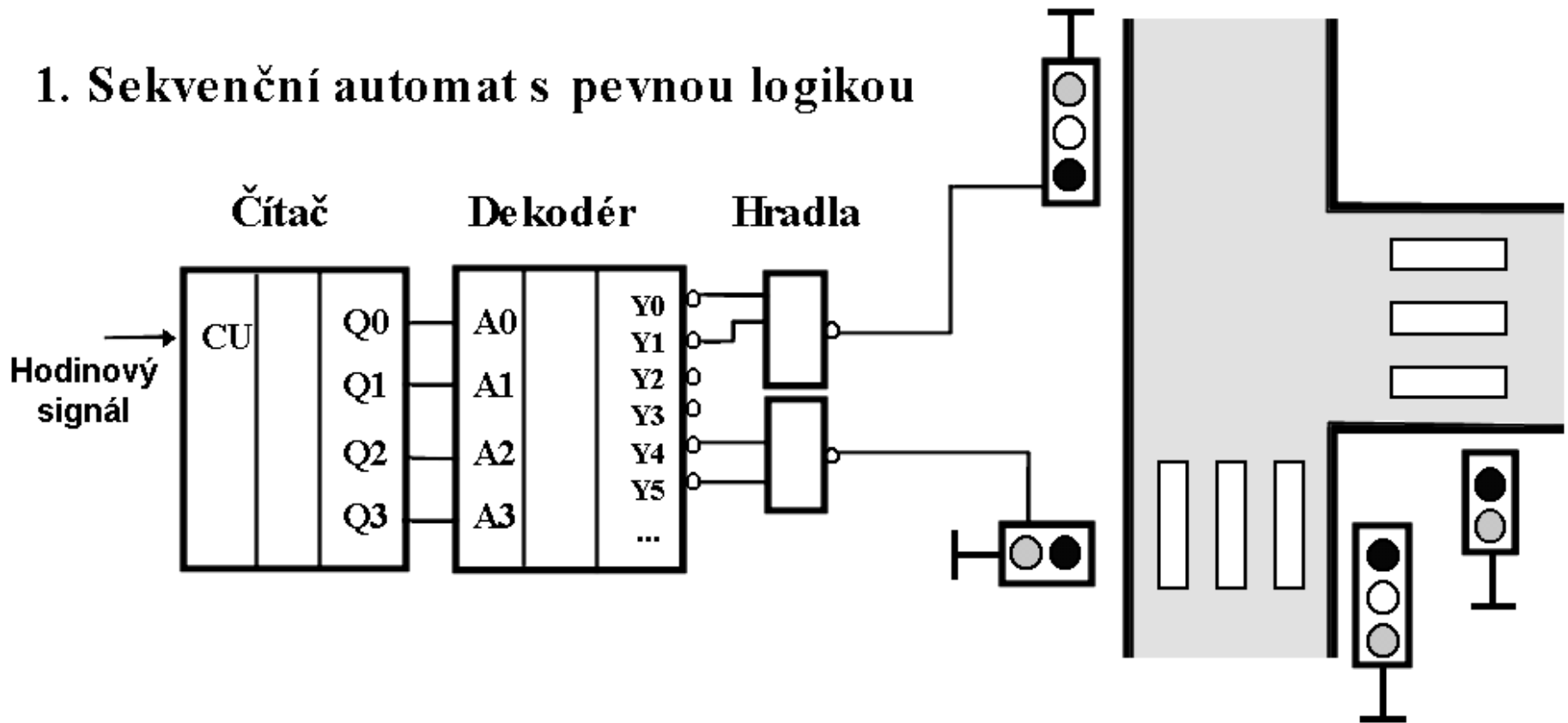


Procesory, mikroprocesory, procesory na FPGA

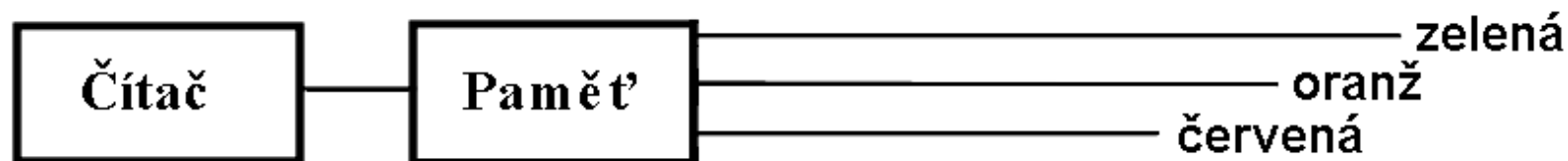
Prof. Ing. Ondřej Novák, CSc.
ITE

Od sekvenčních automatů k mikroprocesorům

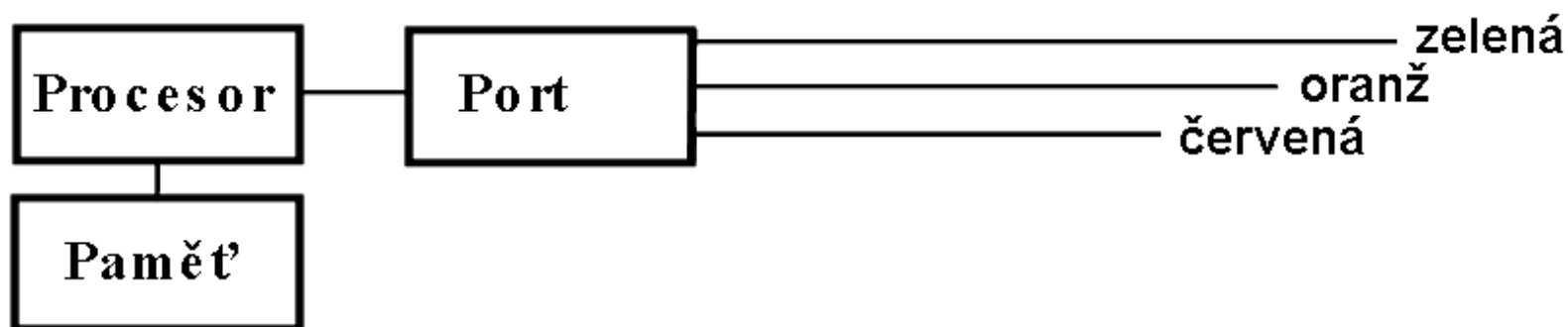
1. Sekvenční automat s pevnou logikou



2. Sekvenční automat s pamětí



3. Řídicí systém s mikroprocesorem



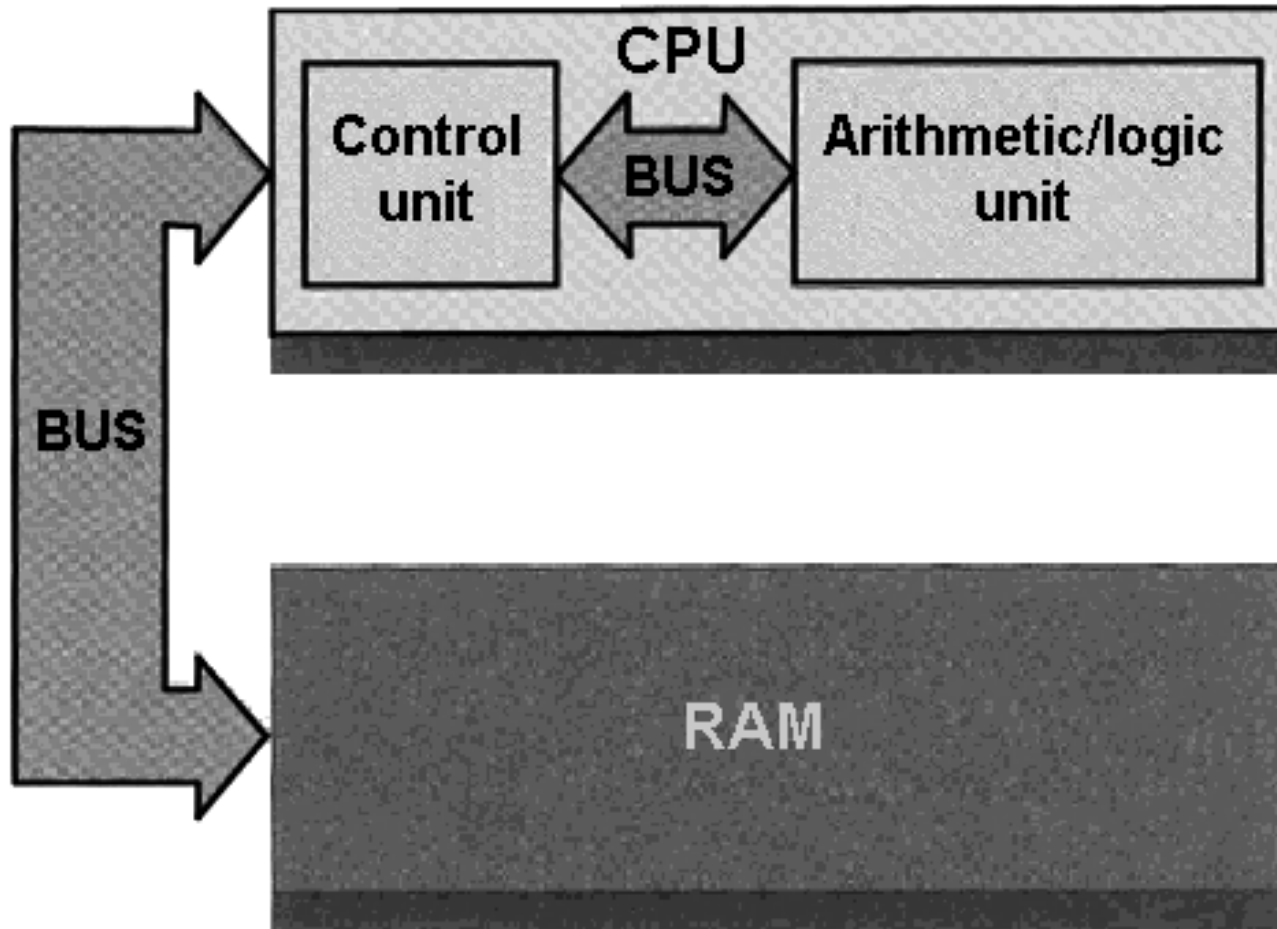
Architektura počítačů

- **Von Neumannovská, harwardská architektura**
- **dělení :**
 - generace
 - počet procesorů
 - proudové zpracování, paralelní zpracování
 - výkon (výkonové mixy)
 - použití (osobní, prac. stanice, mainframe, server, řízení)
 - CISC, RISC
 - netradiční architektury (maticové, řízené tokem dat, systolické systémy, neuronové počítače)

Výpočetní středisko vybavené počítačem typu mainframe

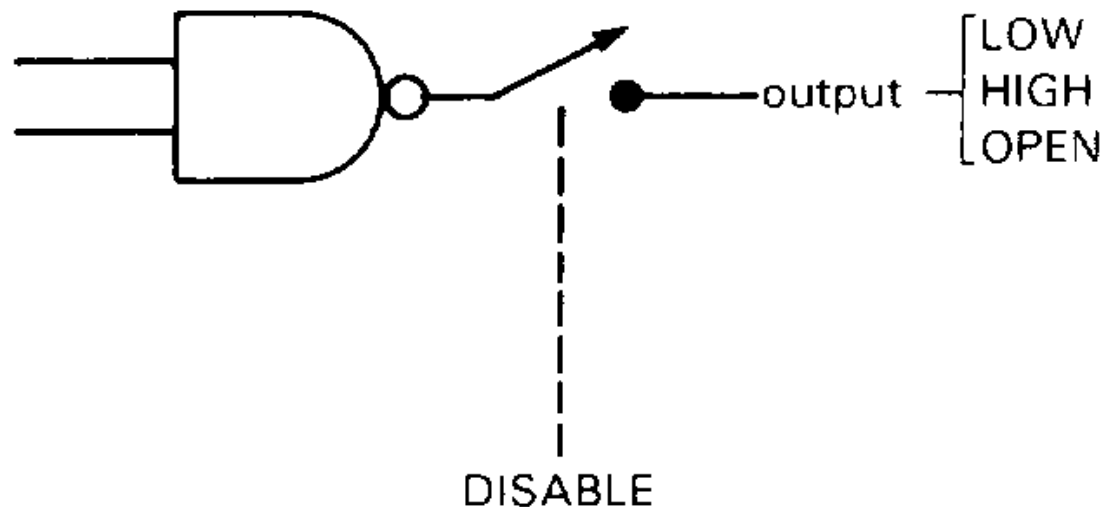


Sběrnice, základ komunikace

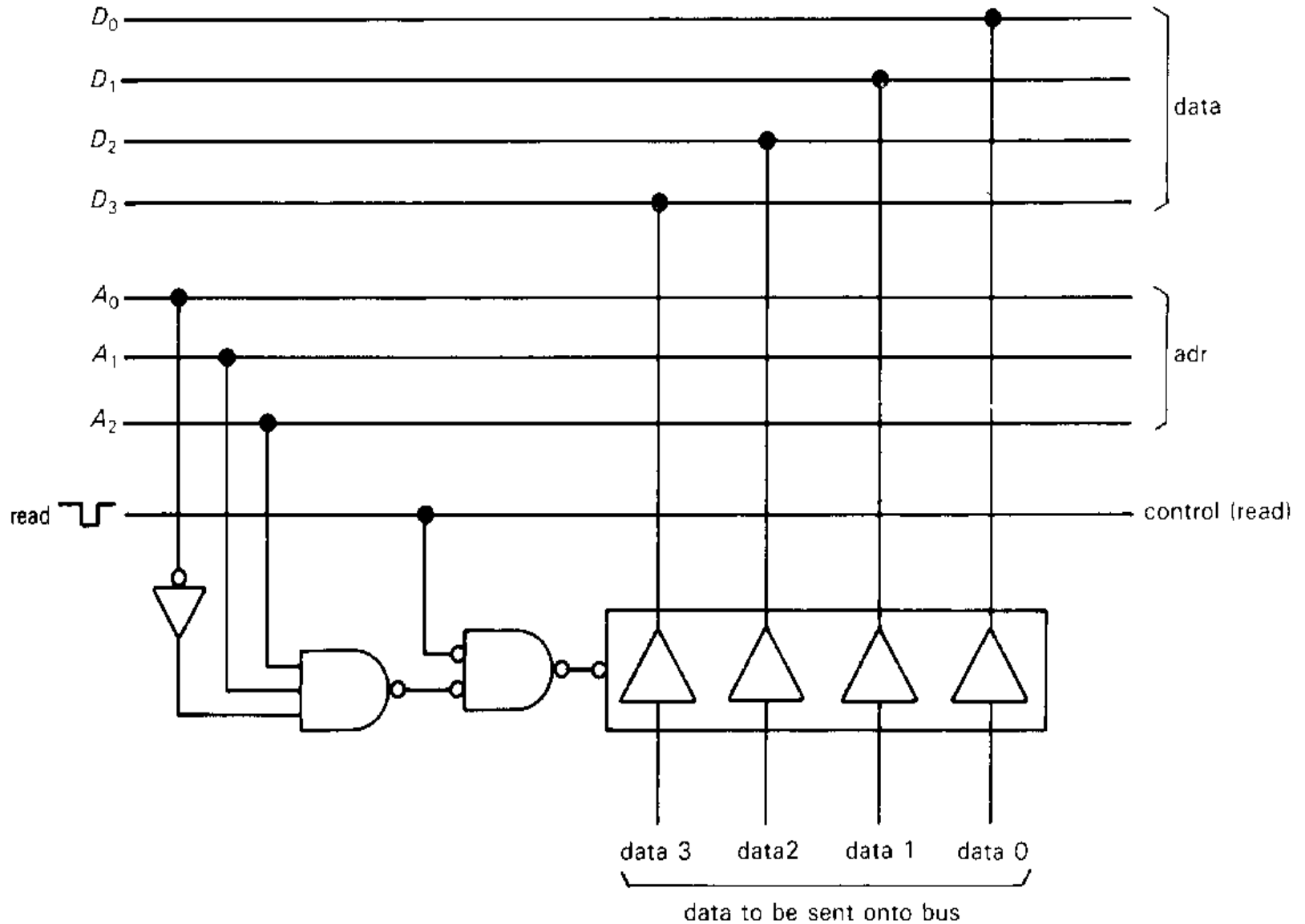


Sběrnice: skupina sdílených vodičů,
která slouží pro výměnu datových
slov

Třístavové výstupy – základ obousměrné
sběrnice



Sběrnice

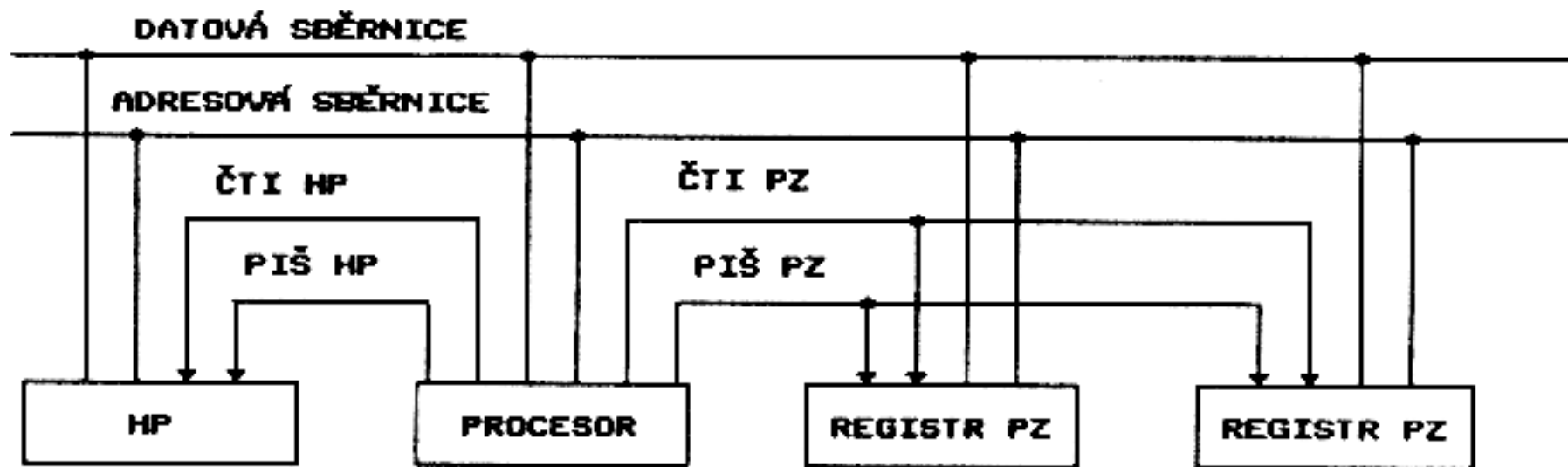


Vlastnosti sběrnice architektury

- + Úspora místa – omezení počtu vodičů
- + Možnost modifikace počtu připojených zařízení na sběrnici – využití nebo nevyužití některých adres
- zhoršení dynamických vlastností sběrnice při velkém počtu připojených zařízení a velké délce vodičů, nutnost dodržovat zásady návrhu (zákaz větvení, zakončovací odpory)

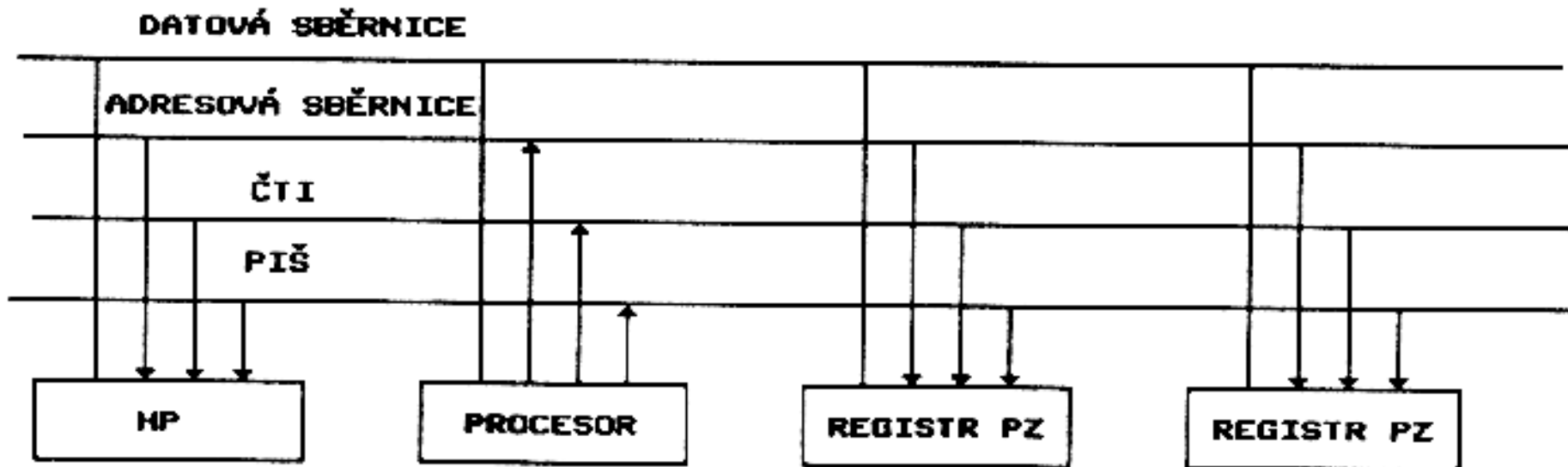
Řízení procesorem při odděleném adresovém prostoru (harvardská architektura)

- adresový prostor PZ oddělen od adres. prostoru HP,
- různé operace pro ovládání HP a PZ
- paměťový prostor je využit dvakrát,
- rozlišení, co se má provést, je dáno řídicími signály (PIŠ HP, PIŠ PZ, ČTI HP, ČTI PZ)
- používá INTEL



Řízení procesorem při společném adresovém prostoru (Von Neumannova architektura)

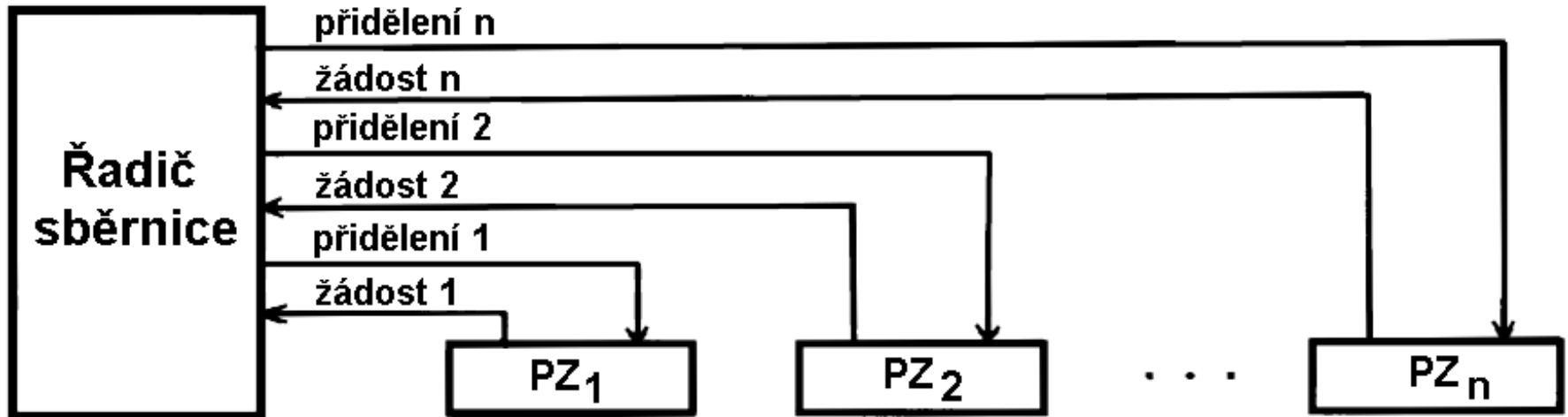
- zjednodušení operačního kódu procesoru,
odpadnou speciální operace pro styk s PZ
- vystačí se s operacemi Čtení a Zápis HP
- používá Motorola, DEC, ...



- přímé řízení VV operací je neefektivní, ztráty čekáním a opakovanými dotazy na to, jestli je zařízení připraveno
- vytvořeny prostředky, umožňující využít dobu čekání k provádění jiných operací
- nejdůležitější:
 - přerušení
 - přímý přístup do paměti
 - přenesení řídicích funkcí na kanál nebo periferní procesor

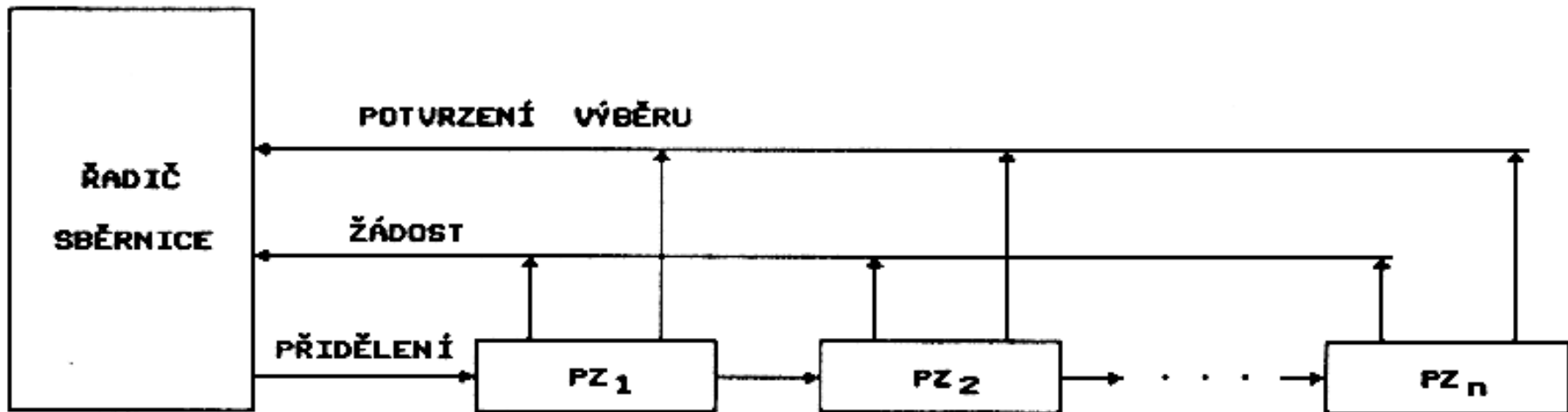
Samostatné žádosti o přidělení sběrnice

- vysílání po radiálních vodičích přímo do procesoru ke každé žádosti je přiřazeno jedno přidělení
- vyhodnocení je možno provést jednoduchým prioritním dekodérem



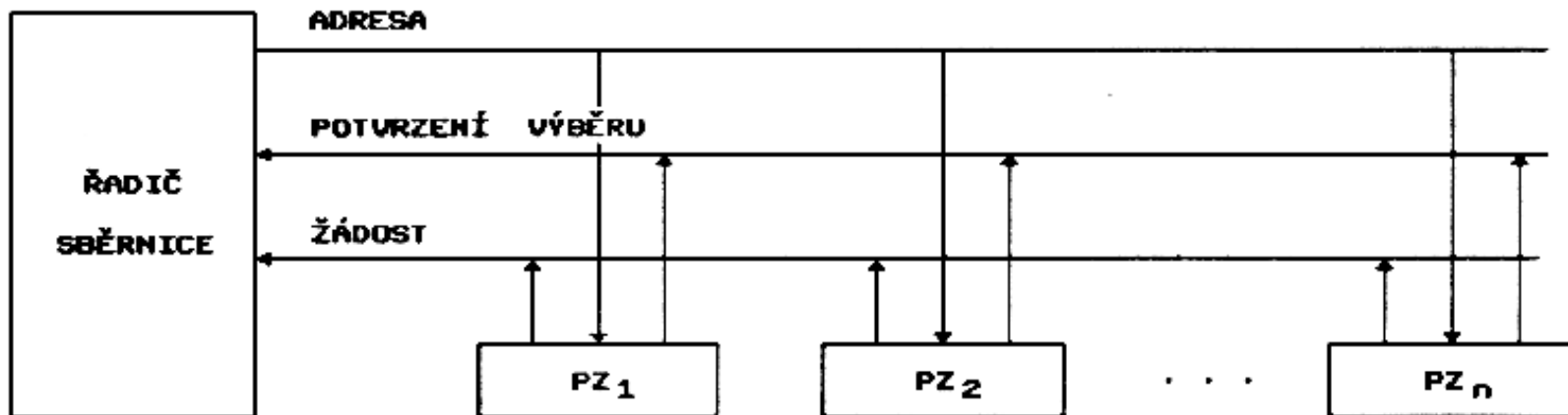
Zřetězení (Daisy Chain)

- žádosti vysílají všechny jednotky po společném vodiči
- žádosti jsou anonymní, není možno je procesorem vyhodnotit
- společný přidělovací impuls, procházející všemi jednotkami pořadí odpovídá pořadí priorit
- pokud jednotka žádala přidělení sběrnice, nepustí signál dál ostatní jednotky musejí čekat

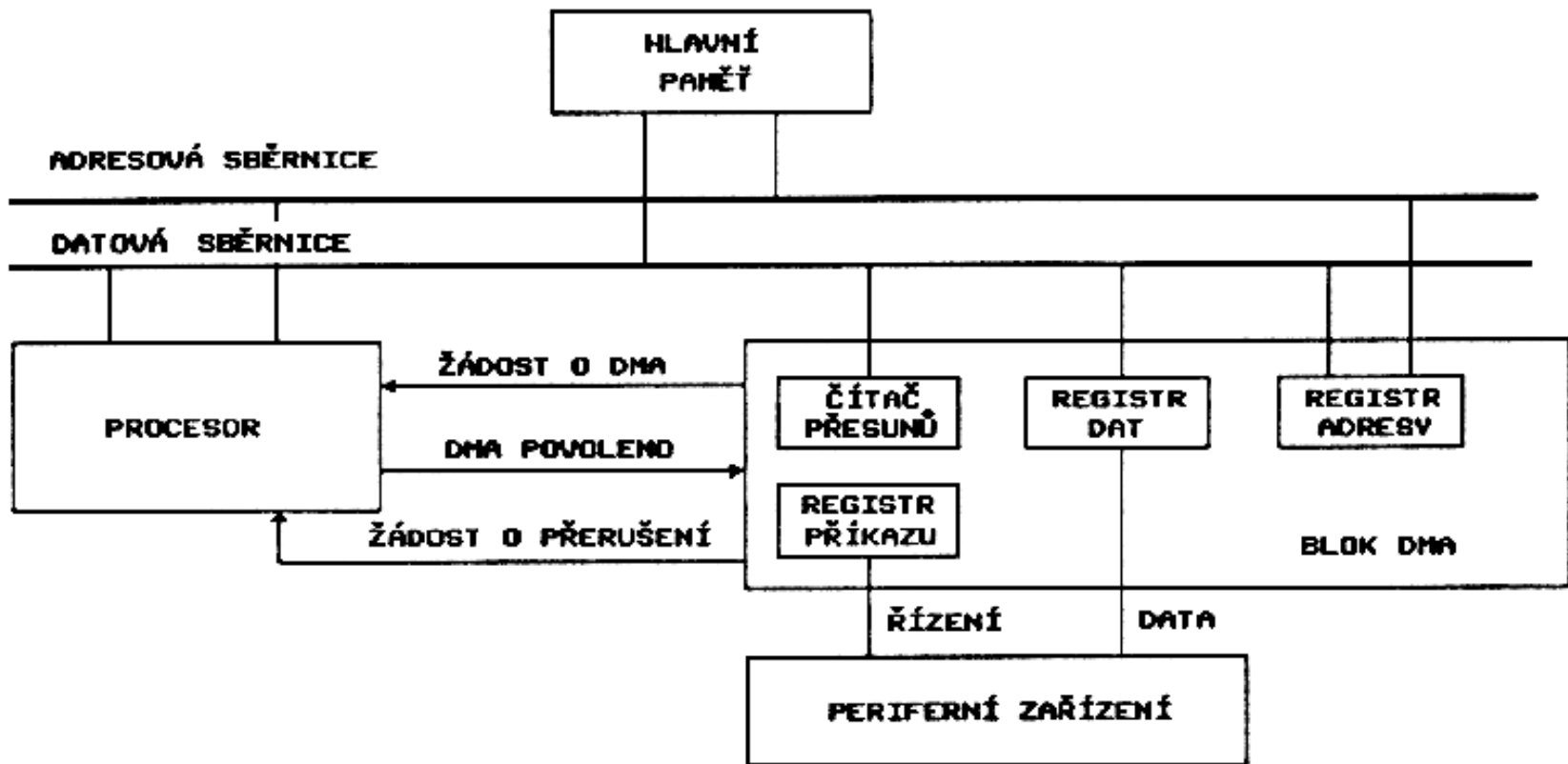


Cyklické výzvy (polling)

- žádosti po jednom vodiči, procesor přijímá žádosti jako anonymní
- jestliže může žádosti vyhovět, začne procesor postupně vysílat adresy jednotlivých PZ.
- žádající jednotka odešle zprávu Potvrzení výběru.
- procesor začne testovat další žádosti po dokončení činnosti PZ
- procesor může začít odpovídat buď od adresy následující nebo adresy s nejvyšší prioritou

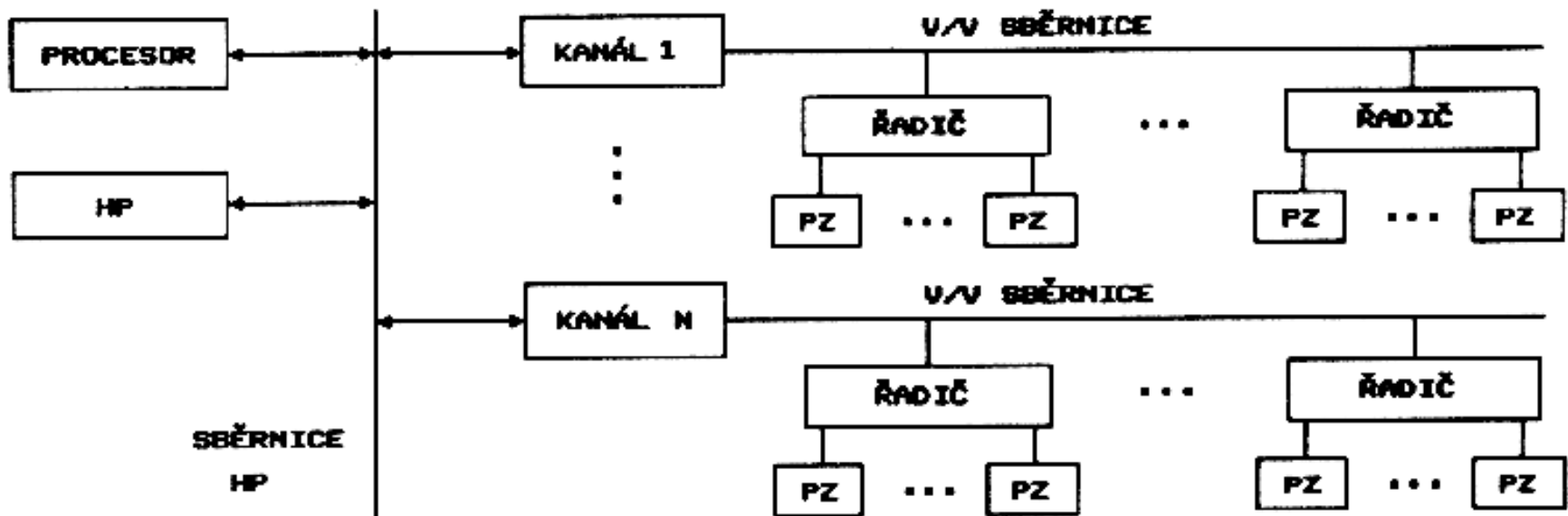


Struktura bloku pro přímý přístup do paměti (DMA)

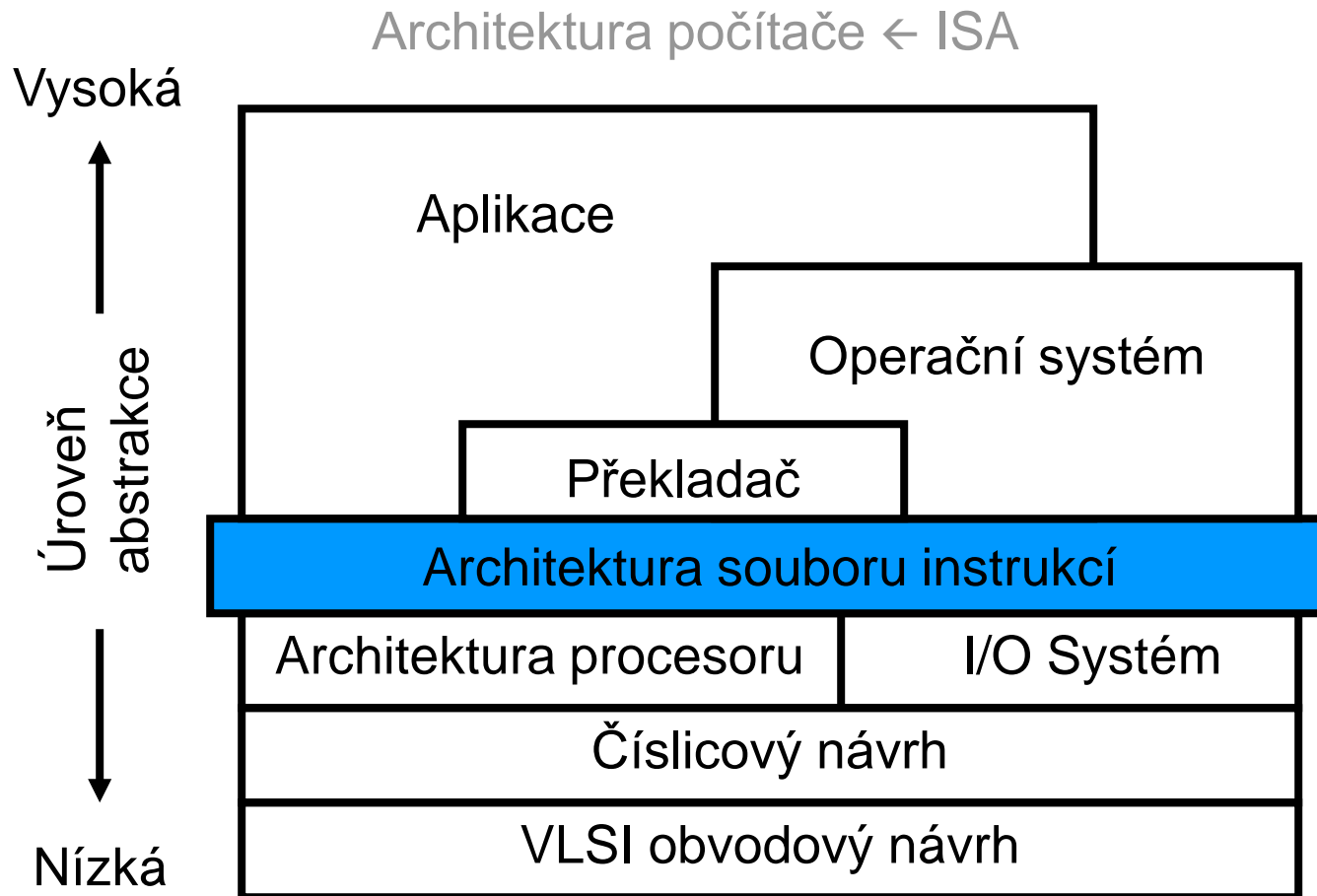


Kanálová architektura počítače

- kanál je specializovaný programově řízený procesor, schopný samostatně řídit vstupní a výstupní operace
- některé řadiče PZ jsou připojeny současně k více kanálům
- kanál většinou řídí operace mezi PZ a HP bez zásahu procesoru
- procesor nemá v operačním kódu V/V operace
- architektura využita hlavně u mainframe počítačů (IBM)



Architektura souboru instrukcí, ISA - Instruction Set Architecture



Architektura souboru instrukcí, ISA - Instruction Set Architecture

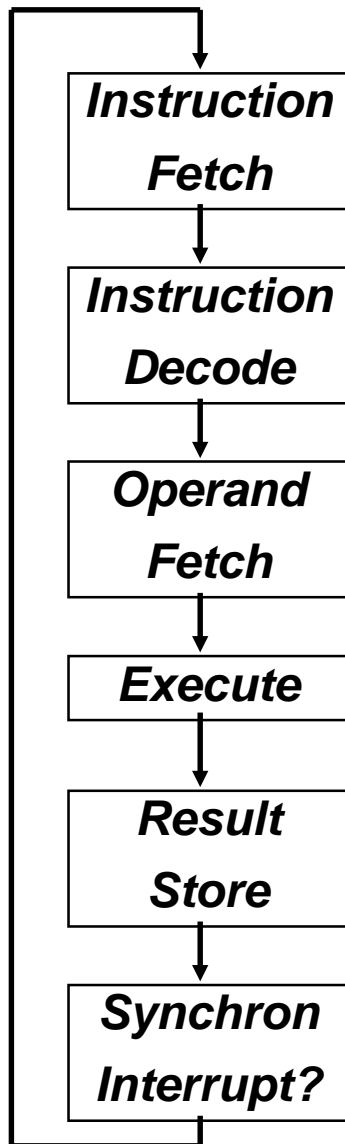
Zahrnuje

- Typy a formáty instrukcí, instrukční soubor
- Datové typy, kódování a reprezentace, způsob uložení dat v paměti
- Módy adresování paměti a přístup do paměti dat a instrukcí
- Mimořádné stavy

Umožňuje

- Abstrakce (výhoda – různé implementace stejné architektury)
- Definice rozhraní mezi nízko-úrovňovým SW a HW
- Standardizuje instrukce, bitové vzory strojového jazyka

ISA: Co musí být definováno?



Formát a kódování instrukcí

- Jak se instrukce dekóduje?

Umístění operandů a výsledku

- Kolik explicitních operandů je v instrukci pro ALU?
- Jak jsou operandy umístěny v paměti nebo jinde?
- Který operand může být v paměti?

Typy dat a velikosti operandů

Operace v ISA

- Které jsou podporovány ?

Výběr další instrukce

- Skoky, větvení programu, volání podprogramů
- „*fetch-decode-execute*“ je implicitní!

Strojový kód

Co musí instrukce obsahovat

instrukce = příkaz, zakódovaný jako číslo

- co se má provést
- s čím se to má provést (operandy)
- kam se má uložit výsledek
- kde se má pokračovat

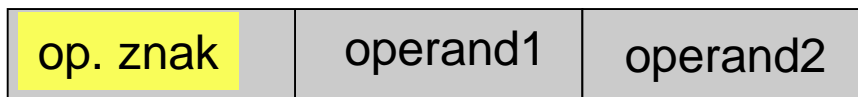
Tyto informace jsou obsaženy v instrukci ... tzn. **explicitně** v instrukci, zčásti dány architekturou počítače, tzn. **implicitně** (von Neumannova architektura)

Příklad: počítač SAPO byl 5-adresový: (1. operand, 2. operand, výsledek, skok při kladném výsledku, skok při záporném výsledku)

Příklady strojových instrukcí



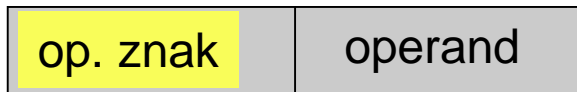
3 adresová instrukce



2 adresová instrukce

výsledek se ukládá na místo prvního operandu, zavedení operace **přesun**

$$\langle x \rangle - \langle y \rangle \rightarrow z \equiv \begin{cases} \langle x \rangle \rightarrow z \\ \langle z \rangle - \langle y \rangle \rightarrow z \end{cases}$$



1 adresová instrukce

zavedení „pracovního“ registru – STŘADAČ, ACCUMULATOR

$$\langle x \rangle - \langle y \rangle \rightarrow z \equiv \begin{cases} \langle x \rangle \rightarrow S \\ \langle S \rangle - \langle y \rangle \rightarrow S \\ \langle S \rangle \rightarrow z \end{cases}$$

ISA: Základní třídy

Akumulátorově (střadačově) orientovaná ISA (1 registr=střadač):

1 operand ADD A $\text{acc} \leftarrow \text{acc} + \text{mem}[A]$
 ADD (A + IX) $\text{acc} \leftarrow \text{acc} + \text{mem}[A + \text{IX}]$
 IX je **indexovací registr**

Zásobníkově orientovaná ISA

0 operandů ADD $\text{stack}(\text{top}-1) \leftarrow \text{stack}(\text{top}) + \text{stack}(\text{top}-1)$
 top--

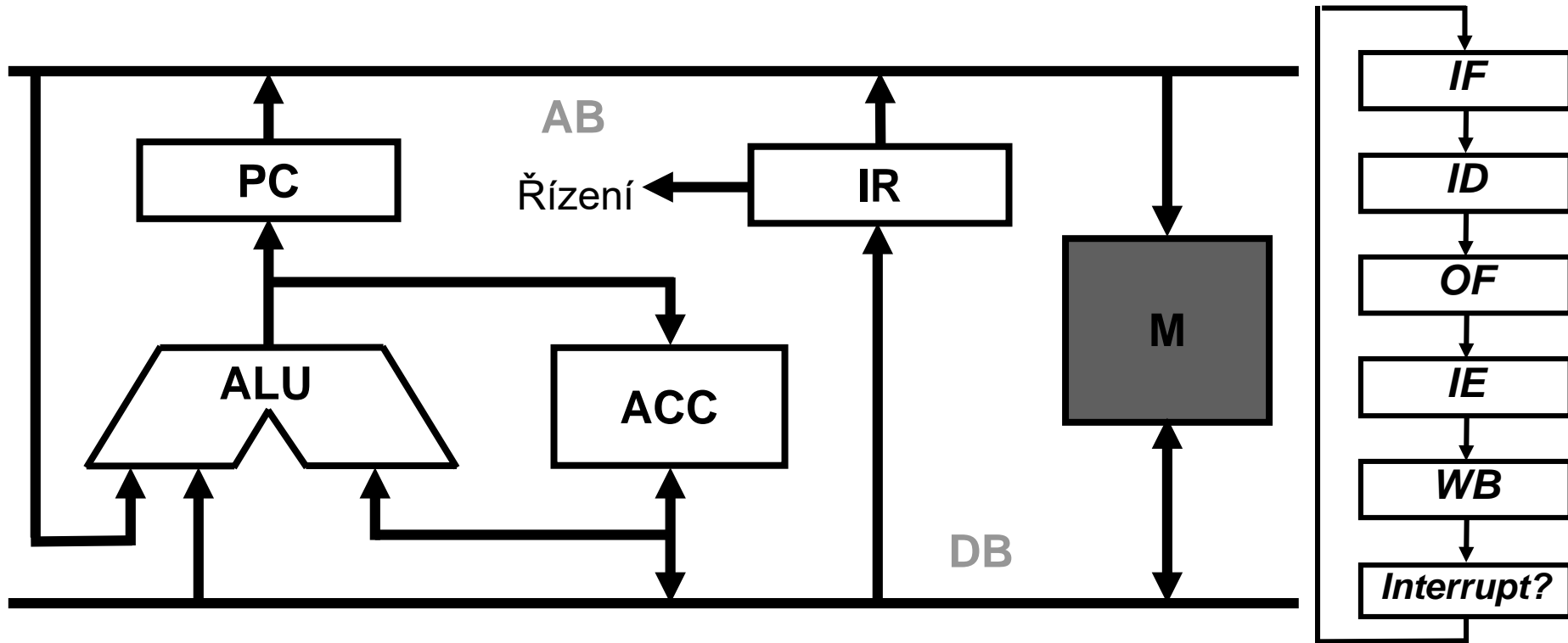
ISA orientovaná na registry pro všeobecné použití
(GPR = General Purpose Registers):

2 operandy ADD A B $\text{EA}(A) \leftarrow \text{EA}(A) + \text{EA}(B)$

3 operandy ADD A B C $\text{EA}(A) \leftarrow \text{EA}(B) + \text{EA}(C)$

EA ... Efektivní adresa (určuje registr, nebo operand v paměti)

Jednoduchý procesor



M – paměť

PC – Program Counter, prog. čítač, obsahuje adresu instrukce, která bude vykonána v dalším cyklu

ACC – Accumulator, střadač, obsahuje zpracovávaná data

ALU – Arithmetic Logic Unit, aritmeticko-logická jednotka, vykonává operace s daty

IR – Instruction Register, instrukční registr, obsahuje kód aktuálně prováděné instrukce – *pro programátora je skrytý*

1971 - první mikroprocesor

- + **výkon**
- + **efektivita**
- + **flexibilita**
- + **uživatelská přítulnost**
- - **menší výkonnost než specializované automaty**

Základní funkce mikroprocesoru

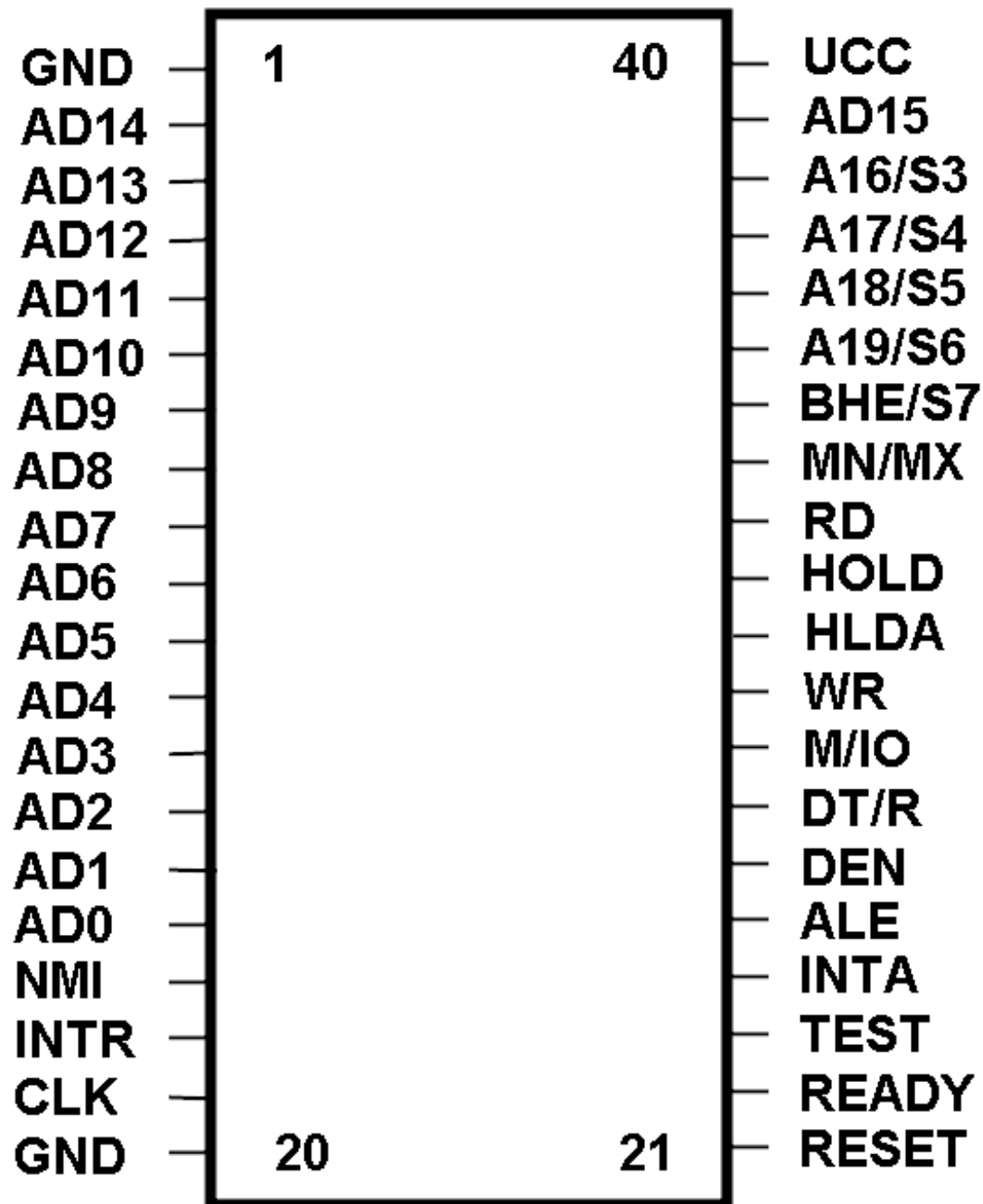
- Matematické a logické operace v ALU
- Přesun dat mezi různými pozicemi
- Rozhodování o skocích v programu a nastavení nových instrukcí na základě těchto rozhodnutí

Procesory Intel - rodina x86

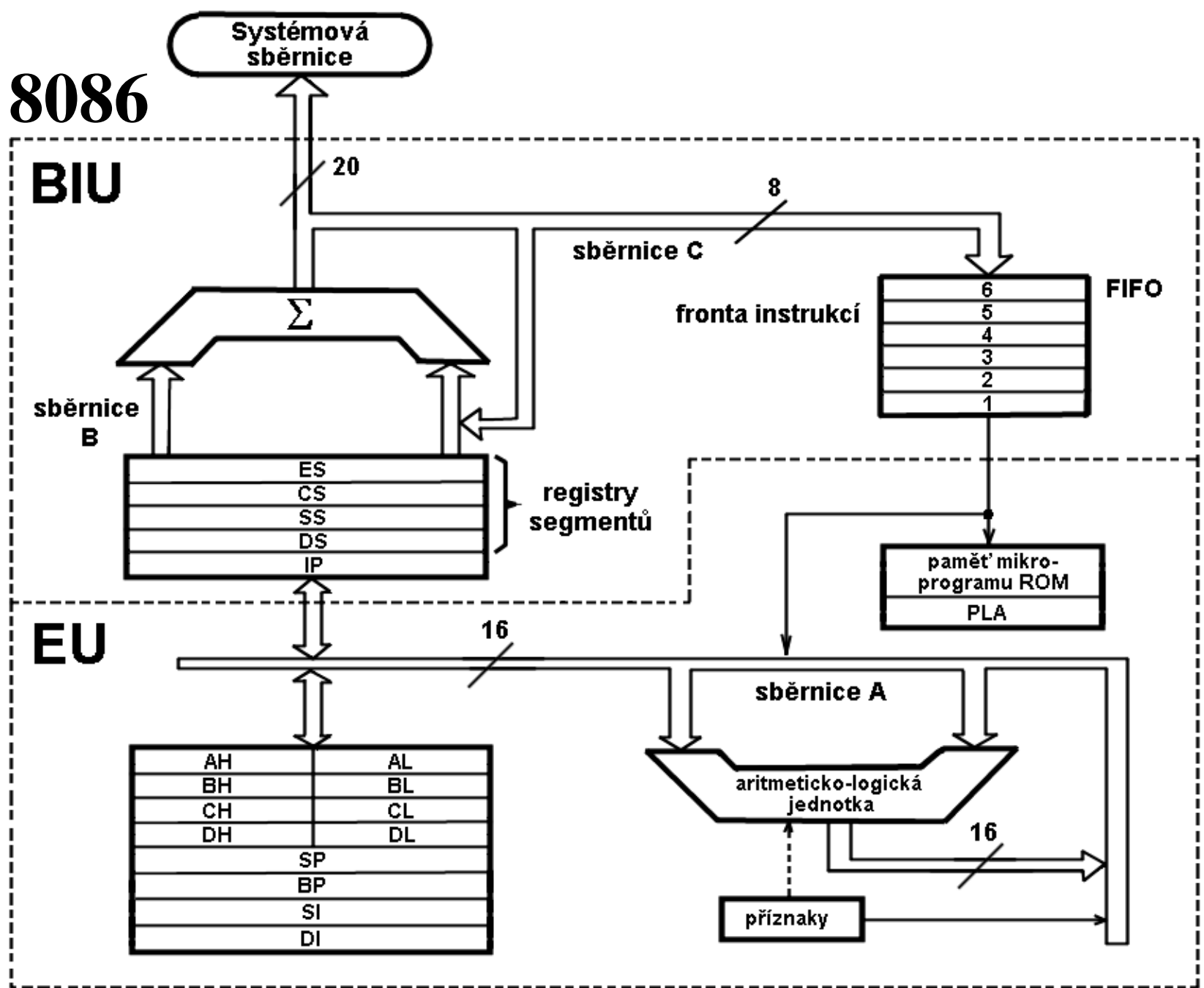
- **i8086**
- **i80486**
- **Pentium**
- **Pentium II, III, IV, ...**

Procesor i8086

- **16 bitová CPU**
- **paměťový adresový prostor 1 MB**
- **periferní adres. prostor: 65 536 v/v zařízení**
- **pouzdro 40 vývodů, dual-in-line**
- **hodinová frekvence 4 MHz a vyšší**



I 8086



Universal Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Counter
DX	DH	DL	Data
SI			Source Index
DI			Dest. Index
BP			Base Pointer
SP			Stack Pointer
	15	0	

Segment Registers

CS		Code seg.
DS		Data seg.
ES		Extra seg.
SS		Stack seg.

Control Registers

IP		Instr. ptr.
F		Flags
	15	0

Registr příznaků (Flags)

- **OF overflow**
- **DF direction**
- **IF interrupt**
- **TF trap**
- **SF sign**
- **ZF zero**
- **AF auxiliary carry**
- **PF parity**
- **CF carry**

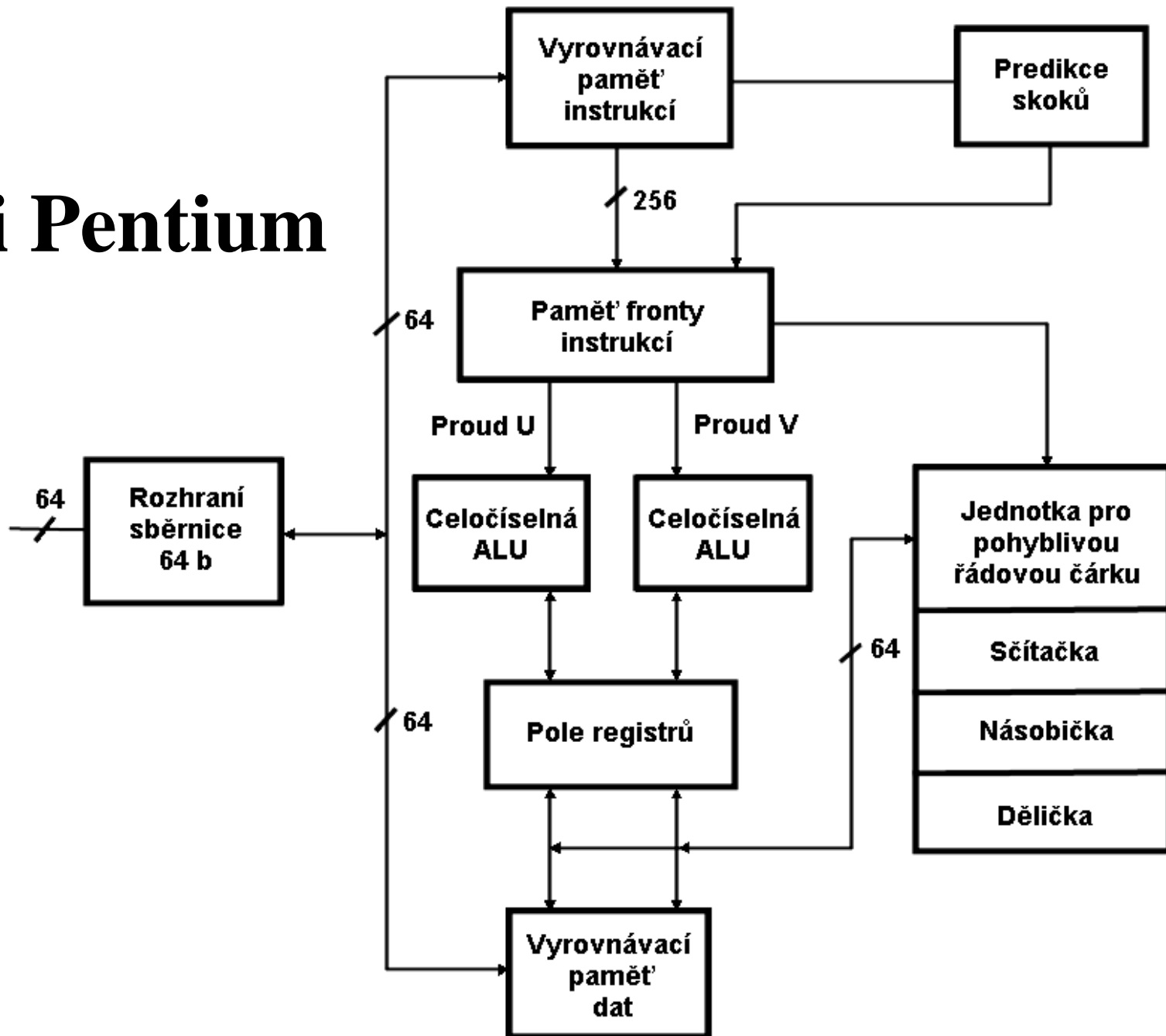
Adresové módy

- **registrový** `MOV ax, bx`
- **bezprostřední** `MOV dh, 5`
- **přímý paměťový** `MOV cl, [Mem]`
- **nepřímý paměťový** `MOV cl, [bx]`

i 80286 - i 80486 růst počtu jednotek

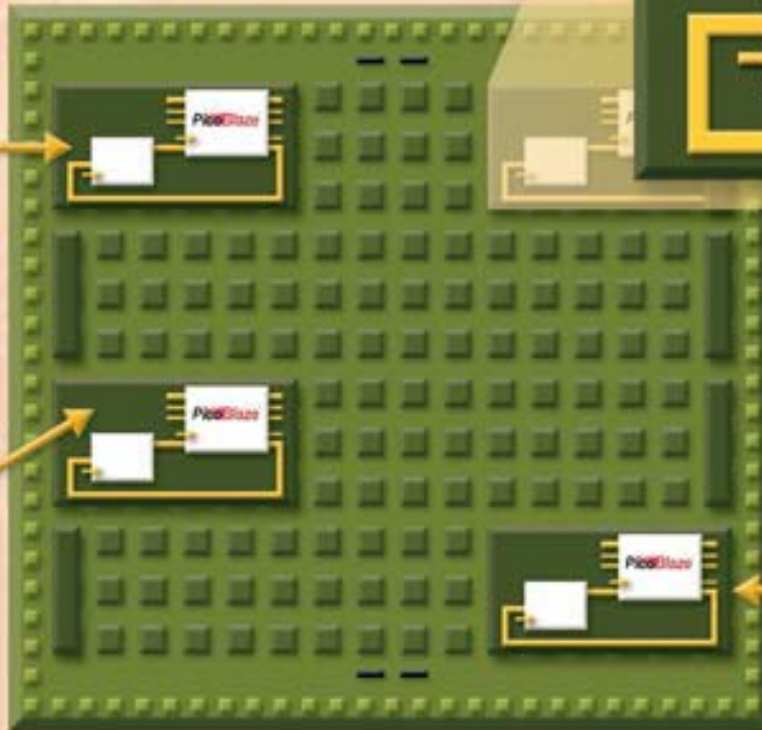
- **Sběrníková jednotka**
- **jednotka předvýběru instrukcí**
- **instrukční jednotka**
- **prováděcí jednotka**
- **segmentační jednotka**
- **stránkovací jednotka**
- **jednotka správy paměti**
- **rychlá vyrovnávací paměť' (cache)**
- **Floating-Point jednotka**

i Pentium

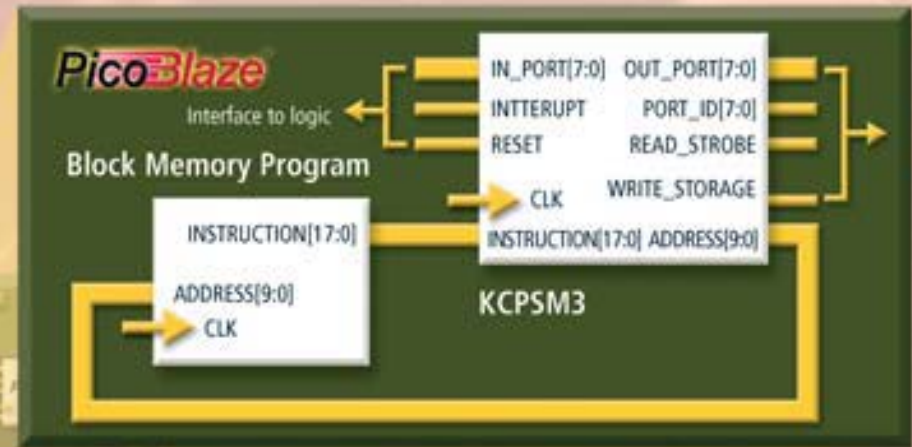


PicoBlaze

Key pad reader with
DTMF dialing and
tone generation



UART communication
providing remote
diagnostic commands

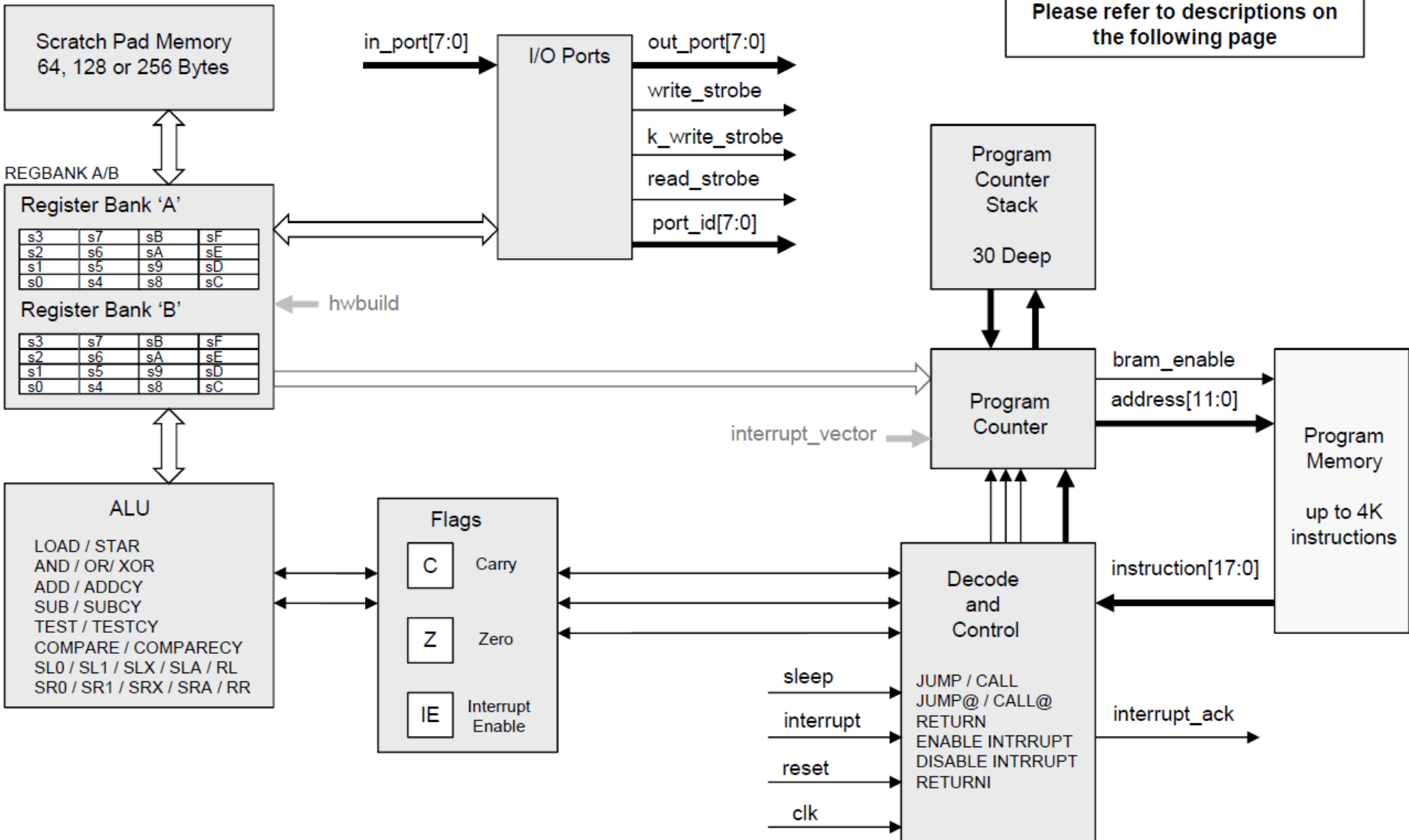


Display Control



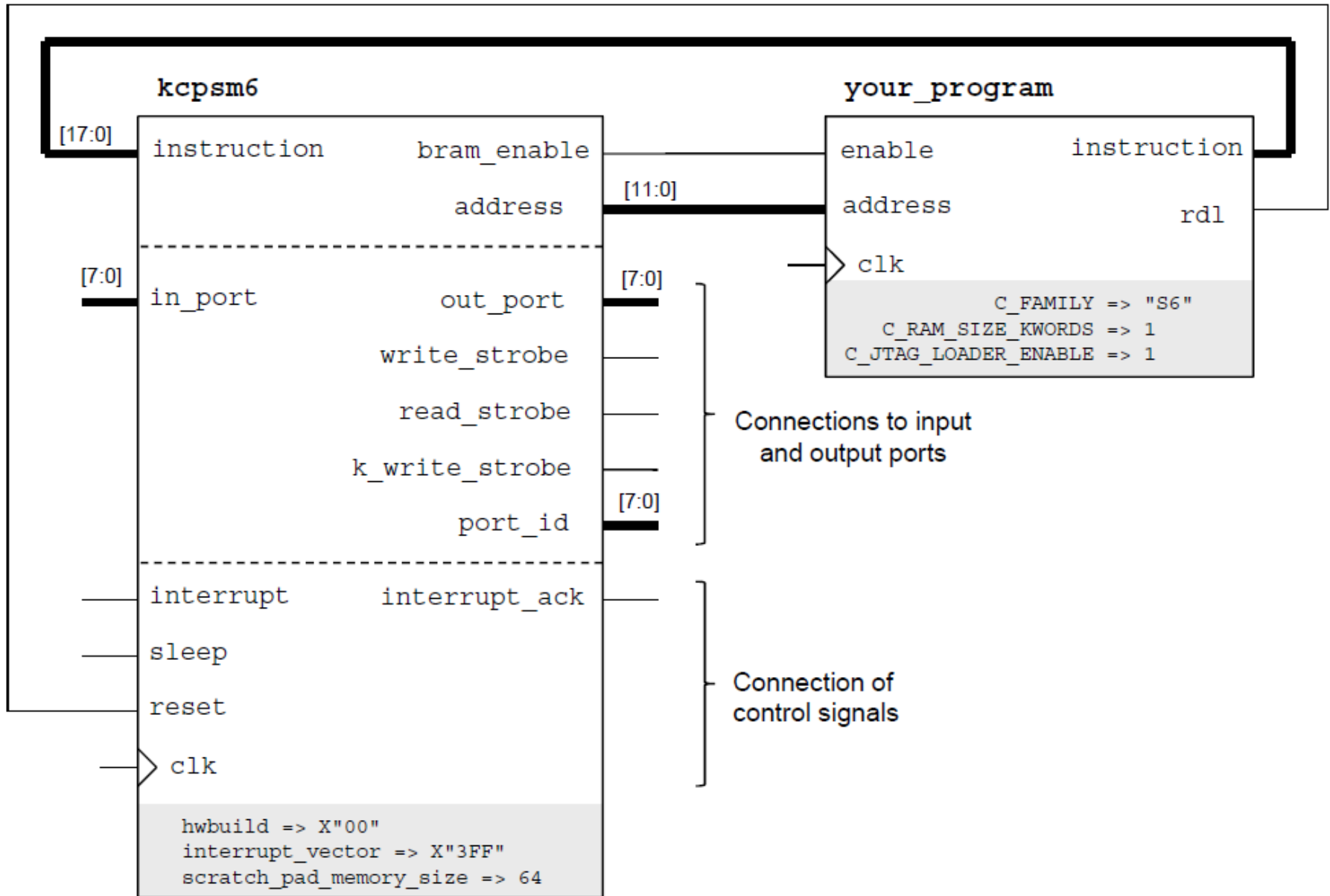
Mikroprocesor PicoBlaze pro Spartan 6

Please refer to descriptions on the following page



PicoBlaze obsahuje

- paměť programu pro 4k instrukcí
- 2x 16 rovnoprávných osmibitových registrů označených s0 až sF, které mohou všechny být použity jako střadač
- aritmeticko-logickou jednotku ALU pro provádění operací obvyklých u jednoduchých procesorů s příznaky nuly a přenosu, které lze použít ke větvení programu
- zápisníkovou paměť s rozsahem 64 -256 byte
- zásobník dovolující provádění podprogramů až do 31 úrovní
- jednoduchý přerušovací systém s jedním programově maskovatelným přerušením
- 256 osmibitových vstupů a tentýž počet osmibitových výstupů se vzorkovacími signály a s přímým nebo nepřímým adresováním;
- nulovací vstup (**reset**) k vyvolání definovaného počátečního stavu procesoru.



Register loading

00xy0 LOAD sX, sY
01xkk LOAD sX, kk
16xy0 STAR sX, sY

Logical

02xy0 AND sX, sY
03xkk AND sX, kk
04xy0 OR sX, sY
05xkk OR sX, kk
06xy0 XOR sX, sY
07xkk XOR sX, kk

Arithmetic

10xy0 ADD sX, sY
11xkk ADD sX, kk
12xy0 ADDCY sX, sY
13xkk ADDCY sX, kk
18xy0 SUB sX, sY
19xkk SUB sX, kk
1Axy0 SUBCY sX, sY
1Bxkk SUBCY sX, kk

Test and Compare

0Cxy0 TEST sX, sY
0Dxkk TEST sX, kk
0Exy0 TESTCY sX, sY
0Fxkk TESTCY sX, kk
1Cxy0 COMPARE sX, sY
1Dxkk COMPARE sX, kk
1Exy0 COMPARECY sX, sY
1Fxkk COMPARECY sX, kk

Shift and Rotate

14x06 SL0 sX
14x07 SL1 sX
14x04 SLX sX
14x00 SLA sX
14x02 RL sX
14x0E SR0 sX
14x0F SR1 sX
14x0A SRX sX
14x08 SRA sX
14x0C RR sX

Register Bank Selection

37000 REGBANK A
37001 REGBANK B

Input and Output

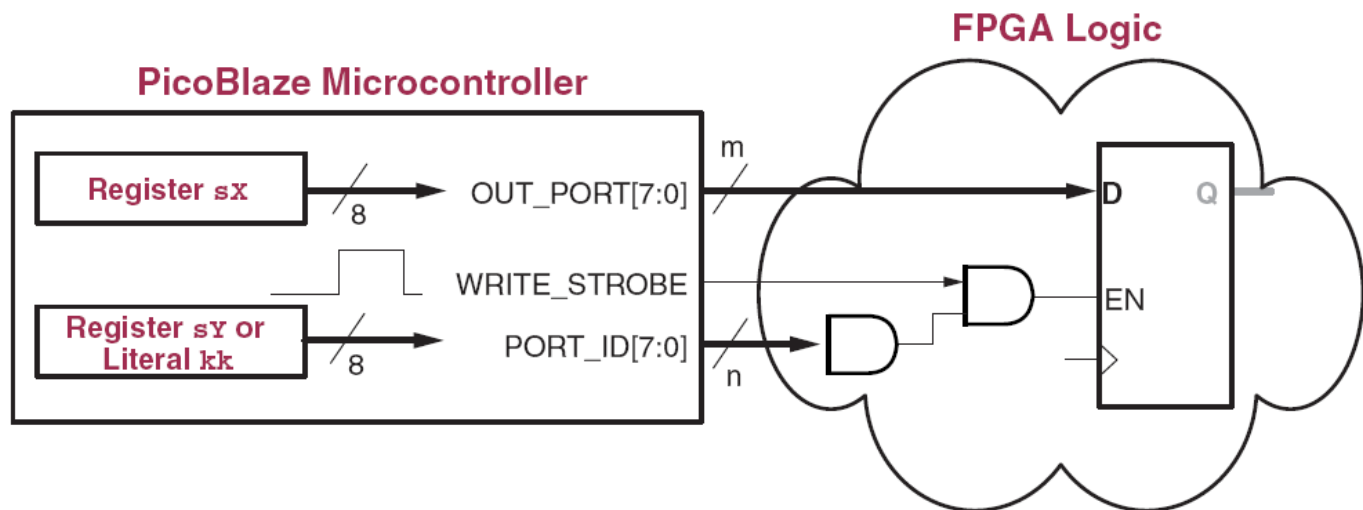
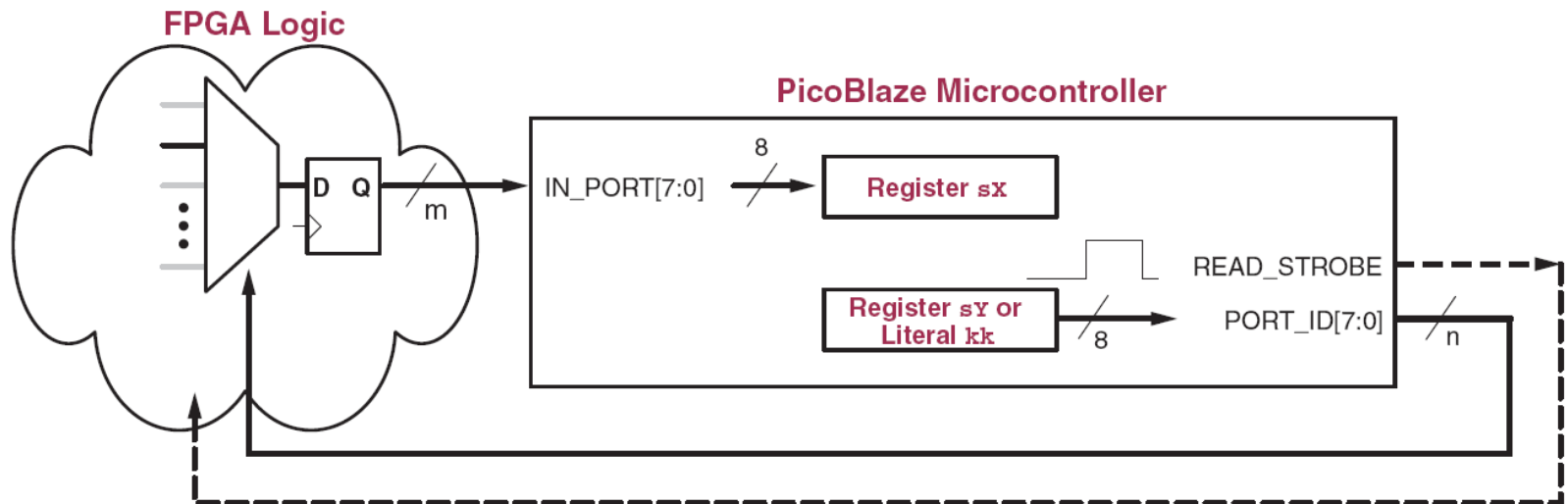
08xy0 INPUT sX, (sY)
09xpp INPUT sX, pp
2Cxy0 OUTPUT sX, (sY)
2Dxpp OUTPUT sX, pp
2Bkkp OUTPUTK kk, p

Scratch Pad Memory

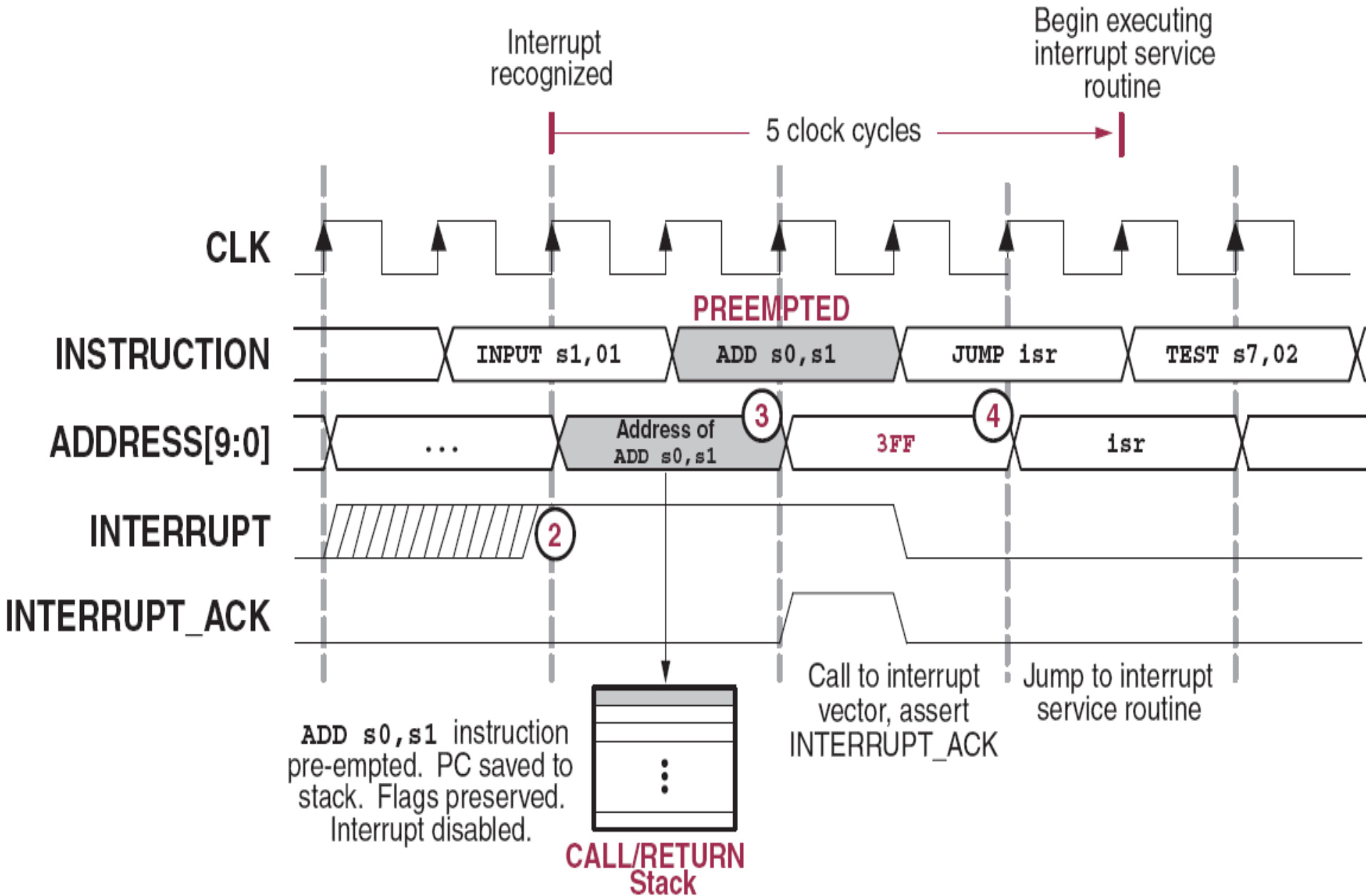
(64, 128 or 256 bytes)

2Exy0 STORE sX, (sY)
2Fxss STORE sX, ss
0Axy0 FETCH sX, (sY)
0Bxss FETCH sX, ss

Řízení vstupních a výstupních operací



Obsluha přerušení



Processor MicroBlaze:

- Procesor s 32bitovou harvardskou architekturou typu RISC. Vývojovým systémem je možno zvolit velikost cache, druh a rozsah jeho výkonných a stykových obvodů. Tento procesor lze implementovat do obvodů FPGA řad Spartan i Virtex firmy Xilinx.
- K použití procesorového jádra MicroBlaze je zapotřebí zakoupit licenci, která je součástí programového nástroje EDK (*Embedded Development Kit*). Tento kit obsahuje programové prostředky k vývoji aplikací procesoru, včetně kompilátoru jazyka C.

Procesor PowerPC:

- 32bitové procesorové jádro typu RISC "natvrdo,, vytvořené ve struktuře některých obvodů FPGA řady Virtex firmy Xilinx, plné označení je IBM PowerPC 405.
- Skalární 5 stupňová struktura pipeliningu
- zvláštní cache pro instrukce a pro data
- brána JTAG
- několik časovačů a jednotka pro řízení paměti;
- v obvodech Virtex-4 je k procesoru PowerPC přidružena také pomocná procesorová řídicí jednotka APU (*Auxiliary Processor Unit*). Při kmitočtu kolem 400 MHz je výkonnost přes 600 Dhrystone MIPS.
- Licence k použití procesoru PowerPC je také součástí nástroje EDK.

Procesory ARM

- součástí rodiny Zync-7000
- multi procesorový systém (1-4 procesory, cache L1 a L2)
- procesorově centrické řešení, programovatelná logika doplňuje funkčnost procesorového systému o neimplementované funkce

RISC-V

- otevřená architektura ISA RISC (na rozdíl od ARM,..)
- kdokoliv může použít a vyrobit vlastní procesory a software
- vhodná pro cloudy – paralelní počítače, mobilní telefony, vnořené systémy
- záměr využívat v hard discích, grafických kartách