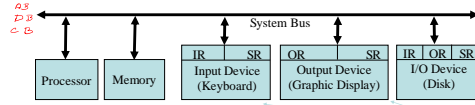


Mikropočítačové Systémy
MIPS
 Distribuované vnorené počítačové systémy
 Distributed Embedded Computer System
 (Microcontrollers)

Prednáška 6.
Prerušená.

Interrupts - Prerušená

Na počiatku bolo všetko jednoduché. Prvé počítače nemali prerušovací podsystem. Postupnosť vykonávania inštrukcií sa riadila pomocou obsahu adresného čítača. Neskôr sa zistilo, že niektoré periférne obvody sú vzhľadom na CPU a pamäť pomalé. Procesor sa musel o ich pozornosť usilovať. Neustále kontroloval stavový register takého pomalého zariadenia, aby zistil, či je alebo nie je pripravené na spoluprácu s procesorom. Týmto sa rýchlosť procesora znižila. Takáto metóda zisťovania stavu periférie sa nazýva: *Polled Method* - opytovacia, testovacia, ...



Tri základné spôsoby komunikácie s I/O podsystemom:
Polled Method - opytovacia metóda
Interrupt Method - prerušenia
DMA - priamy prenos medzi pamäťou a perifériami

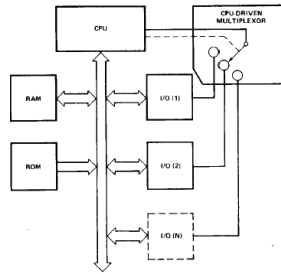
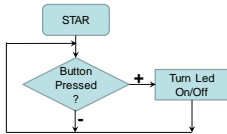
1

2

Obsluha periférií:

Polled Method - opytovacia metóda

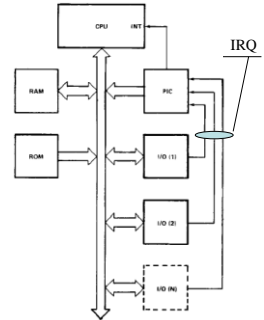
Spôsob komunikácie - protokol je riadený CPU pomocou programu. Nech má I/O(2) „prípravený“ znak pre CPU. CPU adresuje I/O(2) a prečíta znak do ACU. Atď. Ak nie je rýchlosť CPU a I/O(2) rovnaká, CPU preberie znak niekoľko krát. Riešenie je v použití: *Status registra (bitu)*.



Obsluha periférií:

Interrupt - prerušenie

Niekedy je potrebné prerušiť beh programu, napr.: pokles napájania. Prerúšením behu programu môžeme tiež informovať CPU o ukončení udalosti: napr. odovšielanie znaku cez USART. Moderné CPU majú inštrukcie, ktoré sa správajú podobne ako hardwarové prerušenie. T.j. Ak je prerušenie činnosť CPU, riadenie sa dočasne odovzdá inému „programu“. Po skončení sa vráti k pôvodnej činnosti. Viacnásobné prerušenia: Priorita.

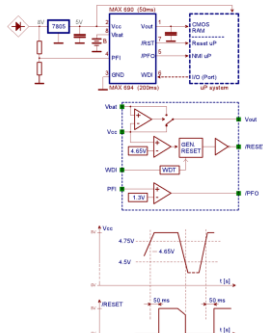
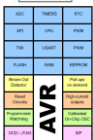


3

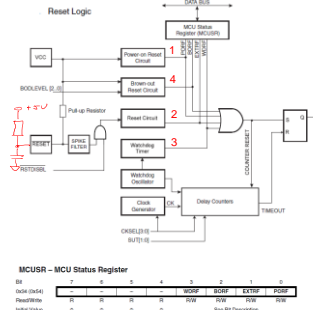
4

Resetovanie MIPS, AVR

MIPS		AVR	
PCNT	PCNT	PCNT	PCNT
(PCNT1) RESET1	PC8	1	28
(PCNT1) RESET2	PC9	2	27
(PCNT1) INT0	PC10	3	26
(PCNT1) INT1	PC11	4	25
(PCNT1) INT2	PC12	5	24
(PCNT1) INT3	PC13	6	23
(PCNT1) INT4	PC14	7	22
(PCNT1) INT5	PC15	8	21
(PCNT1) INT6	PC16	9	20
(PCNT1) INT7	PC17	10	19
(PCNT1) INT8	PC18	11	18
(PCNT1) INT9	PC19	12	17
(PCNT1) INT10	PC20	13	16
(PCNT1) INT11	PC21	14	15



Resetovanie MIPS, AVR



Zdroje Reset-u

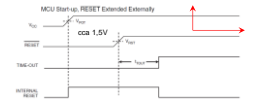
- Power-on Reset.** MCU je resetované, ak napájanie je pod prahovú hodnotu (VPOT).
- External Reset.** MCU je resetované, ak sa na pine RESET objaví nízka úroveň signálu na čas dlhší ako je minimálny.
- Watchdog Reset.** MCU je resetované, ak Watchdog Timer napočíta nastavenú periódu a je jeho činnosť je povolená.
- Brown-out Reset.** MCU je resetované, ak napájacie napätie VCC poklesne pod hodnotu (VBOT) a „Brown-out Detector“ je povolený.
- JTAG AVR Reset.

5

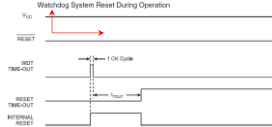
6

Zdroje Reset-u

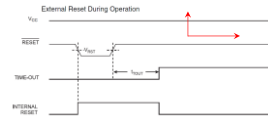
Power-on Reset. MCU je resetované, ak napájanie je pod prahovou hodnotou (VPOT).



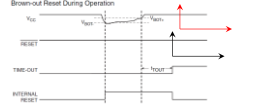
Watchdog Reset. MCU je resetované, ak Watchdog Timer napočíta nastavenú periódu a je jeho činnosť je povolená.



External Reset. MCU je resetované, ak sa na pino RESETE objaví nízka úroveň signálu na čas dlhší ako je minimálny.



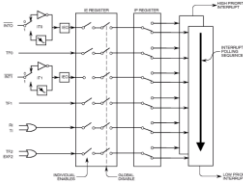
Brown-out Reset. MCU je resetované, a napájacia napätie VCC poklesne pod hodnotu (VBOT) a Brown-out Detector je povolený.



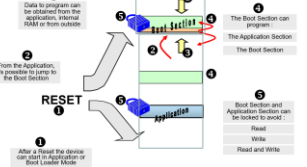
7

Interrupt - Prerušená.

Takto sú zapojeny prenosovacie podsystemy procesora 8051

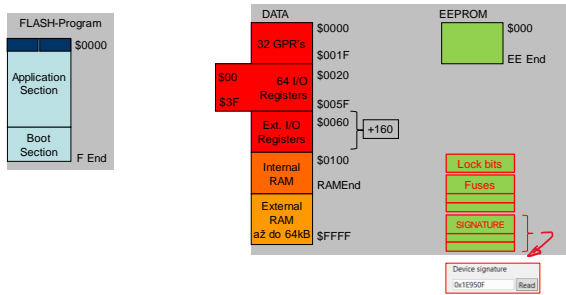


Self Programming Security



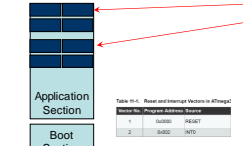
8

AVR mikrokontroler – Mapa pamäte



9

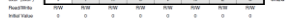
Interrupt - Prerušená.



Interrupt No.	Program Address	Name	Interrupt Definition
1	0x0001	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0004	INT0	External Interrupt Request 0
3	0x0008	INT1	External Interrupt Request 1
4	0x000C	PCINT0	Pin Change Interrupt Request 0
5	0x0010	PCINT1	Pin Change Interrupt Request 1
6	0x0014	PCINT2	Pin Change Interrupt Request 2
7	0x0018	WDT	Watchdog Timeout Interrupt
8	0x001C	TIMERC_COMPB	TimerCounter0 Compare Match B
9	0x0020	TIMERC_COMPA	TimerCounter0 Compare Match A
10	0x0024	TIMERC_OVF	TimerCounter0 Overflow
11	0x0028	TIMERC_CAPT	TimerCounter0 Capture Event
12	0x002C	TIMERC1_COMPA	TimerCounter1 Compare Match A
13	0x0030	TIMERC1_COMPB	TimerCounter1 Compare Match B
14	0x0034	TIMERC1_OVF	TimerCounter1 Overflow
15	0x0038	TIMERC1_COMPA	TimerCounter1 Compare Match A
16	0x003C	TIMERC1_COMPB	TimerCounter1 Compare Match B
17	0x0040	TIMERC1_OVF	TimerCounter1 Overflow
18	0x0044	SPM_SRZ	SPM Error Transfer Complete
19	0x0048	USART_RXC	USART Rx Complete
20	0x004C	USART_TXC	USART Tx Complete
21	0x0050	USART_RXD	USART Receiver Ready
22	0x0054	AD_CONVERT	ADC Conversion Complete
23	0x0058	EE_READY	EEPROM Ready
24	0x005C	ANALOG_COMP	Analog Comparator
25	0x0060	TWI	2-wire Serial Interface
26	0x0064	SPM_READY	Store Program Memory Ready

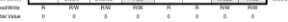
10

SREG - AVR Status Register



Bit 7 - I: Global Interrupt Enable
Ak bit GIE vynútiť sa zaškrtnú všetky prerušenia. Ak je bit GIE nastavený nulu, nie je možné jednotlivé povoliť, resp. zakázať jednotlivé zdroje prerušenia. I bit sa núdže hardwarovo po vyvolaní prerušenia a je nastavený inštrukciou RETI, aby sa umožnila opätovná ďalších čakajúcich prerušenia. Bit I môže byť nastavený a mazať programovo pomocou inštrukcií SEI a CLI.

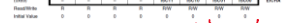
MCUCR - MCU Control Register



Bit 1 - IVSEL: Interrupt Vector Select
Ak je bit IVSEL nastavený na log. 0, prerušovacie vektory sú umiestnené na začiatok Flash pamäte. Ak je bit IVSEL nastavený na log. 1, prerušovacie vektory sú umiestnené na začiatok Boot Loader časti Flash pamäte.

Bit 0 - IVCE: Interrupt Vector Change Enable
Ak chcéme zmeniť IVSEL, bit IVCE musí byť nastavený na log. 1. Bit IVCE sa núdže hardwarovo sťažiť SC po nastavení tohto bitu. Pri nastavení sa zakáza všetky prerušenia.

ISCRA - External Interrupt Control Register A



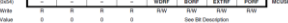
Bit 3 - WDRF: Watchdog System Reset Flag
Tento bit sa nastaví, ak je reset generovaný Watchdog-om. Vynúti sa pri Power-on Reset alebo zpisom log. 0 do tohto bitu.

Bit 2 - BORF: Brown-out Reset Flag
Tento bit sa nastaví, ak je reset generovaný Brown-out. Vynúti sa pri Power-on Reset alebo zpisom log. 0 do tohto bitu.

Bit 1 - EXTRF: External Reset Flag
Tento bit sa nastaví, ak je reset generovaný External Reset. Vynúti sa pri Power-on Reset alebo zpisom log. 0 do tohto bitu.

Bit 0 - PORF: Power-on Reset Flag
Tento bit sa nastaví, ak je reset generovaný Power-on. Tento bit sa vynúti len zpisom log. 0 do tohto bitu.

MCUSR - MCU Status Register

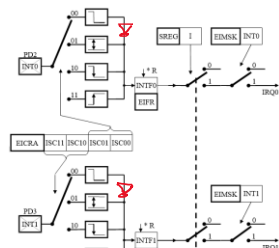


Bit 1 - INT0: External Interrupt Request 0 Enable
Lokálne povolenie/zakázanie externého prerušenia.

11

V podstate máme dva typy prerušenia:

- je vyvolaný úrovňou signálu
- je vyvolaný zmenou signálu.



ISCx1	ISCx0	Description
0	0	The low level of INTx generates an interrupt request.
0	1	Any logical change on INTx generates an interrupt request.
1	0	The falling edge of INTx generates an interrupt request.
1	1	The rising edge of INTx generates an interrupt request.

12

Odozva na prerušenie, AVR

Odozva na požiadavku o prerušenie ktoréhokoľvek prerušenia trvá minimálne 4 SC. Po tomto čase sa začne vykonávať obsluha prerušenia.

Počas štyroch SC sa uloží obsah PC do Stacku.

Do PC sa zapíše adresa vektoru obsluhy prerušenia a vykoná sa inštrukcia **jump**. Toto trvá 3 SC.

Ak sa **interrupt** objaví počas viac cyklovej inštrukcie, táto sa najskôr dokončí a potom sa začne obsluhovať prerušenie.

Ak sa prerušenie objaví počas **SLEEP** módu, odozva sa predlžá o 4 SC. Počas tejto doby „start-up“ MCU zistí z čoho sa to vlastne prebúdzá.

Návrat z podprogramu prerušenia trvá 4 SC. Počas tejto doby sa vyberú zo zásobníka dva byty – obnoví sa PC. **SP** sa inkrementuje o 2 a znovu sa nastaví bit IV registri **SREG**. Ak existuje počas inštrukcie **RETI** nejaká neobslužená požiadavka o prerušenie, najskôr sa vykoná **RETI** potom jedna inštrukcia z hlavného programu a potom sa môže začať realizovať obsluha prerušenia.

13

13

Existuje niekoľko postupností inštrukcií pri vykonávaní ktorých musí byť prerušenie zakázané:
Např.: počas zápisu do **EEPROM**. Zápis do **EEPROM**, vnútornej, pári medzi tzv. atomické operácie.

C Code Example

```
char cSREG;
cSREG = SREG;          /* store SREG value */
/* disable interrupts during timed sequence */
__disable_interrupt(); // CLI(), assembler instr.: cli
EEDR |= (1<<EEMPE); /* start EEPROM write */
EEDR |= (1<<EEMPE); /* restore SREG value (I-bit) */
SREG = cSREG;
```

Ak použijeme inštrukciu **SEI** na povolenie prerušenia, inštrukcia nasledujúca za **SEI** bude vykonaná, a to aj v tom prípade, že bude existovať požiadavka o prerušenie.

C Code Example

```
__enable_interrupt(); /* set Global Interrupt Enable, assembler instr.: sei */
__sleep(); /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
```

Voľne povedané, inštrukcia nasledujúca po inštrukcii „prerušovacej“ sa k požiadavke o prerušenie správa „atomicky“. Najskôr sa vykoná inštrukcia **SEI**, **RETI**, potom sa vykoná nasledujúca inštrukcia a potom sa začne obsluhovať prerušenie ak je povolené.

14

14