Instrukční soubory

= část abstraktního modelu počítače, která definuje, jak je CPU kontrolováno počítačem

# Druhy instrukcí

* Aritmetické a logické – sčítání, odčítání, násobení, logické operátory (ADD, SUB)
* Instrukce pro podmíněné a nepodmíněné větvení - skoky, porovnávání, skoky na základě příznaků (JMP, BREQ)
* Instrukce přesunu – Načtení dat z registru, SRAM, push/pop na zásobníku (MOV – move beetween registers, LDI – load immediate)
* Bitově orientované instrukce – Bitové posuny (SBI – Set Bit in IO register, LSL – logical shift left)
* Řídící instrukce – NOP (no operation), SLEEP

# Struktura instrukce

Operační znak, Adresační režim, Operandy (argumenty; registry nebo konstanty)

# CISC

* =Complex Instruction Set Computing
* rozvinutá instrukční sada
* instrukce mají proměnlivou délku vykonání
* menší počet registrů

# RISC

* Reduced Instruction Set Computing
* provedení instrukcí trvá stejnou dobu
* instrukce mají pevnou délku
* více pracovních registrů
* malý počet instrukcí
* ATmega 64 je RISC

# Load/Store architektura

* S operační pamětí pracují pouze instrukce pro čtení a zápis do paměti
* Všechny aritmeticko-logické operace pracují pouze s registry (žádná ALU operace nemůže sahat přímo do paměti, nejdříve se pomocí operací přesunu musí hodnota nahrát do registru)

# Doba provedení instrukce

1. Načtení operačního znaku z paměti
2. Dekódování
3. Načtení hodnot operandů
4. Provedení operace (execute)
5. Write/Store -výsledek uložen do paměti/registru

# Zdrojový kód v asm

