# TVY – Terminologie

**Data** – Nějaká informace v digitální podobě  
**Informace** – kódovaná data, srozumitelná příjemci, 1bit - 1/0

Program je složen z instrukcí (instrukce = předpis k provedení činnosti)  
**Strojová instrukce** – elementární operace procesoru, Základní jednotka strojového kódu  
**Instrukční sada** (strojoví kód):

* skupina instrukcí kterým rozumí procesor
* Každá instrukce je tvořena ze dvou částí:
* Kód operace (co se bude dělat)
* Kód adresy (s čím se bude pracovat)

Program je v programovacím jazyce a předělá ho do stroj kódu. Dva typy překladačů programovacích jazyků:

* Kompilovaný (po přeložení se spustí)
* Interpretovaný (postupně se překládá a vykonává)

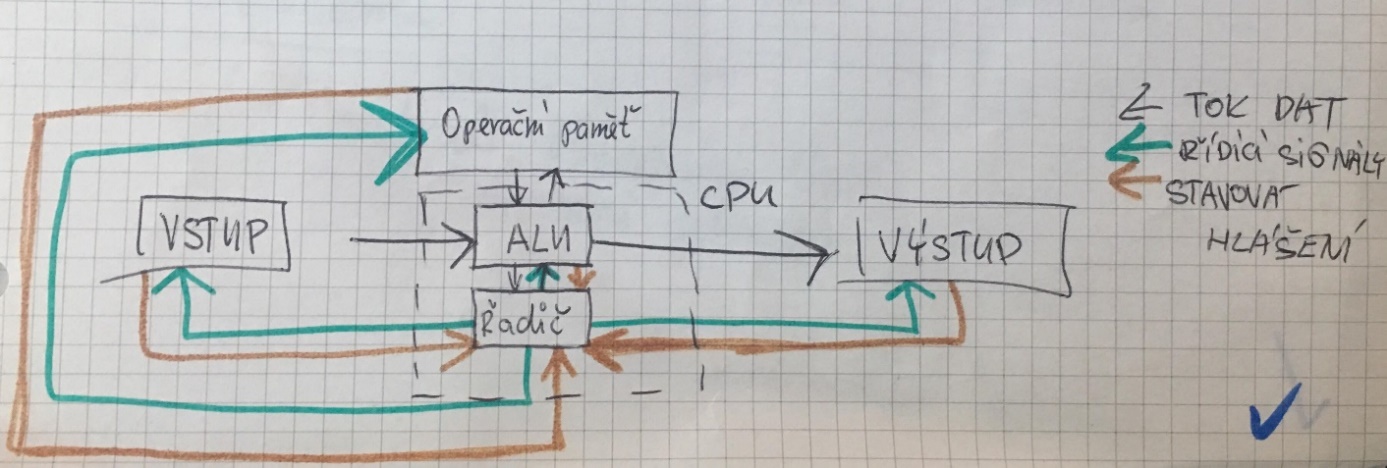
**Asembler** – jazyk symbolických adres

**Instrukce**:

* Aritmetická
* Logická
* Přesun – mezi registry
* Posun
* Řídící instrukce
* Výstupní instrukce

**Takt** – elementární část prováděného cyklu, nejmenší část strojové instrukce

## Von Neumanovo Schéma

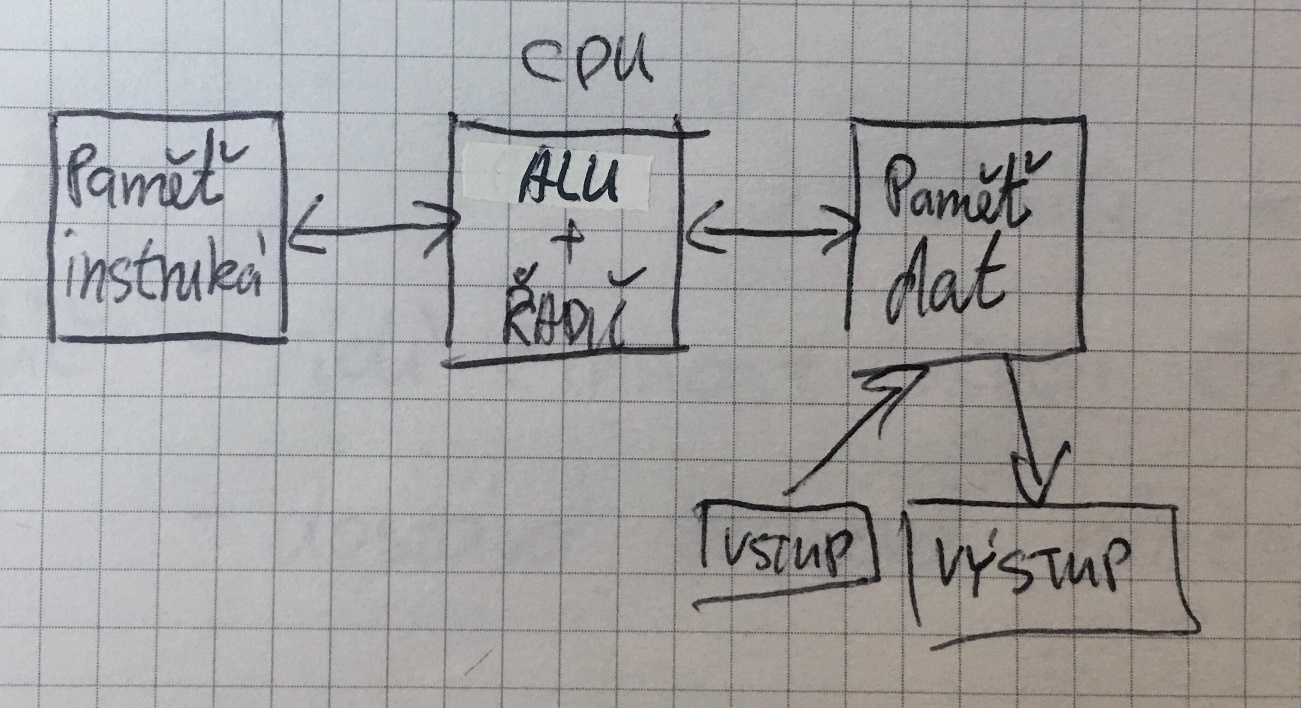
****

**ALU** – provádí aritmetické operace a logické

**Operační paměť** – rozdělená na buňky, **paměť je pro data a instrukce stejná** lze zapisovat jen data nebo instrukce. Nelže současně zapisovat protože používají stejnou sběrnici.

**Řadič** – řídí činnost počítače, odesílá řídící signály a přímá stavová hlášení

## Harvardské schéma



**Paměť pro Instrukce** – programy

**Paměť pro data** – data

* každá paměť se adresuje zvlášť
* číst lze paralelně

Dnešní procesory spojují obě tyto architektury. Uvnitř harvardská struktura, ale zvenku se chová jako Von Neumannovo schéma, protože načítá data pomocí jedné sběrnice.

**CPU** (Central Procesing Unit)

* centrální výpočetní jednotka
* úkolem je načítat instrukce a vykonávat program

**ALU**

* Počítá

**Operační paměť**

* (dočasné ukládání do paměti)
* místo, kam se ukládají programy

**Vstupní / Výstupní zařízení**

* zpracovává a zadává data

**Řadič**

* řídí činnost všech částí PC
* dostává stavová hlášení

# Součásti procesoru



Jádro – řadič + registry + ALU

CACHE = rychlá mezi paměť

* slouží pro načítání dat z pomalejších pamětí

**Řadič**

* dekóduje instrukce
* řadí instrukce(vnitřní pořadí)

**ALU** – aritmetické a logické operace

**Registr** – uchovává data se kterými se právě pracuje

**FPU**

* matematický koprocesor
* stará se o operace s plovoucí řádovou čárkou

**Proces**

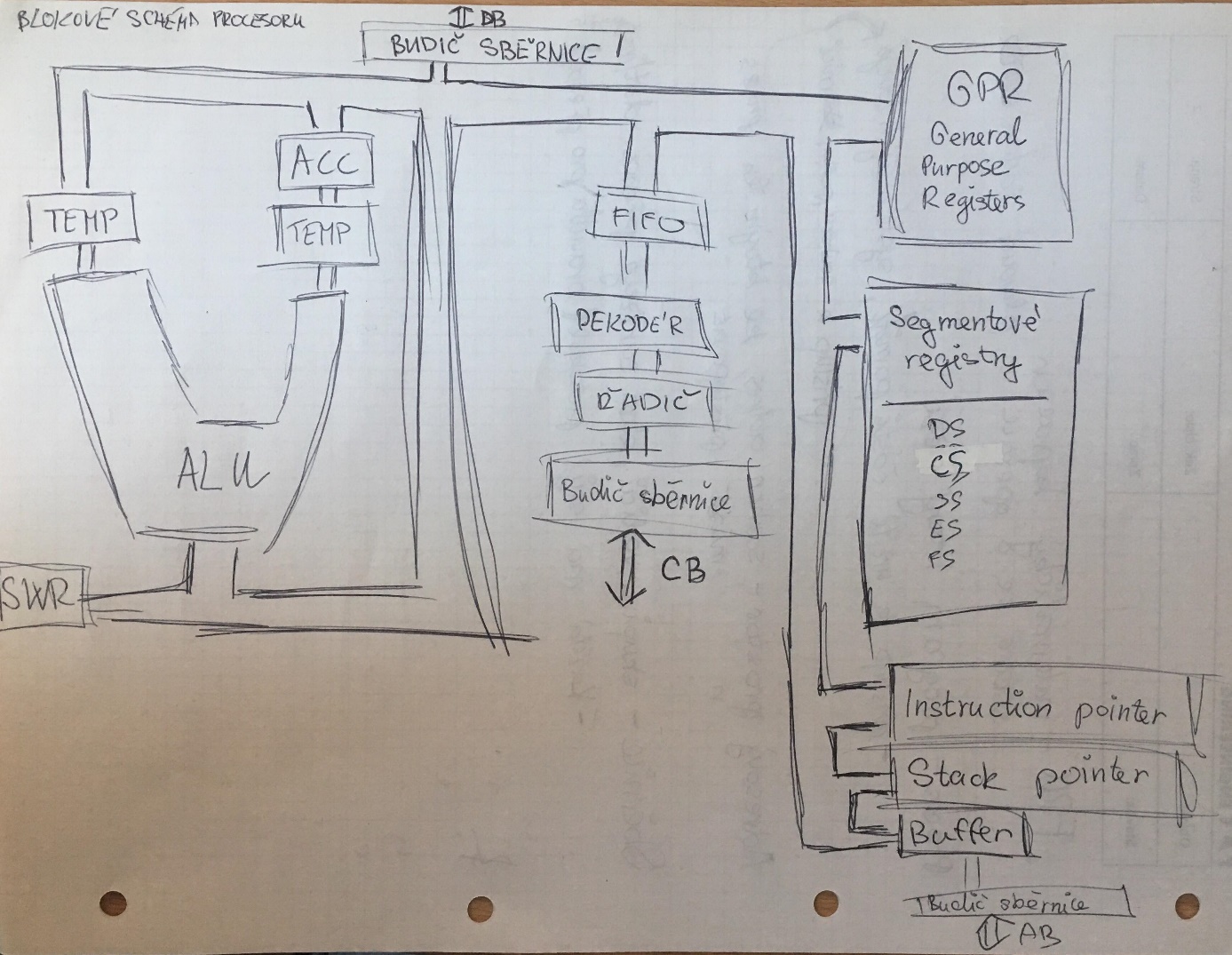
* program který běží
* má přístup k určité části paměti, systémových zdrojů (soubory, porty, sběrnice)

**Adresový prosto**r – souhrn adres, ke kterým sten proces může přistupovat

**Sběrnice**

* skupina vodičů která zajištuje přenos dat/instrukcí
* každá má svůj protokol(pravidla pro přenos)

## Blokové schéma procesoru



**ALU**-slouží k provádění

**SWE**

* Status Words Registrs
* Registr stavového slova
* Ví zda operace proběhla ok
* Je zde zapsán stav akumulátoru (ACC)

**TEMP**

* Registr
* Připojen k ALU, může být připojen k FPU
* Slouží k uložení 1 operandu (operand = hodnota se kterou se počítá)
* Slouží k dočasnému ukládání

**ACC**

* Akumulátor, střadač
* Zapisují se zde výsledky z ALU

Př.: Přičti adresu  
Hodnota z dané adresy se bude přičítat k hodnotě ACC a výsledek se zapíše do ACC

* ACC je u jednodušších procesorů, u složitějších CPU jsou datové registry, mají podobnou funkci
* Používá se jako implicitní operand pro různé operace
* Slouží pro uložení číselných hodnot

**Budič sběrnice**

* Zesiluje signál
* Odděluje provoz, co jde dovnitř a co jde ven

**Datová sběrnice** – umožnuje výměnu dat

**FIFO**

* Fronta, Frst In Frst Out
* Dopředu se nahrává několik instrukcí (většinou 8)
* Když nějakou nepotřebuje tak ji zahodí

**Dekodér** – rozkládá instrukce na mikroinstrukce

**Řadič** – řídí celou činnost

DB = DATA BUS = datová sběrnice

CB = Control Bus= řídící sběrnice

* Posílá příkazy jednotce, která je na adresovaná

**GPE** = General Putopse Register= obecně užitečné registry

* Pro data i instrukce
* Pro výsledky , aby nemusely do hlavní paměti

**Segmentové registry**

* Vypočítá se zde adresa instrukce pak se na CB
  + CS = Control Segment
    - Segment
    - CS +IP => adresa instrukce
  + SS = Stack Segment (stack = zaásobník)
  + DS = Data Segment
  + ES = Extra Segment
    - Např.: pro řetězce
  + FS = Free
    - Pro volné využití

**Instrukční ukazatel**

* ČÍRAČ INSTRUKCÍ <3 (MISS YOU HAJI)
* Obsahuje adresu instrukce, která se bude provádět v dalším taktu
* Automaticky se to o jednotu zvyšuje

**Stack pointer**

* Ukazatel zásobníku
* Ukazuje na vrchol zásobníku
* STACK = odkládací zásobník
* Ukládají se sem adresy návratových instrukcí

**Buffer** = vyrovnávací paměť

**AB** = Adress Bus = adresová sbernice

Code segment + Instruction Pointer = adresa

Cs + IP = číslo se kterým se bude pracovat

### Code segment???

Malé procesory mají místo Code segmentu Program counter, který obsahuje absolutní adresu, která se musí vypočítat.

Vezme se obsah těchto registrů a najde se v operační paměti adresa, z adresy se vezme adresa instrukce, která se bude provádět. Aby se mohla instrukce provádět musí být zapsána v CS a IP, celý program je sled instrukcí, jakmile se instrukce provede o jednotku délky se instrukce CS

Stack pointer + Stack segment = adresa kam byla naposledy zapsána adresa, ukazuje na vrchol zásobníků

Po zapsáni hodnoty na ni ukazuje, tak dlouho dokud, není přečtena

Ukládám-li zmenšuje se pointer

Čtu-li zvyšuje se pointer

Dno má nejvyšší adresu

**Fronata FIFO**

**Zásobník LIFO**

**Stack** = část operační paměti

* Každy program včetně OS má svůj stack

## Přerušení

* Schopnost procesoru přerušit právě vykonávaný program a začít vykonávat jiný(obsluhu přerušení)
* Častá věc při přepínání procesů

Typy přerušení:

* Vnější (asynchronní)
  + nečekané/neplánované
  + zmáčknutí klávesy
* Vnitřní (asynchronní)
  + Vykonává je sám procesor
  + Problém s vykonávanou instrukcí
  + Dělení 0
  + Nečekané
* SW (synchronní)
  + Speciální strojová instrukce
  + Očekávané
  + Vyplívá z podstaty programu
* Maskované
  + Slouží ovladačům
* Nemaskované
  + Výpadek zdrojů
  + Musí se provést