

10) ZÁKLADNÍ CYKLUS POČÍTAČE

Instrukční cyklus je režim, ve kterém pracuje procesor. Od spuštění až po vypnutí se procesor (přesněji jádro procesoru, protože procesor může mít více jader) vždy nachází v jednom bodě instrukčního cyklu.

Jedná se o proces, jakým jádro procesoru zpracovává strojové instrukce. Má-li procesor více fyzických jader, každé z nich pracuje nezávisle na ostatních a v jednom okamžiku se tedy každé může nacházet v jiné fázi instrukčního cyklu.

CPU každého počítače může mít různé cykly založené na různých sadách instrukcí, ale bude podobný následujícímu cyklu:

Kroky :

- 1) **Načtení instrukce** : Další instrukce je načtena z adresy paměti, která je aktuálně uložena v počítačové paměti programu a uložena do registru instrukcí. Na konci operace načtení počítač přejde na další instrukci, která bude přečtena v dalším cyklu.
- 2) **Dekódování instrukce** : Během tohoto cyklu je zakódovaná instrukce přítomná v instrukčním registru interpretována dekodérem.
- 3) **Přečtěte si účinnou adresu** : V případě instrukce paměti (přímá nebo nepřímá) bude fáze provádění během dalšího impulsu hodin. Pokud má instrukce nepřímou adresu, efektivní adresa je přečtena z hlavní paměti a všechny požadované údaje jsou získány z hlavní paměti, která má být zpracována, a poté umístěna do datových registrů (časový impuls: T_3). Je-li instrukce přímá, během tohoto hodinového impulsu se nedělá nic. Pokud se jedná o instrukce I / O nebo instrukce registru, operace se provádí během hodinového impulsu.
- 4) **Provedte instrukci** : Řídící jednotka procesoru předává dekodované informace jako posloupnost řídicích signálů příslušným funkčním jednotkám CPU, aby provedla akce požadované instrukcí, jako jsou hodnoty čtení z registrů, předává je do ALU k provedení matematických nebo logických funkcí a zápis výsledku zpět do registru. Pokud se jedná o ALU, odešle signál stavu zpět do CU. Výsledek generovaný operací je uložen v hlavní paměti nebo odeslán na výstupní zařízení. Na základě zpětné vazby od ALU může být počítač aktualizován na jinou adresu, od které bude načten další instrukce.

SREG-

Registr stavu obsahuje informace výsledku naposledy provedené aritmetické instrukce. Tyto informace lze použít pro změnu toku programu za účelem provádět podmíněné operace. registr stavu je aktualizován po všech ALU operacích, jak je uvedeno v referenci sady instrukcí. To je možná v mnoha případech odstranit potřebu použít vyhrazené instrukce porovnání, což vede k rychlejšímu a rychlejšímu kompaktnější kód.

Stav registru není automaticky uložen při zadávání rutiny přerušení a obnoven při návratu z přerušení. To musí být řešeno softwarem.

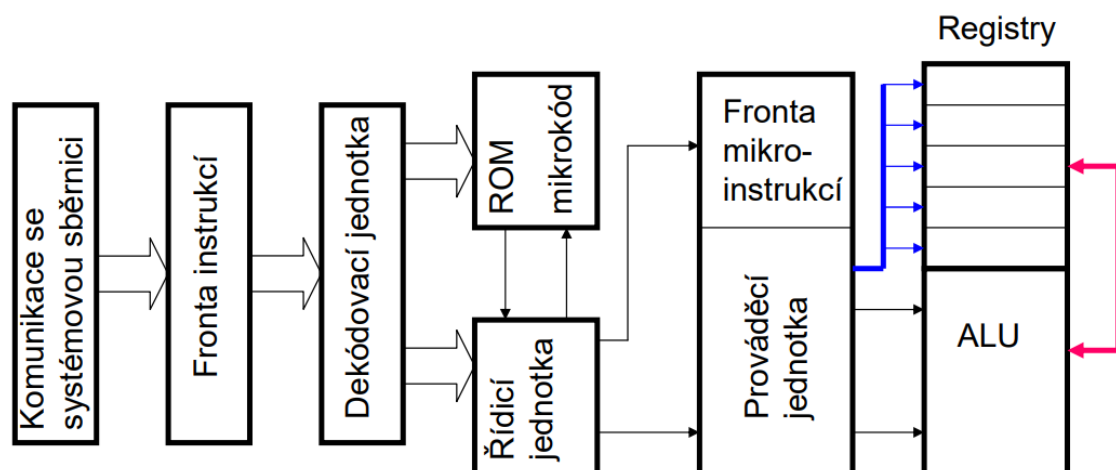
Registr stavu AVR - SREG - je definován jako:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ALU -- aritmetická logická jednotka

Vysoko-výkonový systém AVR ALU pracuje v přímém spojení se všemi 32 univerzálními registry. V jednom hodinovém cyklu jsou provedeny aritmetické operace mezi obecnými registry nebo mezi registrem a okamžitě provedeným. Operace ALU jsou rozděleny do tří hlavních kategorií - aritmetické, logické a bitové funkce. Některé implementace architektury také poskytují silný multiplikátor, který podporuje jak signed tak i unsigned násobení, a zlomkový formát.

Architektura procesoru CISC



Stavový stroj (state machine)

Obsahuje množinu stavů a množinu proměnných

-podle hodnoty proměnné jde buď do jednoho, nebo druhého stavu

*

-

Registr : zachytí výsledek/výstup kombinační logiky

MCU : sekvenční stavový stroj -> paměťové prvky, ukládající výstup z kombinační logiky

*
-

Vykonání instrukce

RISC – proudově zpracovává instrukce -> v každé pipe stahe je instrukce různě zpracována

SRAM – číst/psát po 1 bytu

Přístup do paměti SRAM- ST(store),LD(load),LDD

Přerušování – interrupts -> tabulka vektorů přerušování
-> obsluha přerušování (ISR)