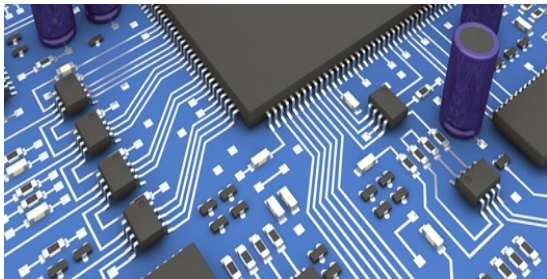


Aplikace Embedded systémů v Mechatronice



Michal Bastl

WatchDog timer

WatchDog Timer (WDT)

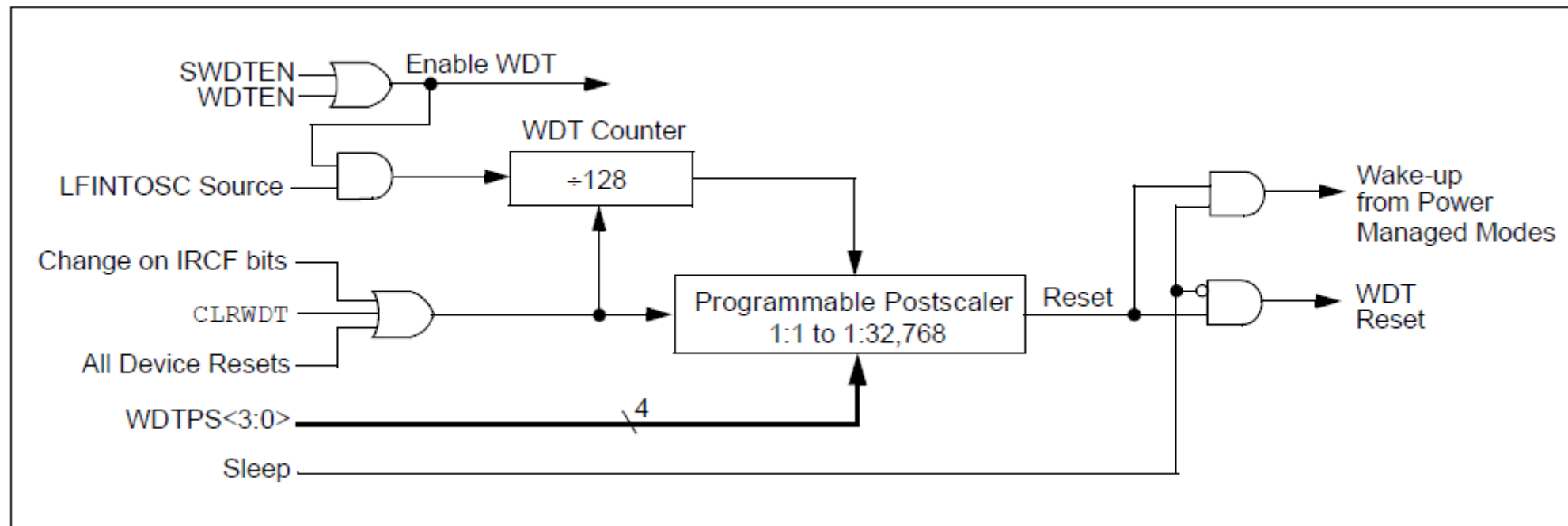
- Jedná se o bezpečnostní periférii MCU.
- Periferie se používá k ochraně proti zaseknutí (zacyklení) programu.
- Timer má vlastní (interní) zdroj hodinových pulzů.
- Pokud dojde k jeho přetečení je vynucen reset MCU.
- Programátor musí WDT pravidelně mazat.



WatchDog timer

- Zdrojem hodin pro WDT je interní LFINOSC v PIC18 má frekvenci 31,25kHz
- WDT se ovládá pomocí konfiguračních bitů
- Existuje i mód, kdy jej můžu ovládat z programu SW
- Doba přetečení lze zhruba ovládat pomocí další děličky
- Perioda WDT je cca 4ms (4,096ms)

FIGURE 24-1: WDT BLOCK DIAGRAM



WatchDog timer

- Pro nastavení WDT slouží konfigurační registr **CONFIH2H**
- Je zde možnost nadefinovat stav
- Zapnut
- SW ovládání umožněno
- Chování ve spánku
- Vypnout
- K nastavení konfiguračních bitů slouží #pragma...

Nastavení pro prekladac XC8:

#pragma config **WDTEN** = ON

#pragma config **WDTPS** = 256

REGISTER 24-3: CONFIG2H: CONFIGURATION REGISTER 2 HIGH

U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	
—	—	WDTPS<3:0>				WDTEN<1:0>		
bit 7								bit 0

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

x = Bit is unknown

bit 7-6

Unimplemented: Read as '0'

bit 5-2

WDTPS<3:0>: Watchdog Timer Postscale Select bits

1111 = 1:32,768

1110 = 1:16,384

1101 = 1:8,192

1100 = 1:4,096

1011 = 1:2,048

1010 = 1:1,024

1001 = 1:512

1000 = 1:256

0111 = 1:128

0110 = 1:64

0101 = 1:32

0100 = 1:16

0011 = 1:8

0010 = 1:4

0001 = 1:2

0000 = 1:1

bit 1-0

WDTEN<1:0>: Watchdog Timer Enable bits

11 = WDT enabled in hardware; SWDTEN bit disabled

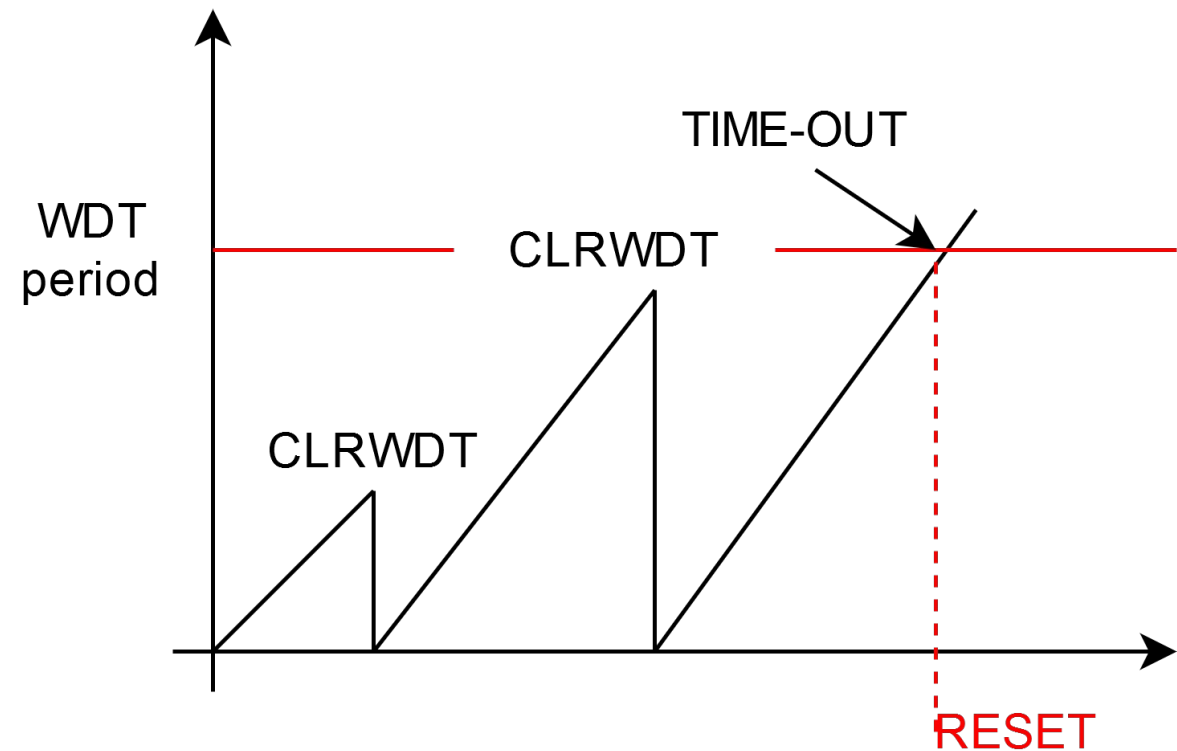
10 = WDT controlled by the SWDTEN bit

01 = WDT enabled when device is active, disabled when device is in Sleep; SWDTEN bit disabled

00 = WDT disabled in hardware; SWDTEN bit disabled

WatchDog timer

- Pokud je jednou WDT nakonfigurován a spuštěn, tak ho musím před vypršením smazat v mém programu.
- WDT se maže pomocí instrukce procesoru „CLRWDT“.
- C program mohou mixovat s příkazy v assembleru pomocí `__asm("CLRWDT");`
- Skutečnost, že došlo k resetu pomocí WDT mohou detekovat.
- V registru RCON je bit TO (time-out)
- Tím mohou kontrolovat, že došlo v předchozím běhu programu k WDT resetu.
- Typicky to dělám ještě v inicializační fázi.



Řízení spotřeby

PIC18 má v podstatě dva módy k snížení spotřeby

- IDLE mod, kdy se zastaví CPU
- SLEEP mód, kdy se zastaví CPU i periferie
- K ovládání slouží IDLEN bit v registru OSCCON
- SCS bity pak mohou nastavit CLOCK, který bude v IDLE režimu pohánět periferie
- Periferie tedy mohou běžet s výrazně nižším clockem a šetřit tak energii
- Konfigurace jsou uvedeny v tabulce

TABLE 3-1: POWER-MANAGED MODES

Mode	OSCCON Bits		Module Clocking		Available Clock and Oscillator Source
	IDLEN ⁽¹⁾	SCS<1:0>	CPU	Peripherals	
Sleep	0	N/A	Off	Off	None – All clocks are disabled
PRI_RUN	N/A	00	Clocked	Clocked	Primary – LP, XT, HS, RC, EC and Internal Oscillator Block ⁽²⁾ . This is the normal full-power execution mode.
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – SOSC Oscillator
RC_RUN	N/A	1x	Clocked	Clocked	Internal Oscillator Block ⁽²⁾
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC
SEC_IDLE	1	01	Off	Clocked	Secondary – SOSC Oscillator
RC_IDLE	1	1x	Off	Clocked	Internal Oscillator Block ⁽²⁾

Note 1: IDLEN reflects its value when the SLEEP instruction is executed.

2: Includes HFINTOSC and HFINTOSC postscaler, as well as the LFINTOSC source.

Řízení spotřeby

IDLEN = 0

- V tomto režimu je přejde s příchodem SLEEP instrukce MCU do spánku
- Vzbudit ho mohu na přetečení WDT, který po SLEEP instrukci změní funkci a nedojde k RESETU, ale k probuzení
- Probuzení by dále bylo možné pomocí pinu MCU (u PIC18 se jedna jen o porty RB)

IDLEN = 1

- V tomto režimu je přejde s příchodem SLEEP instrukce MCU do IDLE režimu
- Vzhledem k tomu, že v tomto módu jsou k dispozici periférie, mohu MCU probouze pomocí interruptu
- Ten může být např. Od TIMERU, ADC, Uartu apod.

ZAP/VYP periférii

- Pomocí registrů PMD0-2 lze vypnout jednotlivé periférie
- Pokud je nepoužívám a potřebuji co nejnižší spotřebu

3.7 Register Definitions: Peripheral Module Disable

REGISTER 3-1: PMD0: PERIPHERAL MODULE DISABLE REGISTER 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7

UART2MD: UART2 Peripheral Module Disable Control bit

1 = Module is disabled, Clock Source is disconnected, module does not draw digital power

0 = Module is enabled, Clock Source is connected, module draws digital power

bit 6

UART1MD: UART1 Peripheral Module Disable Control bit

1 = Module is disabled, Clock Source is disconnected, module does not draw digital power

0 = Module is enabled, Clock Source is connected, module draws digital power

...
