

02 - Instrukční soubor mcu

Druhy instrukcí

Aritmetické a logické (ARITHMETIC AND LOGIC INSTRUCTIONS)

- ADD , SUB

Podmíněné / nepodmíněné větvení (BRANCH INSTRUCTIONS)

- JMP , CPC

Instrukce přesunu (DATA TRANSFER INSTRUCTIONS)

- MOV , LD

Bitově orientované (BIT AND BIT-TEST INSTRUCTIONS)

- SBI , ROL

Řídící (MCU CONTROL INSTRUCTIONS)

- NOP , SLEEP

Struktura instrukce

Operační kód	Operandy (+ adresační režim)	Příznaky
LD	Rd, X	---
ADD	Rd, Rr	Z, C, N, V, H

CISC

- Complex Instruction Set Computing
- rozvinutá instrukční sada
- instrukce mají proměnlivou délku vykonání
- mají menší počet registrů
 - aritmeticko-logické instrukce mohou sahát přímo do paměti

RISC

- Reduced Instruction Set Computing
- **ATmega64 je RISC**
- redukováná instrukční sada
- instrukce jsou optimalizované a **většinou trvají stejnou dobu a mají stejnou délku**
 - uvedeno v seznamu instrukcí v posledním sloupci
- nemusí mít menší počet instrukcí, ale má menší počet možných operací než CISC
- složité instrukce mohou být nahrazeny za více jednodušších
- místo násobení sčítání a bitový posun
- většinou méně adresačních režimů než CISC
 - ATmega64 má 4
 - adresační režim = možný operand instrukce
- **mají více registrů než CISC**
 - **protože mají často architekturu load-store**
 - aritmeticko-logické instrukce pracují pouze s registry, nemohou sahát přímo do paměti

Doba provádění instrukce (fáze provedení instrukce)

1. načtení operačního znaku z paměti
2. dekodování
3. načtení hodnot instrukce
4. provedení instrukce
5. uložení výsledku

- záleží na typu instrukce, některé kroky jsou poté vynechané

Popis příkladu zdrojového kódu v assembleru

- `DSEG` = datový segment
 - definice proměnných
- možná nepovinné