

# Y36SAP

<http://service.felk.cvut.cz/courses/Y36SAP/>

Úvod  
Návrhový proces  
Architektura počítače

2008-Kubátová

Y36SAP-Úvod

1

## Struktura předmětu

- Číslicový počítač, struktura, jednotky a jejich propojení.
- Logické obvody, formy jejich popisu, kombinační obvody a jejich realizace na úrovni hradel.
- Sekvenční obvody a jejich realizace.
- Typické kombinační a sekvenční obvody v číslicových počítačích - jejich realizace (kodéry, sčítačky, čítače, registry)
- Data, jejich zobrazení a zpracování.
- Realizace aritmetických operací.
- Soubor instrukcí a strojový kód.
- Návrh procesoru
- Paměti – struktura paměťového obvodu, paměťový systém počítače.
- Vstupy a výstupy
- Řadiče. Procesory typu CISC a RISC.

2008-Kubátová

Y36SAP-Úvod

2

## Podmínky zápočtu

Předmět 2+2 zápočet, zkouška

- cvičení seminární + laboratorní (1:1)
- zápočet za:
  - 4 fungující laboratorní úlohy (7 bodů logika a 7 bodů - assembler AVR),
  - 2 testy (až 2x15 bodů)
  - celkem minimálně 25 bodů
- zkouška za:
  - 45 (a více) bodů v semestru
  - jinak body ze cvičení + zkouškový test (až 50 bodů)

2008-Kubátová

Y36SAP-Úvod

3

## Cíle předmětu

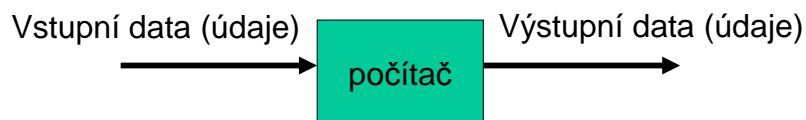
- globální přehled o architektuře počítačů a jejím vývoji (tedy i historické souvislosti)
- odpovídá předmětům „Digital Design“
- tvoří ho základy předmětů SKD, LOB, JPO v programu E+I (X)
- navazuje na Y36ALG a Y31ELI
- využívá programovatelné obvody
- seznamuje s moderními návrhovými prostředky

2008-Kubátová

Y36SAP-Úvod

4

## Co je číslicový počítač



- Zobrazení dat
  - Nespojité – diskrétní (číslíkové, digitální)
  - Spojité - analogové
- Počítač
  - Analogový – spojité zobrazení dat
  - Číslíkový – nespojité zobrazení dat
  - Hybridní – obojí + A/D, D/A převodníky

2008-Kubátová

Y36SAP-Úvod

5

## Technologie – dramatický rozvoj, Mooreuv zákon

- Procesory
  - Logická kapacita: o 30% za rok
  - Hodinová frekvence: o 20% za rok
- Hlavní paměť
  - DRAM kapacita: o 60% za rok (4x každé 3 roky)
  - Rychlost – přístupová doba: o 10% za rok
  - Cena za bit: snížení o 25% za rok
- Disk
  - Kapacita: o 60% za rok
  - Využití dat: o 100% každých 9 měsíců
- Počítačové sítě –
  - šířka pásma o 100% za rok!

Každých 18 měsíců  
zdvojnásobení  
hustoty integrace

2008-Kubátová

Y36SAP-Úvod

6

## Historie – vývojové mezníky

- víc než 50 let uplynulo od vytvoření 1. univerzálního elektronického počítače
- dnešní PC jsou výkonnější než počítač z r. 1980 za miliony
- HW průlom: VLSI technologie a mikroprocesory (70. léta)
- SW průlom: univerzální na výrobci nezávislé OS (UNIX) a přechod od programování v SOJ (strojový jazyk) k programování ve vyšších jazycích
- nástup RISC (Reduced Instruction Set Computer) - důsledek:
  - paralelizmus na úrovni zpracování instrukcí – ILP (Instruction Level Parallelism), tj. proudové zpracování instrukcí, super-skalární architektury atd.
  - používání vnitřních skrytých pamětí (cache)
- průlom v navrhování: vývoj *kvantitativního přístupu k návrhu a analýze počítačů*, který využívá empirické pozorování, experimentování a simulace

## Chronologie v datech

- 60. léta: dominantní velké sálové počítače s aplikacemi jako
  - zpracování dat ve finanční sféře
  - rozsáhlé vědeckotechnické výpočty
- 70. léta: mikropočítače pro aplikace ve vědeckých laboratořích
- 80. léta: příchod stolních počítačů založených na mikroprocesorech (osobní počítače a pracovní stanice)
- dále se objevují servery a lokální sítě pro větší úlohy s větší pamětí a výkonem
- 90. léta: Internet a WWW technologie
- současnost: rozdělení počítačového trhu na 3 oblasti charakterizované rozdílným použitím, požadavky a počítačovou technologií:
  - osobní, stolní a přenosné počítače
  - servery a výkonné paralelní počítače a superpočítače
  - vestavné (embedded) a řídicí počítače v jednoúčelových zařízeních

## Reprezentace systému

- Funkční (behavioral or functional representation)
  - Popis **funkce** ne implementace **Co to má dělat**
  - Black-box + závislosti výstupů na vstupech v čase
- Strukturní
  - Popis **implementace** bez zvláštního popisu funkce (ta vyplývá ze vzájemného spojení bloků o známé funkci)
  - Vnitřek black-boxů **Jak**
- Fyzikální
  - Popisuje fyzikální vlastnosti každého black-boxu
  - Popisuje přesné vztahy mezi bloky (velikost, hmotnost, spotřebu, zahřátí, a to v každém bodě, vstupním i výstupním pinu) **Jak to vyrobit**

2008-Kubátová

Y36SAP-Úvod

9

## Úrovně abstrakce

Funkční, strukturní i fyzikální reprezentace může být použita na různém úrovní abstrakce (granularity) podle použitých typů objektů.

Dále na obr. – návrh elektronických systémů:

1. Transistor
2. Hradlo
3. Registr
4. Procesor

2008-Kubátová

Y36SAP-Úvod

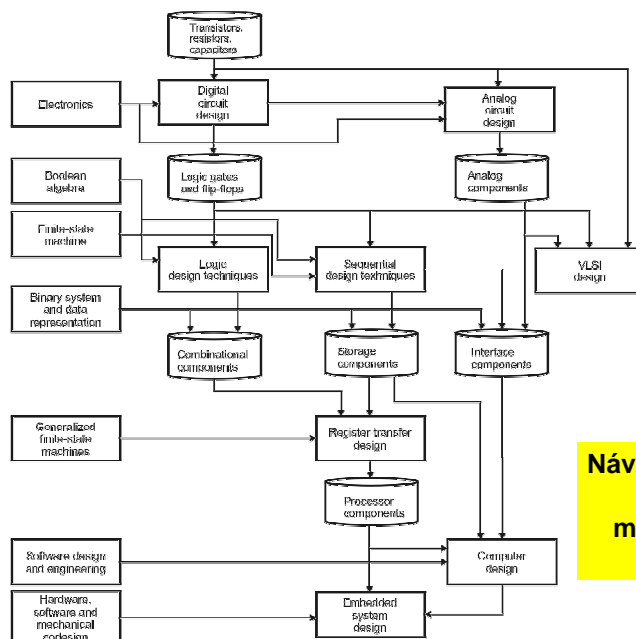
10

Úroveň abstrakce	Funkční popis	Strukturní bloky	Fyzikální objekty
transistor	Diferenciální rovnice, volt-ampérová charakteristika	Transistor, odpor, kondenzátor	Analogové a číslíkové buňky - layout
hradlo	Boolovské rovnice, konečný automat	Hradlo, klopný obvod	Moduly, bloky
registr	Algoritmus, vývojový diagram, soubor instrukcí	Sčítačka, komparátor, čítač, registr	Mikročipy
procesor	Specifikace funkce, program	Procesor, řadič, paměť	Desky plošných spojů, vícečipové moduly

2008-Kubátová

Y36SAP-Úvod

11

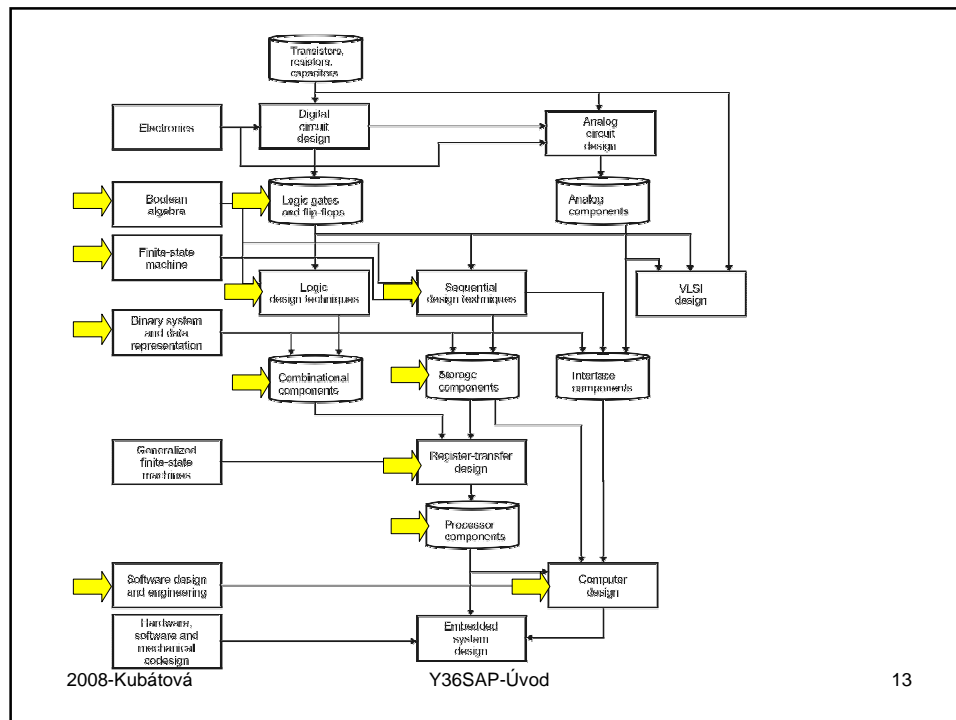


# Návrh počítačových systémů metodou „zdola nahoru“

2008-Kubátová

Y36SAP-Úvod

12



## Pohled zdola-nahoru x zhora-dolů

- Návaznost na Elektroniku - ELI  
transistory, jak vypadá hradlo  
zdola
- Algoritmizace  
programování, typy dat a jejich struktury  
shora
- SAP  
jak data uložit a jak je zpracovat a jak mají vypadat ty  
jednotky, které data zpracovávají – jak se postaví z  
hradel  
mezi

## Počítačový software

- **Firmware**  
BIOS, adresní módy, architektura souboru instrukcí – Instruction Set Architecture - **ISA**, jazyk symbolických instrukcí - **assembler**
- **Operační systém**  
Struktura souboru na disku, privilegia a ochrana, přepínání úloh, jádro, správa paměti a zařízení
- **Vývojářský SW**  
Assembler-překladač a linker; simulátor a debugger; knihovny
- **Aplikace**  
Programovací jazyky, editory, prohlížeče, hry, ...

2008-Kubátová

Y36SAP-Úvod

15

## Počítačový hardware

- **Architektura počítače**  
Architektura procesoru, provádění instrukcí, tok dat, řízení, predikce větvení
- **Paměťová hierarchie**  
vyrovnávací paměť (cache), správa paměťového systému, segmentace a stránkování
- **Uživatelské rozhraní**  
displej, grafické rozhraní, klávesnice, myš, porty
- **Další rozhraní**  
Přerušovací systém, DMA (Direct Memory Access), komunikační protokoly

2008-Kubátová

Y36SAP-Úvod

16



## Von Neumannova architektura (1)

- 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).

2008-Kubátová

Y36SAP-Úvod

17

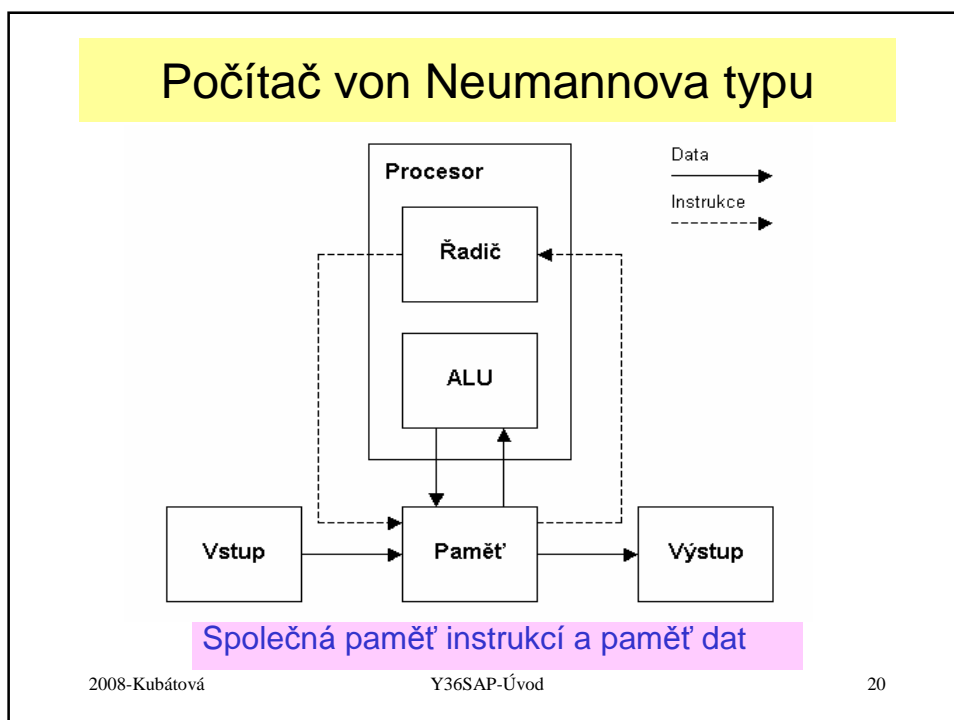
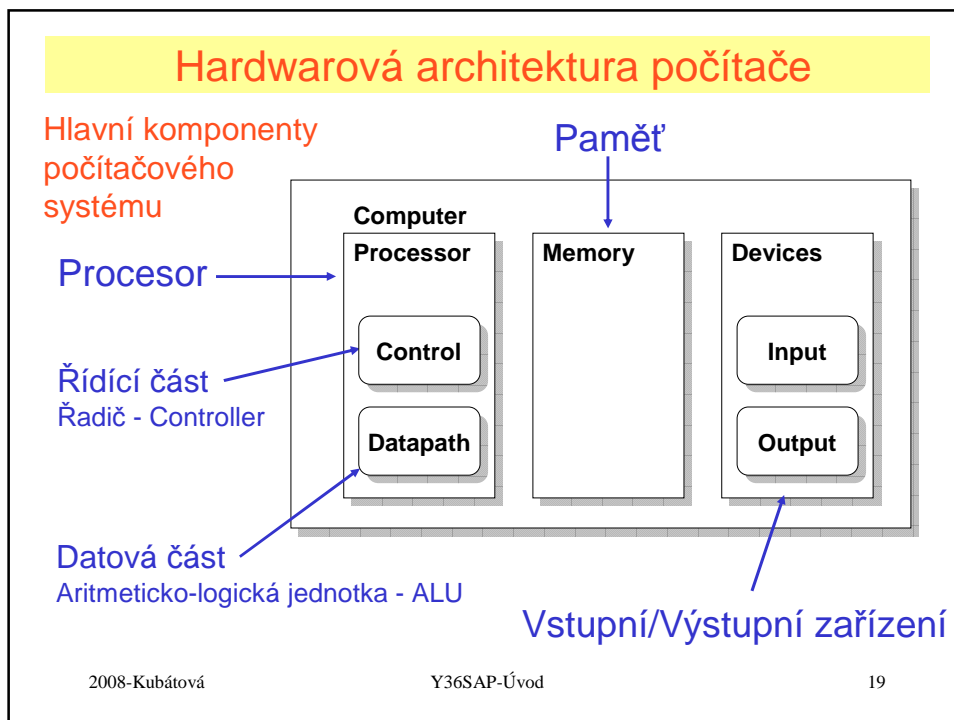
## Von Neumannova architektura (2)

- **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

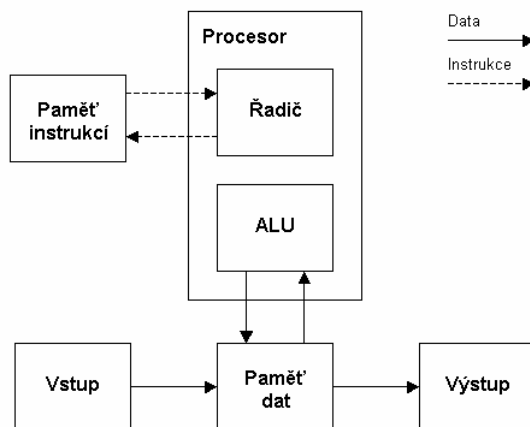
2008-Kubátová

Y36SAP-Úvod

18



## Harvardská architektura



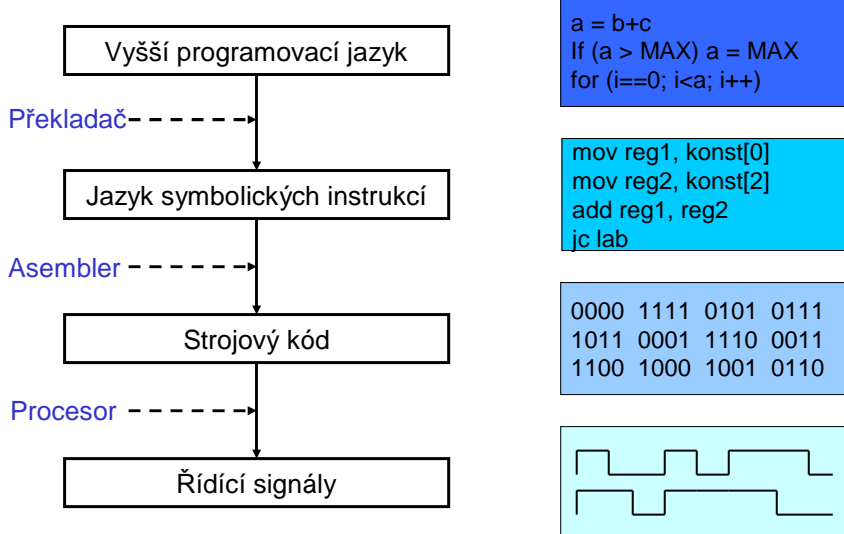
Oddělena paměť instrukcí a paměť dat

2008-Kubátová

Y36SAP-Úvod

21

## Vývoj softwaru – úrovně abstrakce



2008-Kubátová

Y36SAP-Úvod

22

## 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, 32b, 50b, b označuje bit)
  - slabika B – byte, 8b = 1B, obvykle 2 nebo i více slabik tvoří slovo, např. u procesorů Intel 80x86 – 1 slovo = 2B
- Obsah paměťového místa na adrese *adr* bývá někdy označován *<adr>*; nehrozí-li nedorozumění píše se však často *adr* místo *<adr>*.

2008-Kubátová

Y36SAP-Úvod

23

## 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

1. **big-endian** (IBM 360, Motorola 68000)

2. **little-endian** (Intel 80x86, DEC Alpha)

Oba způsoby (Motorola 88 110)

2008-Kubátová

Y36SAP-Úvod

24

## Zobrazení dat v paměti

### Numerická data – čísla:

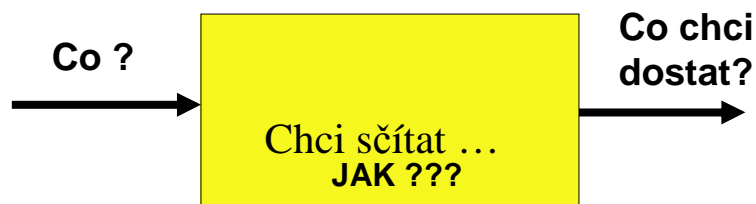
- V pevné řádové čárce – **fix point**, obvykle celá čísla (integer, byte, word ...)
- V pohyblivé řádové čárce – **floating point**, racionální čísla (real, float, ...)
- Dvojková – **binary**
- Desítková – **decimal**
- Šestnáctková – **hexadecimal**
- Bez znaménka – **unsigned**, pouze nezáporná (byte,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

2008-Kubátová

Y36SAP-Úvod

25

## Příklad 1

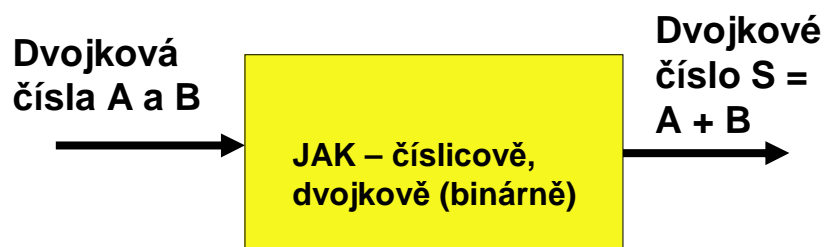


2008-Kubátová

Y36SAP-Úvod

26

## Příklad 1



Dvojková čísla budou nejprve 1 bitová, tzn.:

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

2008-Kubátová

Y36SAP-Úvod

27

## Příklad 1



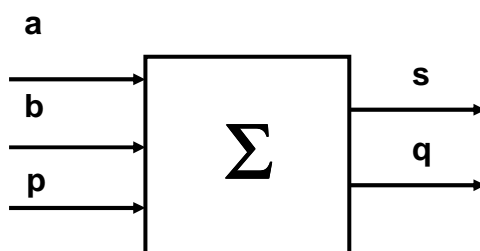
A co přenos ???

2008-Kubátová

Y36SAP-Úvod

28

## Příklad 1



2008-Kubátová

Y36SAP-Úvod

29

## Příklad 1 - intuitivně

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

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

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

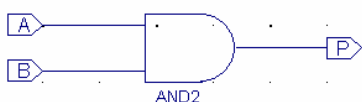
Úpravy výrazů později

2008-Kubátová

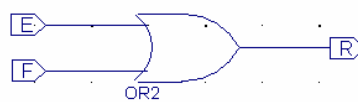
Y36SAP-Úvod

30

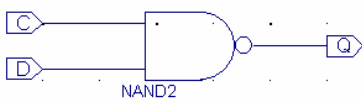
## Realizace ???



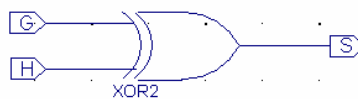
$$A \cdot B = P \dots \text{and}$$



$$E + F = R \dots \text{or, V}$$



$$\overline{C \cdot D} = P$$



$$G \text{ xor } H = S$$

2008-Kubátová

Y36SAP-Úvod

31

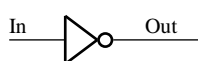
## Funkce hradel, Booleova algebra

NAND Gate



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

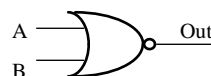
Invertor



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

In	Out
0	1
1	0

NOR Gate



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

2008-Kubátová

Y36SAP-Úvod

32



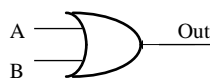
## Funkce hradel, Booleova algebra

**XOR Gate**



G	H	S
0	0	0
0	1	1
1	0	1
1	1	0

**OR Gate**



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

**AND Gate**



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