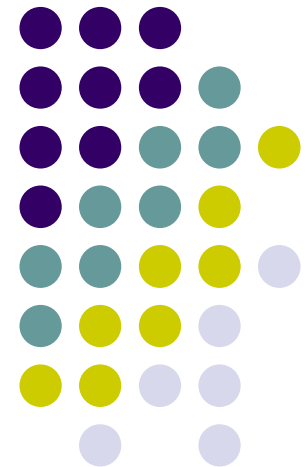
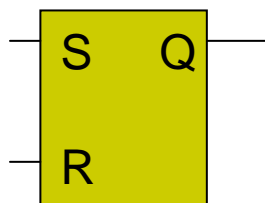


# PPI

26.10.2011



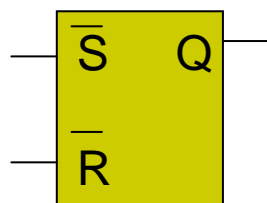
asynchronny PO SR



	S		R		
	_____			_____ Q	
0	0	1	X	0	0
1	1	1	X	0	1

	S	R	
P	1	x	0

asynchronny PO  $\overline{S}\overline{R}$



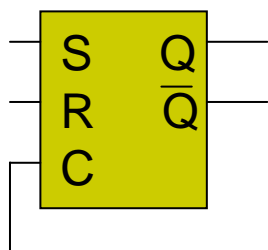
	$\overline{S}$		$\overline{R}$		Q
0	X	0	0	1	0
1	X	0	1	1	1

	$\overline{S}$	$\overline{R}$	
X	0	P	1





synchronný PO SR



	S		R		Q
0	0	1	X	0	0
1	1	1	X	0	1

	S		R
P	1	X	0

synchronný PO JK

	J		K		Q
0	0	1	1	0	0
1	1	1	0	0	1

univerzálny

všetky typy správania

	J		K
P	1	K	0

synchrónny PO-D

	<u>D</u>		Q
0	0	1	0
1	0	1	1

<u>D</u>	
0	1

synchrónny PO-T

	<u>T</u>		Q
0	0	1	0
1	1	0	1

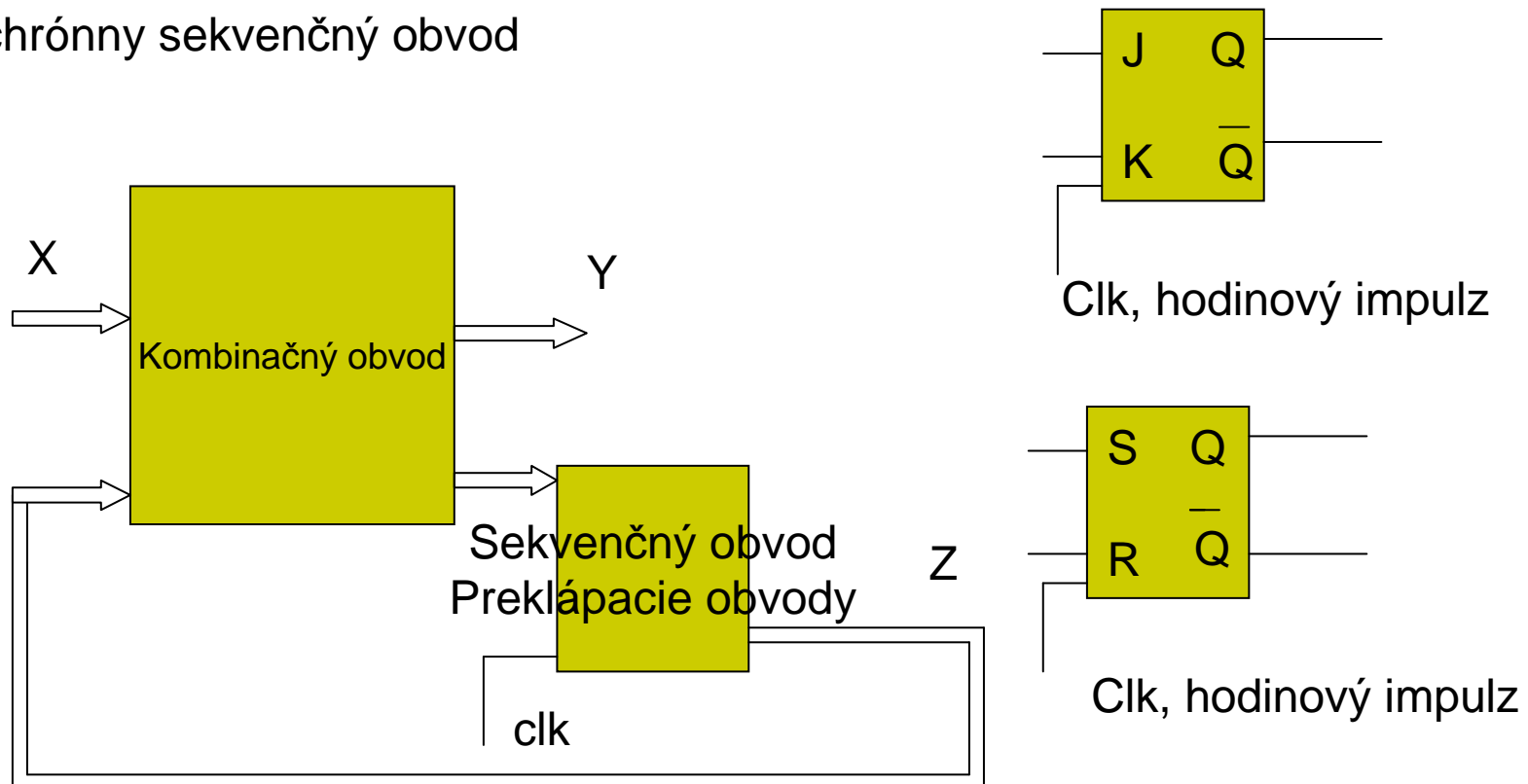
<u>T</u>	
P	K



**Podľa prednášky LOGICKÉ SYSTÉMY**, Návrh digitálnych systémov na úrovni logických obvodov, Norbert Frištacký, Katedra informatiky a výpočtovej techniky, FEI-STU, 2003

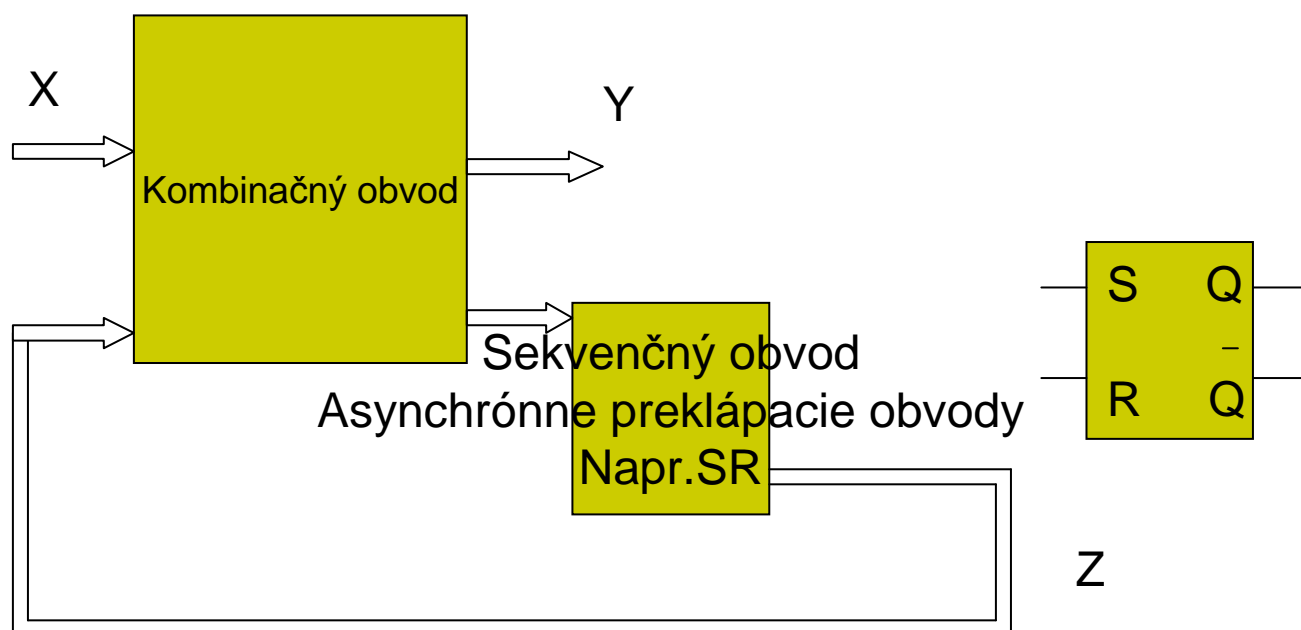


## Synchrónny sekvenčný obvod





## Asynchrónny sekvenčný obvod



# Postup pri návrhu asynchrónneho sekvenčného obvodu.



- Kritérium optimálnosti-minimálny počet preklápacích obvodov, minimálny počet logických členov v kombinačnej časti alebo maximálna operačná rýchlosť.

## Postup

- Návrh fundamentálneho automatu 1, rádu, kódovanie stavov, kontrola kritických a nekritických súbehov
- Návrh budiacich funkcií pre stavové premenné
- Návrh výstupných funkcií
- Skupinová minimalizácia budiacich funkcií.



Základným formálnym špecifikačným prostriedkom je tzv. **fundamentálny stavový stroj** (FSM, automat)

$$M = (A, Q, U, p, v),$$

ktorý má nasledujúcu vlastnosť: pri každom stave  $q \in Q$  a pri každom vstupnom symbole  $v \in A$  platí:

vstupné slovo:  $v \quad v \quad v \quad .. \quad v \quad v \quad v \quad v \quad v \quad ....$

postupnosť stavov:  $q \quad q_1 \quad q_2 \quad .. \quad q_k \quad q' \quad q' \quad q' \quad q' \quad ....$ ,

kde  $k \geq 0$  je pevné číslo pre danú dvojicu  $(q, v)$

**Formálne:** Pri dostatočne veľkom čísle "n" pre každý stav  $q$  a vstupný symbol  $v$  platí:

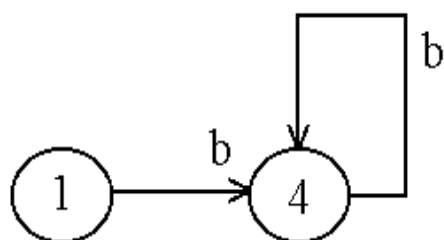
$$p(q, vn) = p(q, vn+1), \quad n \geq k$$





Stav "q" v danom FA sa nazýva **stabilný pre daný vstupný symbol "v"** práve vtedy ak platí  $p(q, v) = q$ .

Vo fundamentálnom FSM pre každý stav q a vstupný symbol v platí, že jeho dostatočnom opakovaní sa automat **dostane do stabilného stavu** pre vstupný symbol v.

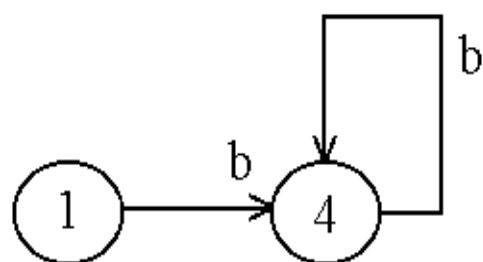


V prech.  
grafe

M1	a	b	c	d	
1	①	4	①	①	u
2	1	②	②	4	w
3	③	2	1	③	r
4	3	④	1	④	h
S	p				v

$p(1, a) = p(1, aa) = 1$   
 $p(1, b) = p(1, bb) = 4$   
 $p(2, d) = p(2, dd) = 4$   
 $p(2, a) = p(2, aa) = 1$   
 atd.

fundamentálny automat prvého rádu

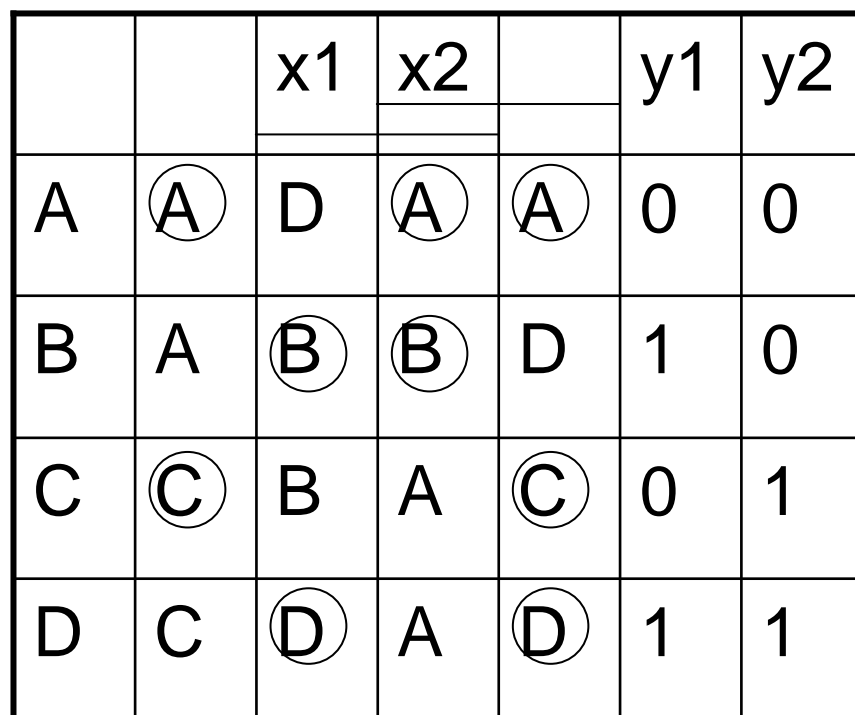


V prech.  
grafe

M1	a	b	c	d	
1	①	4	①	①	u
2	1	②	②	4	w
3	③	2	1	③	r
4	3	④	1	④	h
S	p				v

$p(1, a) = p(1, aa) = 1$   
 $p(1, b) = p(1, bb) = 4$   
 $p(2, d) = p(2, dd) = 4$   
 $p(2, a) = p(2, aa) = 1$   
 atd.

FFSM 1.rádu rozpoznať podľa toho, že prechodovej funkcií musí platiť: Ak je pri niektorom vstupnom symbole (vektore) t.j. v príslušnom stĺpci v prechodovej tabuľke niektorý stav q nezakrúžkovaný, potom musí byť pri tomto vstupnom symbole (v tomto stĺpci) aj stav q zakrúžkovaný.



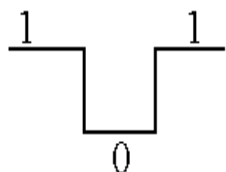
Prechod A->C-----v tabuľke nie je  
Prechod C->A ---- 11 -> 00 pri vstupe 11  
11->01->00 CBB, CBA je.. kritický súbeh  
11->10->00.....CDA nekritický súbeh  
Prechod D->B---v tabuľke nie je  
Prechod B->D-----01->10 pri vstupe 01  
01->00->10....BAA kritický súbeh  
01->11->10.....BCC kritický súbeh



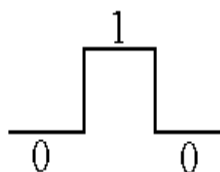
A	B
C	D

A	C	2	B
D	3		1

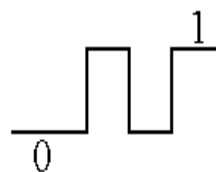
Univerzálny kód



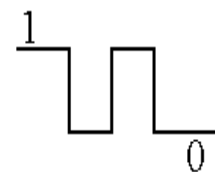
**statický  
hazard v 1**



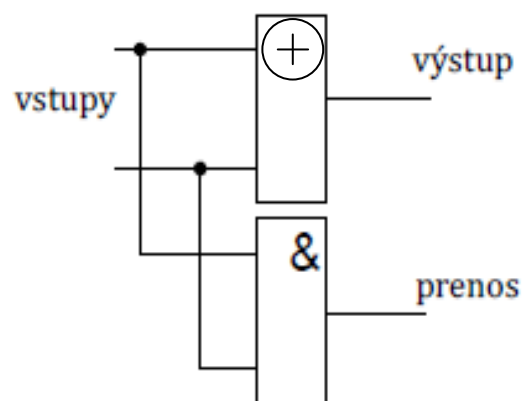
**statický  
hazard v 0**



**dynamický  
hazard pri 0→1**



**dynamický  
hazard pri 1→0**

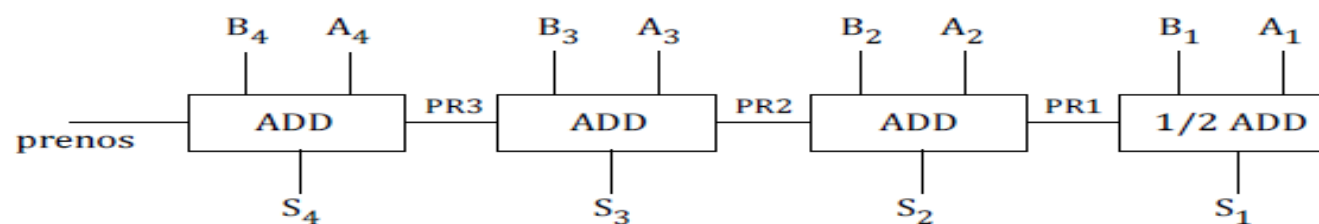
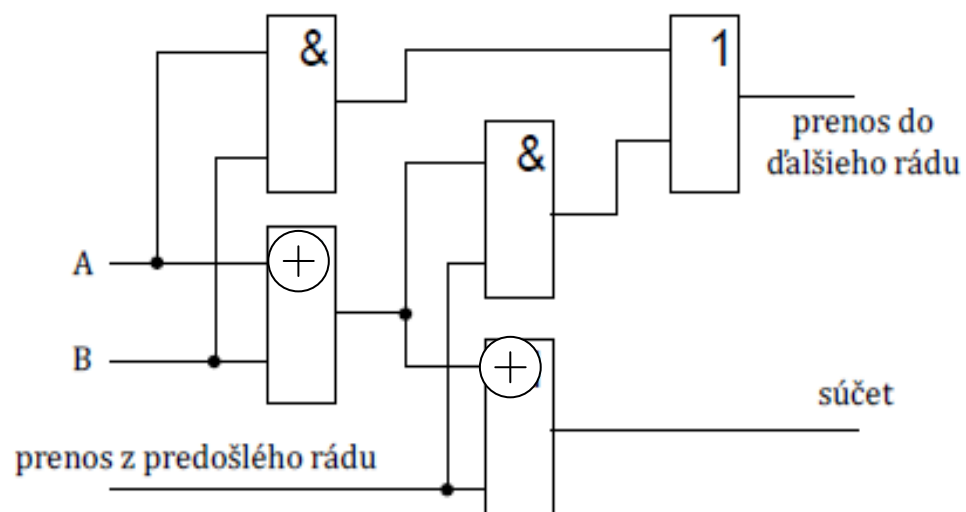


Polovičná sčítacia

$\oplus$  XOR

Úplná sčítacia

Sériová sčítacia



# Prepojovací podsystém počítača

**Zbernice** (prepojený každý podsystém počítača s každým)  
(v jednom okamihu len 1 vysielateľ)



## Rozdelenie zberníc

### 1. Podľa spôsobu riadenia

Single master iba 1 nadriadený podsystém-master

Multi-master – každé zariadenie môže riadiť zbernicu, ale v danom okamžiku iba jedno

### 2. Podľa synchronizácie prenosu

Synchrónne zbernice synchronizované synchronizačným impulzom

Asynchrónne zbernice-prenos synchronizovaný odpoveďou  
podriadeného, pomalšie

# Rozdelenie zberníc



## 3. Podľa časového multiplexu

Multiplexované zbernice-druh informácie sa mení s časom (adresa, inštrukcia, údaj)

Nemultiplexované zbernice – význam a druh informácie sa s časom nemení

## 4. Podľa tvaru prenášaných údajov

Paralelné zbernice

Sériové zbernice (prenos bit po bite)

Dnešné počítačové zbernice sú paralelné, asynchrónne, nemultiplexované a skladajú sa z nasledujúcich sekcií – adresová (adresa pamäte, v/v zariadenia), údajová (inštrukcie), riadiaca (povely, žiadosti)



## Processor

- Interpretuje inštrukcie programu
  - Výber inštrukcie z pamäte
  - Vykonanie operácie s operandami
  - Realizuje sa prenos informácií medzi časťami počítača

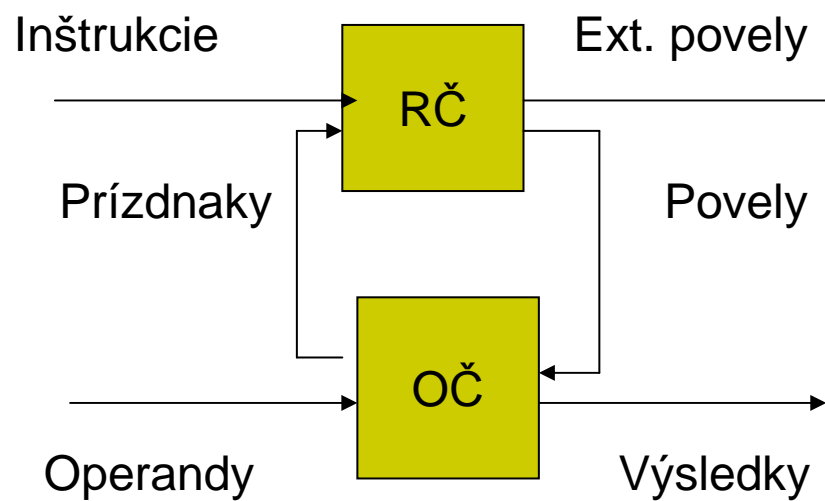
Univerzálne procesory (bohatý, úplný inštrukčný súbor)

Problémovo-orientované procesory špecializované funkcie



- **Hlavné časti procesora**

- Operačná časť
- Riadiaca časť





## Operačná časť

- Aritmeticko logická jednotka, ALU
- Registre
- Komunikačné obvody pre medziregistrové prenosy

### ALJ

- Paralelná dvojková sčítačka
- Sériové sčítačky
- Funkčné jednotky pre logické operácie
- Posúvacie obvody, logický posun, aritmetický posun, kruhový posun