



Mikroprocesorová technika

Prednáška č. 3

**Systemový reset, inicializácia procesora, prerušenia,
režimy činnosti procesora a nízkopríkonové aplikácie**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA RÁDIOELEKTRONIKY
Laboratórium DSP a mikroradičov



:: Signály resetu POR a PUC

Signál resetu POR (Power-On Reset)

- predstavuje samotný reset procesora ako obvodu
- je generovaný len v prípade, že nastane jedna z nasledujúcich udalostí:

- na napájacie piny procesora privedieme menovité napájacie napätie
- na pin !RST/NMI (Non Maskable Interrupt), ktorý je konfigurovaný ako vstup vonkajšieho resetu, privedieme logický signál s nízkou úrovňou a príslušnou dĺžkou trvania
- ak je signál SVS_POR (Supply Voltage Supervisor) na nízkej úrovni a zároveň bit PORON = 1

Signál resetu PUC (Power-Up Clear Reset)

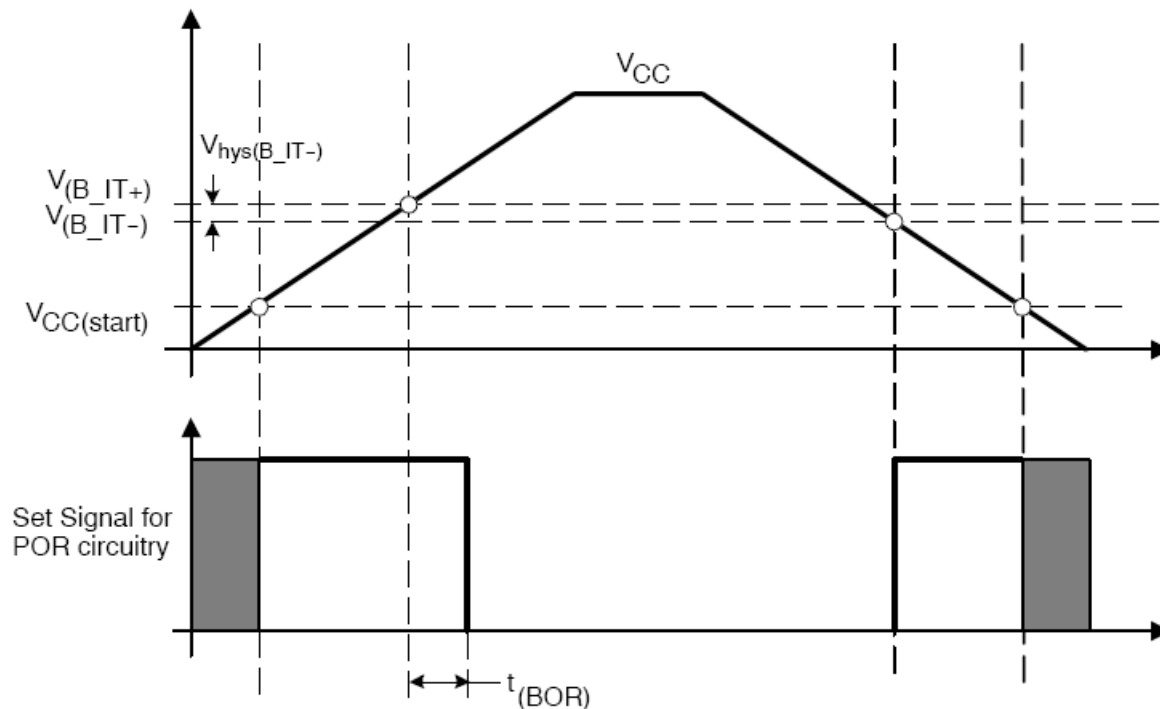
- je naviazaný na signál resetu POR, t.j. ak je generovaný POR je vždy generovaný aj PUC, spätne to však neplatí, t.j. ak je generovaný PUC, signál POR ním nie je generovaný. Signál PUC je spúšťaný nasledujúcimi udalosťami:

- nastavením signálu POR
- pretečením časovača watchdogu, ak procesor pracuje v režime aktívneho watchdogu
- zápisom chybného bezpečnostného kľúča watchdogu
- chybným bezpečnostným kľúčom pamäte FLASH
- v prípade, kedy CPU načíta inštrukciu z rozsahu adries prislúchajúcim periférnym modulom, t.j. z adries 0h - 01FFh



:: Podpäťový reset BOR (Brownout reset)

Obvod BOR detekuje nízke napájacie napätie na pine V_{CC} napríklad pri pripojení alebo odpojení napájacieho napätia a vyvolá reset procesora spustením signálu POR. Brownout - znamená pôvodne zníženie intenzity osvetlenia mesta vo vojne, pri leteckých náletoch. Blackout - úplné vypnutie elektriny.





:: Stav po systémovom resete

Po vykonaní systémového resetu generovaním signálu POR dôjde k **automatickej počiatočnej inicializácii procesora**, počas ktorej sa jeho základné nastavenia uvedú do **preddefinovaného stavu**. Stav po systémovom resete je nasledovný:

- pin !RST/NMI je nastavený ako vstup resetu
- všetky I/O piny sú nastavené ako vstupné piny
- riadiace registre (bity) periférnych modulov a ostatné registre sú nastavené do preddefinovaného stavu
- stavový register (SR) je resetovaný
- aktivuje sa časovač watchdogu
- do registra programového počítadla (PC) sa uloží adresa zapísaná na pozícii vektora resetu (0FFFEh). CPU začne vykonávať program od adresy uloženej v PC
- v prípade, že na pozícii vektora resetu sa nachádza hodnota 0FFFFh, dôjde k uvedeniu procesora do stavu s minimálnou spotrebou



:: Softvérová inicializácia



Po systémovom resete musí aplikačný program zabezpečiť inicializáciu procesora tak, aby nastavenie spĺňalo požiadavky aplikácie:

- inicializácia registra zásobníka (SP - Stack Pointer); zásobník typicky umiestňujeme na začiatok RAM pamäte
- inicializácia watchdogu podľa požiadaviek aplikácie
- konfigurácia všetkých periférnych modulov, ktoré v aplikácii používame
- správne nastavenie vektorov prerušení
- nastavenie portov (I/O pinov) a inicializácia premenných
- navyše je možné vyhodnotiť príznak pretečenia časovača watchdogu, príznak chyby oscilátora a príznaky FLASH pamäte, aby sme určili zdroj predchádzajúceho systémového resetu a aplikácia mohla príslušne reagovať



:: Spracovanie prerušenia



Akceptácia prerušenia

Oneskorenie vykonania prerušenia predstavuje 6 cyklov, začínajúc akceptovaním požiadavky na prerušenie a končiac počiatkom vykonávania prvej inštrukcie obslužnej rutiny prerušenia - ISR (Interrupt **S**ervice **R**outine).

Logické obvody prerušovacieho podsystemu procesora pritom vykonajú nasledujúcu postupnosť:

- akákoľvek práve vykonávaná inštrukcia je úplne dokončená
- údaj programového počítadla PC, ktoré ukazuje na inštrukciu, ktorá by sa mala vykonať v nasledujúcom cykle, je uložená do zásobníka
- údaj stavového registra SR je uložený do zásobníka
- v prípade, že počas vykonávania poslednej inštrukcie nastalo viacero udalostí požadujúcich prerušenie a čakajúcich na obsluhu, vyberie sa prerušenie s najvyššou prioritou
- príznak prerušenia, ktoré pochádza z jediného zdroja je automaticky zmazaný. Príznaky prerušenia, ktoré môžu pochádzať z viacerých zdrojov zostávajú nastavené, aby bolo možné obslužiť ich v programe
- je zmazaný obsah SR. Tým dôjde k ukončeniu akéhokoľvek nízkoopríkonového režimu. Pretože pri tejto operácii je zmazaný aj bit GIE, je zakázané vykonanie ďalších prerušení.
- obsah vektora prerušenia akceptovaného prerušenia (adresa ISR) je uložený do PC a program pokračuje vykonávaním ISR od tejto adresy



:: Spracovanie prerušenia



Návrat z prerušenia

ISR je vždy ukončená inštrukciou RETI (Return from an Interrupt Service Routine). Návrat z prerušenia si vyžaduje 5 cyklov, vykoná sa pritom nasledujúca postupnosť operácií:

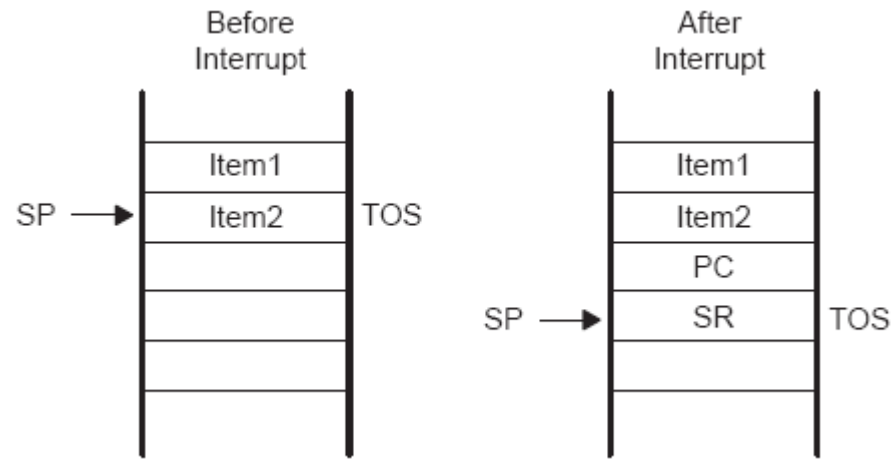
- pôvodný obsah (nastavenie bitov) registra SR, ktorý obsahoval pred vykonaním prerušenia, a ktorý bol uložený do zásobníka, je načítaný zo zásobníka a uložený do SR. Všetky predchádzajúce nastavenia bitov GIE, CPUOFF, atď. sú v tomto okamihu platné bez ohľadu na nastavenia uskutočnené počas vykonávania ISR.
- pôvodný obsah registra PC je načítaný zo zásobníka a uložený do PC, t.j. program pokračuje vo vykonávaní inštrukcií na adrese, ktorá nasleduje za adresou inštrukcie, počas ktorej nastala požiadavka na prerušenie.

Vnorené prerušenia

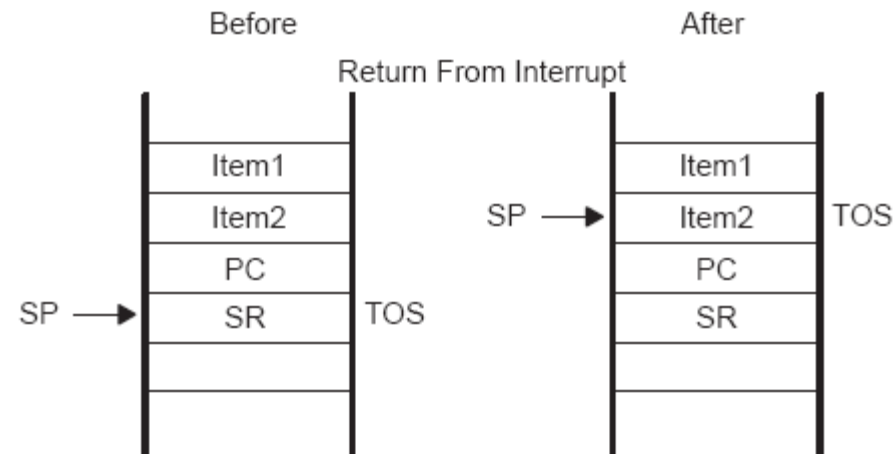
Ak počas vykonávania ISR nastavíme bit GIE, povolíme tým vnorenie prerušení. Ak v takomto prípade nastane prerušenie, bude vykonávanie ISR prerušené bez ohľadu na prioritu prerušení.



:: Spracovanie prerušenia



- **Top Of the Stack**
vrch zásobníka





:: Vektory prerušení

- vektory prerušení sú mapované v rozsahu adres 0FFFFh až 0FFC0h
- všetky vektory prerušenia sú programovateľné, pričom programátor do nich ukladá 16-bitovú adresu príslušnej obslužnej rutiny prerušenia
- doporučuje sa v zdrojovom kóde uviesť ISR pre každý vektor prerušenia
- nepoužívané vektory prerušenia potom môžu odkazovať na jednu ISR, ktorá obsahuje iba inštrukciu RETI
- nepriradené vektory prerušení predstavujú v podstate voľnú programovú pamäť, ktorú môžeme využiť pre zdrojový kód aplikácie



:: Tabuľka vektorov prerušení

INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-Up External Reset Watchdog Timer+ Flash key violation PC out-of-range ⁽¹⁾	PORIFG RSTIFG WDTIFG KEYV ⁽²⁾	Reset	0FFFEh	31, highest
NMI Oscillator fault Flash memory access violation	NMIIFG OFIFG ACCVIFG ⁽²⁾⁽³⁾	(non)-maskable (non)-maskable (non)-maskable	0FFFCh	30
			0FFFAh	29
			0FFF8h	28
			0FFF6h	27
Watchdog Timer+	WDTIFG	maskable	0FFF4h	26
Timer_A2	TACCR0 CCIFG ⁽⁴⁾	maskable	0FFF2h	25
Timer_A2	TACCR1 CCIFG, TAIFG ⁽²⁾⁽⁴⁾	maskable	0FFF0h	24
			0FFEEh	23
			0FFEC	22
ADC10 ⁽⁵⁾	ADC10IFG ⁽⁴⁾⁽⁵⁾	maskable	0FFEAh	21
USI	USIIFG, USISTTIFG ⁽²⁾⁽⁴⁾	maskable	0FFE8h	20
I/O Port P2 (two flags)	P2IFG.6 to P2IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE6h	19
I/O Port P1 (eight flags)	P1IFG.0 to P1IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE4h	18
			0FFE2h	17
			0FFE0h	16
See ⁽⁶⁾			0FFDEh to 0FFC0h	15 to 0, lowest

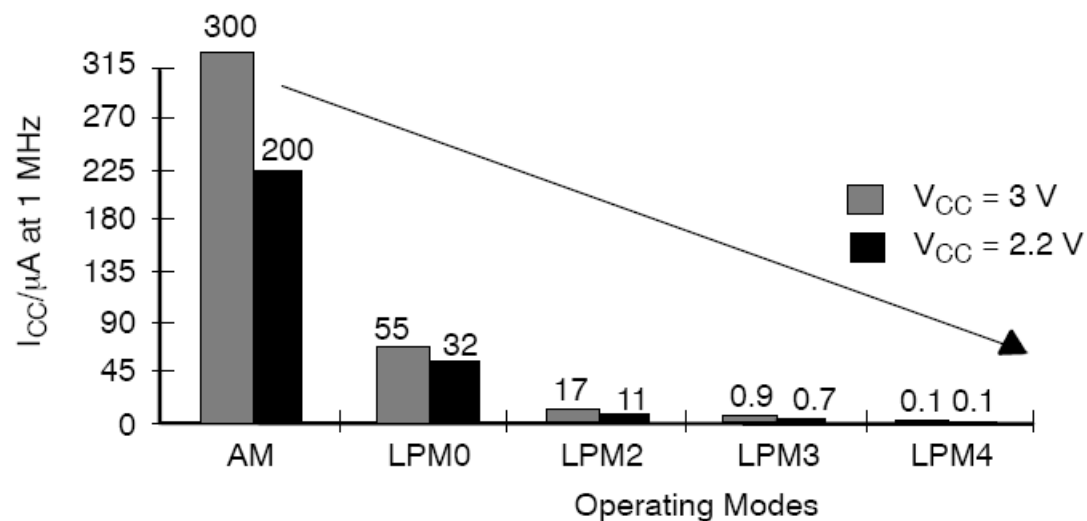
- (1) A reset is generated if the CPU tries to fetch instructions from within the module register memory address range (0h to 01FFh) or from within unused address ranges.
- (2) Multiple source flags
- (3) (non)-maskable: the individual interrupt-enable bit can disable an interrupt event, but the general interrupt enable cannot.
- (4) Interrupt flags are located in the module.
- (5) MSP430G2x31 only
- (6) The interrupt vectors at addresses 0FFDEh to 0FFC0h are not used in this device and can be used for regular program code if necessary.



:: Režimy činnosti MSP430

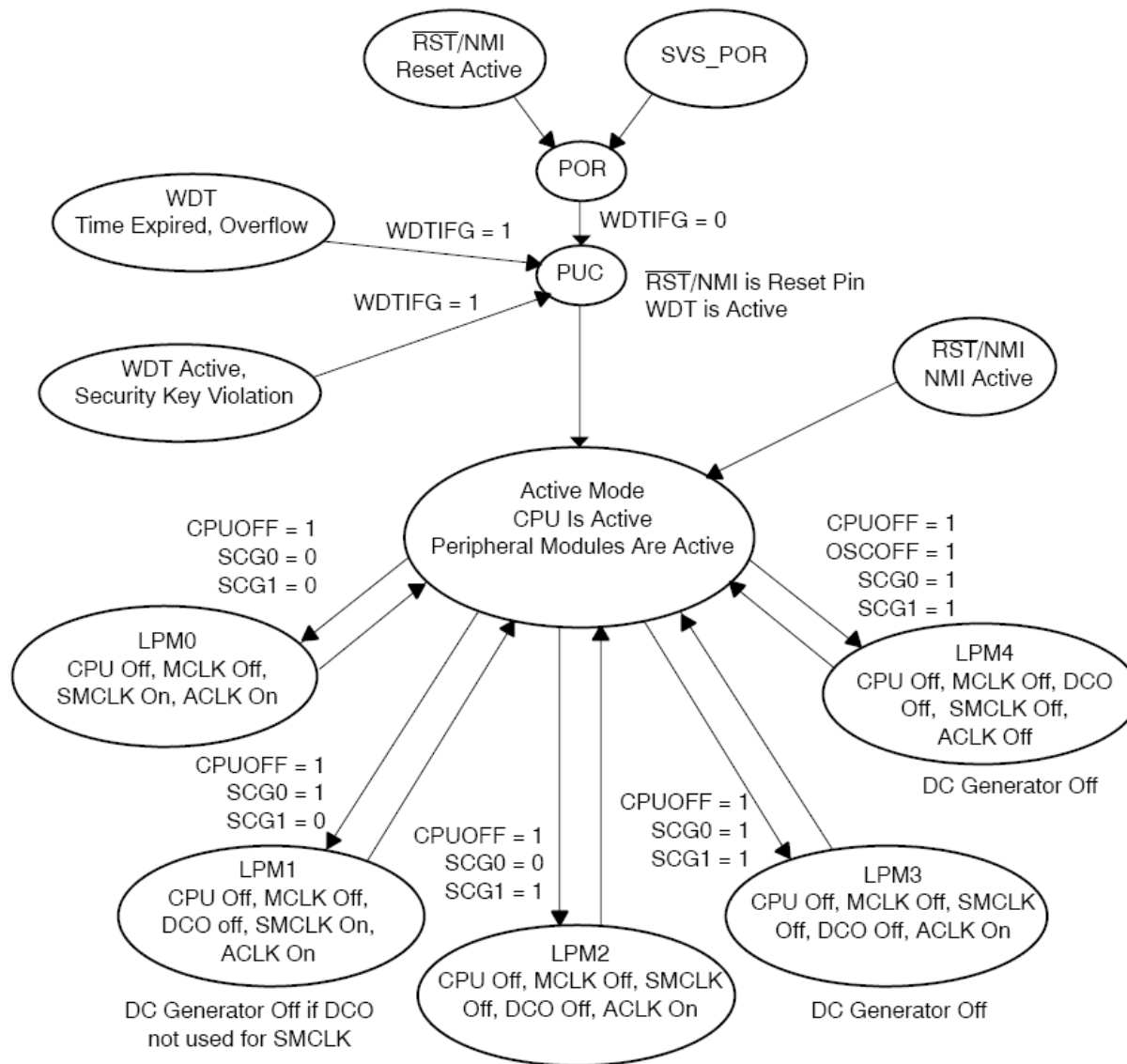
Procesory platformy MSP430 sú navrhnuté špeciálne pre **nízkopríkonové aplikácie** a z tohto dôvodu disponujú rozličnými **režimami činnosti**, ktoré berú do úvahy tri odlišné potreby:

- nízky príkon
- rýchlosť a dátovú priepustnosť
- minimalizáciu prúdovej spotreby jednotlivých periférií





:: Režimy činnosti MSP430





:: Režimy činnosti MSP430

SCG0	SCG1	OSCOFF	CPUOFF	Režim	Stav CPU a hodín
0	0	0	0	Aktívny	CPU je aktívna; všetky povolené hodinové signály sú aktívne
0	0	0	1	LPM0	CPU a MCLK sú deaktivované; SMCLK a ACLK sú aktívne
0	1	0	1	LPM1	CPU a MCLK sú neaktívne; DCO a DC generátor sú neaktívne, ak DCO nie je zdrojom pre SMCLK; ACLK je aktívny
1	0	0	1	LPM2	CPU, MCLK, SMCLK a DCO sú neaktívne; DC generátor zostáva aktívny; ACLK je aktívny
1	1	0	1	LPM3	CPU, MCLK, SMCLK a DCO sú neaktívne DC generátor je neaktívny; ACLK je aktívny
1	1	1	1	LPM4	CPU a všetky hodinové signály sú neaktívne

Pozn.: bity SCG0, SCG1, OSCOFF a CPUOFF sa nachádzajú v stavovom registri!



:: Prechod do a z nízkopríkonových režimov

Ak nastane udalosť, ktorá vyvolá povolené prerušenie, dôjde k prechodu procesora do aktívneho režimu z ktoréhokoľvek nízkopríkonového režimu!

Programový tok je nasledovný:

▪ začne vykonávanie ISR:

- *registre PC a SR sú uložené do zásobníka,*
- *bity CPUOFF, SCG1, a OSCOFF sú automaticky zmazané,*

▪ môžeme zvoliť spôsob návratu z ISR:

- *zo zásobníka je načítaný pôvodný obsah SR a teda dôjde k nastaveniu pôvodného režimu činnosti, alebo*
- *počas vykonávania ISR modifikujeme príslušné riadiace bity registra SR uloženého v zásobníku a teda po návrate z ISR (vykonaním inštrukcie RETI) pokračuje procesor v odlišnom režime činnosti.*



:: Prechod do a z nízkopríkonových režimov

Príklad prechodu do režimu LPM0

```
BIS #GIE+CPUOFF,SR           ; Vstup do režimu LPM0  
; ...                         ; Tu sa zastaví vykonávanie programu
```

Nastavenie ukončenia režimu LPM0 v ISR

```
BIC #CPUOFF,0(SP)           ; Ukončenie režimu LPM0 po vykonaní RETI  
RETI
```

; Príklad prechodu do režimu LPM3

```
BIS #GIE+CPUOFF+SCG1+SCG0,SR ; Vstup do režimu LPM3  
; ...                         ; Tu sa zastaví vykonávanie  
; programu
```

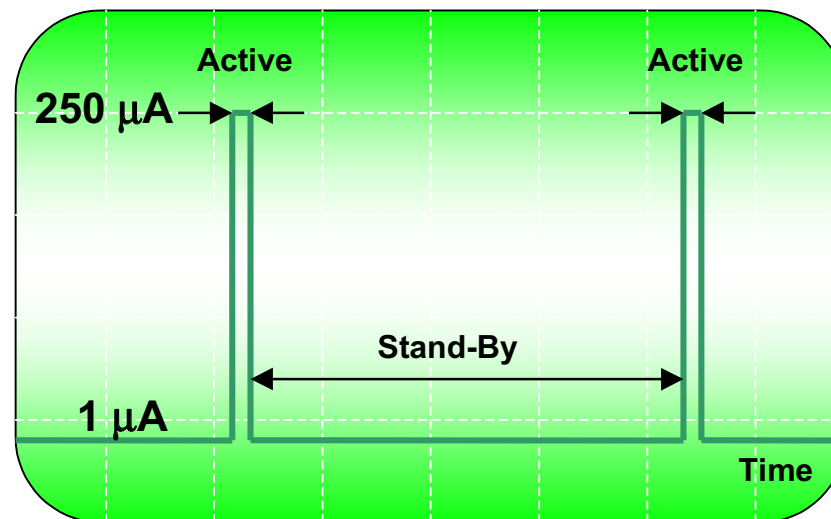
; Nastavenie ukončenia režimu LPM3 v ISR

```
BIC #CPUOFF+SCG1+SCG0,0(SP) ; Ukončenie režimu LPM3  
; po vykonaní RETI  
RETI
```



:: Princípy nízko príkonových aplikácií

- snažíme sa využiť hodinový systém procesora MSP430 tak, aby sme maximalizovali časový interval, počas ktorého zotrvá procesor v režime LPM3
- počas režimu LPM3 má procesor spotrebu menej ako $2 \mu\text{A}$ pričom zostáva aktívna funkcia RTC aj všetky prerušenia
- takt ACLK je odvodený od 32-kHz kryštálu a CPU je taktovaná z DCO (normálne odpojený), ktorému trvá prechod do aktívneho stavu $6 \mu\text{s}$





:: Princípy nízkopříkonových aplikací

Niekoľko zásad, ktoré by sme mali pri návrhu nízkopříkonovej aplikácie dodržať:

- používame prerušenia k prechodu procesora do aktívneho stavu
- periférne moduly by mali byť aktívne len v prípade potreby
- namiesto programového riešenia funkcií sa snažíme čo najviac využívať nízkopříkonové integrované periférne moduly. Napr. časovače môžu automaticky generovať PWM a zachytávať externé udalosti bez potreby účasti zdrojov CPU.
- namiesto sledovania príznakov a dlhých programových výpočtov sa snažíme používať vopred vypočítané skoky a rýchle hľadanie v tabuľkách
- snažíme sa zabrániť častému volaniu podprogramov a funkcií
- v prípade dlhých programových rutín, by sme mali používať registre CPU s jednocyklovým prístupom



:: Otázky ku skúške



- Aké dva druhy signálu resetu generuje systém resetu procesora MSP430 a aký je medzi nimi základný rozdiel?
- Ktoré udalosti spúšťajú reset POR?
- Ktoré udalosti spúšťajú reset PUC?
- Aká je úloha systému generovania podpäťového resetu BOR?
- Stručne opíšte stav procesora MSP430 po systémovom resete!
- Ktoré základné operácie musí vykonať aplikačný softvér v rámci inicializácie po systémovom resete?
- Definujte pojmy prerušenie, príznak prerušenia, vektor prerušenia a priorita prerušenia a uveďte, ktoré tri typy prerušení existujú!
- Stručne opíšte proces akceptácie prerušenia!
- Stručne opíšte proces návratu z obsluhy prerušenia!
- Definujte pojem nízkopríkonový režim činnosti procesora, uveďte koľko nízkopríkonových režimov je k dispozícii na procesore MSP430, čím sa vzájomne líšia a akým spôsobom tieto režimy môžeme aktivovať!



Koniec prednášky č. 3

**Systemový reset, inicializácia mikroradiča, prerušenia,
režimy činnosti mikroradiča a nízkopríkonové aplikácie**