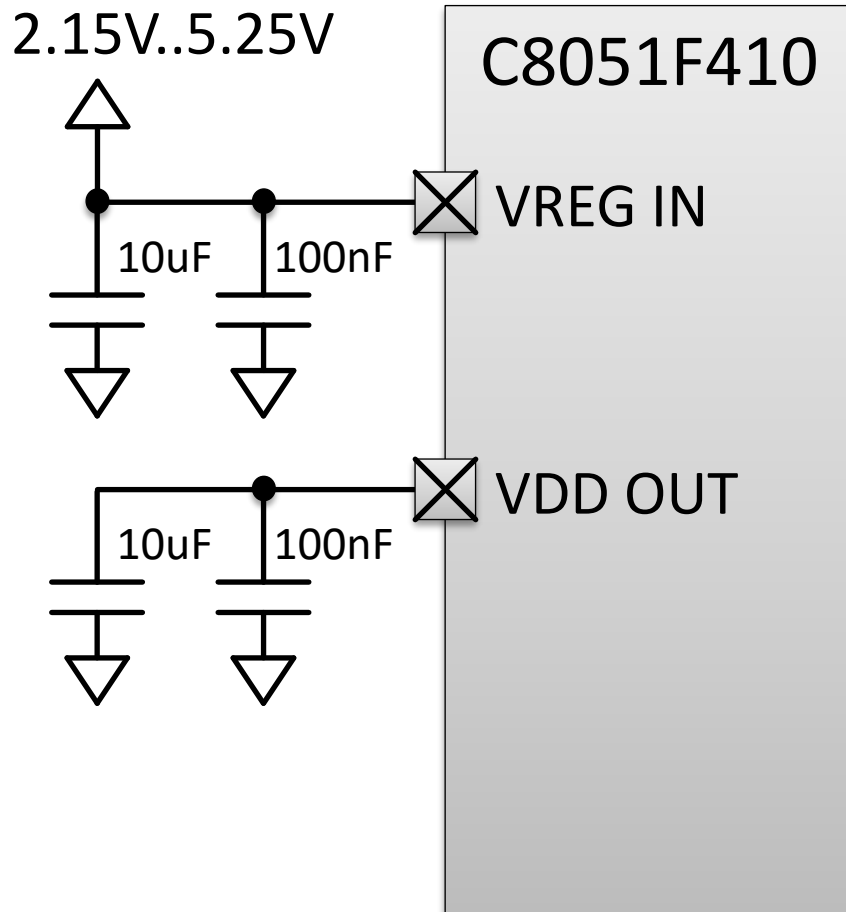
*Output Voltage = Nominal Output Voltage – (Load Regulation x Load Current)



Feszültségstabilizátor ▶ Adatlap

Paraméter	leírás
Input Voltage Range	A bemeneti feszültség értéktartománya A minimális értéket az határozza meg, mekkora a kimeneti feszültség (VDD). A terheléstől függ, hogy a bemeneti feszültségnek mennyivel kell nagyobbnak lennie. Ha ezt nem tartjuk be, a kimeneti feszültség is esik
Load Current	Terhelő áram, amit a processzor és perifériák összesen igényelnek Jelentősen függ az rendszerórajel frekvenciájától, közel lineárisan
Load Regulation	Azt adja meg, mennyit esik a kimeneti feszültség a terhelőáram függvényeként. Lineárisnak tekinthető.
Output Voltage (VDD)	A stabilizátor kimeneti feszültsége. A processzor és belső perifériák mellett külső áramkörök is használhatják, ki van vezetve

Feszültségstabilizátor ▶ Kapcsolások



$$V_{REGIN} - V_{DD} > I_{DD} * 15 \text{mV/mA}$$

Feszültségstabilizátor ▶ Mire figyeljünk?

- ▶ *A bemeneti feszültségkorlátot ne lépjük túl*
- ▶ *Csatolásmentesítő kondenzátorok kellenek*
 - ▶ *ki- és bemeneteken*
 - ▶ *100nF és 10uF párhuzamosan kötve a GND-re*
- ▶ *A tápáram korlátos*
 - ▶ *A kimenetek összárámára figyeljünk*
 - ▶ *Ha külső áramköröket táplálunk, ezek áramfelvételére is figyelni kell*

Watchdog timer

Watchdog timer (WDT)

- ▶ RESET generálása, ha a program hibásan működik
 - ▶ végtelen ciklus
 - ▶ hibás kódra futás
 - ▶ verem felborulás
 - ▶ elektromágneses zavar
 - ▶ ...
- ▶ **RESET után automatikusan bekapcsolva**
 - ▶ ha nem használjuk (teszteléskor), azonnal ki kell kapcsolni
 - ▶ végleges kódnál használni kell

Watchdog timer (WDT)

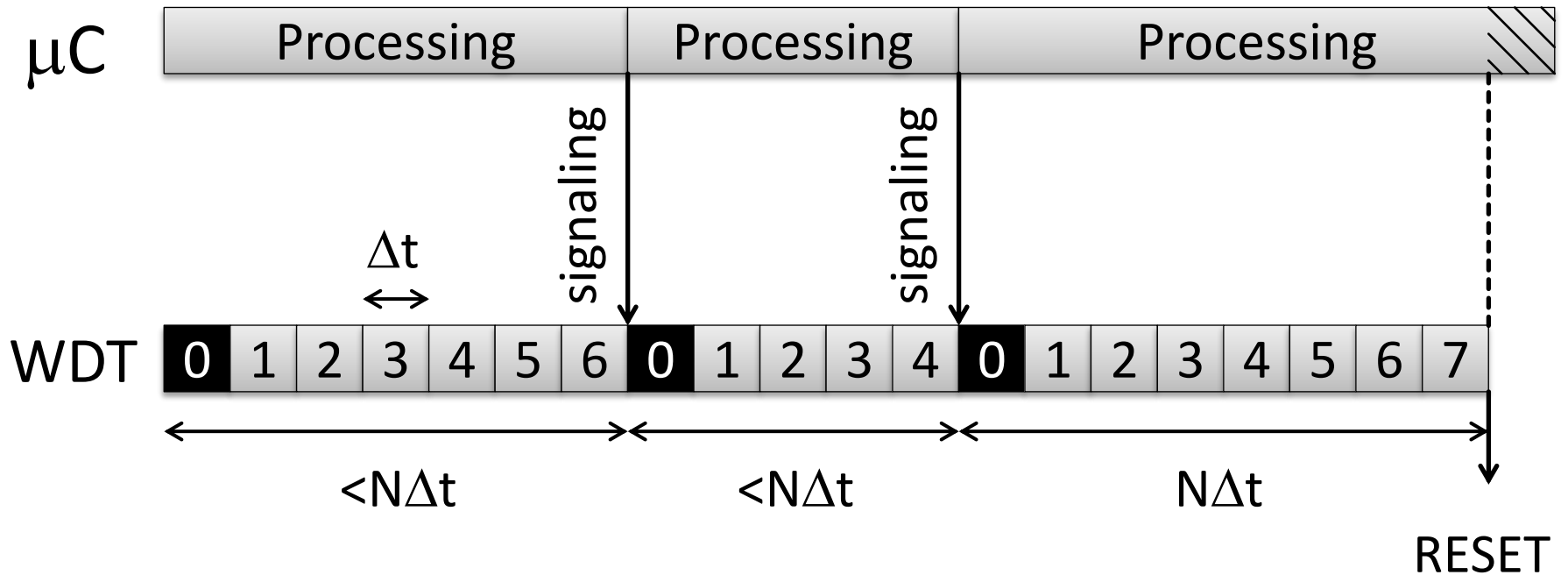
▶ Működési elv:

- ▶ a programnak adott időközönként jelezni kell a WDT-nek
- ▶ ellenkező esetben RESET generálódik

▶ Megoldás:

- ▶ egy számláló adott órajellel lép (Δt időnként)
- ▶ N lépés megtétele után RESET
- ▶ ha jelez a program neki, újra N lépés van a RESET-ig
- ▶ így a programnak $N\Delta t$ -nél rövidebb időnként kell jeleznie a WDT-nek, ellenkező esetben RESET

Watchdog timer ▶ Példa: N=8



Watchdog timer ▶ F410 példa

- ▶ Részletek egy későbbi fejezetben
- ▶ Alapállapot: $\Delta t = 256 * 12 / 191406$ s
- ▶ **N=125, timeout: 2,006 s**

```
PCA0CPL5=125; // 125 lépés -> 2 másodperc
```

```
...
```

```
void main(void)  
{
```

```
...
```

```
    while (1)  
    {  
        Process(); // feladat elvégzése <2s  
        PCA0CPH5=0; // jelzés a WDT-nek  
    }  
}
```

Watchdog timer ▶ Megfontolások

- ▶ RESET után automatikusan aktív
 - ▶ Ha nem használjuk, azonnal le kell kapcsolni
 - ▶ A C start-up kód túl hosszú lehet
- ▶ A jelzés lehet megszakítási rutinban
 - ▶ Nem véd a főprogram hibái ellen
 - ▶ Kevésbé robusztus
- ▶ Gondos programtervezést igényel
- ▶ A legjobb, ha be van kapcsolva
- ▶ Kikapcsolás csak tesztelés, prototípus esetén
- ▶ A kikapcsolás is tiltható! Csak RESET hatástalanít