

Instrukční soubor mcu

Druhy instrukcí, popis. Struktura instrukce v paměti, CISC, RISC, architektura load-store, doba provádění instrukce. Popis příkladu zdrojového kódu v programu asm.

- K dispozici bude dokumentace + Instrukční soubor

Druhy instrukcí

- Nadpisy instrukční sady
 - Instrukce přesunu
 - Nejedná se o přesun, ale o kopírování
 - Aritmeticko logické
 - Pracují pouze s registry
 - Instrukce pro podmíněné a nepodmíněné větvení
 - Instrukce řídicí
 - Bitově orientované instrukce

Popis struktury instrukce v paměti

- Na začátku je operační znak (kód), pak jsou argumenty (operandy) nebo adresační režim
- Standardní operandy
 - Registry (RD)
 - Konstanty (K)
 - U větvení vede konstanta do paměti programu (k)
- Adresační režimy
 - Přímá adresa
 - X, Y, Z – registry pro nepřímou adresaci
 - Post inkrement
 - Predecrement
 - Displacement

CISC, RISC (Complex, Reduced Instruction Set Computer)

- RISC
 - 130 Instrukcí (u jednoduchých procesorů hodně, ale u složitějších málo)
 - Instrukce trvají stejnou dobu
 - Většina instrukcí mají stejnou délku
 - 32 pracovních registrů
 - Omezený počet adresačních registrů
 - Většinou kombinovaná s load-store (nemusí vždycky)
- CISC
 - Provedení instrukcí trvá rozdílnou dobu
 - Nemají stejnou délku
 - Mohou mít malý počet pracovních registrů, protože aritmeticko logické operace sahají přímo do paměti

Load-store

- Pokud chceme provést aritmeticko logické operace, tak musíme nejdříve pomocí operací přenosu hodnotu nahrát do registrů procesoru

Doba provedení instrukce

- Načtení operačního znaku a operandu z paměti (LOAD)
- Dekódování
- Načtení hodnot operandů (LOAD, READ)
- Provedení operace (EXECUTE)
- Výsledek zpětně ukládá do paměti, či registru (WRITE/STORE)

Popis příkladu zdrojového kódu v programu asm.

```
.DEVICE ATMEGA64

.DEF pocet = R16
.DEF oper1 = R23
.DEF oper2 = R24
.DEF temp = R5

.SET OPAK = 8

.DSEG
cislo1: .BYTE 1 ;obsahuje 0xB5
cislo2: .BYTE 1 ;obsahuje 0x6A
vysledek: .BYTE 1

.CSEG; paměť programu
.ORG 0x0000
LDS oper1, cislo1
LDS oper2, cislo2
CP oper1, oper2
BRSH preskoc01
PUSH oper1
MOV oper1, oper2
POP oper2
preskoc01: ;oper1 = oper2 =
ANDI oper1, 0x0F
ANDI oper2, 0xF0
OR oper1, oper2
NEG oper1
LDI pocet, OPAK
CLR temp ;tato instrukce byla chybně az po navesti zacatek
zacatek:
ROL oper1
BRCS preskoc02
INC temp
preskoc02:
DEC pocet
BRNE zacatek
STS vysledek, temp
;oper1 = vysledek =
.EXIT
```

- Pojmenování registrů:

```
.DEF pocet = R16
.DEF oper1 = R23
.DEF oper2 = R24
.DEF temp = R5
```

- Konstanta

```
.SET OPAK = 8
```

- DSEG - Datový segment

```
.DSEG  
cislo1: .BYTE 1 ;obsahuje 0xB5  
cislo2: .BYTE 1 ;obsahuje 0x6A  
vysledek: .BYTE 1
```

-
- Definice proměnných (jméno a kolik to má zabírat bytů)
- LDS (Z instrukční sady)
 - Load direct from SRAM
- BRSH
 - Branch if Same or Higher