



# *Mikroprocesorová technika*

Prednáška č. 5

**Watchdog, hardvérová násobička**

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

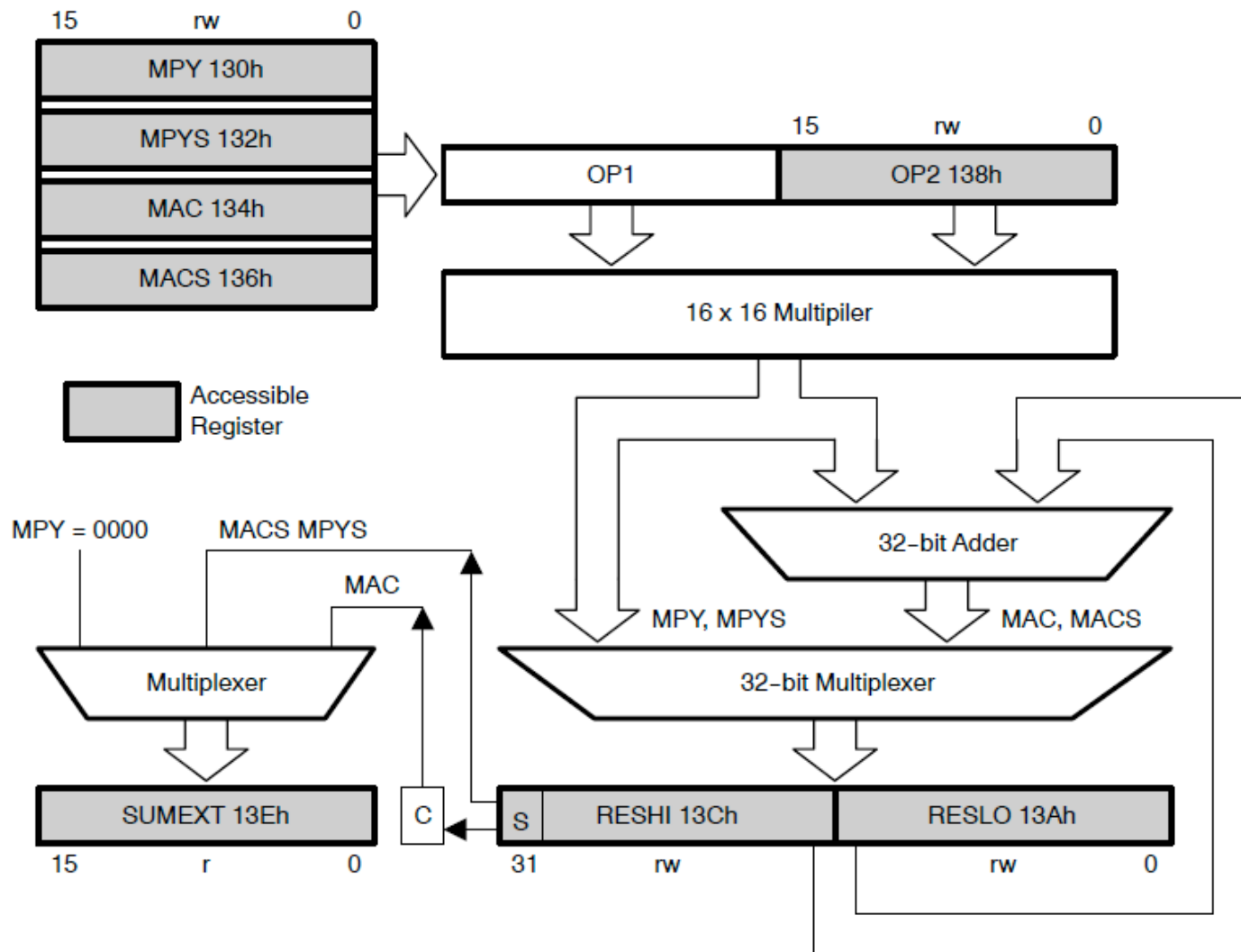


## :: Hardvérová násobička – základné vlastnosti

- hardvérová násobička procesorov MSP430 je periférny modul
- keďže nie je časťou CPU, činnosť násobičky neovplyvňuje činnosť CPU
- periférny modul násobičky riadime príslušnými registrami
- násobička podporuje:
  - *neznamienkové a znamienkové násobenie*
  - *neznamienková a znamienková akumulácia súčinov*
  - *formáty 16×16 bitov, 16×8 bitov, 8×16 bitov, 8×8 bitov*
- typ formátu násobenia je daný použitým registrom prvého operandu



## :: Hardvérová násobička – funkčná bloková schéma





## :: Hardvérová násobička – činnosť

- násobička disponuje
  - *dvomi 16-bitovými registrami operandov, OP1 a OP2*
  - *tromi výsledkovými registrami RESLO, RESHI, a SUMEXT*
- RESLO obsahuje dolné slovo výsledku
- RESHI obsahuje horné slovo výsledku
- SUMEXT obsahuje informáciu o výsledku
- výsledok násobičky je k dispozícii po troch cykloch MCLK, t.j. ďalšia inštrukcia nasledujúca po zápise operandu OP2 môže prečítať výsledok násobenia s výnimkou prípadu, kedy na prístup k výsledku využívame nepriame adresovanie
- v prípade použitia nepriameho adresovania je nevyhnutné použiť inštrukciu NOP pred prístupom k výsledku



## :: Hardvérová násobička – registre operandov

- register prvého operandu OP1 má štyri adresy
- zápisom prvého operandu na zvolenú adresu vyberieme typ násobenia
- zápisom druhého operandu do registra OP2 začne proces násobenia operandov OP1 a OP2
- výsledok násobenia bude uložený v troch výsledkových registroch RESLO, RESHI a SUMEXT
- pokiaľ vykonávame opakované násobenie, nie je potrebné znovu ukladať prvý operand do registra OP1

Adresa prvého operandu OP1	Inštrukcia násobenia	Režim násobičky
0130h	MPY - multiply	Neznamienkové násobenie
0132h	MPYS - multiply signed	Znamienkové násobenie
0134h	MAC - multiply and accumulate	Neznamienková suma súčinov
0136h	MACS - multiply and accum. signed	Znamienková suma súčinov



## :: Hardvérová násobička – registre výsledku

- register RESLO obsahuje dolných 16-bitov výsledku násobenia
- obsah registra RESHI závisí od režimu násobičky:

Režim násobičky	Obsah registra RESHI
MPY	Horných 16-bitov výsledku
MPYS	Bit MSB reprezentuje znamienko výsledku, ostatné bity reprezentujú horných 15-bitov výsledku. Výsledok má tvar dvojkového doplnku.
MAC	Horných 16-bitov výsledku
MACS	Horných 16-bitov výsledku. Výsledok má tvar dvojkového doplnku.



## :: Hardvérová násobička – registre výsledku

- obsah registra SUMEXT závisí od režimu násobičky:

Režim násobičky	Obsah registra SUMEXT
MPY	<b>SUMEXT je vždy 0000h.</b>
MPYS	<b>SUMEXT obsahuje znamienkové rozšírenie výsledku:</b> 00000h      výsledok násobenia bol kladný alebo nula 0FFFFh      výsledok násobenia bol záporný
MAC	<b>SUMEXT obsahuje carry bit výsledku:</b> 0000h      nedošlo k prenosu do vyššieho rádu 0001h      došlo k prenosu do vyššieho rádu
MACS	<b>SUMEXT obsahuje znamienkové rozšírenie výsledku:</b> 00000h      výsledok násobenia bol kladný alebo nula 0FFFFh      výsledok násobenia bol záporný



## :: Hardvérová násobička

### – režim MACS, pretečenie a podtečenie

- v režime MACS násobička nevykonáva automatickú detekciu podtečenia ani pretečenia
- rozsah akumulátora je
  - *pre kladné čísla: 0 až 7FFF FFFFh*
  - *pre záporné: 0FFFF FFFFh až 8000 0000h*
- podtečenie nastane, keď sčítanie dvoch záporných čísiel vedie k výsledku, ktorý sa nachádza v rozsahu pre kladné čísla
- pretečenie nastane, keď sčítanie dvoch kladných čísiel vedie k výsledku, ktorý sa nachádza v rozsahu pre záporné čísla
- v oboch prípadoch, obsahuje register SUMEXT znamienko výsledku:
  - *0FFFFh pri pretečení*
  - *0000h pri podtečení*
- tieto stavy musí detekovať a vhodne ošetriť aplikačný program





## :: Hardvérová násobička – príklady



- v príkladoch všetky režimy typu  $8 \times 8$  používajú absolútne adresovanie registrov, pretože assembler neumožňuje prístup typu `.B` k 16-bitovým registrom keď používame návestia so štandardného definičného súboru
- nie je potrebné softvérovo aktivovať znamienkové rozšírenie, keďže prístup k násobičke pomocou `.B` inštrukcie počas znamienkovej operácie automaticky zabezpečí znamienkové rozšírenie bytu v rámci modulu násobičky



## :: Hardvérová násobička – príklady

```
; Neznamienkové násobenie 16x16
      MOV      #01234h, &MPY      ; Uloženie prvého operandu
      MOV      #05678h, &OP2     ; Uloženie druhého operandu
;      ...                       ; Spracovanie výsledku
; Neznamienkové násobenie 8x8, absolútne adresovanie
      MOV.B    #012h, &0130h     ; Uloženie prvého operandu, MPY
      MOV.B    #034h, &0138h     ; Uloženie druhého operandu, OP2
;      ...                       ; Spracovanie výsledku
; Znamienkové násobenie 16x16
      MOV      #01234h, &MPYS    ; Uloženie prvého operandu
      MOV      #05678h, &OP2     ; Uloženie druhého operandu
;      ...                       ; Spracovanie výsledku
; Znamienkové násobenie 8x8, absolútne adresovanie
      MOV.B    #012h, &0132h     ; Uloženie prvého operandu, MPYS
      MOV.B    #034h, &0138h     ; Uloženie druhého operandu
;      ...                       ; Spracovanie výsledku
```



## :: Hardvérová násobička – príklady

; Neznamienkové násobenie 16x16 a akumulácia

MOV #01234h, &MAC ; Uloženie prvého operandu

MOV #05678h, &OP2 ; Uloženie druhého operandu

; ... ; Spracovanie výsledku

; Neznamienkové násobenie 8x8 a akumulácia, absolútne adresovanie

MOV.B #012h, &0134h ; Uloženie prvého operandu, MAC

MOV.B #034h, &0138h ; Uloženie druhého operandu, OP2

; ... ; Spracovanie výsledku

; Znamienkové násobenie 16x16 a akumulácia

MOV #01234h, &MACS ; Uloženie prvého operandu

MOV #05678h, &OP2 ; Uloženie druhého operandu

; ... ; Spracovanie výsledku

; Znamienkové násobenie 8x8 a akumulácia, absolútne adresovanie 8x8

MOV.B #012h, &0136h ; Uloženie prvého operandu

MOV.B #034h, R5 ; Dočasné umiestnenie druhého operandu

MOV R5, &OP2 ; Uloženie druhého operandu

; ... ; Spracovanie výsledku



## :: Hardvérová násobička

### – nepriame adresovanie registra RESLO

- ak pri prístupe k registrom výsledku násobičky využívame nepriame alebo nepriame autoinkrementačné adresovanie, musí sa medzi inštrukciou ukladajúcou druhý operand a inštrukciou, ktorá pristupuje k jednému z výsledkových registrov, nachádzať minimálne jedna inštrukcia:

```
MOV    #RESLO, R5      ; adresa reg. RESLO v reg. R5
MOV    &OPER1, &MPY    ; ulozenie prveho operandu
MOV    &OPER2, &OP2    ; ulozenie druhého operandu
NOP                    ; potrebný aspon jeden cyklus
MOV    @R5+, &xyz      ; ulozenie reg. RESLO
MOV    @R5, &xyz       ; ulozenie reg. RESHI
```



## :: Hardvérová násobička

### – využitie prerušení

- ak nastane po zápise prvého operandu OP1 a pred zápisom druhého operandu OP2 prerušenie a zároveň v obsluhu daného prerušenia využívame násobičku, dôjde k strate pôvodného nastavenia režimu násobičky a výsledok násobenia bude nepredvídateľný
- tomuto stavu je možné zabrániť zakázaním prerušení pred použitím násobičky alebo nepoužívaním násobičky v obsluhu prerušenia

```
DINT                ; zakazanie preruseni
NOP                 ; oneskorenie pozadovane pre DINT
MOV    #xxh, &MPY   ; prvý operand
MOV    #xxh, &OP2   ; druhy operand
EINT                ; povolenie preruseni je mozne uz
                   ; pred spracovanim vysledku nasobenia
```

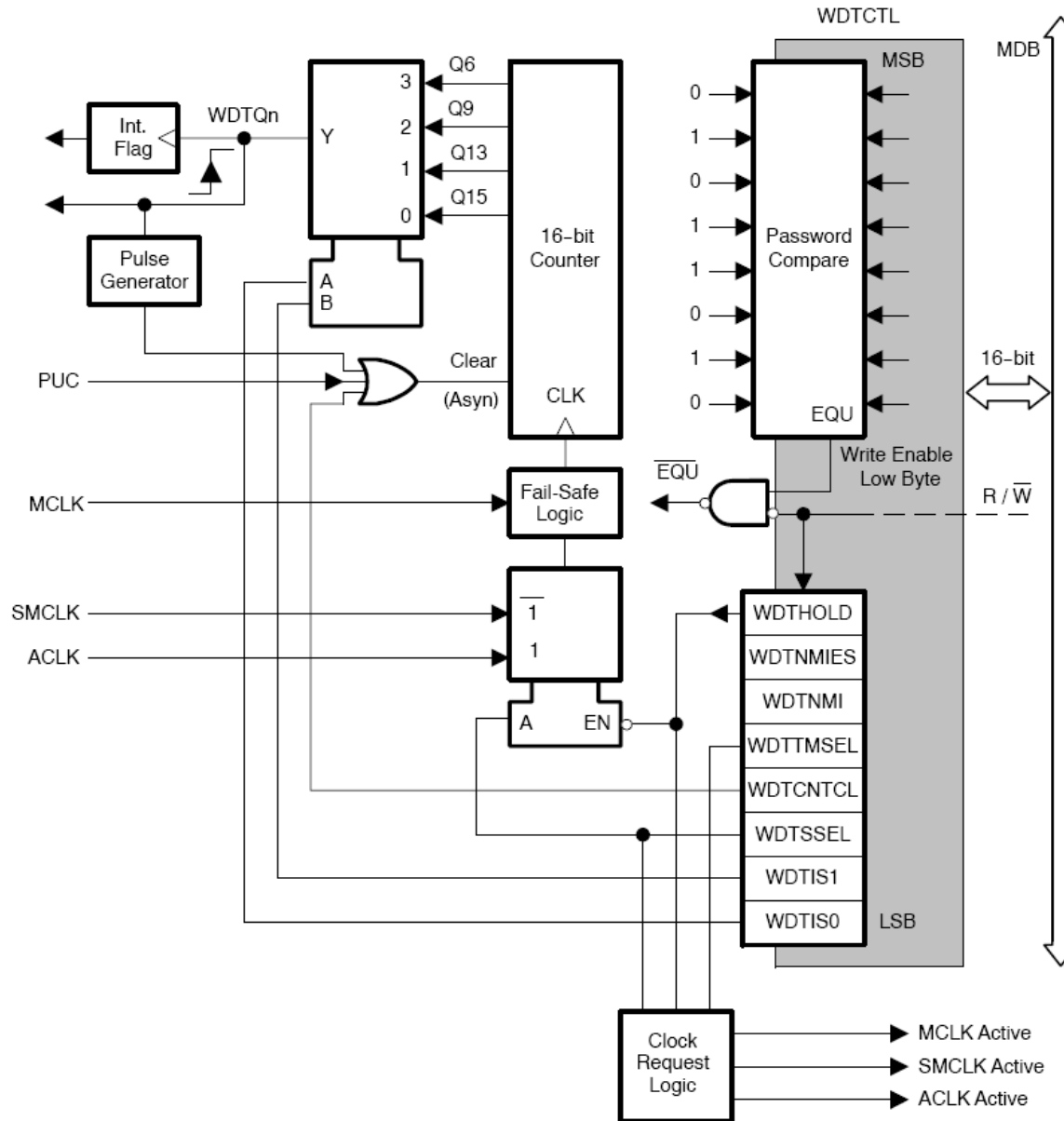


## :: Watchdog – základné vlastnosti

- hlavnou funkciou časovača watchdog-u je vykonať kontrolovaný reštart procesora ak dôjde počas behu aplikačného programu k neočakávanému softvérovému problému
- k aktivácii systémového resetu PUC dôjde po uplynutí zvoleného časového intervalu
- ak funkciu watchdog-u v aplikácii nepotrebujeme, môžeme modul konfigurovať ako bežný intervalový časovač, ktorý môže generovať prerušenia vo zvolených časových intervaloch
- základné vlastnosti modulu watchdog-u:
  - *štyri časové intervaly voliteľné softvérovo*
  - *režim watchdog / intervalový časovač*
  - *prístup k riadiacemu registru watchdog-u je chránený heslom*
  - *modul watchdog-u zároveň umožňuje konfiguráciu funkcie pinu RST/NMI*
  - *voliteľný zdroj hodín pre modul watchdog-u*
  - *WDT môžeme kvôli úspore spotreby zastaviť*
  - *zabezpečenie proti zlyhaniu hodín*



# :: Watchdog – funkčná bloková schéma





## :: Watchdog – činnosť



- časovač WDT môžeme konfigurovať pomocou riadiaceho registra WDTCTL
- WDTCTL je 16-bitový register chránený bezpečnostným kľúčom
- akýkoľvek prístup k registru je potrebné vykonať pomocou .W inštrukcie, pričom pri zápise do registra musí horný byte obsahovať bezpečnostný kľúč 05Ah
- ak pri zápise do registra WDTCTL nebude horný byte obsahovať bezpečnostný kľúč 05Ah vyhodnotí modul WDT situáciu ako chybu pri zápise bezpečnostného kľúča a dôjde k vyvolaniu resetu PUC bez ohľadu na nastavený režim časovača
- pri čítaní registra WDTCTL bude horný byte vždy obsahovať číslo 069h
- frekvencia hodinového signálu časovača WDT by mala byť buď nižšia alebo rovnaká ako systémové hodiny (MCLK)





## :: Watchdog – počítadlo časovača

- počítadlo časovača WDT (WDTCNT) je 16-bitové počítadlo s počítaním smerom hore, ktoré nie je priamo prístupné aplikačnému softvéru
- riadenie počítadla WDTCNT ako aj výber časového intervalu watchdog-u zabezpečuje riadiaci register WDTCTL
- zdrojom hodinového signálu pre počítadlo WDTCNT je buď SMCLK alebo ACLK



## :: Watchdog – režim WDT



- po vykonaní resetu PUC je modul WDT konfigurovaný v režime watchdog s počiatočným intervalom resetu každých 32768 cyklov DCOCLK
- aplikačný softvér musí nastaviť, zastaviť alebo zmazať WDT predtým ako dôjde k uplynutiu nastaveného intervalu resetu, inak dôjde k vyvolaniu resetu PUC
- v režime WDT vyvolá zápis do registra WDTCTL s nekorektným bezpečnostným kľúčom okamžite reset procesora
- reset PUC automaticky vykoná pôvodné nastavenie WDT a nastaví pin RST/NMI ako vstup resetu



## :: Watchdog – režim intervalového časovača

- nastavením bitu WDTTMSEL = 1 aktivujeme režim intervalového časovača
- režim je vhodný pre generovanie periodických prerušení
- vždy po uplynutí zvoleného časového intervalu dôjde k nastaveniu príznaku WDTIFG (IFG1.0)
- v tomto režime nedôjde k resetu PUC a teda nedôjde ani k zmene nastavenia bitu povoľujúceho prerušenie WDTIE (IE1.0)
- ak sú bity WDTIE a GIE nastavené, vyvolá nastavenie príznaku WDTIFG prerušenie
- príznak prerušenia WDTIFG je mazaný automaticky keď dôjde k obsluhu prerušenia, resp. môžeme ho zmazať aj softvérovo
- adresa vektora prerušenia v režime intervalového časovača sa líši od adresy vektora prerušenia v prípade režimu WDT



## :: Watchdog – modifikácia časovača WDT

- ak potrebujeme modifikovať časovač WDT, k zmene by malo dôjsť v rámci jednej inštrukcie, pričom zároveň by mal byť nastavený bit  $WDTCNTCL = 1$
- uvedený postup zamedzí vykonaniu nežiaduceho okamžitého resetu PUC, resp. prerušenia podľa aktuálne nastaveného režimu
- aby sme zabránili generovaniu nekorektného časového intervalu pri zmene zdroja hodín pre modul WDT, je vhodné modul pred zmenou zastaviť



## :: Watchdog – zabezpečenie proti zlyhaniu hodín

- modul WDT obsahuje zabudovanú ochranu proti zlyhaniu hodín, ktorá zabezpečuje, aby nebolo možné zakázať hodinový signál používaný pre modul WDT, pokiaľ je tento v režime WDT
- z toho vyplýva, že voľba hodín pre modul WDT môže ovplyvniť nízkopríkonové režimy. Napr. ak zdrojom hodín pre WDT je signál ACLK, potom režim LPM4 nie je možné použiť, pretože modul WDT zabráni zakázaniu signálu ACLK.
- ak zdrojom hodín WDT sú signály ACLK alebo SMCLK a dôjde k ich zlyhaniu modul WDT automaticky zmení zdroj hodín na MCLK
- ak zdrojom signálu MCLK je oscilátor s externým kryštálom a tento zlyhá, modul WDT automaticky aktivuje DCO a použije ho ako zdroj signálu MCLK
- ak používame modul WDT v režime intervalového časovača, systém zabezpečenia proti zlyhaniu hodín nie je aktívny

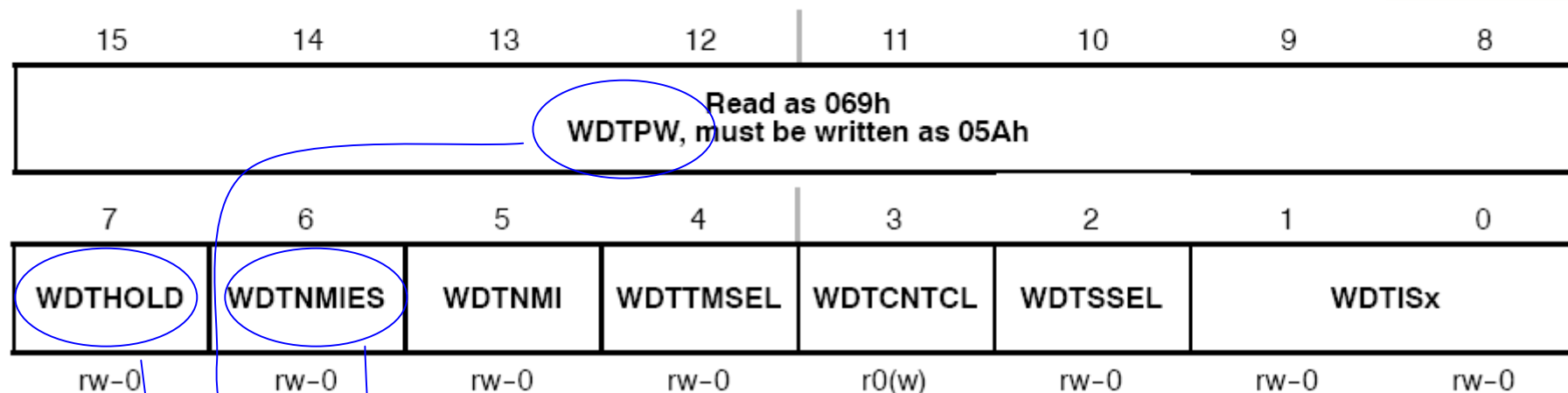


## :: Watchdog – príklady

```
; Periodické mazanie aktívneho watchdogu
    MOV    #WDTPW+WDTCNTCL, &WDTCTL
;
; Zmena intervalu časovača watchdog-u
    MOV    #WDTPW+WDTCNTL+WDTSSSEL, &WDTCTL
;
; Zastavenie watchdog-u
    MOV    #WDTPW+WDT HOLD, &WDTCTL
;
; Nastavenie režimu intervalového časovača, interval CLK/8192
    MOV    #WDTPW+WDTCNTCL+WDTTMSSEL+WDTIS0, &WDTCTL
```



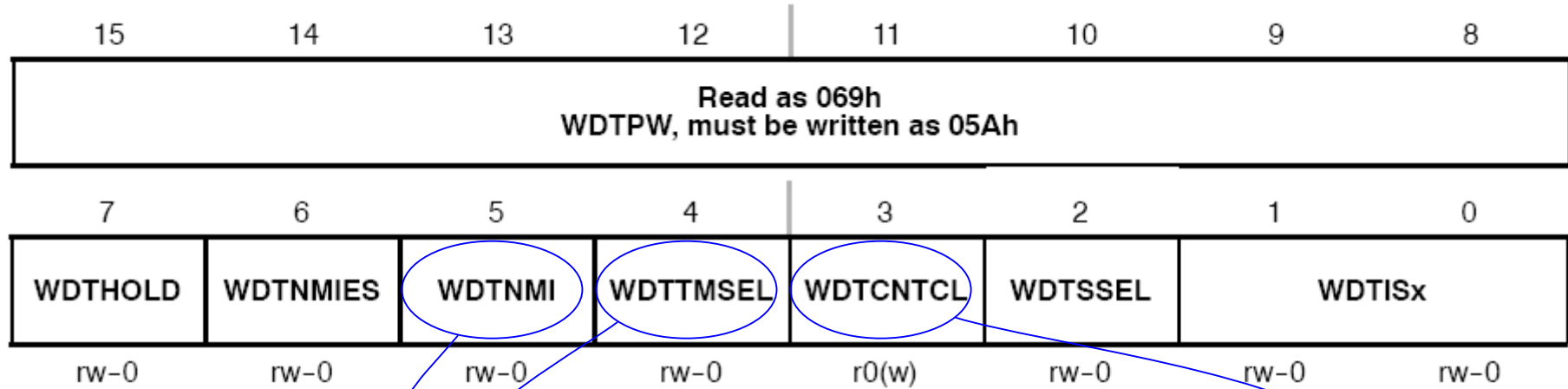
## :: Watchdog – riadiaci register WDTCTL



<b>WDTPW</b>	Bity 15-8	<b>Watchdog timer password.</b> Pri čítaní má vždy hodnotu 069h. Pri zápise musí mať hodnotu 05Ah, inak dôjde ku generovaniu resetu PUC.
<b>WDT HOLD</b>	Bit 7	<b>Watchdog timer hold.</b> Nastavením WDT HOLD = 1, keď nepoužívame WDT, znížime spotrebu. 0 časovač WDT nie je zastavený 1 časovač WDT je zastavený
<b>WDTNMIES</b>	Bit 6	<b>Watchdog timer NMI edge select.</b> Nastavenie aktívnej hrany NMI prerušenia, keď WDTNMI = 1. Modifikácia tohto bitu môže zapríčiniť aktiváciu NMI. Aby sa zabránilo nežiaducemu spusteniu NMI, je vhodné pred modifikáciou tohto bitu nastaviť WDTIE = 0. 0 NMI nastane pri nábežnej hrane 1 NMI nastane pri zostupnej hrane



## :: Watchdog – riadiaci register WDTCTL

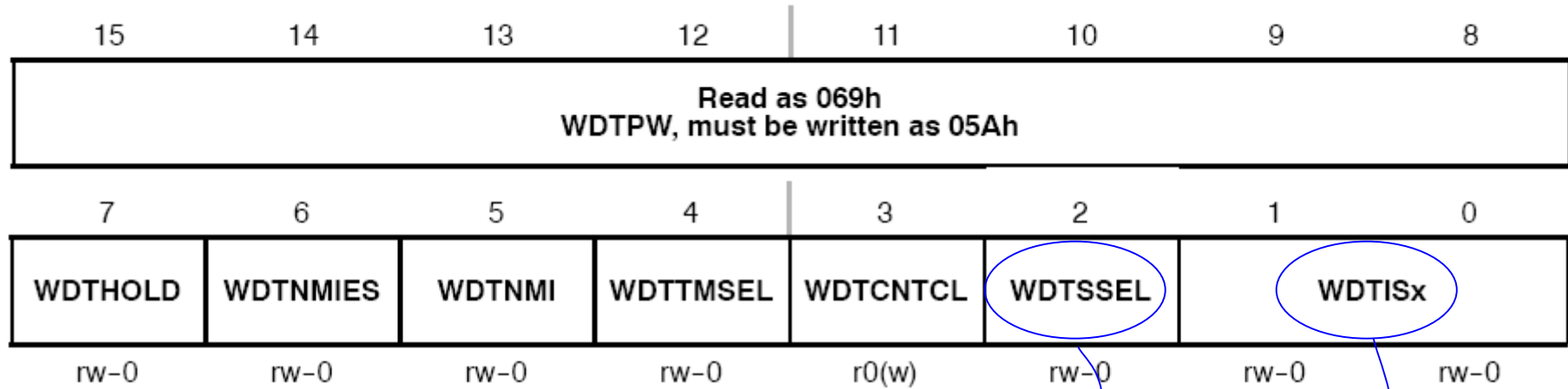


<b>WDTNMI</b>	Bit 5	<b>Watchdog timer NMI select.</b> Výber funkcie pinu RST/NMI. 0 pin RST/NMI konfigurovaný ako vstup resetu 1 pin RST/NMI konfigurovaný ako vstup NMI prerušenia
<b>WDTTMSSEL</b>	Bit 4	<b>Watchdog timer mode select.</b> 0 režim watchdog 1 režim intervalového časovača
<b>WDTCNTCL</b>	Bit 3	<b>Watchdog timer counter clear.</b> 0 bez vplyvu 1 WDTCNT = 0000h





## :: Watchdog – riadiaci register WDTCTL



<b>WDT SSEL</b>	Bit 2	<b>Watchdog timer clock source select</b> 0 SMCLK 1 ACLK
<b>WDT ISx</b>	Bits 1-0	<b>Watchdog timer interval select.</b> Výber intervalu časovača watchdog-u pre generovanie prerušenia alebo PUC. 00 zdroj hodín WDT /32768 01 zdroj hodín WDT /8192 10 zdroj hodín WDT /512 11 zdroj hodín WDT /64



## :: Otázky ku skúške



- Aké typy a formáty podporuje hardvérová násobička procesorov MSP430?
- Opíšte registre operandov a výsledkov hardvérovej násobičky procesora MSP430!
- Akým spôsobom vyberáme typ násobenia a ako sa násobička správa v prípade opakovaných násobení?
- Vysvetlite pojem pretečenia a podtečenia pri režime MACS a akým spôsobom mu predchádzame!
- Ako sa správa hardvérová násobička pri prerušení a ako to môže ovplyvniť beh programu?
- Uveďte primárnu funkciu časovača watchdog-u, jeho základné vlastnosti a možné zdroje hodín pre watchdog!
- Uveďte, ako je watchdog konfigurovaný po resete, čo je to bezpečnostný kľúč watchdog-u a ako ho používame.
- Ako pracuje watchdog v režime intervalového časovača a na aký účel môžeme tento režim použiť?
- Ako pracuje systém zabezpečenia proti zlyhaniu hodinového signálu pre watchdog?



**Koniec prednášky č. 5**

**Watchdog, hardvérová násobička**