

06 - Přerušovací podsystém mcu

Přerušení

- přerušovací systém reaguje na určitou událost
 - **hardwarové události (v mcu častější)**
 - změna hodnoty na I/O pinu - externí přerušení
 - přetečení čítače / časovače - interní přerušení
 - softwarové události
 - neplatná adresa
 - neplatný operační znak

Obsluha

- reaguje na událost
- přeruší se běžící program
- řízení se předá na nějakou speciální adresu (na vektor)
 - na této adrese je buď řídicí podprogram
 - nebo se na adrese nachází skok na tento podprogram (častější)

Přerušovací vektory

- **vektor = adresa, kam se předá řízení**
- interrupt vektory
- stupňují se po dvou adresách
 - aby se tam vešel skok na místo podprogramu
 - adresy 0x0000, 0x0002, 0x0004
- dělí se na:
 - **každá událost má vlastní interrupt vektor - ATmega64**
 - jen některé události mají vlastní interrupt vektor, některé ho sdílejí (kombinovaný režim) - Intel
 - existuje pouze jeden interrupt vektor pro celé mcu, poté nutnost softwarově vyhodnotit - Microchip

Povolení

- každá událost se musí povolit
- po resetu jsou všechny defaultně zakázané
- v příznakovém registru je bit, který globálně ovládá všechny události v mcu (povolit/zakázat)
 - použije se v případě, že je nějaká událost vyvolána
 - aby ji nenahradila nějaká další, než se dokončí její řízení

Příznak

- každá událost má vlastní interrupt flag (příznak)
 - jeden příznak je povolení přerušení
 - druhý příznak je informace, že došlo k přerušení
- příznaky lze nulovat hardwarově i softwarově
 - softwarově lze smazat kdykoliv
 - hardwarově ho dokáží některé systémy mazat samy

Priorita

- pokud vznikne událost když probíhá řízení jiné
- 2 možnosti řešení:
 - hardwarově
 - softwarově
- u některých systému lze každému interruptu nastavit prioritu

- u **ATmega64** a **většiny mcu** - pořadí interrupt vektorů v paměti určuje prioritu
- pokud přijde žádost na událost s větší prioritou když probíhá událost s menší, některé systémy ji přestanou řešit
 - nejprve vyřeší tu s větší prioritou

Kontext

- po přerušení se program dostane na určitou adresu
- na této adrese se musí na přerušení softwarově reagovat
 - probíhají určité instrukce
 - mohou mít vliv na obsah registrů
- nastává problém, **že se po skončení řešení problému vrátíme zpět s modifikovanými registry**
- obsluha přerušení musí zajistit, že obsah registrů, se kterými pracuje, zůstane nezměněný
- obsluha přerušení na začátku řešení události uloží obsah registrů, se kterými bude pracovat
 - původní obsah registrů se nazývá **kontext**
- na konci řešení vrátí hodnoty zpět

Zpracování

- během každého instrukčního cyklu se vyhodnocuje, zda nastala nějaká událost
- pokud vzniklo a je povolené, obsah následující instrukce se uloží
- do program counteru se nahraje příslušný **interrupt vektor**
 - přijde skok na adresu obsluhy přerušení
- zapíše se 0 do **global interrupt enable** (zakázání)
- provede se obsluha přerušení
- zapíše se 1 do **global interrupt enable** (povolení)
- zpětně se najde poslední instrukce, kde jsme přerušili program a kam se máme vrátit
- program pokračuje dál

Přerušení u ATmega64

- každá událost má vlastní interrupt vektor
- pořadí interrupt vektorů v paměti určuje jejich prioritu