

# Mikrokontroléry (mcu) - základní pojmy

Vysvětlení pojmu mikrokontrolér,  
harvardská a von Neumanova  
architektura, hlavní vlastnosti  
mikrokontrolerů Atmel AVR, popis  
blokového schématu

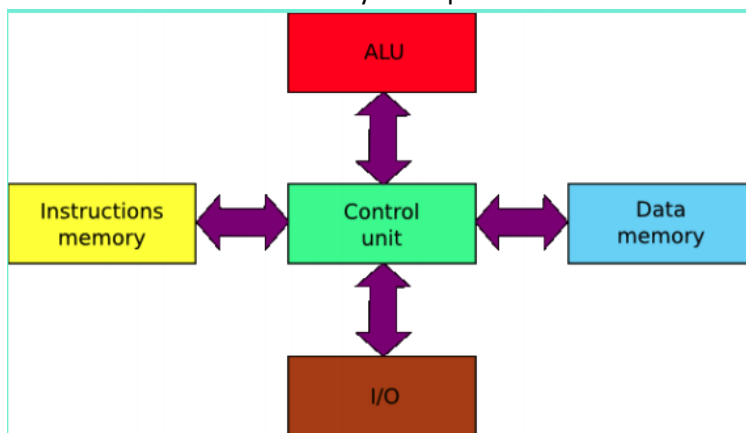
## Vysvětlení pojmu mikrokontrolér

- Jednočipový počítač je většinou monolitický integrovaný obvod obsahující kompletní mikropočítač
- Jednočipové počítače se vyznačují velkou spolehlivostí a kompaktností, proto jsou určeny především pro jednoúčelové aplikace jako je **řízení, regulace** apod.
- Často jsou součástí vestavěných systémů (embedded)
- Za jednočipový počítač je možno označit i hlavní integrovaný obvod v současných mobilních telefonech
- Jednočipový počítač je integrovaný obvod, který v sobě zahrnuje zpravidla vše potřebné k tomu, aby mohl dosáhnout celou aplikaci, aniž by potřeboval další podpůrné obvody.
- Především jde o paměť pro uložení programu (FLASH, EEPROM, nebo ROM) a operační paměť RAM pro uložení aplikačních proměnných a zásobníku
- Většina jednočipových počítačů také obsahuje rozsáhlou sadu podpůrných obvodů (typicky jde o bloky pro logické a analogové vstupy/výstupy, pro komunikační linky, pro rozdělení strojového času a další aplikační logiku)

## Harvardská architektura

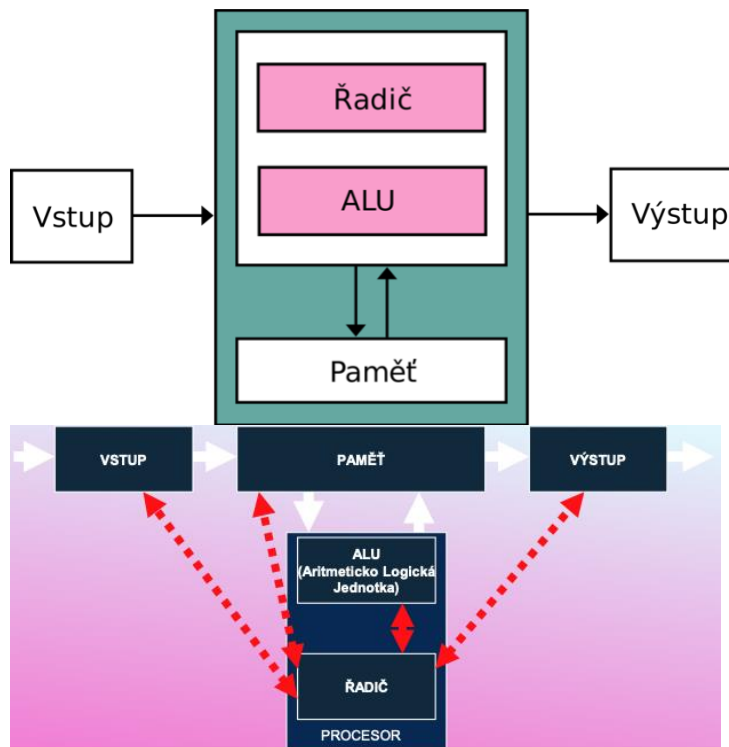
- Počítačová architektura, která fyzicky odděluje paměť programu a dat a jejich spojovací obvody.
- **Princip:**
  - Paměť je rozdělena na paměť programu a paměť dat tak, aby současně mohly dva operandy vstupovat do ALU
  - Pro zvýšení výpočetní výkonnosti se používá pipelining (zřetěžené zpracování instrukcí)
    - Rozdělení zpracování jedné instrukce mezi různé části procesoru a tím i umožnění zpracovávat více instrukcí najednou.
    - Fáze zpracování je rozdělena na 2 úseky:
      - Načtení a dekodování instrukce
      - Provedení instrukce a případné uložení výsledku
- Řízení jádra procesoru je odděleno od řízení vstupních a výstupních jednotek.
- **Paměť**
  - U harvardské architektury není potřeba mít paměť stejných parametrů a vlastností pro data a program
  - Paměti mohou být naprosto odlišné, mohou mít různou délku slova, časování, technologii a způsob adresování
  - V některých systémech se pro paměť programu používá paměť ROM (Read only memory), přičemž paměť dat vyžaduje typ paměti RWM (Read-Write Memory)
  - Dvojí paměť umožňuje paralelní přístup k oběma pamětím, což zvyšuje rychlost zpracování
  - Umístění programu v paměti ROM může významně přispět k bezpečnosti systému (program nelze modifikovat)
- Pro rychlé zpracování velkého toku dat se čím dál tím více používají počítače s paralelním zpracováním (Harvardská architektura)
- U této architektury, díky odděleným propojovacím obvodům a separátní paměti programu a dat, může procesor zároveň číst/zapisovat z/do paměti programu a paměti dat najednou

- **Rychlost:**
  - V posledních letech se rychlost procesorů zněkolikanásobila v poměru k rychlosti přístupové doby k hlavní paměti.
  - => Tendence zredukovat počet přístupů do hlavní paměti (pokud by každá instrukce běžící v procesoru vyžadovala přístup do paměti, zrychlení procesoru by nemělo žádný přínos pro výkon počítače, kvůli dlouhé přístupové době k paměti)
  - Paměť může být rychlejší (mnohem vyšší cena), řešením je paměť známa jako **cache**, která je velmi rychlá, ale je jí také mnohem méně než hlavní paměti.
  - Velikost vyrovnávací paměti (cache) je jeden z hlavních aspektů při určování rychlosti procesoru
- **Použití:**
  - Specializovaných DSP (digital signal processor) procesorech, obvykle audio/video technice.
    - Například procesor Blackfin od společnosti Analog Devices, Inc.
  - Převážně malé jednoúčelové mikrokontroléry, používané v mnoha běžných aplikacích, jako je například PIC od firmy Microchip Technology, Inc., a AVR společnosti Atmel Corp. Tyto procesory jsou charakteristické svojí malou kapacitou pamětí, ale především těží z výhod harvardské architektury a redukované instrukční sady (RISC), které zajišťují, že většina instrukcí může být vykonána během jednoho strojového cyklu.
  - Výhoda rozdělené paměti spočívá v možnosti použití různých typů pamětí, především však různé bitové šířky obou pamětí



## Von Neumanova architektura

- Jednoduché schéma programovatelného počítače, které **používá jednu sběrnici, na kterou jsou připojeny všechny aktivní prvky** (procesor, paměť, vstupy a výstupy)
- Popisuje počítač (automat) **se společnou pamětí pro instrukce i data** => zpracování instrukcí a dat je sekvenční oproti harvardské architektuře, která je typickým představitelem paralelního zpracování
- Procesor počítače se skládá z **řídící a výkonné** (aritmeticko-logické) jednotky
- **Řídící jednotka** zpracovává jednotlivé instrukce uložené v paměti
- Vlastní činnost s instrukčními daty provádí a má na starosti **aritmeticko-logická jednotka**



- Bílé šipky = tok dat
  - Červené šipky = řídící pokyny řadiče
  - Data putují ze vstupu do paměti, po případném zpracování ALU mohou být poslána na výstup
  - V paměti nejsou nejen data, ale i příkazy programu
  - ALU – si z paměti vybírá příkazy programu a vykonává je
  - Při jejich vykonávání může z paměti vybírat i další údaje (data) a může data do paměti i zapisovat
  - Řadič – koordinuje činnost celého systému
- Rychlost zpracování instrukcí dnešními procesory je výrazně vyšší než rychlost komunikace s pamětí
- **Komunikace s pamětí** se tak stává nejslabším článkem řetězu ve VN architektuře
- Řešením jsou paměťové **cache, což jsou rychlé mezi paměti**, do kterých se potřebná data a instrukce z pomalejší hlavní paměti načítají dříve, než jsou při zpracování potřeba
- **Princip:**
  - Do operační paměti se pomocí vstupních zařízení přes ALU umístí program, který bude provádět výpočet
  - Stejným způsobem se do operační paměti umístí data, která bude program zpracovávat
  - Proběhne vlastní výpočet, jehož jednotlivé kroky provádí ALU. Tato jednotka je v průběhu výpočtu spolu s ostatními moduly řízena řadičem počítače. Mezivýsledky výpočtu jsou ukládány do operační paměti
  - Po skončení výpočtu jsou výsledky poslány přes ALU na výstupní zařízení

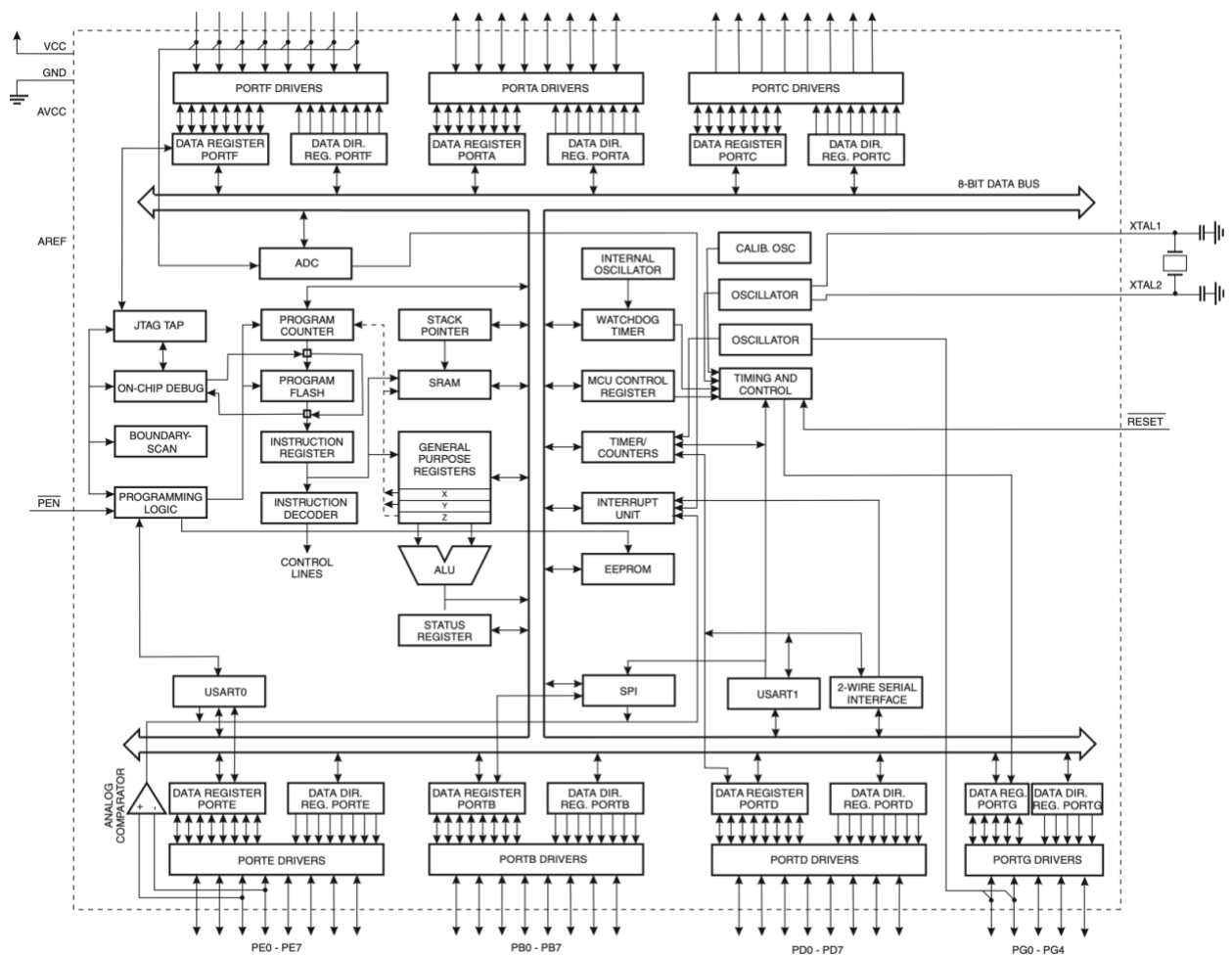
## Hlavní vlastnosti mikrokontrolérů Atmel AVR

- Architektura typu Harvard a RISC

- 32 identických 8bitových registrů pro všeobecné použití, které jsou všechny použitelné jako akumulátor
- Ukazatel na zásobník (stack pointer) a tři adresní ukazatele X, Y, Z, které jsou vytvářeny z registrů R26/27=X, R28/29=Y, R30/31=Z
- Pro registry ukazatelů Y a Z je možná indexace polí pomocí 6bitové relativní adresy (displacement)
- Všechny registry jsou adresovatelné prostřednictvím standardních paměťových přístupů
- Bitové adresování pro všechny registry
- S výjimkou čtyř 32bitových instrukcí mají všechny délku 16 bitů
- Lineární adresní prostor pro datovou paměť navzdory funkčnímu rozdělení
- Interní paměť:
  - FLASH pro program
  - SRAM pro registry, oblast I/O a data
  - EEPROM pro ukládání dat
- Externí paměť:
  - RAM rozšiřitelná v závislosti na typu MCU
- 64 paměťových míst vstupu/výstupu (prostřednictvím instrukce I/O nebo adresní přístup 0x20 až 0x5F)
- Ochrana před poklesem napájecího napětí (brown-out)
- Žádné další dělení hodinového cyklu v krystalu
- K dispozici jsou jádra RTOS
- AVR 1200 má jen 3úrovňový zásobník a žádnou přídavnou RAM
- Mnoho funkčních bloků a periférií podle typu MCU

## Popis blokového schématu

- **DOKUMENTACE BUDE K DISPOZICI**



- 
- 32 general purpose working registers přímo připojeny do ALU, dva nezávislé registry (mohou se použít za jeden cyklus oba)

	7	0	Addr.	
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

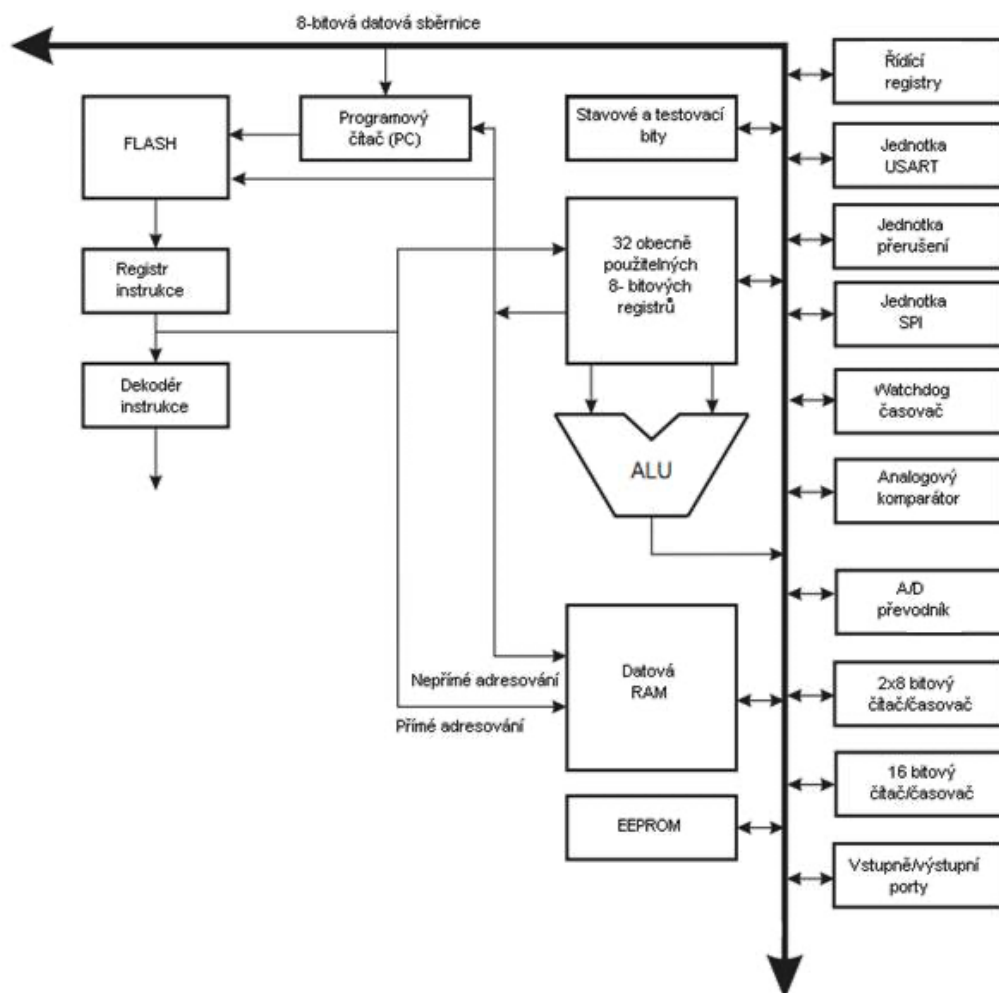
- 
- Program FLASH Read While Write
- 53 linek (pinů na připojení)
- Real Time Counter
- 8-channel
- 10-bit ADC (Analog to digital converter)
- Watchdog Timer - periferie, která resetuje systém při jeho zacyklení
- Sériová rozhraní:

- SPI
- IEEE std. 1149.1 (JTAG test interface – testování plošných spojů)
- Boundary scan - testování konektivity linek
- Status register:

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	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	

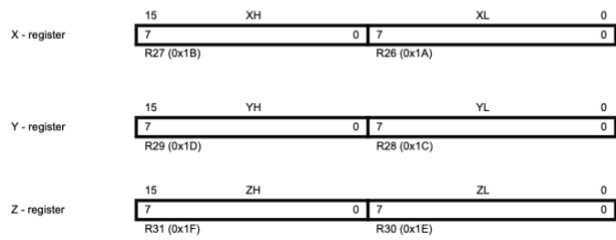
- Každý z jeho bitů má zvláštní význam
- Soubor příznaků
- Př. Instrukce podmíněných skoků, které provedou skok v závislosti na stavu konkrétních příznaků – tedy v závislosti na výsledky předchozích instrukcí
- Dělení příznaků:
  - Aritmeticko logické:
    - **Přenosu:** + a -, dojde-li k přenosu z nejvýznamnější bitu
    - **Vynulování:** výsledkem operace 0
    - **Znaménka:** Je nastavován, má-li výsledek operace nastaven nejvýznamnější bit (znaménkový), tedy je-li při interpretaci jako číslo ve dvojkovém doplňku záporný
    - **Přečtení:** pokud se nevejde do registru při počítání ve dvojkovém doplňku
    - **Parity:** Počet nastavených bitů výsledku poslední operace sudý, či lichý
  - Řídící
    - Krokování: po každé instrukci zavoláno přerušení
    - Povolení přerušení: povolení maskovatelných přerušení
- Přerušení vykonání instrukcí
- Data Bus:
  - Real time data



- 
- Harvard architektura:
  - Rozdílná paměť programu a dat
- Single level pipelining:
  - Když se zpracovává jedna instrukce, tak se následující instrukce předběžně načte
  - => to umožňuje že při každém cyklu se provést jednu instrukci
- Paměť programu je v Re programovatelný Flash paměti
- Fast-access register file obsahuje 32x 8bitových registrů (single clock access time)
  - Single-cycle ALU operation – jedna instrukce za jeden cyklus
  - V normální ALU, 2 operandy jsou výstup z registrů, provede se instrukce a výsledek se zapíše zpět do registru - v jednom cyklu
- 6 z 32 registrů se mohou použít jako ukazatele na adresování dat, jeden z nich se použít jako ukazatel na Flash paměť
- Tyto registry jsou 16bitový X-, Y- a Z- registry



Figure 5. The X-, Y-, and Z-Registers



- 
- ALU podporuje logické operace