

Přerušovací podsystemy mcu

Přerušení, obsluha, vektor, povolení,
příznak, priorita, kontext, zpracování,
přerušovací systém Atmel AVR

Přerušení

- Událost
- Pomocí přerušovacího systému reagujeme na události
 - Hardwarové (nejčastější)
 - Změn hodnoty na pinu (externí přerušení)
 - Přetečení čítače/časovače
 - Překročení hodnoty u AD komparátoru
 - Dokončení převodu u analogově digitálního převodníku
 - Příjem nebo odeslání znaku u sériové komunikace
 - Softwarové
 - Neplatná adresa
 - Neplatný operační znak
- U PC přes přerušení můžeme volat funkce OS
- Reagujeme na událost tím, že se přeruší program a předá se řízení na speciální adresu
 - Na adrese je buď přímo uložený obslužný podprogram anebo skok nebo adresa, kde se program nachází

Obsluha

- Podprogram, co reaguje na událost

Vektor

- Adresa kam se předá řízení

Povolení

- Každý interrupt (událost na kterou se dá reagovat pomocí přerušení) se musí povolit a po resetu je zakázána
- Každá událost má speciální povolovací bit a zároveň po resetu jsou všechny interrupty zakázány
- Global Interrupt Enable
 - V registru Status (často v příznakových registrech)
 - Jeden z bitů slouží ke globálnímu zakázání přerušení
 - Když vynulujeme, tak zakážeme všechny
 - Používá se v okamžiku, kdy se spustí obsluha přerušení a nechceme aby obsluha přerušení byla přerušena jinou obsluhou přerušení

Příznak

- Každý zdroj přerušení má i interrupt flag
- Pokud vzniklo přerušení, tak se nastaví na 1
- Nulovat se dá nulovat SW (kdykoliv) a HW (některé systémy umí mazat automaticky, pokud se spustí obsluha přerušení)
- Kdybychom se nenuloval, tak by se přerušení volalo furt dokola

Priorita

- Pořadí, ve kterém se zpracovávají události a zároveň pokud probíhá obsluha přerušení s nižší prioritou a nastane přerušení s vyšší prioritou, tak se obsluha s nižším přerušením přeruší
- Dvě možnosti:
 - HW
 - Registry, kde interruptu můžeme nastavit prioritu
 - Pořadí, v jakém jsou interrupt vektory umístěny v programu, určuje prioritu
 - SW

Kontext

- Při běhu programu vznikne událost, tak se zastaví program a je předán na adresu, kde se má na přerušení SW reagovat, instrukce mohou mít vliv na obsah registrů (status), poté se vrátíme do programu s modifikovanými registry
- Hlavní program s tím obsahem mohl pracovat, ale je přepsán
- Obsluha by měla zajistit, že po skončení obsah zůstane nezměněn => kontext
- Na začátku uloží do zásobníku obsah registrů jejichž obsah modifikuje a před návratem původní obsah (kontext) obnoví, aby nenarušila přerušený program

Zpracování

- Během každého instrukčního cyklu se vyhodnocuje jestli nevzniklo přerušení, pokud ano a je povolené, tak se obsah instrukce uloží na zásobník a do program counteru se nahraje příslušný interrupt vektor, poté skok na obsluhu přerušení, zároveň se zakáže global interrupt enable, provede se obsluha přerušení, součástí instrukce návrat z přerušení je povolení global interrupt enable, ze zásobníku se vytáhne adresa kam se máme vrátit, ta se nahraje do program counter a provede se jedna následující instrukce
- Pokud instrukce zabírá více než jeden instrukční cyklus, tak se vždycky dokončí a nezastaví se uprostřed

Přerušovací podsystémy u Atmega

- Přerušovací vektory (adresy kam se předává řízení na začátku paměti programu)
- Přerušovací vektory (adresy vedoucí do paměti programu) jsou odstupňované po dvou

1	0x0000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	INT2	External Interrupt Request 2
5	0x0008	INT3	External Interrupt Request 3

-
- Z důvodu, aby se tam vešel nepodmíněný skok na místo, kde je umístění podprogramu
- Z hlediska počtu a umístění interrupt vektorů dělíme přerušovací systémy dělíme na 3 skupiny
 - Každý zdroj přerušení (událost) má vlastní interrupt vektor (Atmega)
 - Většina událostí má vlastní interrupt vektor, ale sériová komunikace sdílí vektor jak pro odesílání, tak pro přijímání
 - Jeden interrupt vektor (vždy se skáče na stejnou adresu), vyhodnocuje se SW kdo událost vyvolal (pomalejší, ale kvůli vyhodnocování lze stanovit prioritu)

