

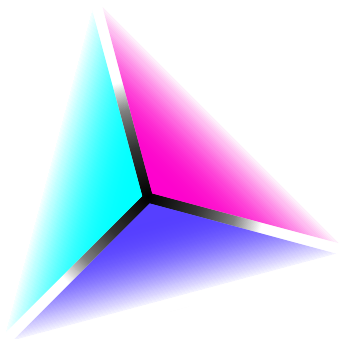
Struktura a architektura počítačů

Katedra číslicového návrhu
Fakulta informačních technologií
České vysoké učení technické

© Hana Kubátová, 2021

Úvod

BI-SAP, únor 2021



Cíle předmětu

- Vysvětlit co je číslicový počítač a jak pracuje
- Co je hardware a co je software ... a není náhodou ještě něco mezi tím?
- Vysvětlit nezbytné principy, tzn. „proč“ počítač něco dělá (nejen „počítá“), a „jak“ ho to naučit
- Uvědomit si, proč je číslicová technika v současné době všude, a co může být dál
- Uvědomit si kvantitativní souvislosti, tzn. že „na velikosti a rychlosti záleží“
- Spojit si matematické (exaktní) zákonitosti s realitou (inženýrský přístup)
- Zkusit si všechno v laboratoři s nezbytnou podporou pro domácí přípravu

Obsah první přednášky

- Úvod - k čemu je mi tento předmět?
- Co je číslicový počítač, kde všude najdeš (mikro)procesor (počítač)?
- Co je vestavný systém a FPGA?
- Jak navrhnout elementární blok v počítači - sčítačku (a dále ji budeme zjednodušovat, zrychlovat, používat, apod.).

Podklady a všechny informace k předmětu hledejte zde:

<https://courses.fit.cvut.cz/BI-SAP/>

Prerekvizity, způsob výuky

- Něco historie z oblasti architektury počítačů a jejího vývoje
- Učebnice na téma „Digital Design“
- Využití znalostí z BI-CAO, BI-PA1, BI-MLO, i z BI-LIN
- Zaměření na praktické úlohy
- Využití rekonfigurovatelných a programovatelných obvodů (FPGA, AVR) a moderních (a dostupných) návrhových prostředků (EDA tools)
- Testy a zkouška: důraz nejen na teorii (přednášky + doporučená literatura), ale zejména na její praktickou aplikaci (konkrétní návrhy a realizace: prosemináře + laboratoře)

Podmínky zápočtu a zkoušky

Předmět **2+1+2** zápočet, zkouška

- Přednáška a laboratorní cvičení každý týden (**pozor na nutné proškolení z bezpečnostních předpisů!!**)
- Prosemináře 1 krát za 14 dní (liché týdny) + konzultace, odpovědi na dotazy a testy (sudé týdny)
- Zápočet za:
 - fungující laboratorní úlohy (až 20 bodů) + aktivita,
 - 2 testy (až 2x15 bodů) ... (je třeba získat alespoň 20 bodů na A)
 - celkem minimálně 25 bodů na zápočet
- Zkouška za:
 - 45 (a více) bodů v semestru (A)
 - jinak body ze cvičení + zkouškový test (až 50 bodů)

Aktuality 2021

- Laboratoře v 10 patře (laboratoř počítačového inženýrství Jana Hlavičky), 24 +1 PC, rozvrh úterý až pátek
- Po dobu krizové situace vše virtuálně (Teams), ale ***podle rozvrhu***
- Samoobsluha v respiriu (PC + přípravy) ... *podle situace*
- Testy (dva) v době proseminářů a v Moodle
- Skriptum: ***Struktura a architektura počítačů s řešenými příklady*** (autor Kubátová ... 131,-Kč,
https://eobchod.cvut.cz/skripta_cvut/skripta_cvut/struktura_a_architektura_pocitacu_s_resenymi_prikklady-150031738
kopie kapitol postupně na courses)

Co se naučíte

1. Číslicový počítač, struktura, jednotky a jejich propojení.
2. Logické obvody, formy jejich popisu, kombinační obvody a jejich realizace na úrovni hradel.
3. Sekvenční obvody a jejich realizace.
4. Typické kombinační a sekvenční obvody v číslicových počítačích - jejich realizace (kodéry, sčítačky, čítače, registry).
5. Data, jejich zobrazení a zpracování.
6. Realizace aritmetických operací.
7. Soubor instrukcí, strojový kód a jazyk symbolických instrukcí (assembler).
8. Návrh procesoru.
9. Paměti: struktura paměťového obvodu, paměťový systém počítače.
10. Vstupy a výstupy.
11. Řadiče. Procesory typu CISC a RISC.

Realizace: jak je možné, že to „počítá“?

- Technologie - realizace operací pomocí *hradel*
- Pomocí hradel realizují jakoukoli logickou funkci – jak?
- Dnes CMOS, ale v budoucnosti??? (dříve relé, elektronky, ..., kvantové, biologické počítače?)
- Bloky v počítači ... architektura
- Základem je sčítačka
- Co je procesor a počítač?
- Jak zařídit universalitu?
- Program? Jde to i jinak?

Proč vznikly počítače?

- Universalita – využití jedné techniky pro více výpočtů (aplikací)



- Průlom je vznik přepisovatelné paměti, kam je možné ukládat různé programy a data

Architektura



- Co je na vstupu, co na výstupu a co uvnitř?
 - Zobrazení dat: diskrétní (číslicové) nebo spojitě (analogové)
 - Počítač
 - Analogový – spojitě zobrazení dat
 - Číslicový – nespojitě zobrazení dat
 - Hybridní – obojí + A/D, D/A převodníky

Základ číslicových počítačů

- Booleova algebra
 - Realizace logické funkce z hradel, co je úplný soubor logických funkcí
 - Přizpůsobení více úlohám (bloky v počítači, paměť) ... universalita
 - Na velikosti a rychlosti záleží ... technologie, komunikace
 - Jak počítači říct, co má dělat?
 - Jak zobrazit zpracovávaná data?
 - Jak poznat správný výsledek?
 - Jak to dát celé dohromady?
-
- The diagram uses curly braces to group the list items and link them to lecture numbers in boxes on the right:
- A large brace groups the first two items (Booleova algebra, Realizace logické funkce...) and points to a box labeled "Přednášky 2 a 3".
 - A brace groups the third item (Přizpůsobení více úlohám...) and points to a box labeled "Přednášky 4, 6, 7 a 10, 11".
 - A brace groups the fourth item (Na velikosti a rychlosti záleží...) and points to a box labeled "Přednášky 3, 10 a 11".
 - A brace groups the fifth item (Jak počítači říct, co má dělat?) and points to a box labeled "Přednášky 8 a 9".
 - A brace groups the sixth and seventh items (Jak zobrazit zpracovávaná data?, Jak poznat správný výsledek?) and points to a box labeled "Přednášky 5, 6 a 10".
 - A brace groups the eighth item (Jak to dát celé dohromady?) and points to a box labeled "Přednášky 1 a 12".

Co je software?

- **Firmware**

- BIOS, adresní módy, architektura souboru instrukcí: Instruction **S**et **A**rchitecture - **ISA**,
- jazyk symbolických instrukcí/adres: JSI/JSA/... **assembler**

- **Operační systém**

- Struktura souborů na disku, privilegia a ochrana, plánování a přepínání úloh, jádro, správa paměti a zařízení

- **Vývojářský SW**

- Assembler-překladač a linker; simulátor a debugger; knihovny

- **Aplikace**

- Programovací jazyky, editory, prohlížeče, hry, ...

Co je počítačový hardware?

- **Struktura a architektura počítače, jednotlivých bloků a komunikace mezi nimi**

procesor, jak se provádějí instrukce, tok dat, řízení, predikce větvení

- **Paměťová hierarchie**

Skrytá paměť (cache), správa paměťového systému, segmentace a stránkování a tomu odpovídající bloky (řeší to OS ... tedy SW)

- **Uživatelské rozhraní**

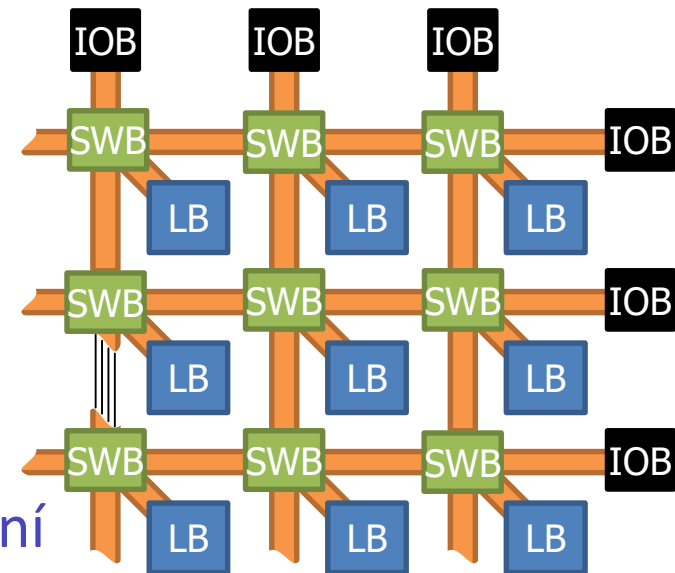
Displej, grafické rozhraní, klávesnice, myš, porty

- **Další rozhraní**

Přerušovací systém, DMA (Direct Memory Access), komunikační protokoly

FPGA

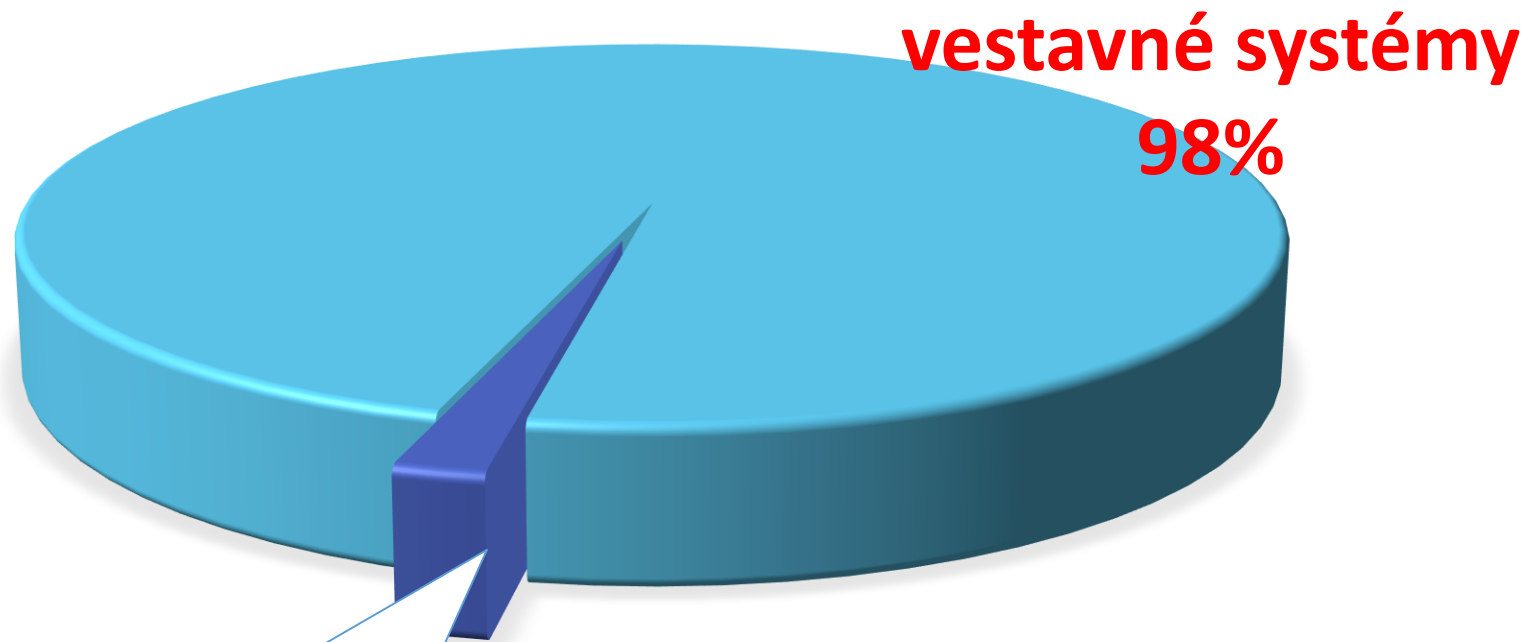
- Rekonfigurovatelný
(programovatelný) hardware
- Jak je možné „programovat“ hardware?
programuje se funkce pomocí „tabulek“
(LUT = malá RAM paměť) a jejich propojení
- Pomocí softwarového nástroje pro návrh (tzv. EDA tool) je možné navrhnout realizaci – zde propojení prvků v „hradlovém poli“
- Jak se to návrhovému nástroji řekne?
 - nakreslí se schema na úrovni hradel
 - popíše se to v jazyce pro popis hardwaru (VHDL, Verilog, nebo i v C, v Matlabu, ..)
- Je možné „naprogramovat“ procesorové jádro (IP core)



Co je vestavný systém?

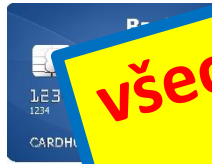
- „embedded systems“ ... něco „malého“, „jednoduchého“ se „snadným ovládáním“, „universálně použitelného“, co řídí nějaké reálné zařízení (pračku)
- proč je to možné? ... číslicové řízení, technologie, Moorův zákon
- co znamená programovatelnost?
 - nejen program
 - existuje i programovatelný (*rekonfigurovatelný*) HW – laboratoře SAP

Procesory na této planetě



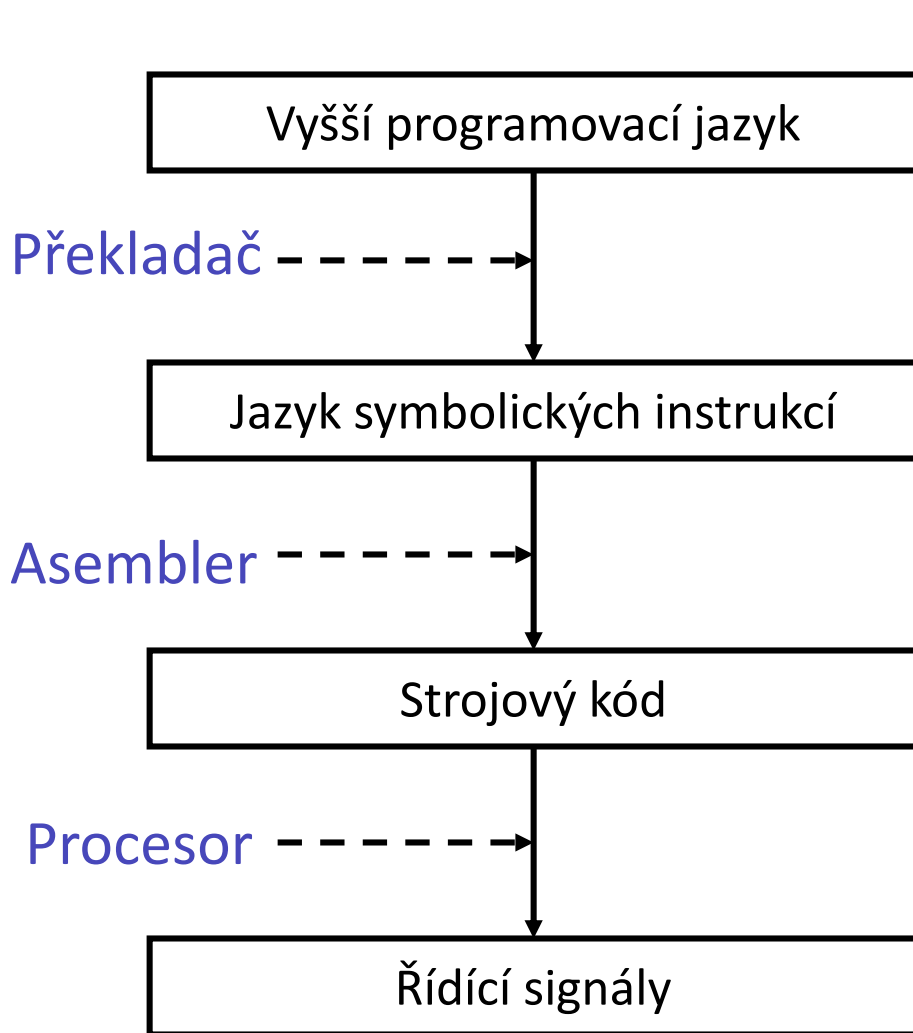
počítače, tablety
2%

Vestavné systémy



**všechna tato zařízení jsou ovládána zabudovanými
(= vestavěnými počítači):
vestavné systémy (angl. embedded systems)**

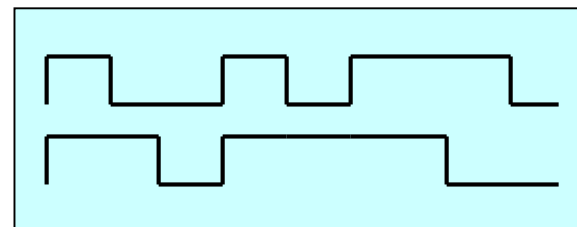
Vývoj softwaru – úrovně **abstrakce**



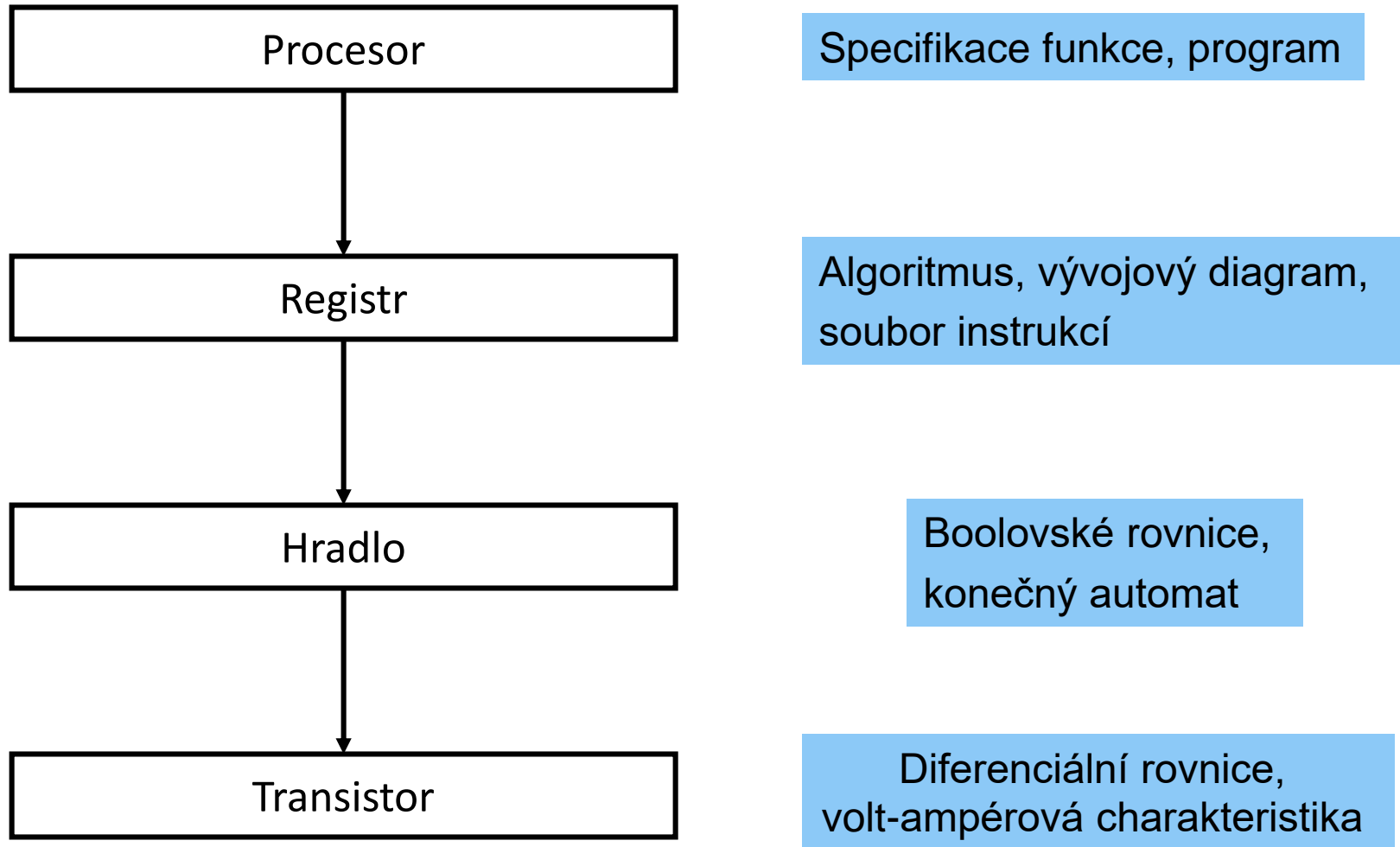
```
a = b+c  
If (a > MAX) a = MAX  
for (i==0; i<a; i++)
```

```
mov reg1, konst[0]  
mov reg2, konst[2]  
add reg1, reg2  
jc lab
```

```
0000 1111 0101 0111  
1011 0001 1110 0011  
1100 1000 1001 0110
```



Úrovně abstrakce v hardwaru



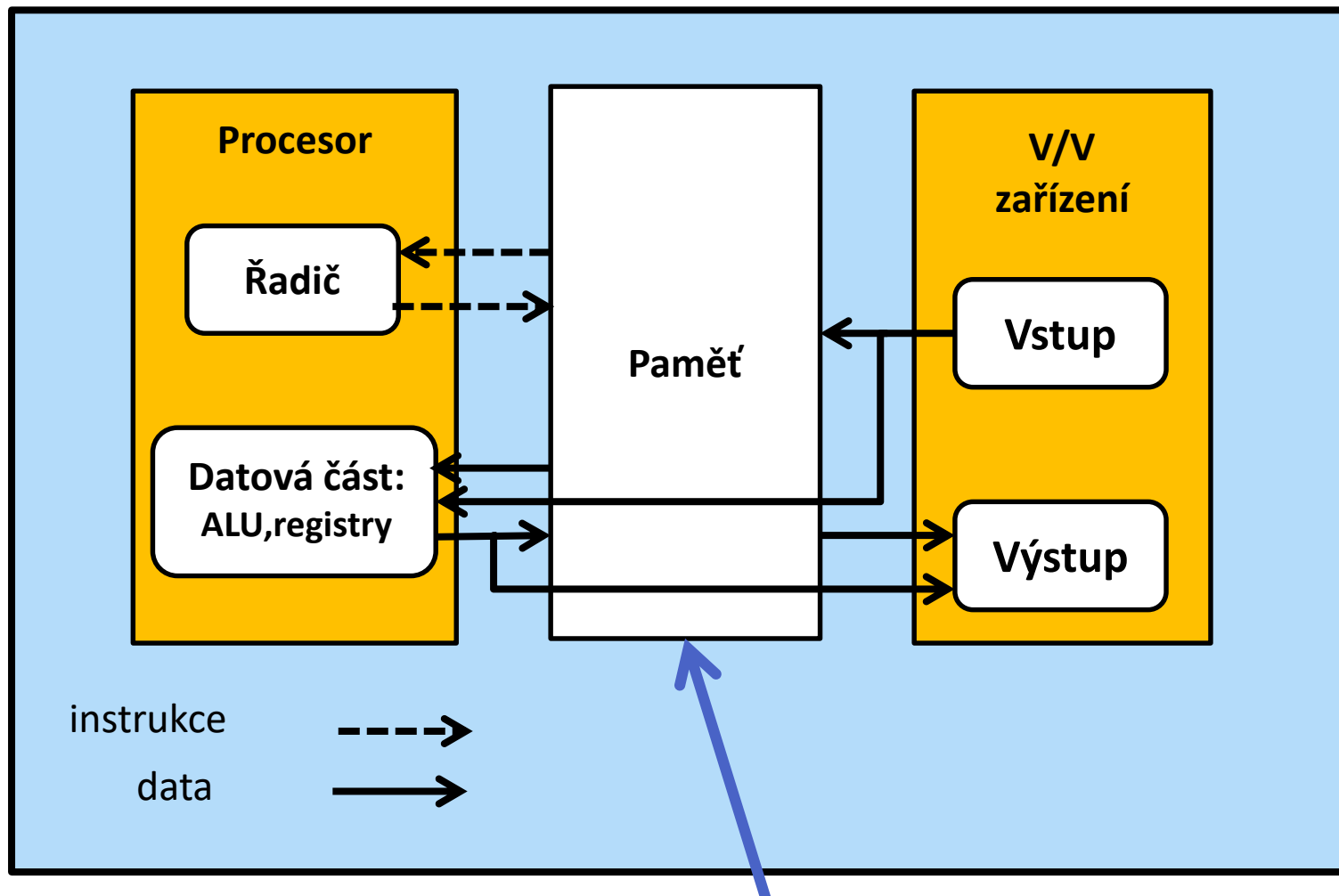
Von Neumannova architektura

- Instrukce a data jsou uloženy **v téže paměti**.
- **Paměť** je organizována **lineárně** (tzn. jednorozměrně) a je rozdělena na stejně velké buňky, které se adresují celými čísly (zprav. 0, 1, 2, 3, ...).
- **Data ani instrukce** nejsou explicitně označeny.
- Explicitně nejsou označeny ani různé **datové typy**.
- Pro reprezentaci dat i instrukcí se používají **dvojkové signály**.
- V instrukci zpravidla není uváděna **hodnota operandu**, ale jeho **adresa**.
- Instrukce se provádějí jednotlivě, a to **v pořadí**, v němž jsou zapsány **v paměti**, pokud není toto pořadí změněno speciálními instrukcemi (nazývanými **skoky**).

Von Neumannova architektura (2)

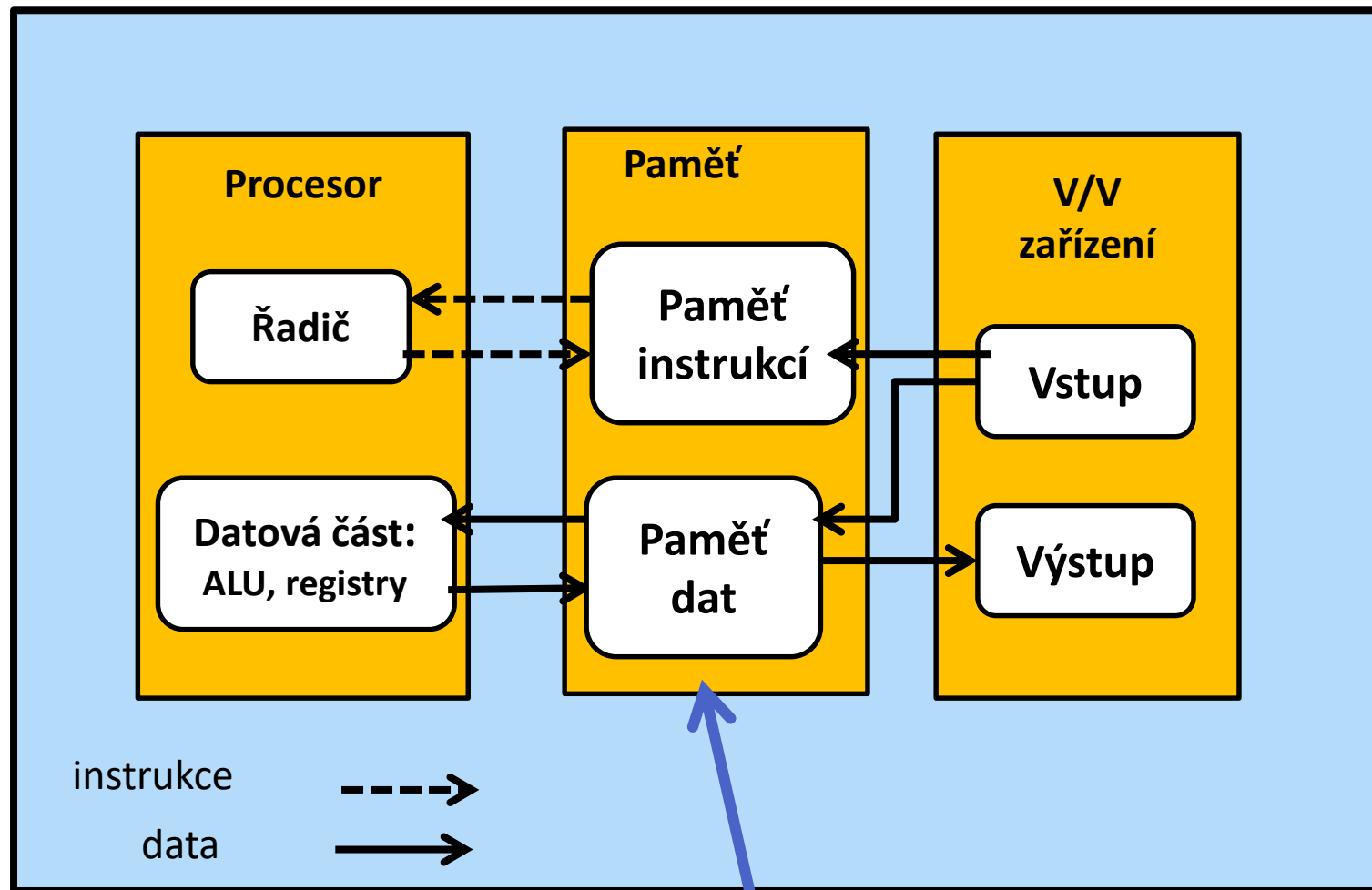
- **Důsledek:** podle výpisu paměti nelze poznat, zda jde o instrukce nebo o data (ani o jaká data) → je třeba znát kontext
- Počítač tvoří:
 - hlavní paměť (main memory)
 - procesor:
 - datová část
 - ALU – aritmeticko-logická jednotka
 - Registry
 - řídící část
 - Řadič – control unit (controller)
 - vstupní/výstupní zařízení

Počítač von Neumannova typu



Společná paměť pro instrukce i pro data

Harwardská architektura



Oddělené paměti pro instrukce a pro data

Organizace hlavní paměti

- Hlavní paměť je rozdělena na buňky – **paměťová místa**, kterým jsou přiřazena nezáporná čísla nazývaná **adresy**
- Obsah paměťového místa je **slovo**
 - slovo – (word) – velikost závisí na procesoru (např. 16b, 37b, 50b, b označuje bit)
 - **slabika B** – byte, $8b = 1B$, obvykle 2 nebo i více slabik tvoří slovo, např. u procesorů Intel 80x86 jedno slovo = 2B
- Obsah paměťového místa na adrese **adr** bývá někdy označován **<adr>**; nehrozí-li nedorozumění píše se však často **adr** místo **<adr>**.

Slabiková organizace paměti

Př. 1 slabika = 1B

1 slovo = 2B

1 dvojité slovo = 4B [DW – Double Word] – tedy 32b

Od adresy 5678 má být uloženo dvojité slovo 1234ABCD:

	1. způsob	2. způsob
5678	12	CD
5679	34	AB
567A	AB	34
567B	CD	12

big-endian (IBM 360, Motorola 68000)

little-endian (Intel 80x86, DEC Alpha)

Oba způsoby (Motorola 88 110)

Zobrazení dat v paměti

- **Numerická data – čísla:**
 - V pevné řádové čárce – **fix point**, obvykle celá čísla (integer, byte, word ...)
 - V pohyblivé řádové čárce – **floating point**, racionální čísla (real, float, ...)
- Dvojková – **binary**
- Desítková – **decimal**
- Šestnáctková – **hexadecimal**
- Bez znaménka – **unsigned**, pouze nezáporná (byte, word, unsigned ...)
- Se znaménkem – **signed** (integer, short int, signed ...)
- Různě dlouhá, různý rozsah hodnot (short int, integer, long int, byte, word, ...)

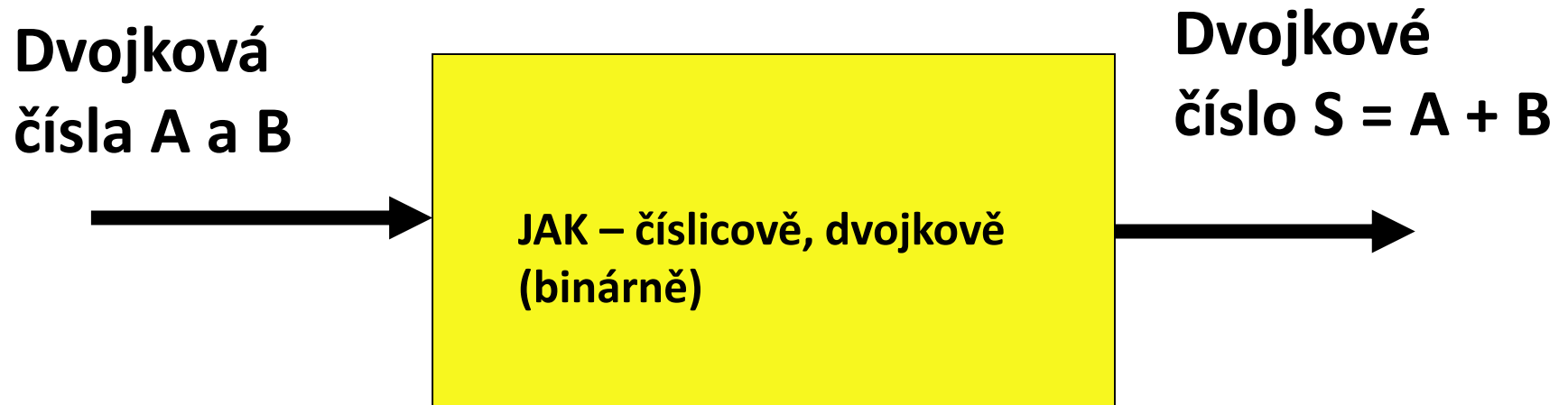
Vždy existuje **omezení**: registry mají určitou velikost, paměť je slabikově organizovaná, šířka sběrnic je daná

Příklad 1 - sčítačka

Chci sčítat ...



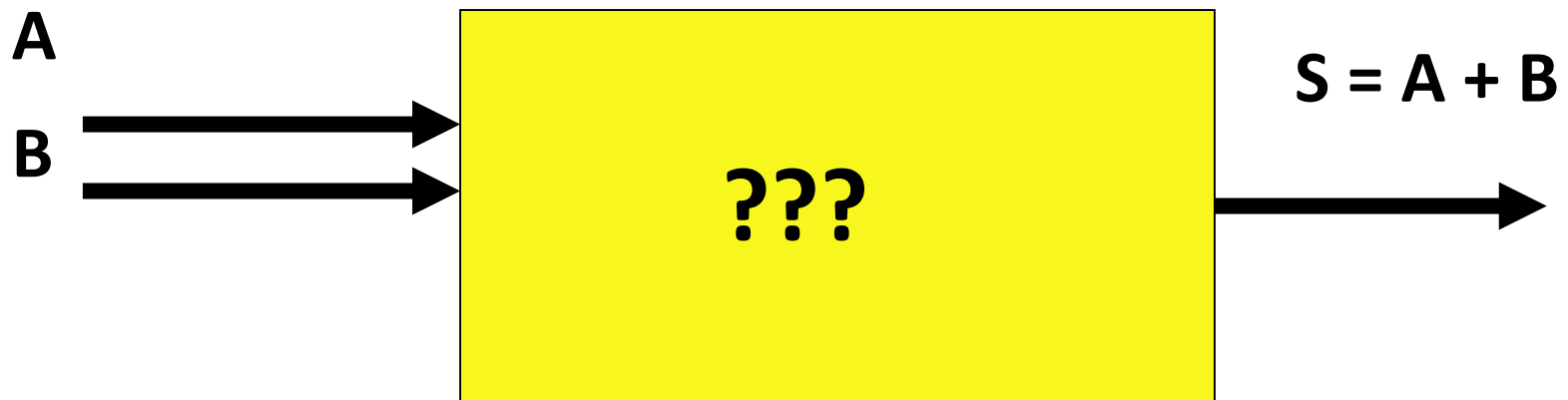
... příklad 1



Dvojková čísla budou nejprve 1 bitová, tzn.:

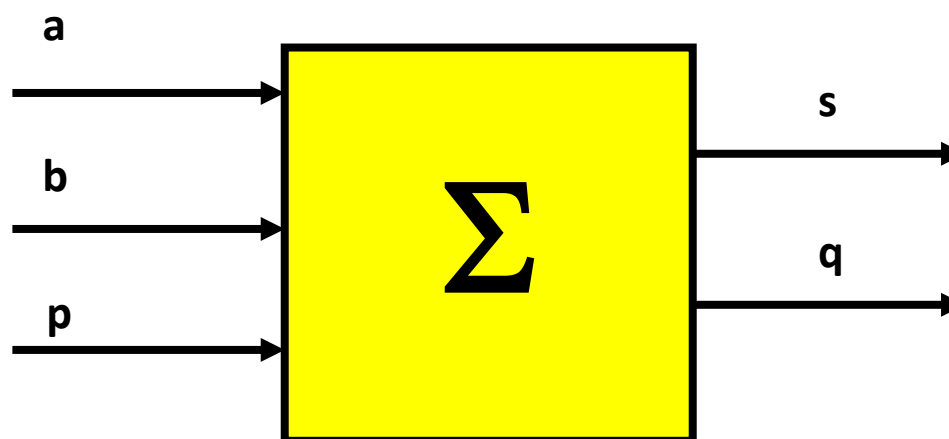
0+0=0, 0+1=1, 1+0=1, ale pozor 1+1=10 !!! Přenos !!!

... příklad 1



A co ten přenos ???

Určíme vstupy a výstupy



Řešení: tabulka

a	b	p	q	s
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

1. popis všech možností, tzn.
 2^3 řádků
2. výsledek musí odpovídat součtu, tzn. až dva bity (S a q) ...
tři jedničky dají binárně trojku
(11)
3. zapíšeme algebraickým
výrazem
4. realizujeme pomocí vybrané
technologie, zde hradel

Řešení - tabulka

a	b	p	q	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Řešení - tabulka

a	b	p	q	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$s = \bar{a}\bar{b}p + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp$$

$$q = \bar{a}bp + a\bar{b}p + ab\bar{p} + abp$$

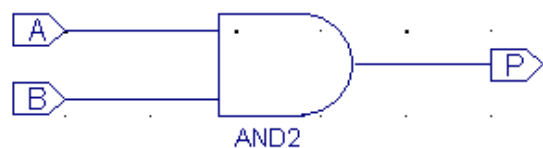
$$q = ap + bp + ab$$

Úpravy výrazů viz BI-MLO:
mapy, Booleova algebra
tabule ... další přednáška
a proseminář

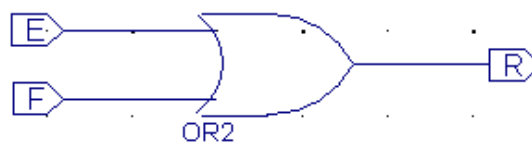
Realizace ???

Hradla (XILINX EDA tool: prostředí Vivado 2018.2)

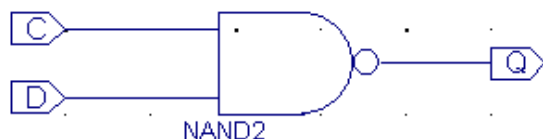
$A \cdot B = P$ and



$E + F = R$... or, V



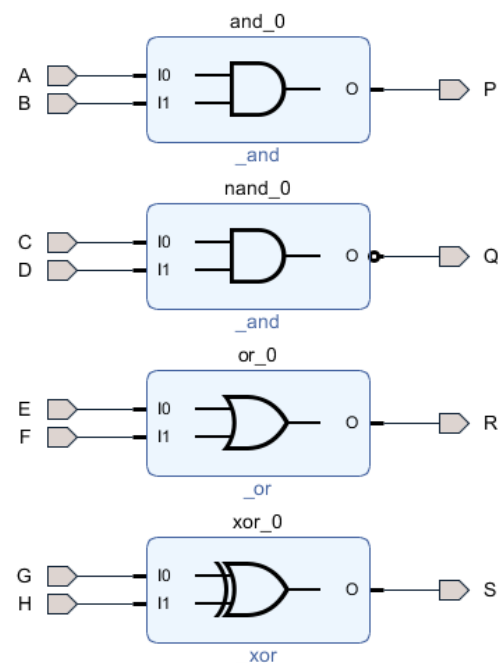
$C \cdot D = Q$



$G \text{ xor } H = S$

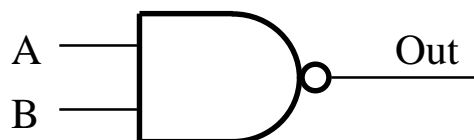


$G \oplus H = S$



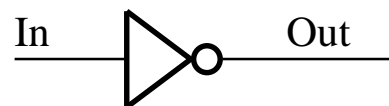
Funkce hradel, Booleova algebra

NAND Gate



A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

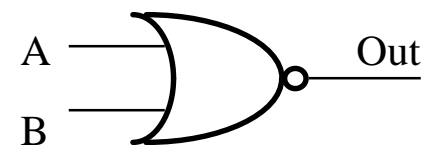
Invertor



$$\text{Out} = \overline{\text{In}}$$

In	Out
0	1
1	0

NOR Gate



A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

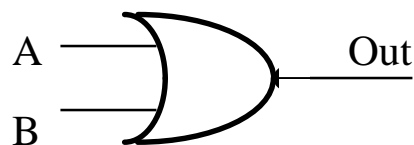
Funkce hradel, Booleova algebra

XOR hradlo



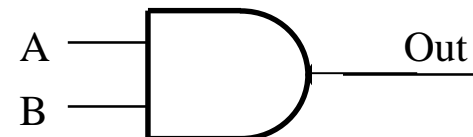
A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

OR hradlo



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

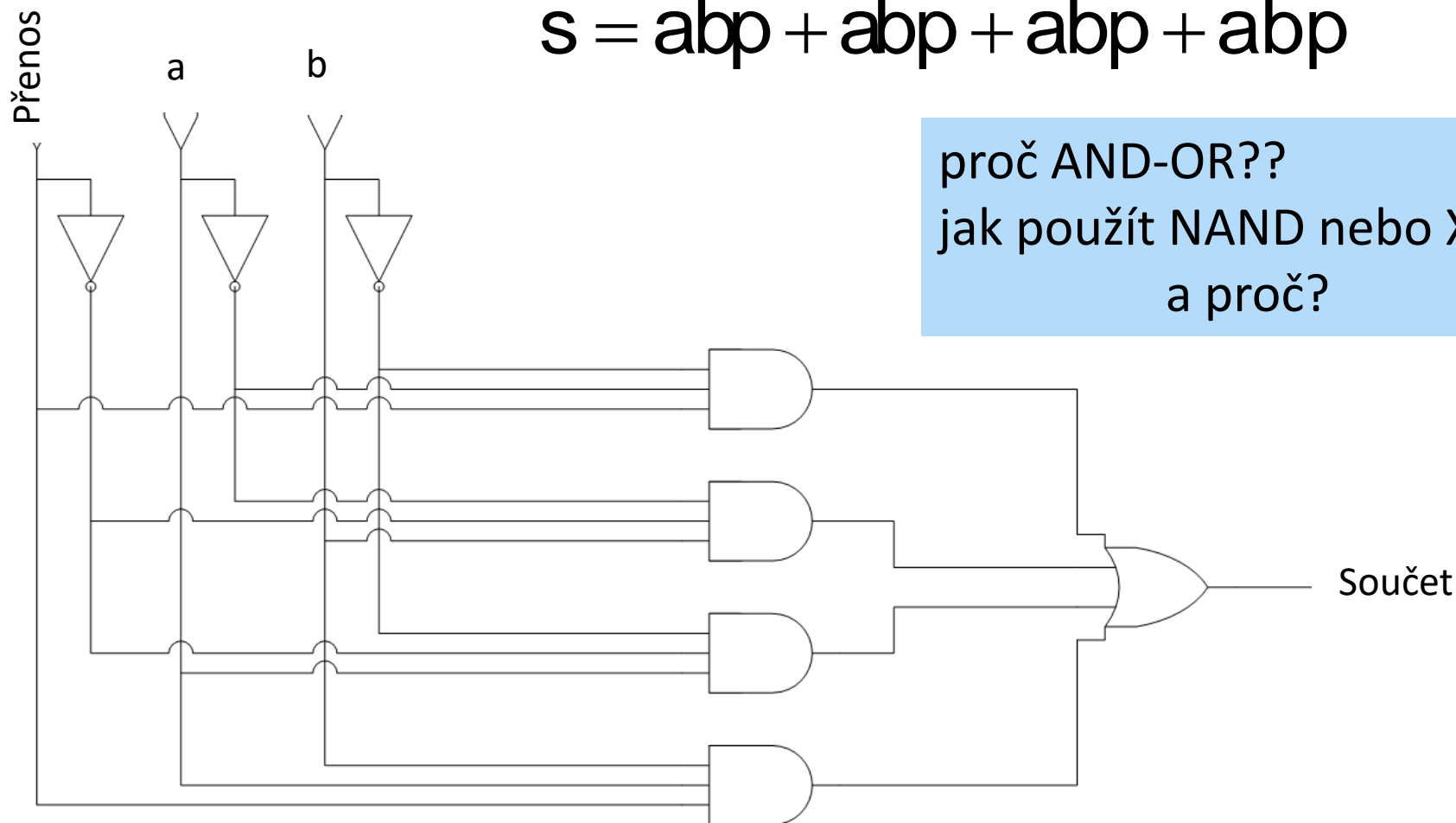
AND hradlo



A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

Sčítačka - realizace

$$s = \bar{a}\bar{b}p + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp$$

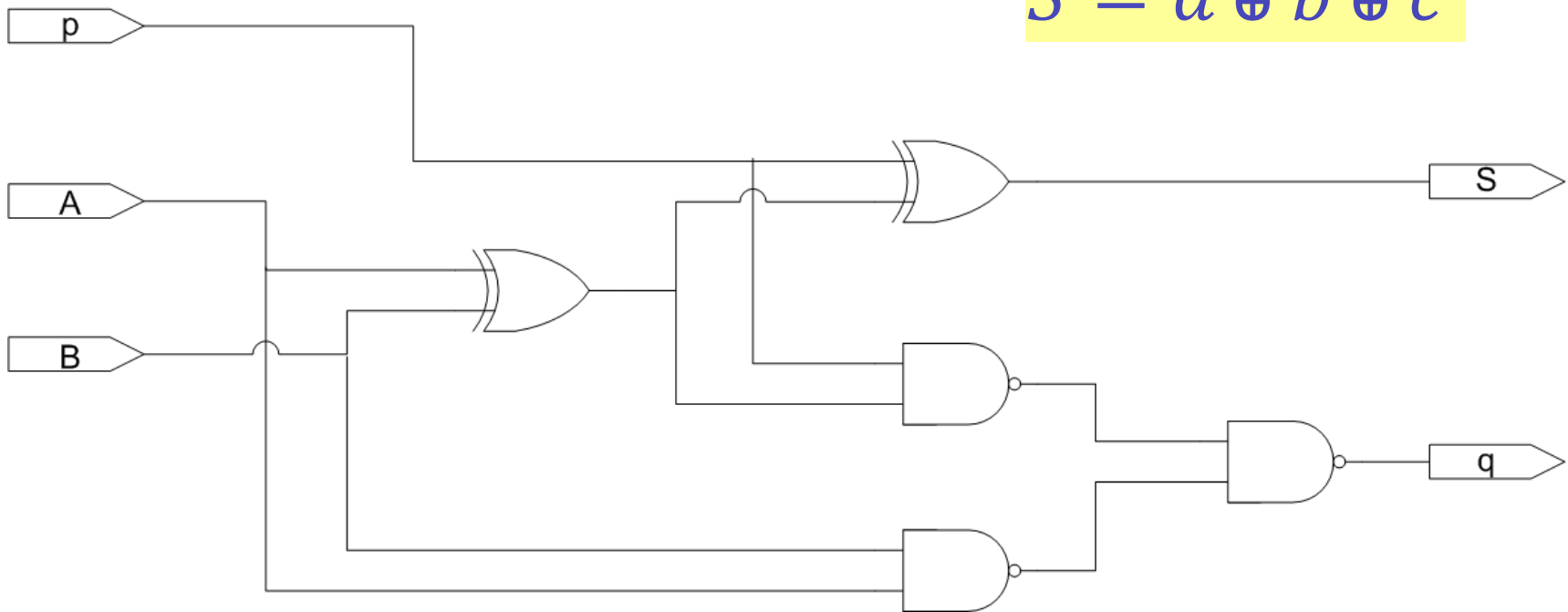


Realizace sčítačky jinak

$$s = \bar{a}\bar{b}p + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp$$

po úpravách lze zapsat i takto:

$$S = a \oplus b \oplus c$$



$$q = \bar{a}bp + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp$$

ale také: $q = ab + p(a \oplus b)$

$$q = ap + bp + ab$$

Sčítačka: ještě jiné řešení

