

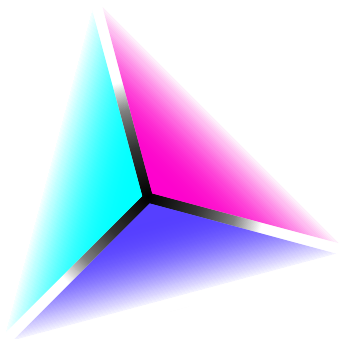
Struktura a architektura počítačů

Katedra číslicového návrhu
Fakulta informačních technologií
České vysoké učení technické

© Hana Kubátová, 2021

Sekvenční obvody

BI-SAP, březen 2021



Obsah

- Logické obvody sekvenční
- Formy popisu
- Postup návrhu až k realizaci
- Příklady

Cíl:

naučit se základní principy týkající se sekvenčních obvodů a jak navrhnout takový (řídící) automat

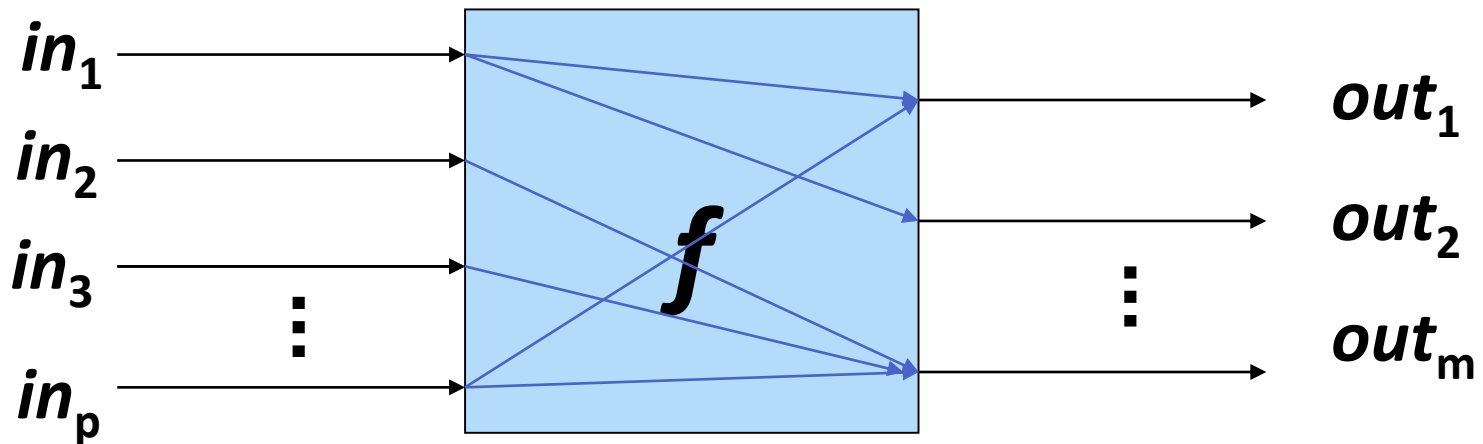
Kombinační x sekvenční obvody

- Kombinační: výstup je dán **kombinací** vstupů, „nezáleží“ na čase
- Sekvenční: výstup závisí na **posloupnosti** (sekvenci) hodnot na vstupech, „zapamatování“ se realizuje **zpětnou vazbou**
- Vše lze matematicky popsat
 - Logická funkce – ***f*** (*Booleova algebra*)
 - Konečný automat – **FSM** (*Finite State Machine*)

Kombinační funkce

Kombinační funkce:

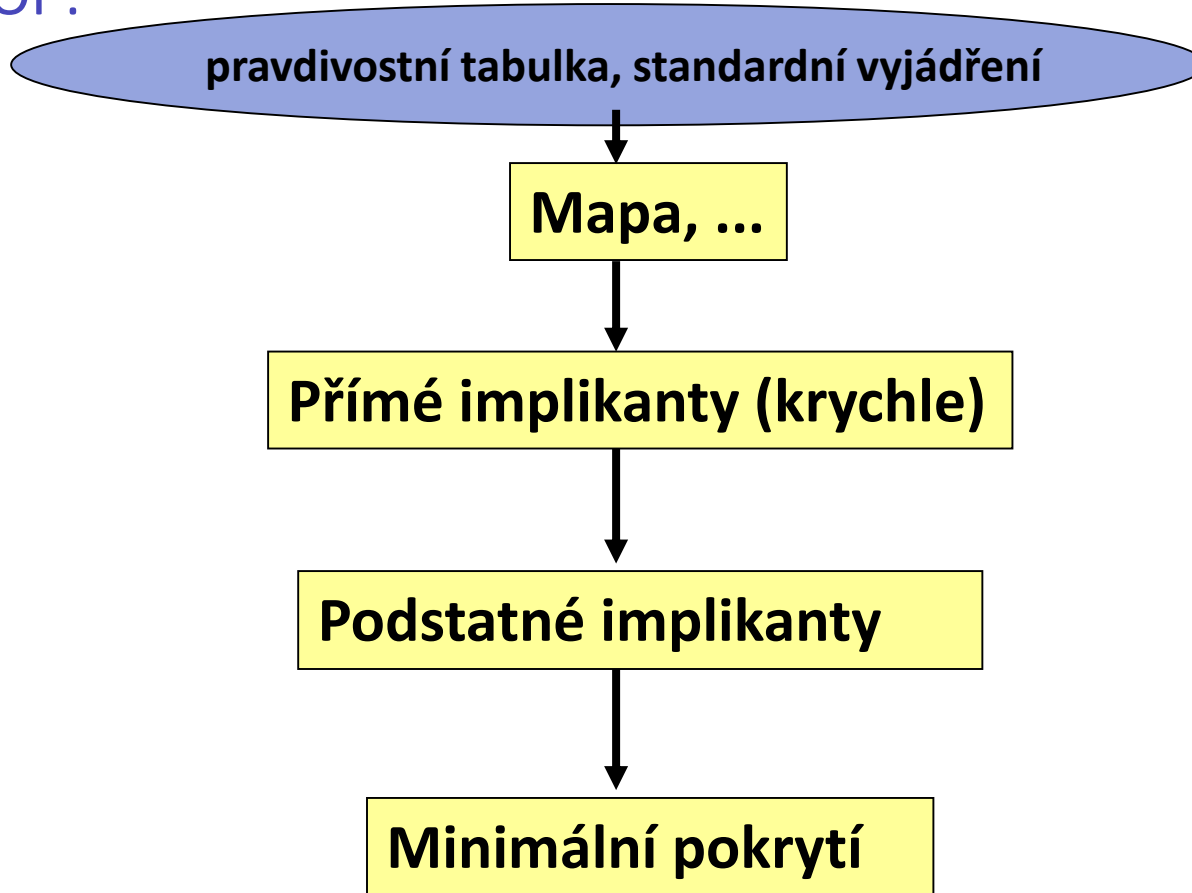
$$out_k = f_i(in_1, in_2, in_3, \dots in_p), k=1,2,\dots,m, i=1,2,\dots,p$$



Algoritmus minimalizace

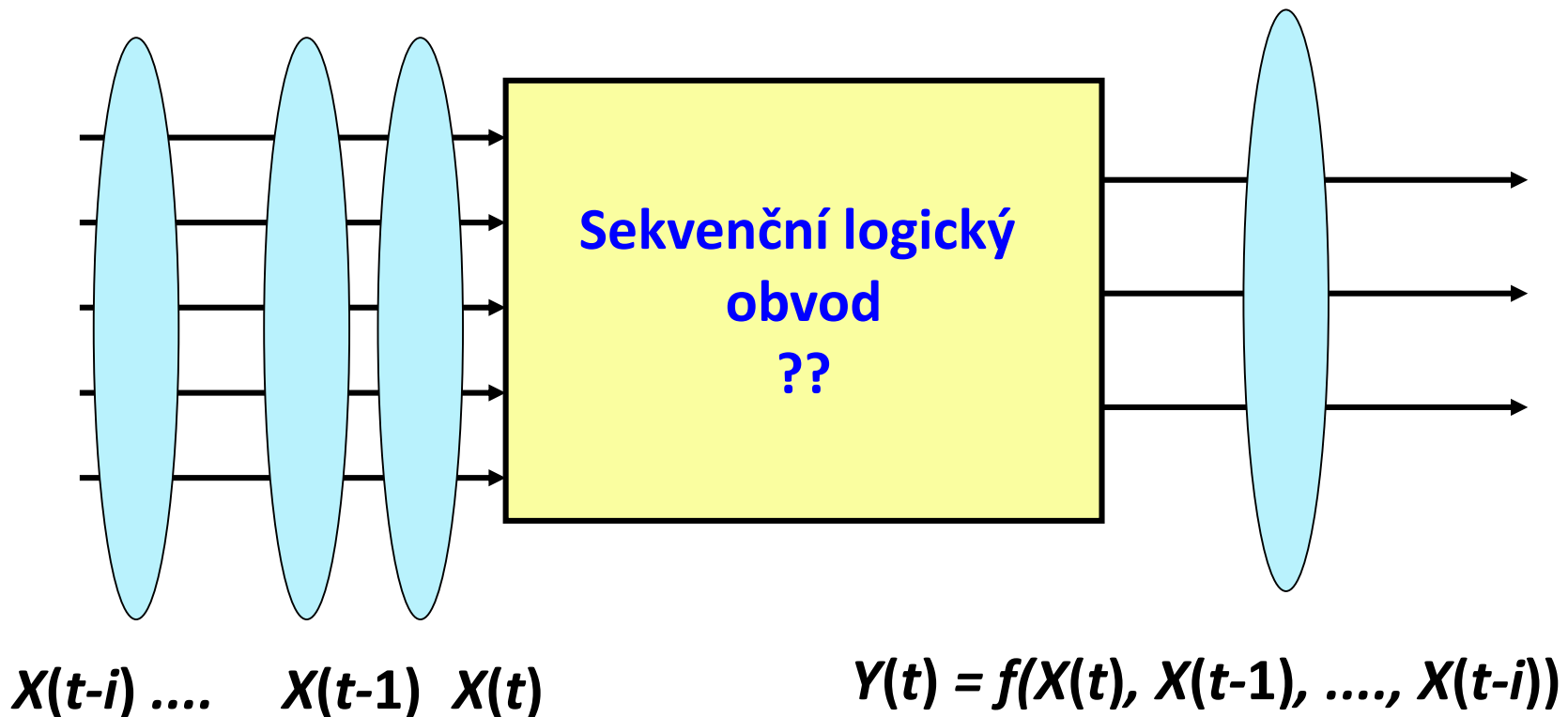
... opakování, ale bude se hodit i pro návrh sekvenčních obvodů.

zde pro SOP:



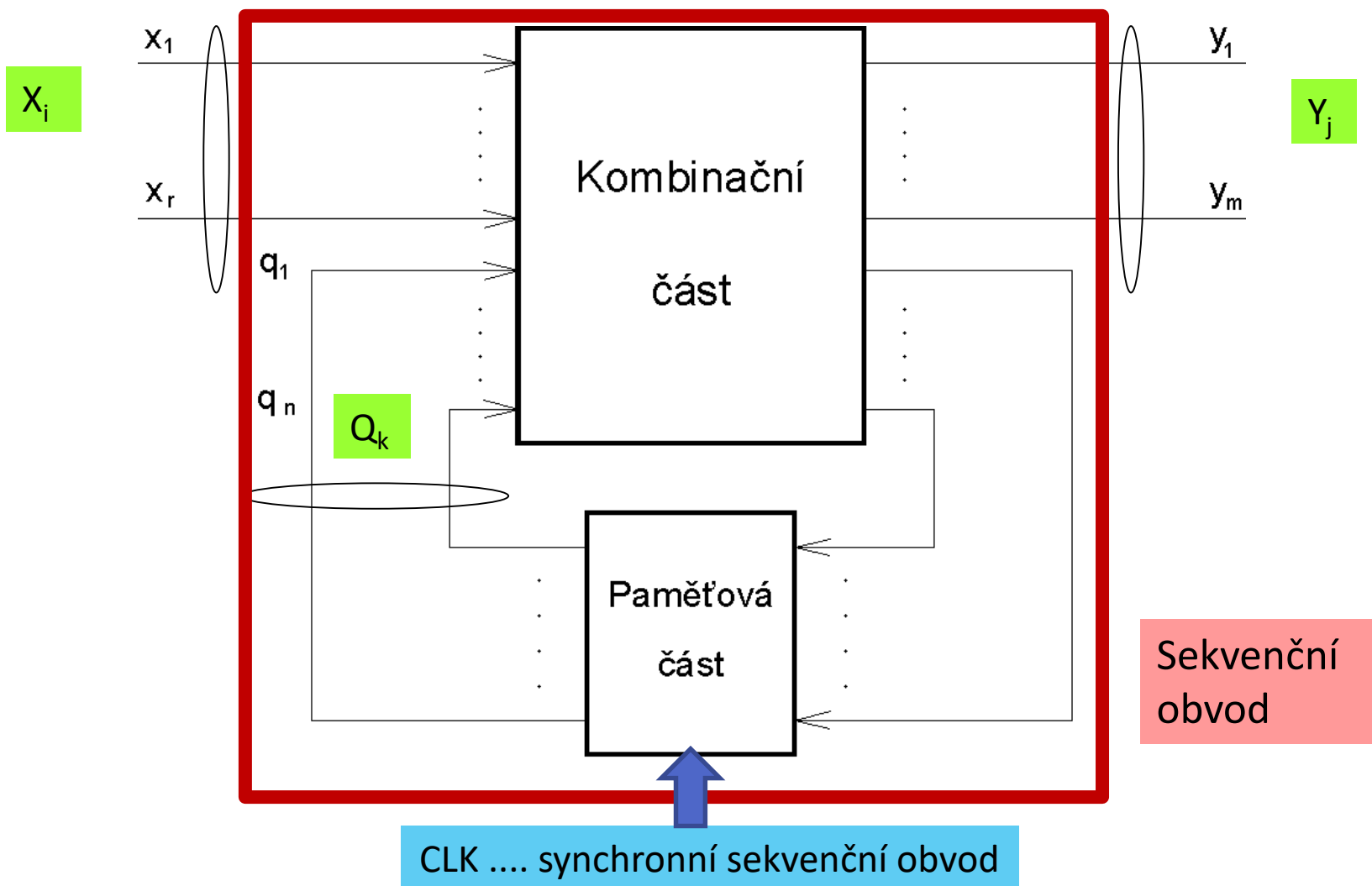
Sekvenční chování

- výstup závisí na posloupnosti (sekvenci) hodnot na vstupech



Obecný model sekvenčního obvodu

též *Huffmannův*



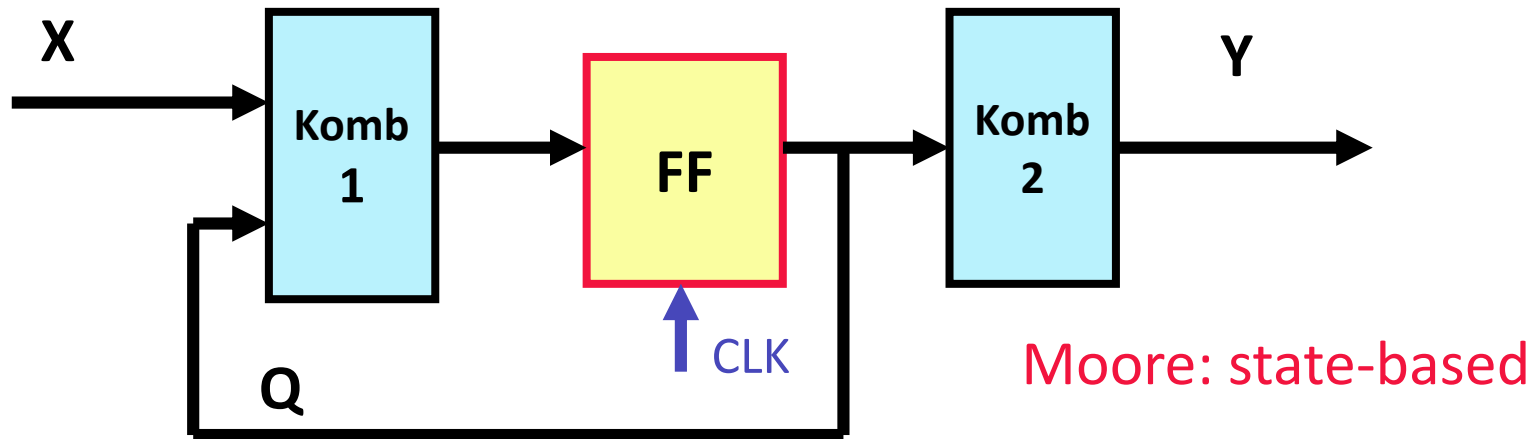
Sekvenční logický obvod

Model, matematický popis: konečný automat KA
(*FSM: finite state machine*)

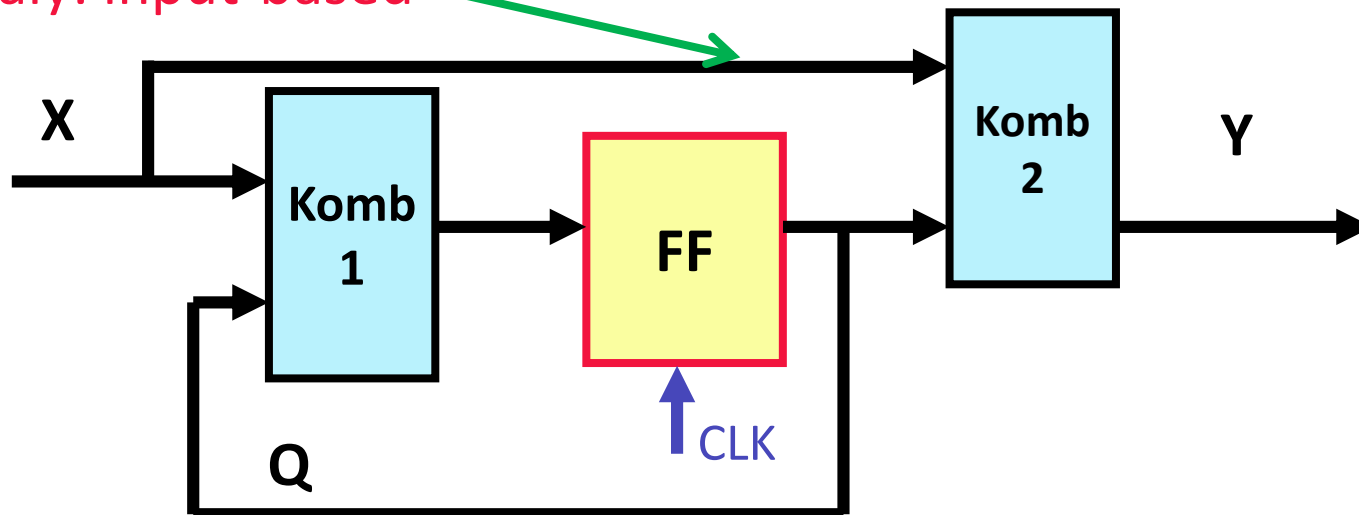
- X** ... Množina přípustných kombinací hodnot vstupních proměnných KA;
př: pro 3 vstupní proměnné $\Rightarrow X$ může obsahovat až $2^3=8$ kombinací, ale všechny nemusí být možné (přípustné)
- Y** ... Množina možných kombinací výstupních hodnot KA
- Q** ... Množina kombinací hodnot vnitřních proměnných KA (množina stavů)
- Q_0** ... Počáteční stav (kombinace hodnot vnitřních proměnných KA v počátečním stavu)
- δ** ... Stavově přechodová funkce:
 $\delta : X \times Q \rightarrow Q$... definuje příští vnitřní stav(y) KA
- λ** ... Výstupní funkce ... definuje výstup(y) KA:
 $\lambda : a) X \times Q \rightarrow Y$... typ Mealy
b) $Q \rightarrow Y$... typ Moore

Mealy, Moore

Rozdíl: přímá vazba na výstup buď je (Mealy) nebo není (Moore)



Mealy: input-based



Postup návrhu sekvenčního obvodu

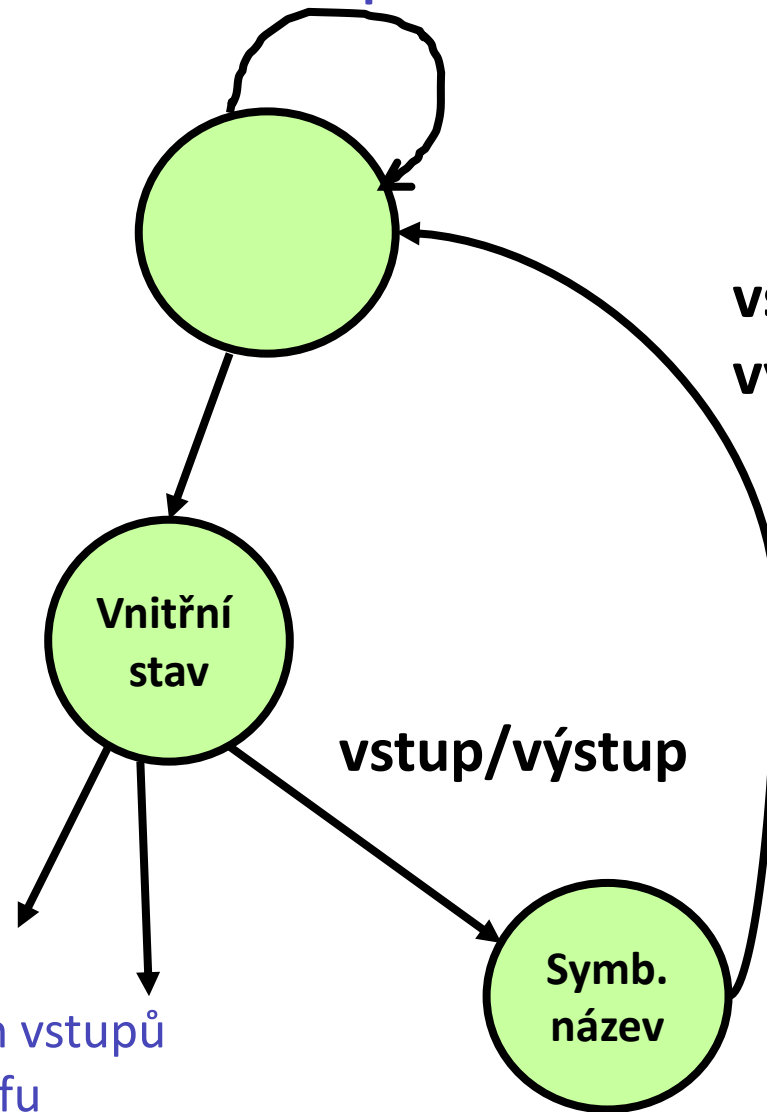
1. Slovní popis
2. Graf přechodů („*state-transition graph* - STG“)
3. Tabulky přechodů a výstupů
4. Zakódování vstupů, výstupů a vnitřních stavů
5. (Zakódované tabulky přechodů a výstupů)
6. Minimalizace výrazů pro budící vstupy vybraného typu klopných obvodů (mapy)
7. Minimalizace výrazů pro výstupní funkce
8. Realizace z (předepsaného typu) hradel
9. Výpočet hodinové frekvence

Graf přechodů

Uzly: vnitřní stavy

Hrany: ohodnoceny vstupem a výstupem mezi stavy, pro automat Moore jen vstupem, výstup se projevív až v cílovém stavu

Čím více přípustných vstupů tím větší větvení grafu



vstupní symbol/
výstupní symbol

vstup/výstup

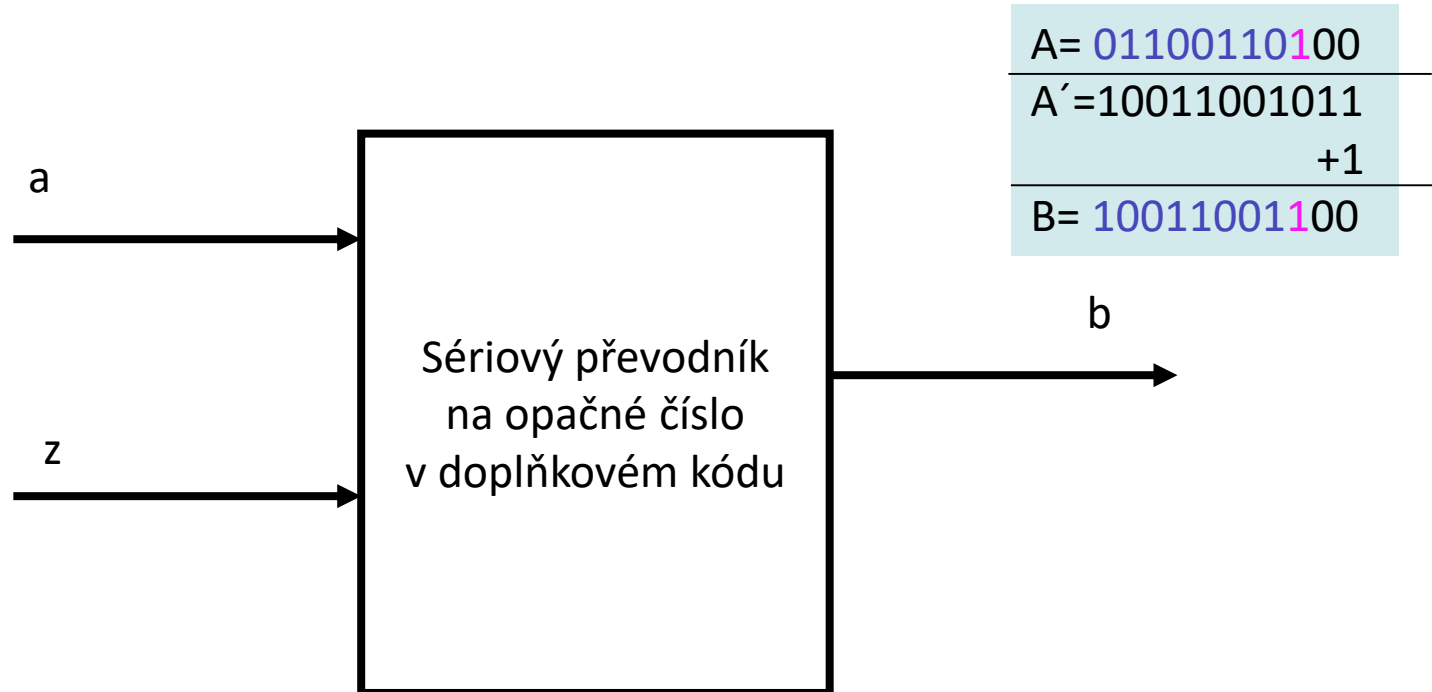
Podle zadání může
být vše (i vstupy a výstupy)
v symbolické formě.

Příklady

Příklad 1:

- Navrhněte SSO se dvěma vstupy a , z a jedním výstupem b , který bude převádět sériově vstupující binární číslo A v doplňkovém kódu na číslo B opačné k A . A vstupuje nejnižším řádem napřed, z indikuje začátek čísla A . (jestliže je $z=1$, na vstupu je nejnižší řád A).
- Poznámka: automat není iniciální

Postup řešení př. 1



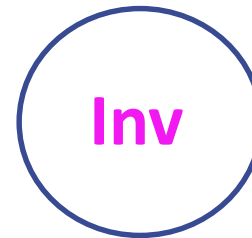
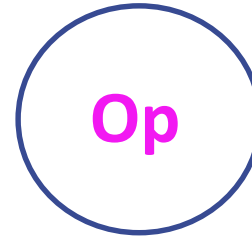
Postup řešení př. 1: vnitřní stavy

Hledáme princip funkce:
buď číslice opisují nebo
negují:

alespoň dva vnitřní stavy:

Q_0 : opiš (**Op**)

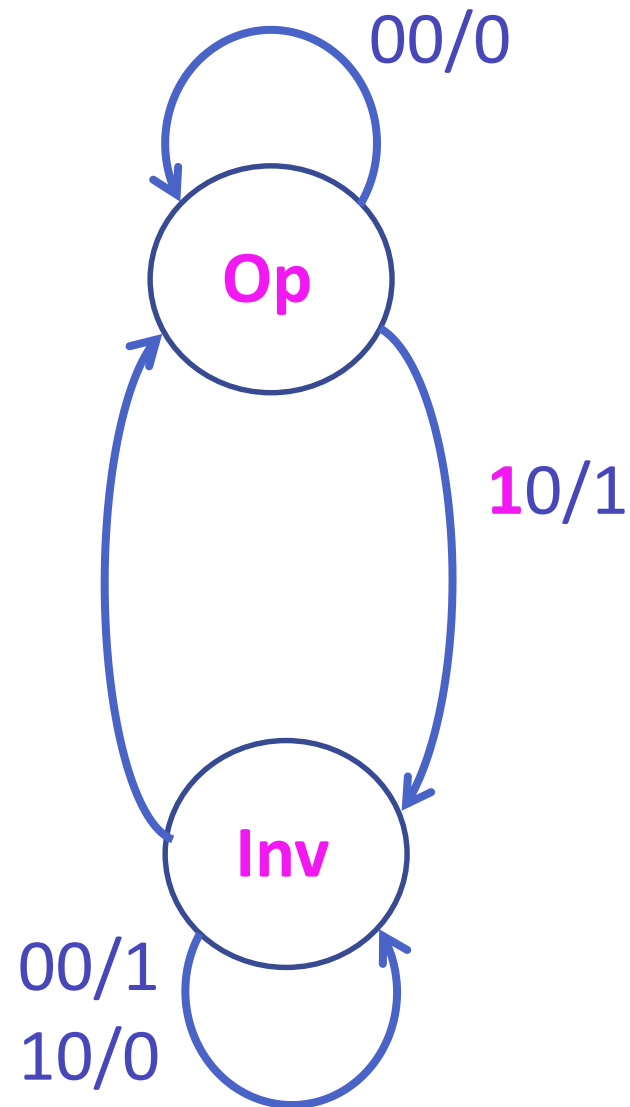
Q_1 : invertuj (**Inv**)



Postup řešení př. 1: přechody mezi stavy

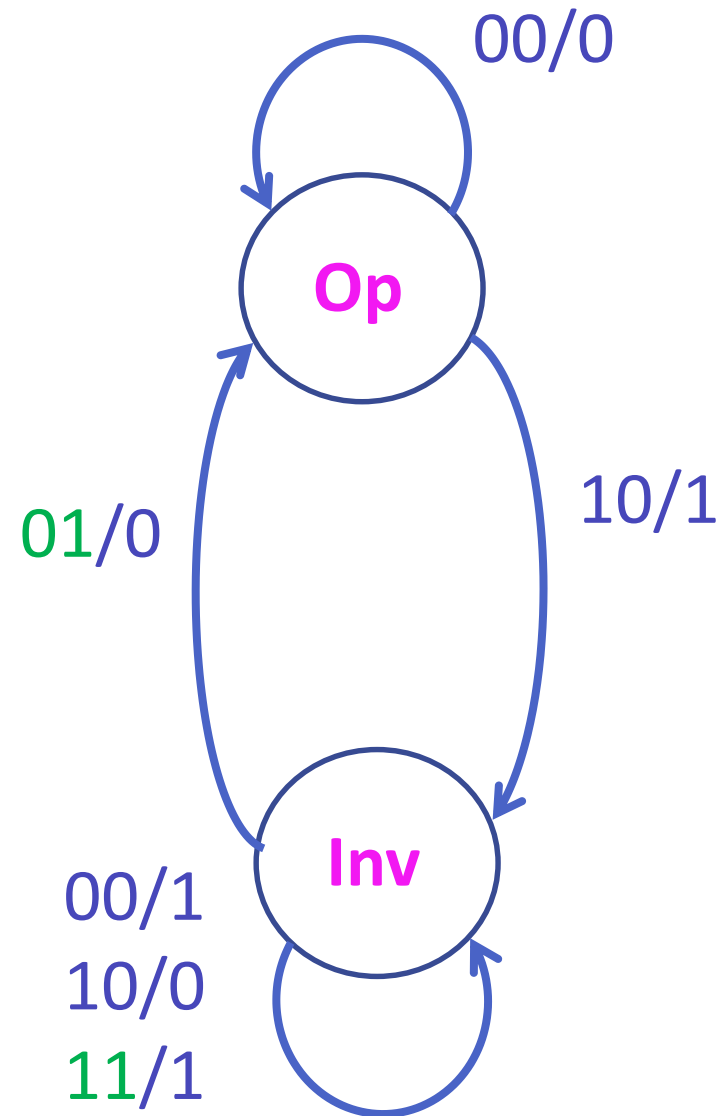
Hledáme princip funkce:
opisujeme dokud nepřijde
první jednička na a , pak
invertujeme
(zatím $z=0$)

přechody podle
vstupů, označíme
vstupy/výstup: az/b



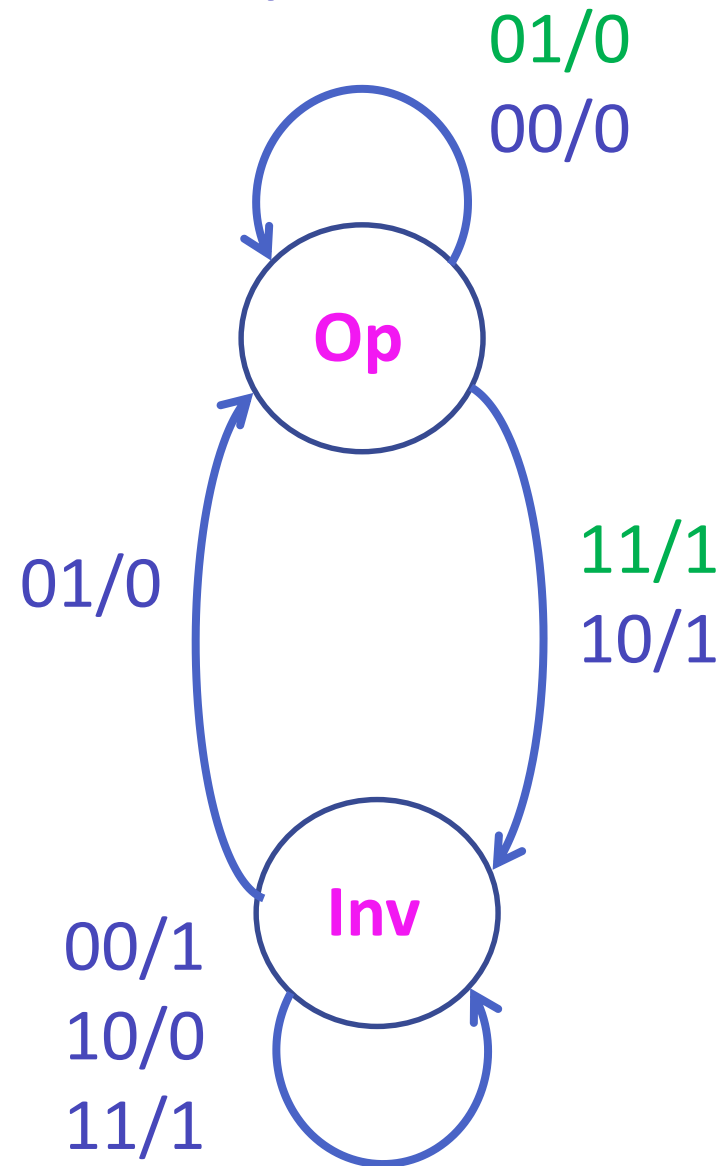
Postup řešení př. 1: doplnění všech možností

z=1 znamená, že přišlo další číslo a začínáme znovu, tzn. pro $a=0$ se vracíme do Op a pro $a=1$ zůstáváme v Inv



Postup řešení př. 1: všechny možnosti

$z=1$ znamená, že přišlo další číslo a doplníme zbylé možnosti (zeleně), tak aby z každého uzlu vedly všechny možné kombinace vstupů (výstupy už máme)



Řešení př. 1: výsledný graf

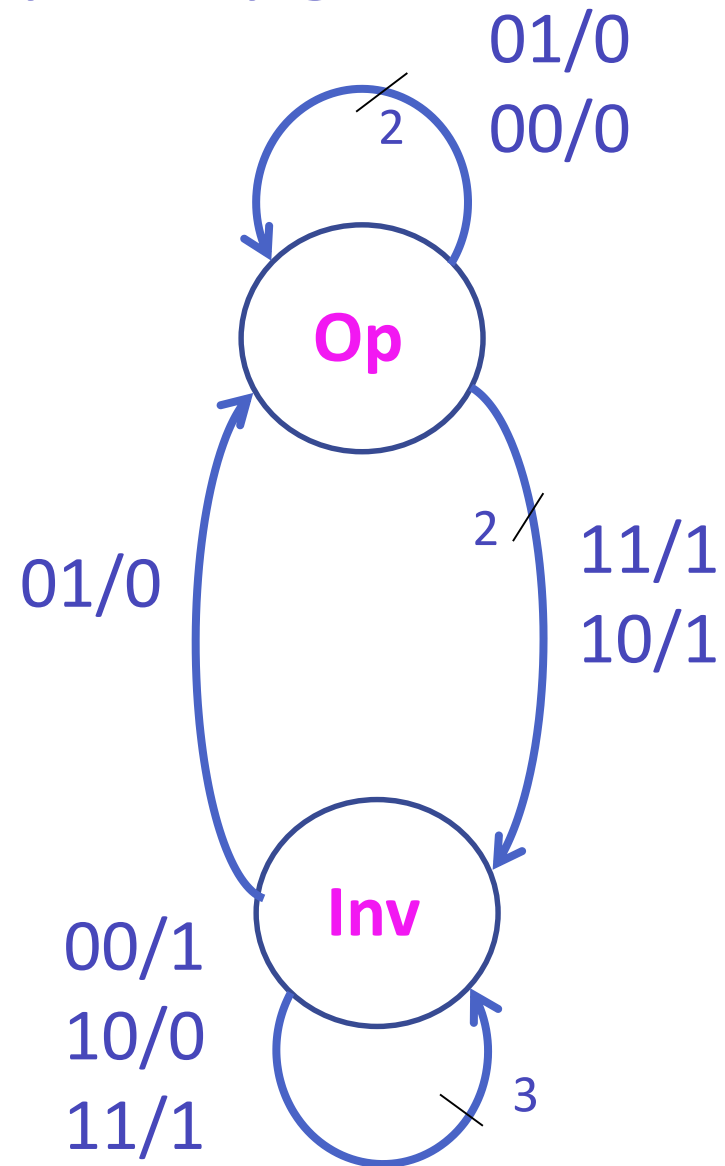
2 vnitřní stavy:

Q_0 : opiš (**Op**)

Q_1 : invertuj (**Inv**)

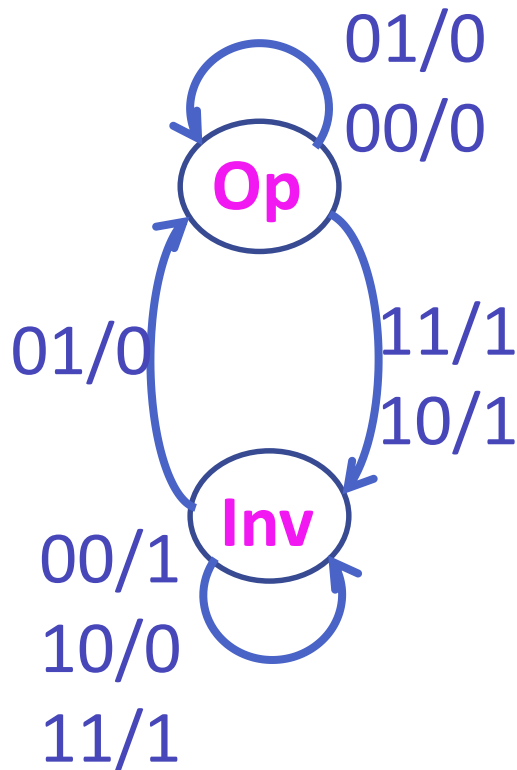
vstupy/výstup: az/b

Mealy nebo Moore ???



Tabulka přechodů a výstupů

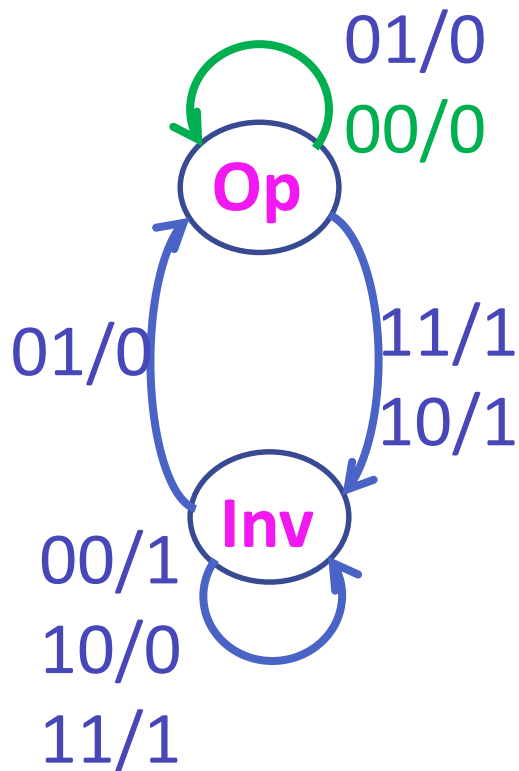
postupně přepisujeme graf do tabulky:



	az			
Q_{next}	00	01	11	10
Op				
Inv				

b	00	01	11	10
Op				
Inv				

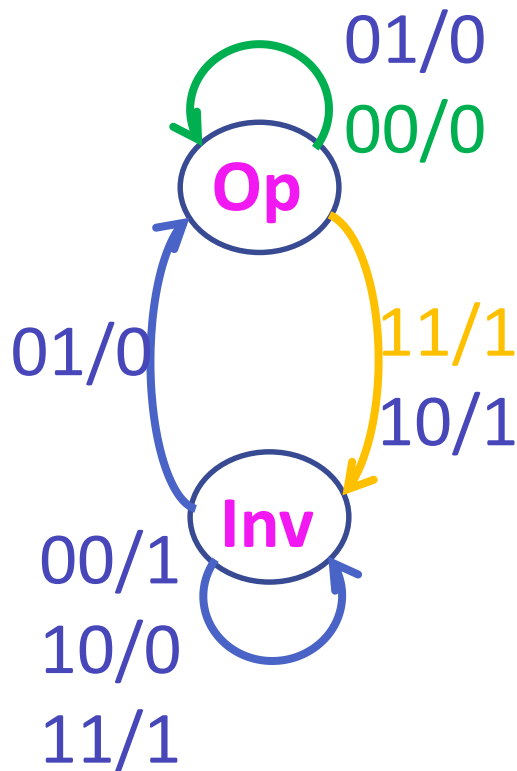
Tabulka přechodů a výstupů



	az			
Q_{next}	00	01	11	10
Op	Op			
Inv				

b	00	01	11	10
Op	0			
Inv				

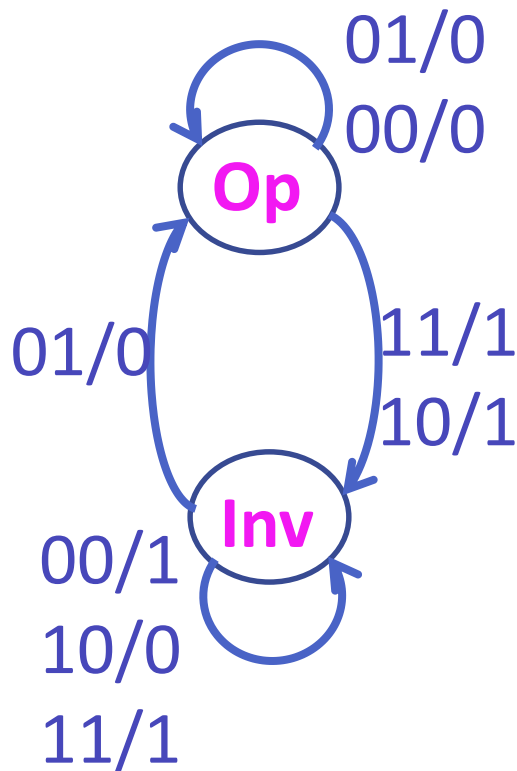
Tabulka přechodů a výstupů



	az			
Q_{next}	00	01	11	10
Op	Op		Inv	
Inv				

b	00	01	11	10
Op	0		1	
Inv				

Tabulka přechodů a výstupů



	az			
Q_{next}	00	01	11	10
Op	Op	Op	Inv	Inv
Inv	Inv	Op	Inv	Inv

b	00	01	11	10
Op	0	0	1	1
Inv	1	0	1	0

Kódování vnitřních stavů

(vstupy a výstup už zakódované jsou)

Q _{next}	00	01	11	10
Op	Op	Op	Inv	Inv
Inv	Inv	Op	Inv	Inv

b	00	01	11	10
Op	0	0	1	1
Inv	1	0	1	0

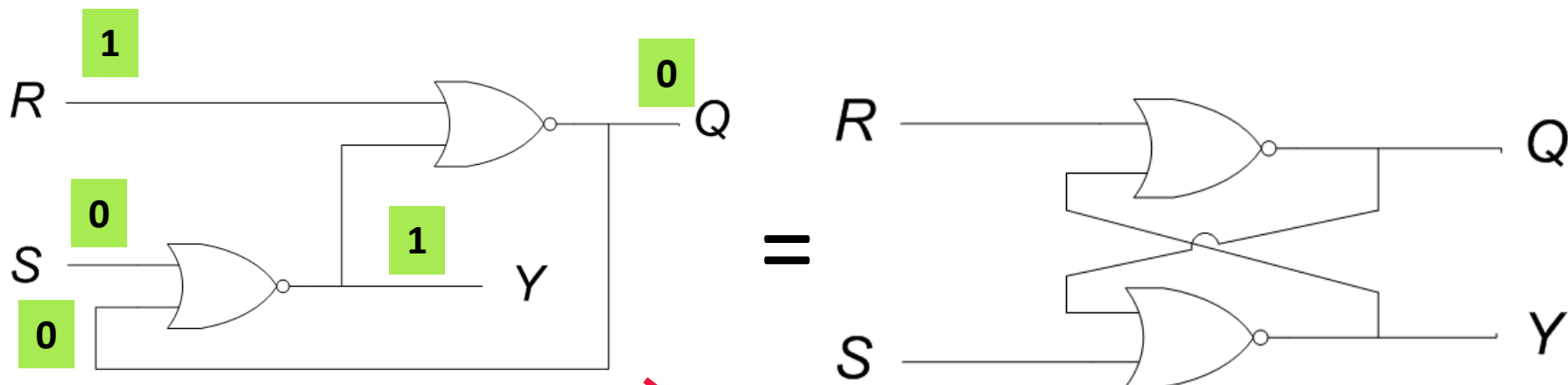
2 stavy ... pro rozlišení stačí 1 bit,
zvolme např. Op ... 0, Inv ... 1

Q _{next}	00	01	11	10
0	0	0	1	1
1	1	0	1	1

Následuje realizace pomocí hradel
a klopných obvodů ale co je to
klopný obvod?

Klopný obvod ... paměť

Klasické zapojení asynchronního R-S klopného obvodu:

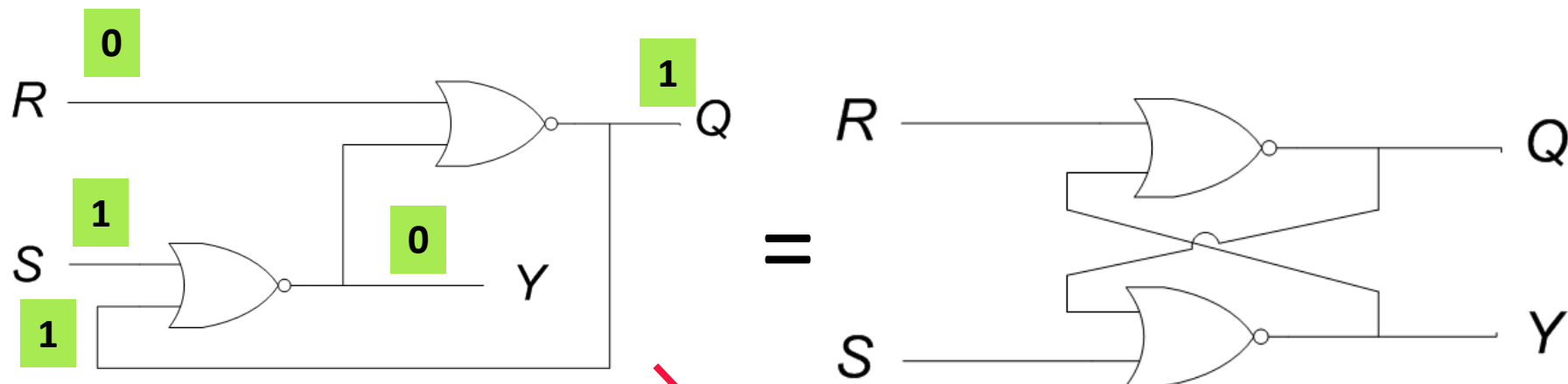


A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

RS	Q	Y
1 1	0	0
1 0	0	1
0 1	1	0
0 0	?	?

Klopné obvody - asynchronní

Klasické zapojení asynchronního R-S klopného obvodu:



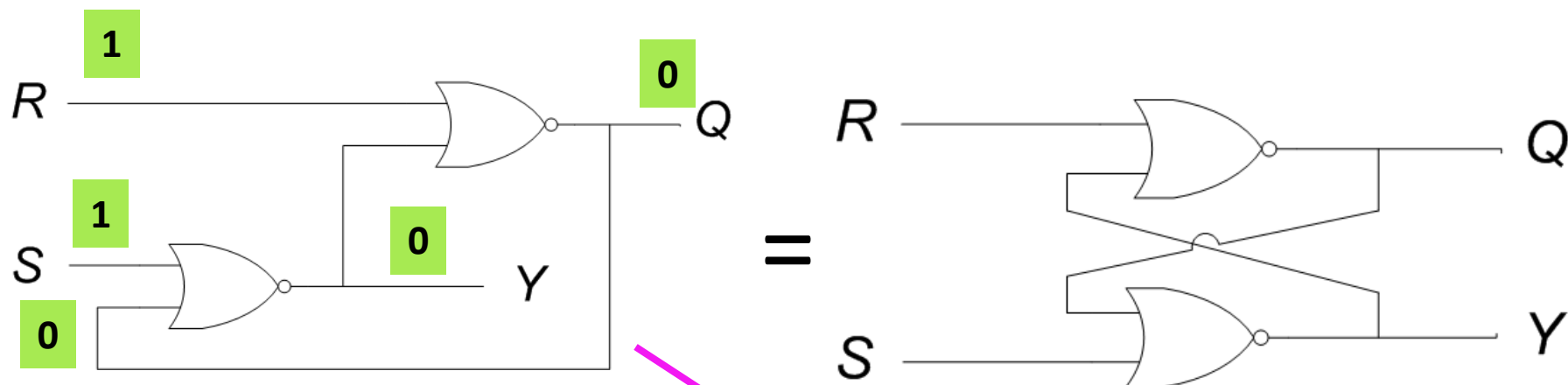
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

RS	Q	Y
1 1	0	0
1 0	0	1
0 1	1	0
0 0	?	?

S ... set
R ... Reset

Klopné obvody - asynchronní

Klasické zapojení asynchronního R-S klopného obvodu:



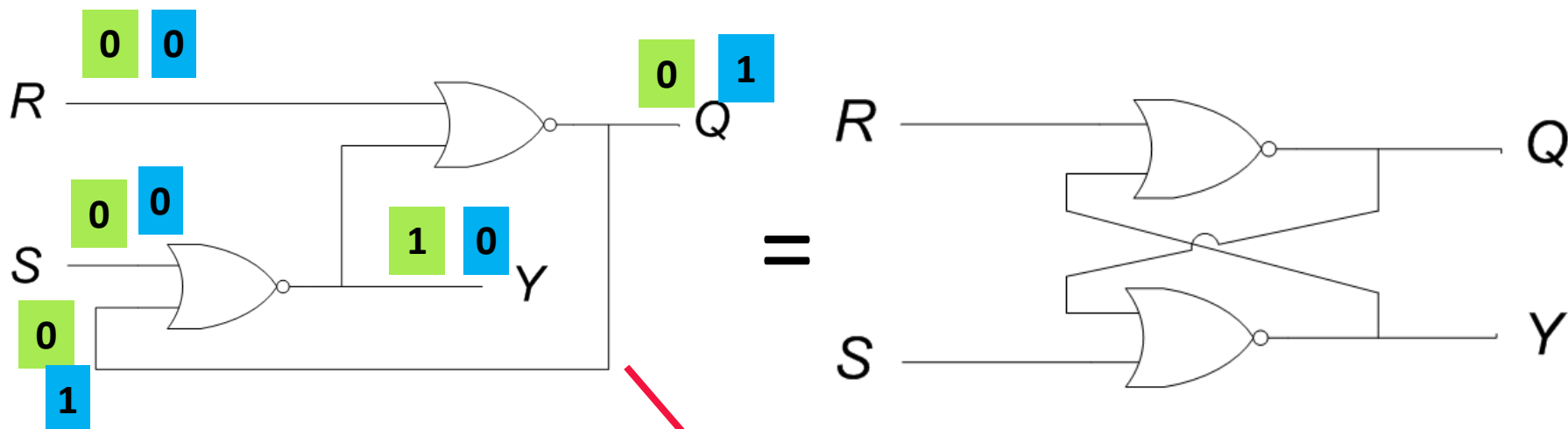
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

RS	Q	Y
1 1	0	0
1 0	0	1
0 1	1	0
0 0	?	?

!!!

Klopné obvody - asynchronní

Klasické zapojení asynchronního R-S klopného obvodu:


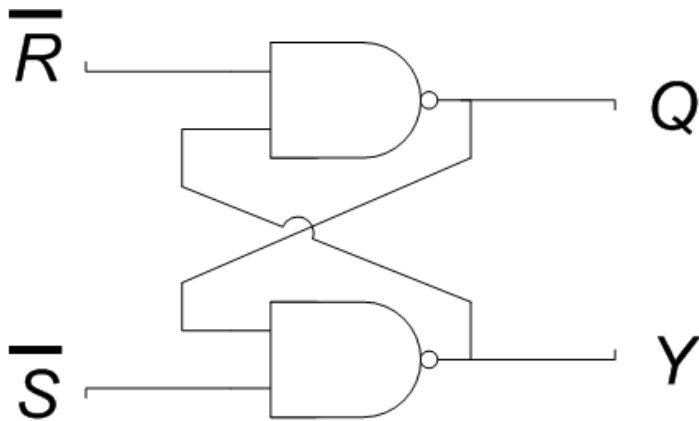


A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0


RS	Q	Y	
1 1	0	0	!!!
1 0	0	1	
0 1	1	0	
0 0	?	?	závisí na předchozím stavu, „pamatuje si ho“

Klopné obvody - asynchronní

„Podobné“ chování má obvod složený z hradel NAND:



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

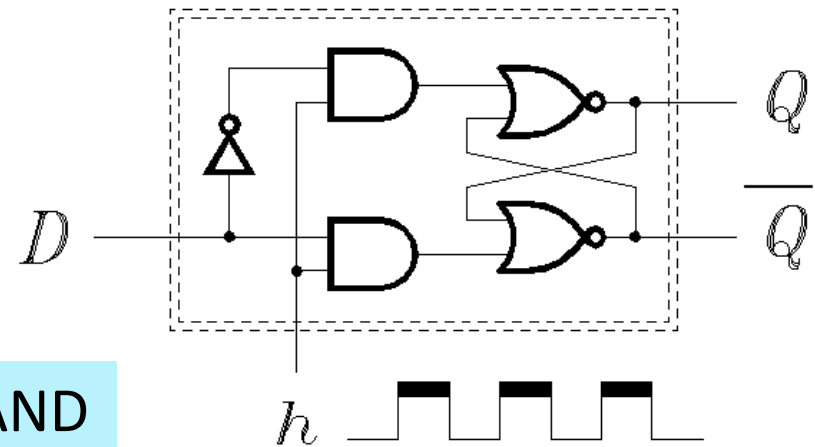
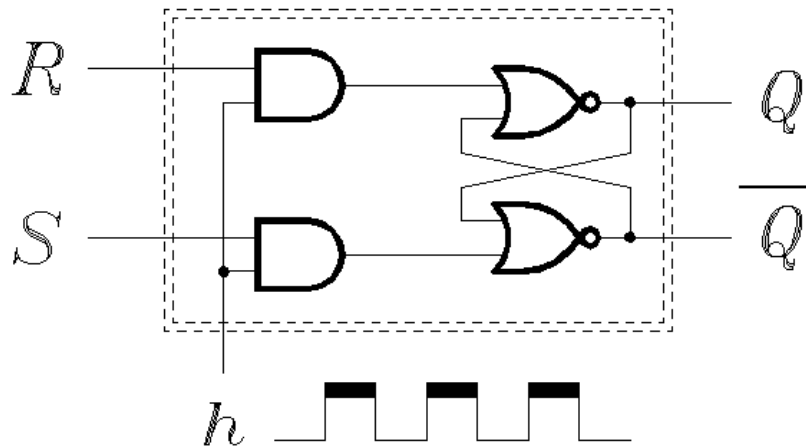


Jde také o asynchronní R-S klopný obvod, ale s *inverzními vstupy*, tzn.:

- pamatuje při vstupech 11 a
- při kombinaci 00 jsou na obou výstupech 1,
- 0 na S („set“) nastavuje Q do 1.

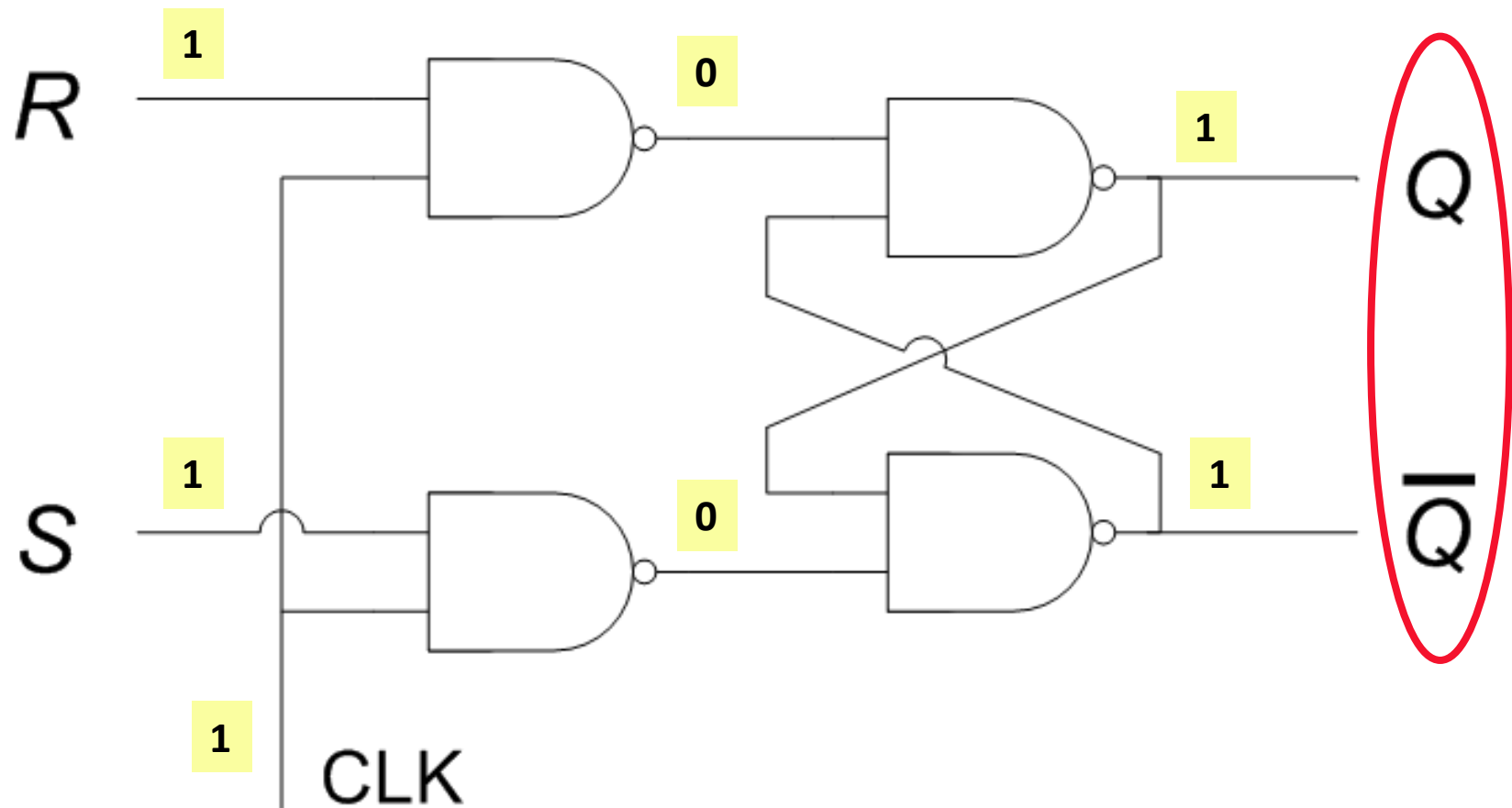
Klopné obvody – úrovnňové (latch)

změna stavu (překlopení) nastane jen při $h=1$, při $h=0$ se výstup nemění

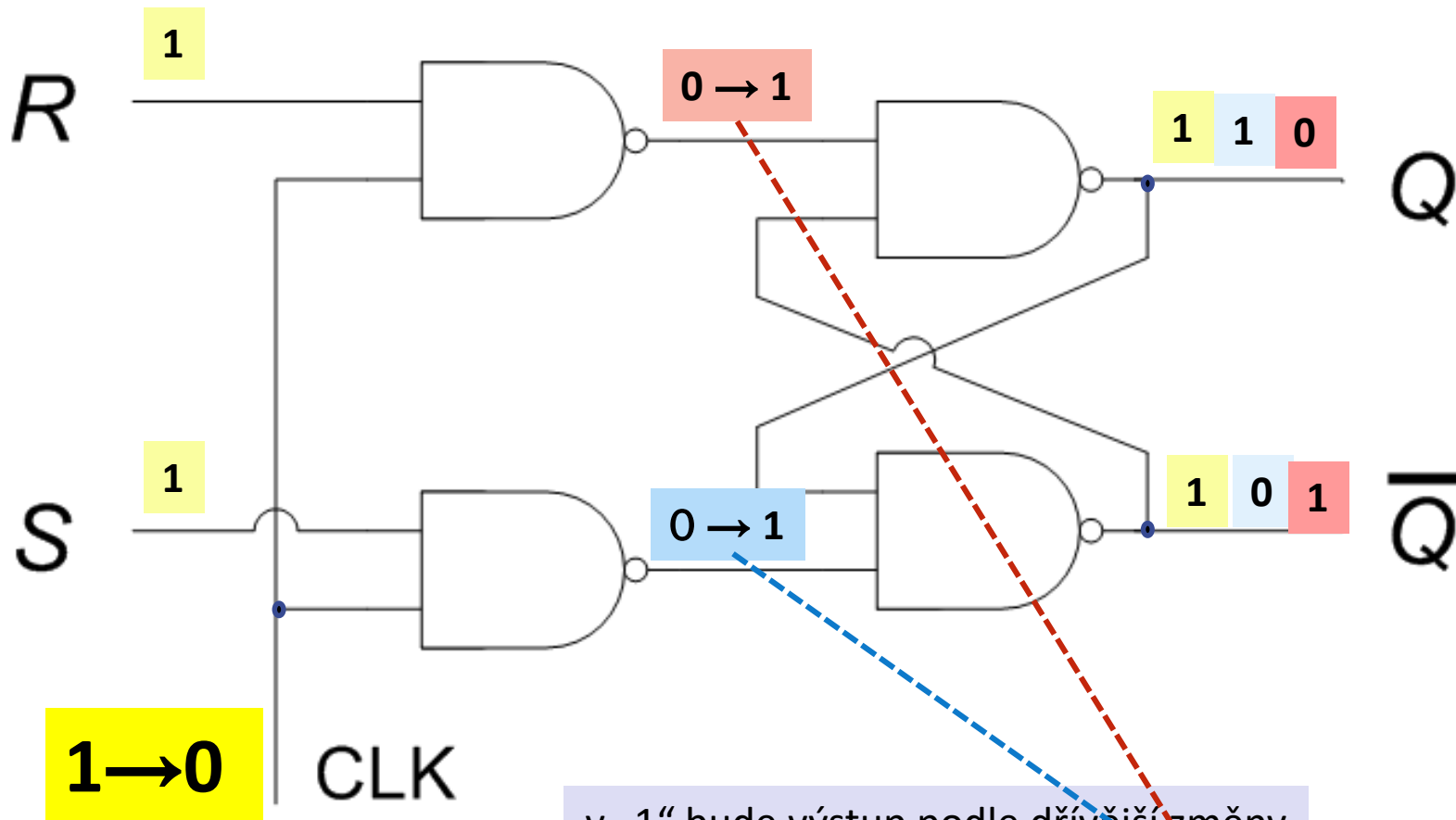


Pozn. „výhodnější“ je použít 4x NAND

„Zakázaný stav“



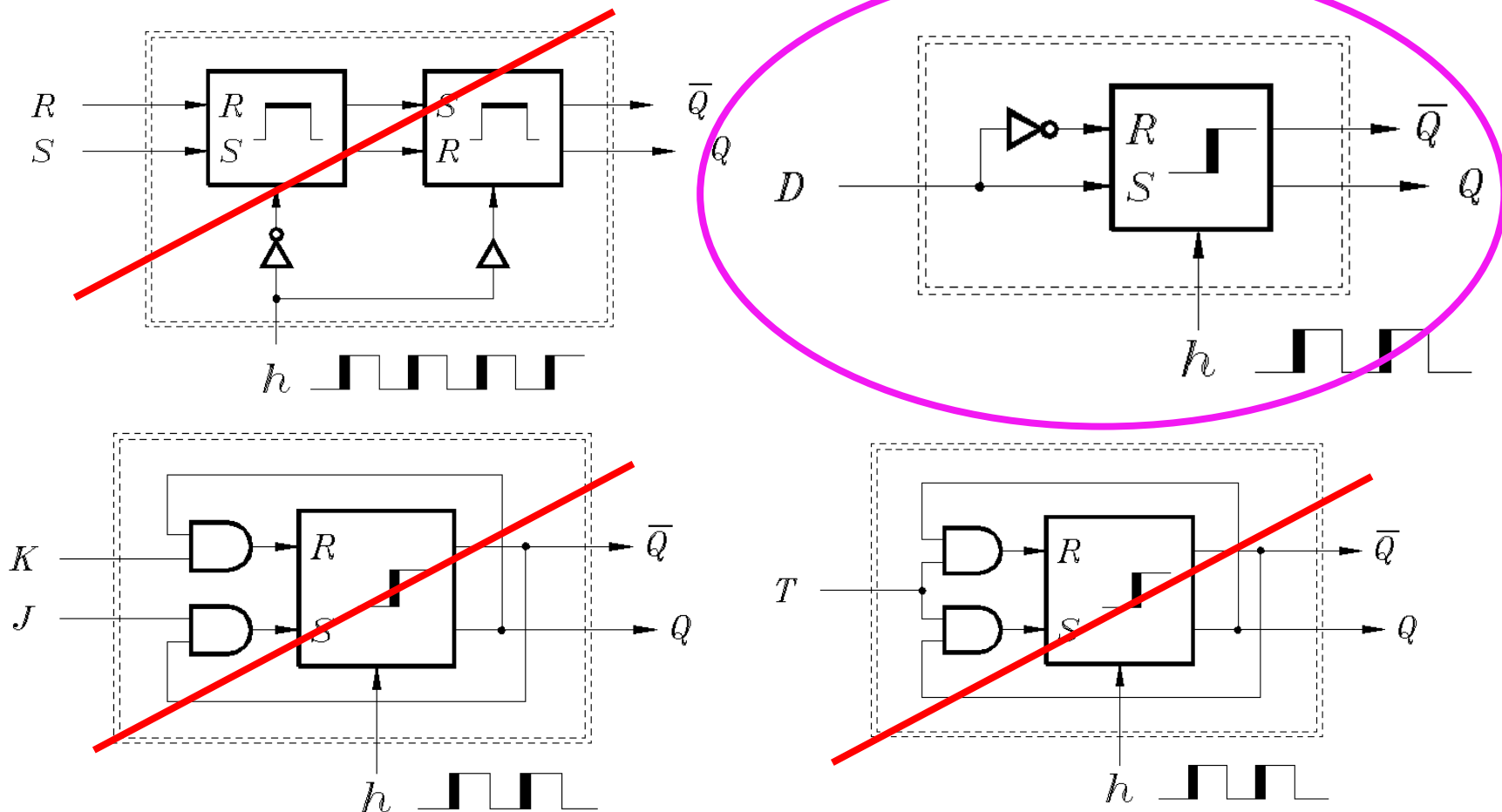
„Zakázaný stav“.....



v „1“ bude výstup podle dřívější změny vstupu posledního hradla do „0“

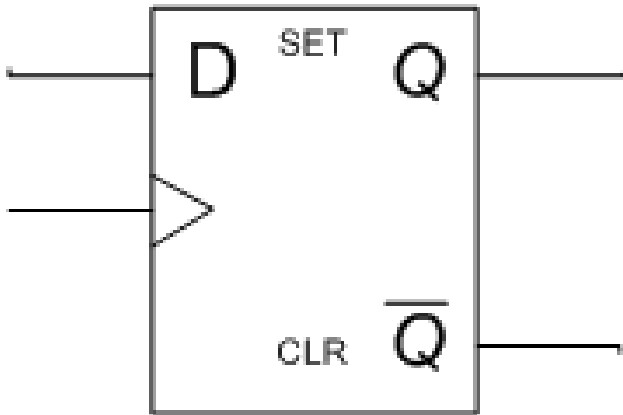
Klopné obvody hranové

Master – Slave, flip-flop (FF)



D klopný obvod

- budeme používat jen typ D



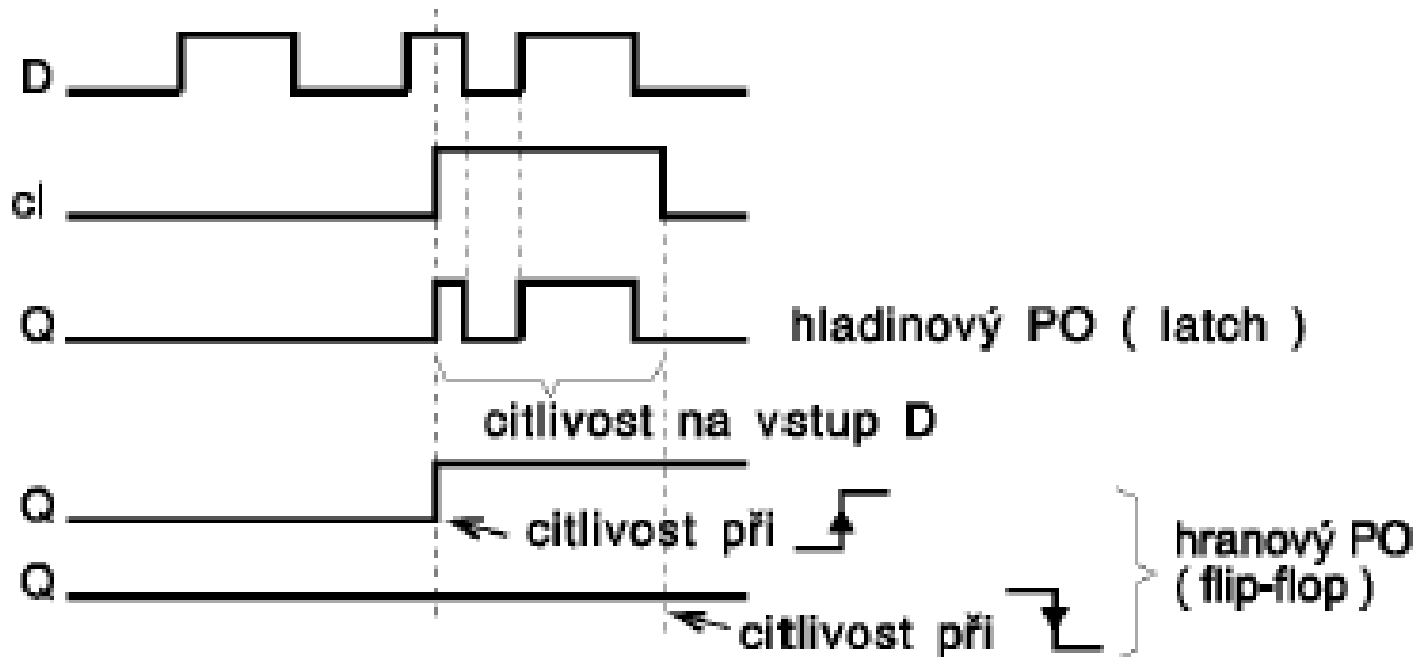
D	Q(next)
0	0
1	1

$$Q(next) = D$$

Q	Q(next)	D
0	0	0
0	1	1
1	0	0
1	1	1

D-KO

Rozdíl v chování hladinového a hranového D-KO



Poznámka: Klopný, též někdy paměťový obvod, angl. často jen ***latch*** pro úrovnňový nebo ***FF*** pro hranový klopný obvod

Příklad 2

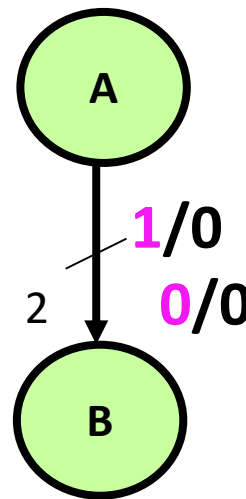
- Navrhněte SSO s jedním vstupem x a jedním výstupem y , který bude detekovat, zda jsou v sériově přicházejících tříbitových vstupních posloupnostech binární čísla 4 nebo 5. Počáteční podmínky – na vstupu je nejnižší řád prvního tříbitového čísla.
- Poznámka1: automat je iniciální

Graf přechodů, začátek

4: 100 nebo 5: 101

Tentokrát nevíme, kolik vyjde stavů a nemáme indikaci začátku, tzn. že musíme počítat do 3 v návrhu (tzn. nesmíme zůstat v žádném stavu na žádný vstup).

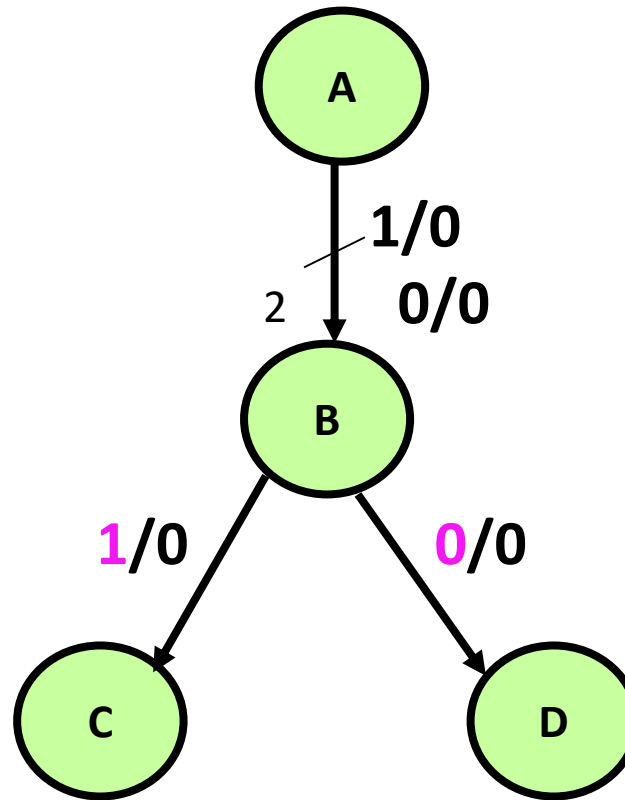
první bit správné posloupnosti je 0 nebo 1, tzn. na obě možnosti jdeme dál, přicházející posloupnost může být ta, kterou hledáme:



Graf přechodů, pokračování

4: 100 nebo 5: 101

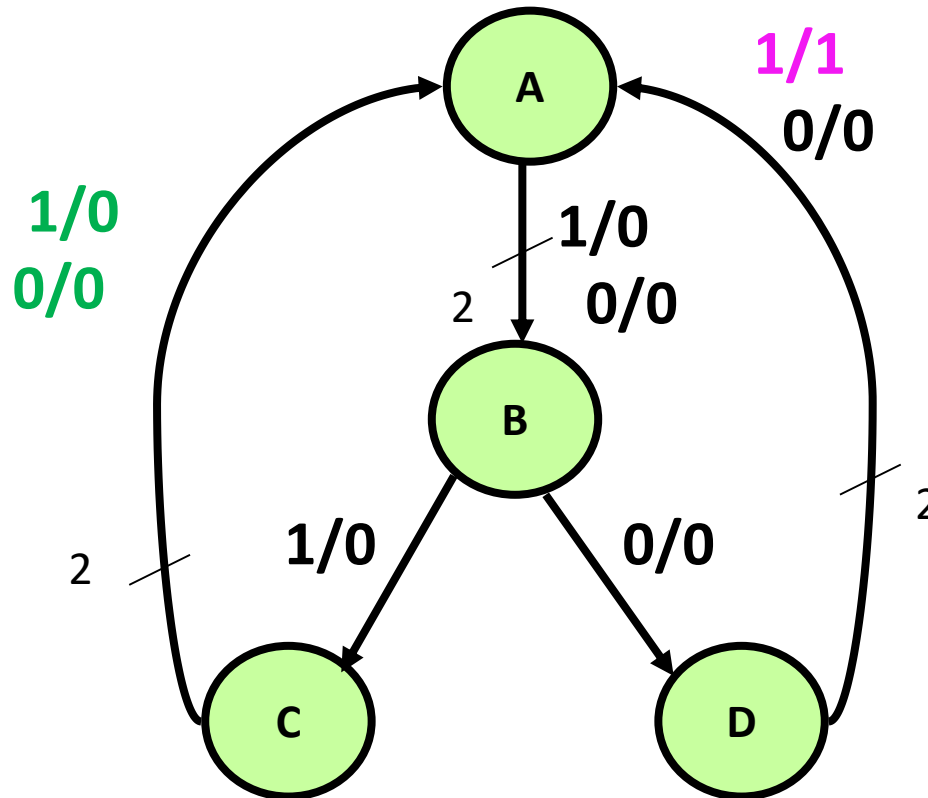
správně je jen 0, ale musíme dopočítat do 3, tzn. přidat i stav pro špatnou cestu (C)



Graf přechodů

4: **1**00 nebo 5: **1**01

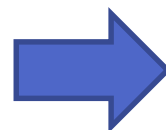
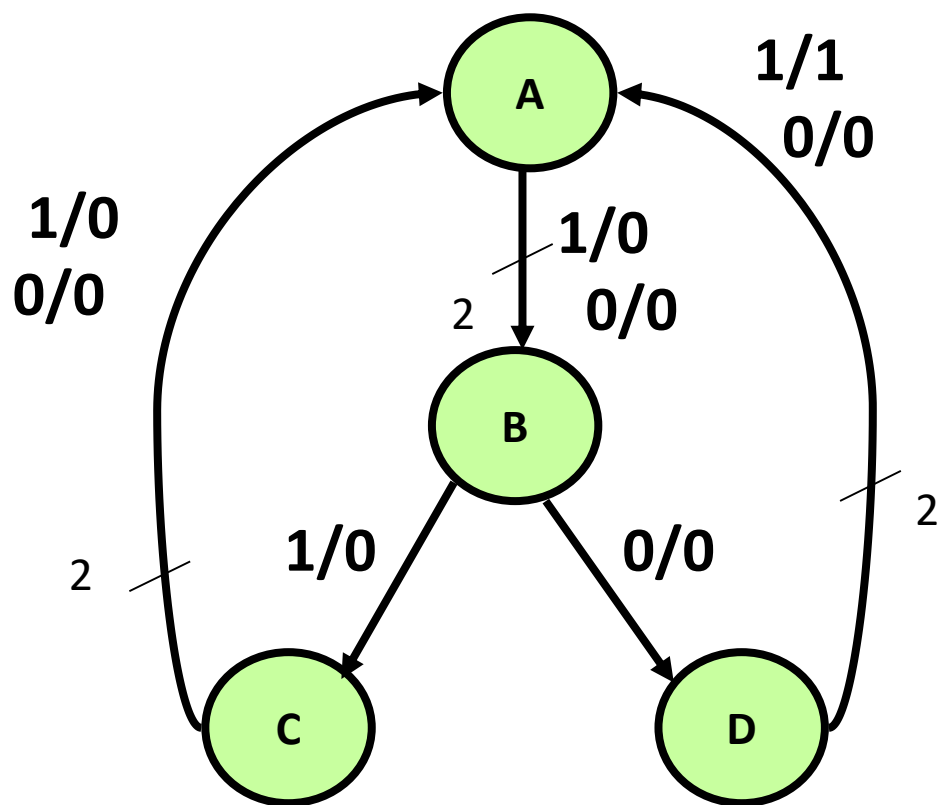
správně je jen 1 (výstup bude 1), ale musíme dopočítat do 3 znaků a pokračovat v detekci další trojice



Další postup

2. Z grafu tabulky pro přechodovou a výstupní funkci
3. Kódování a zakódované tabulky
4. Mapy,
5. Minimalizace
6. Budící funkce pro vstupy klopných obvodů a pro výstupy
7. Realizace
8. Časování – výpočet maximální hodinové frekvence

Tabulky přechodové a výstupní funkce



	0	1	0	1
A	B	B	0	0
B	D	C	0	0
C	A	A	0	0
D	A	A	0	1

2. Tabulka přechodů a výstupů

	0	1	0	1
A	B	B	0	0
B	D	C	0	0
C	A	A	0	0
D	A	A	0	1



	b	a
A	0	0
B	0	1
C	1	1
D	1	0



Zakódovaná tabulka
přechodů a výstupů:

	0	1	0	1
00	01	01	0	0
01	10	11	0	0
11	00	00	0	0
10	00	00	0	1

4.

\overline{a} \overline{b}

	\overline{a}	\overline{b}	$a\overline{b}$	$\overline{a}b$
$\overline{a}\overline{b}$	1	0	0	0
$a\overline{b}$	1	1	0	0

X

\overline{a} \overline{b}

	\overline{a}	\overline{b}	$a\overline{b}$	$\overline{a}b$
$\overline{a}\overline{b}$	0	1	0	0
$a\overline{b}$	0	1	0	0

X

5.

$$D_a = \overline{a}\overline{b} + \overline{b}X$$

$$D_b = a\overline{b}$$

\overline{a} \overline{b}

	\overline{a}	\overline{b}	$a\overline{b}$	$\overline{a}b$
$\overline{a}\overline{b}$	0	0	0	0
$a\overline{b}$	0	0	0	1

X

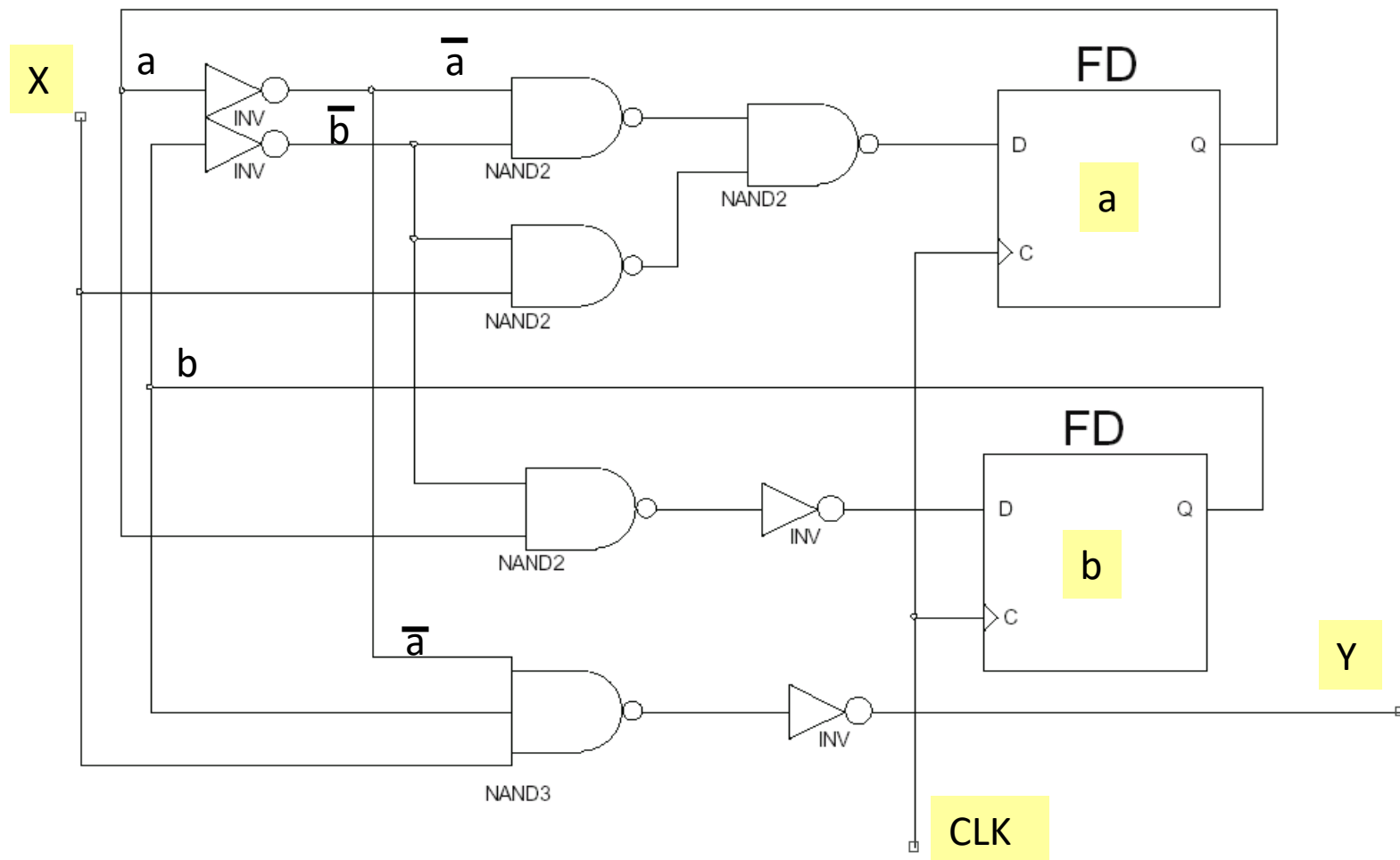
$$Y = \overline{a}bX$$

Schema podle funkcí:

$$D_a = \bar{a}\bar{b} + \bar{b}X$$

$$D_b = a\bar{b}$$

$$Y = \bar{a}bX$$



Časování

Výpočet maximální hodinové frekvence

- Záleží na:
 - Technologii
 - Typu hradel
 - Počtu vstupů
 - Větvení
 - Klopných obvodech (v podstatě nyní jen D-KO)
 - Délce spojů (vodičů)
- Návrhové systémy

Podklady pro výpočet

Na dalších snímcích:









Tabulka 1: knihovna základních hradel

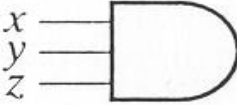
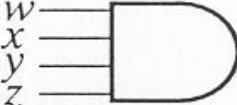
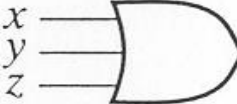
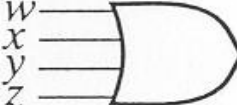
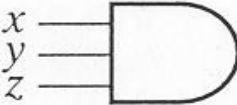
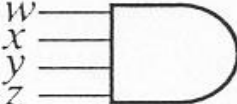


Tabulka 2: standardní logická hradla s více vstupy

Popis tabulek (technologie CMOS):

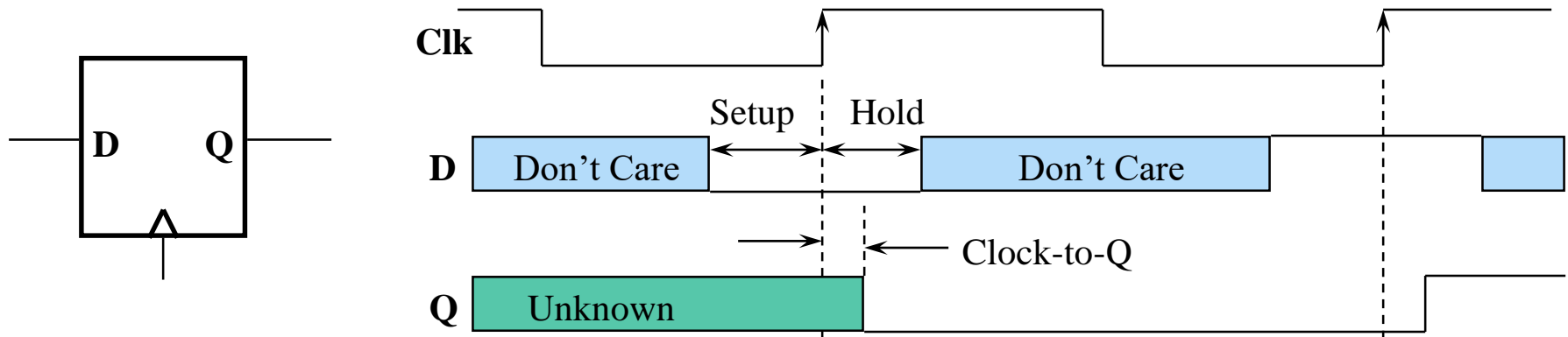
Název hradla	grafický symbol	funkce	Cena (počet transistorů)	Zpoždění normalizované (ns)
-----------------	--------------------	--------	--------------------------------	-----------------------------------

*Pozn. hodnoty zpoždění závisí na technologii, zde je příklad
konkrétních hodnot užitých pro výpočet*

Inverter		$F = x'$	2	1
Driver		$F = x$	4	2
AND		$F = xy$	6	2.4
OR		$F = x + y$	6	2.4
NAND		$F = (xy)'$	4	1.4
NOR		$F = (x + y)'$	4	1.4
XOR		$F = x \oplus y$	14	4.2
XNOR		$F = x \odot y$	12	3.2

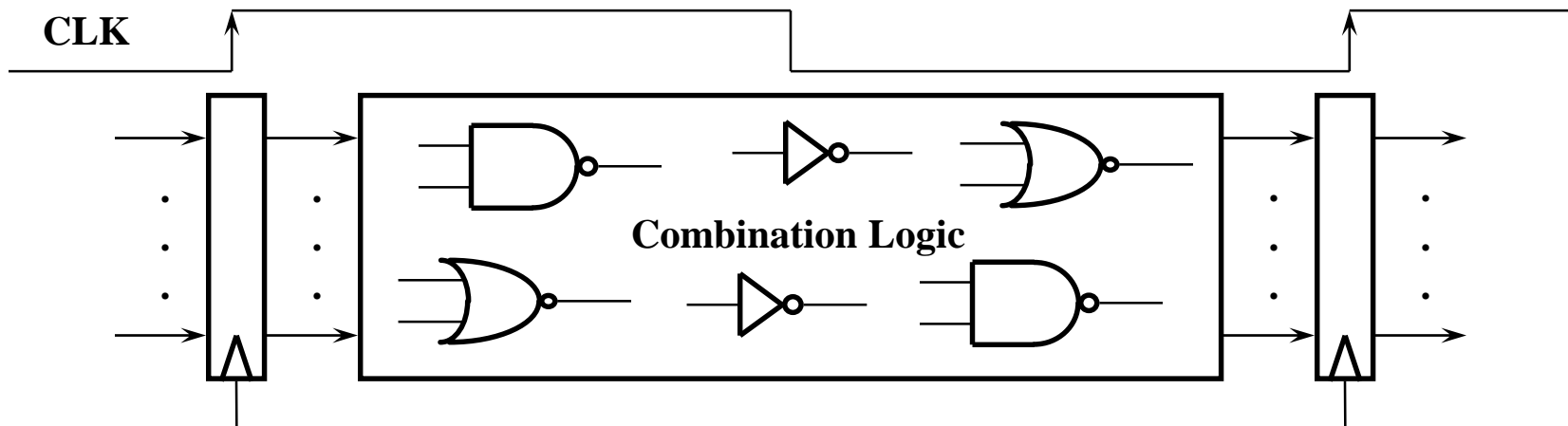
3-input AND		$F = xyz$	8	2.8
4-input AND		$F = xyzw$	10	3.2
3-input OR		$F = x + y + z$	8	2.8
4-input OR		$F = x + y + z + w$	10	3.2
3-input NAND		$F = (xyz)'$	8	1.8
4-input NAND		$F = (xyzw)'$	10	2.2
3-input NOR		$F = (x + y + z)'$	8	1.8
4-input NOR		$F = (x + y + z + w)'$	10	2.2

Časování klopného obvodu



- **Předstih** (Setup Time): vstup musí být stabilní (ustálený) **PŘED** aktivní hodinovou hranou
- **Přesah** (Hold Time): vstup musí zůstat stabilní (ustálený) **PO** aktivní hodinové hraně
- **Zpoždění** klopného obvodu (Clock-to-Q Time): doba mezi přechodem aktuálních dat z D na Q odvozená od aktivní hrany hodin

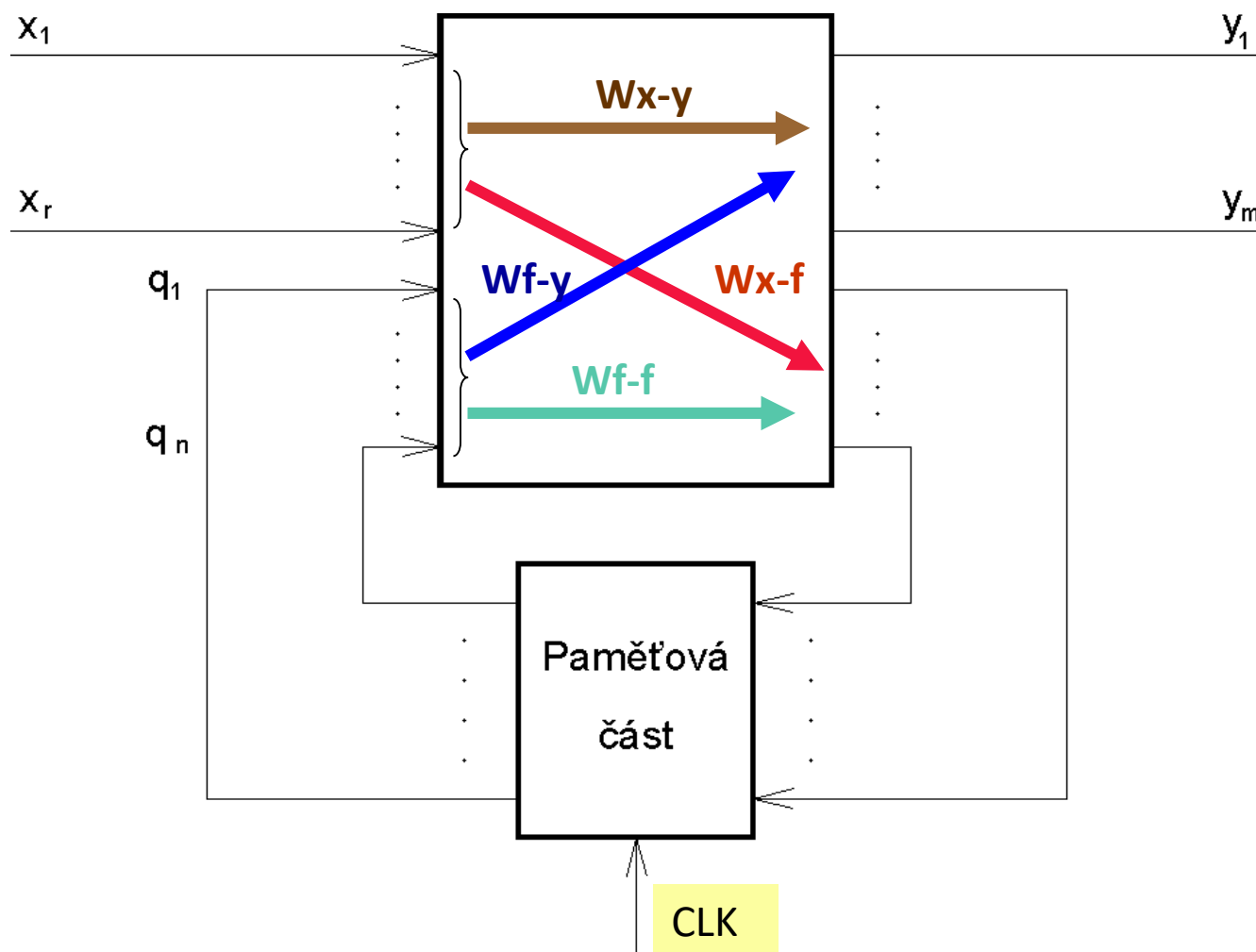
Maximální hodinová frekvence



- Všechny klopné obvody jsou řízeny stejnou hodinovou frekvencí
- Kombinační logické bloky:
 - Vstupy jsou aktualizovány při každém taktu
 - Všechny výstupy kombinační části musí být stabilní (nastaveny na správnou hodnotu) před dalším taktem

Kritická cesta

Nejdelší možná cesta mezi každým vstupem a výstupem kombinační části:

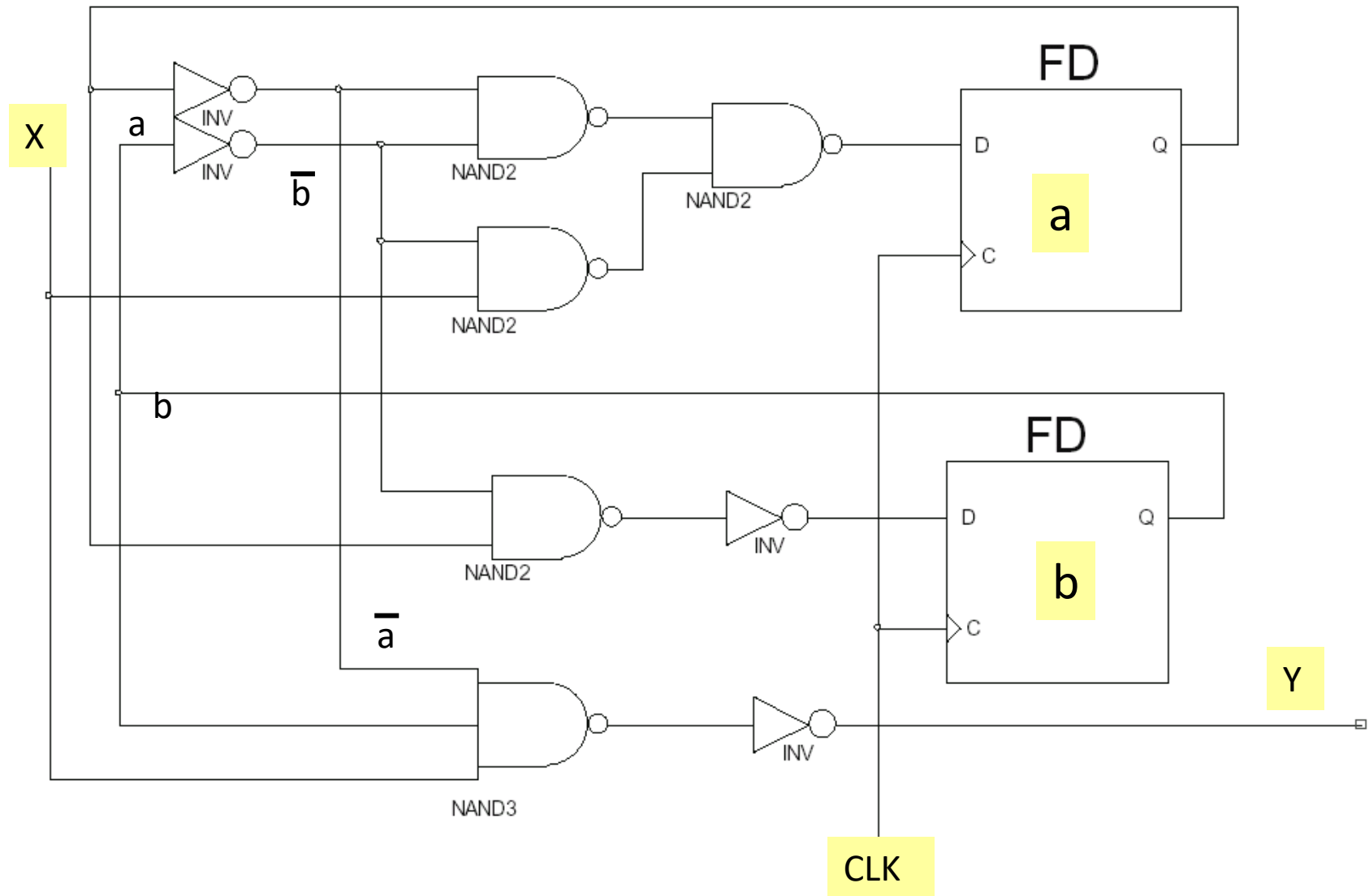


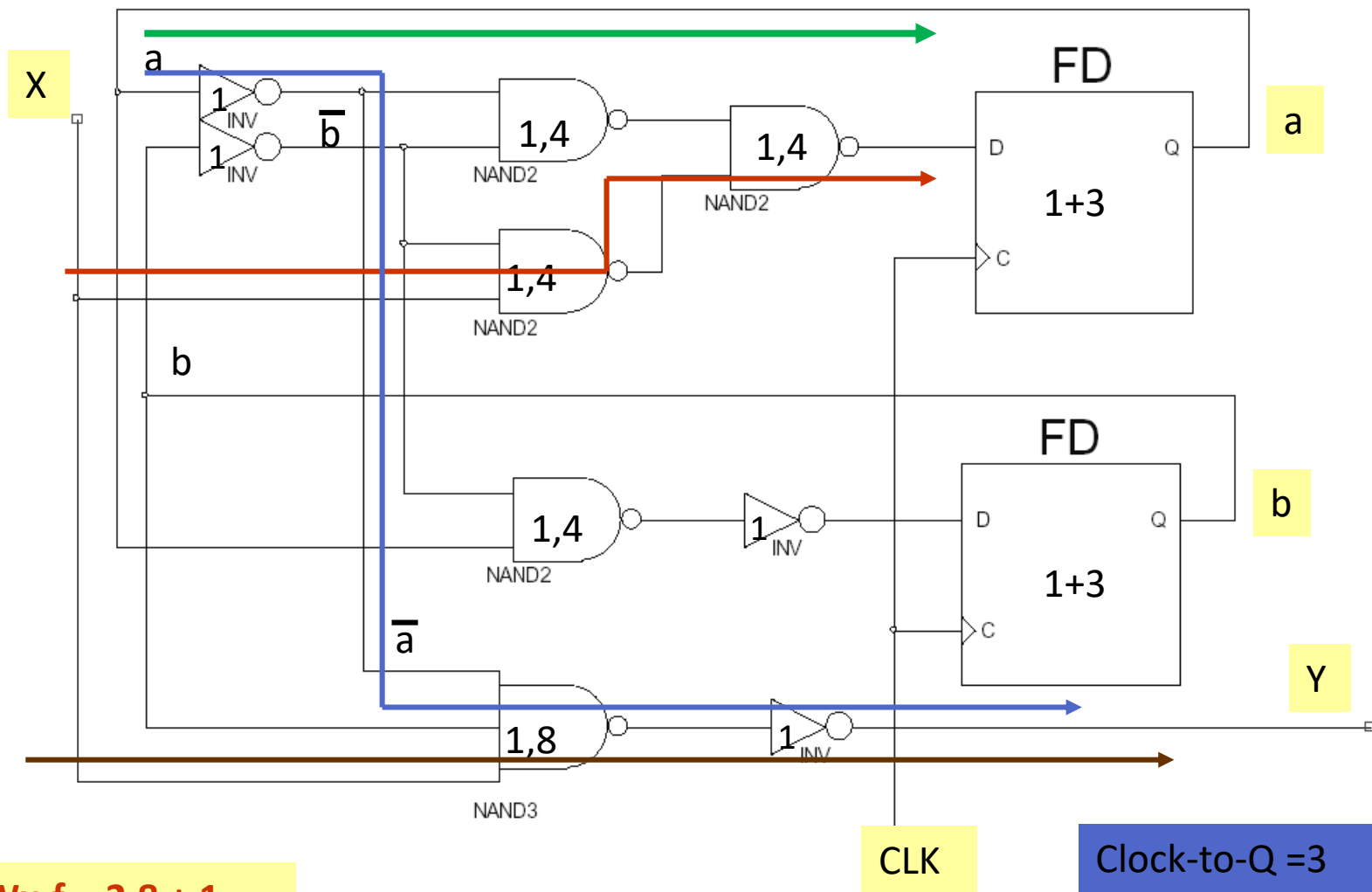
Výpočet

Hodinová frekvence = $1/\max W_{i-j}$, kde ex. 4 cesty:

- **W_{x-f}** : ze vstupu X na vstup KO, zpoždění na hradlech + nestabilita vstupů + předstih
- **W_{f-y}** : výstup KO výstup Y, zpoždění na hradlech + zpoždění KO + požadavek na stabilitu výstupu
- **W_{x-y}** : ze vstupu X na výstup Y, zpoždění na hradlech + nestabilita vstupů + požadavek na stabilitu výstupu
- **W_{f-f}** : mezi dvěma KO, zpoždění na hradlech + předstih + zpoždění KO (Clock-to-Q)

Schema





$$W_{x-f} = 2,8 + 1$$

$$W_{f-y} = 3 + 3,8$$

$$W_{x-y} = 2,8$$

$$W_{f-f} = 1 + 3 + 3,8$$

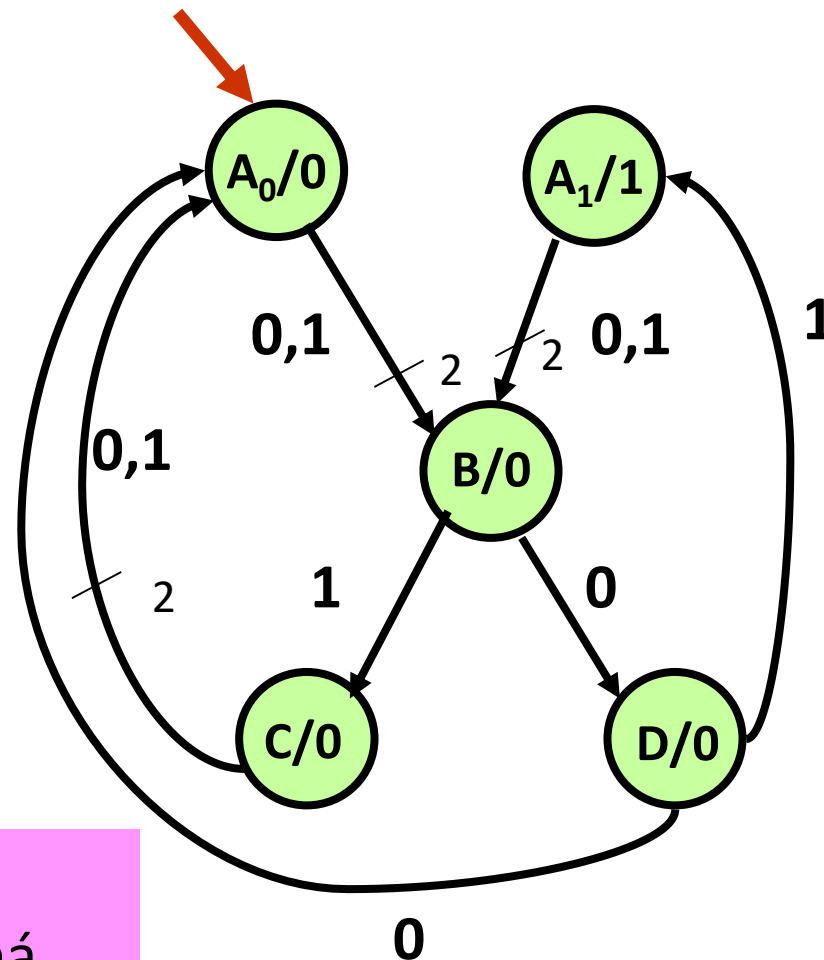
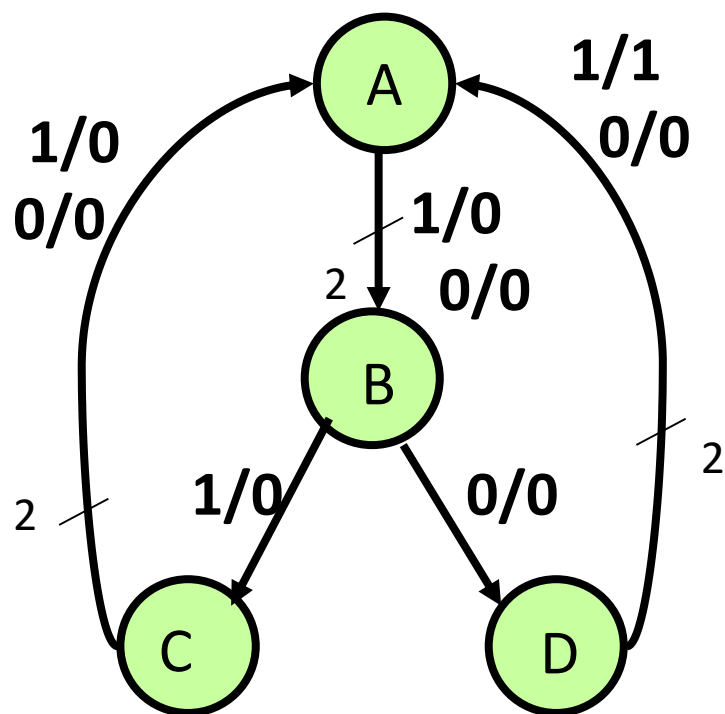
$$f_{\max} = 1/7,8\text{ns} = 128 \text{ MHz}$$

Clock-to-Q = 3
Setup = 1

Převody Mealy → Moore

- Uzly grafu, do něhož vstupují hrany ohodnocené **stejným výstupním symbolem** ponecháme
- Každý uzel, který nemá uvedenou vlastnost nahradíme toliko uzly, kolika výstupními symboly jsou ohodnoceny hrany **do něho vstupující**
- Připojíme vstupní a výstupní hrany, uzly ohodnotíme příslušnými výstupními symboly

Příklad: Mealy → Moore



Poznámka: indikace správné posloupnosti bude opožděná

Moore → Mealy

- Nejlépe z tabulky přechodů a výstupů – jde jen o přiřazení výstupu podle následného stavu (výstup má reagovat na vstup dříve):

Moore

A:

X/Q	X1 X2	0
Q1	Q3 Q1	Y3
Q2	Q1 Q2	Y1
Q3	Q2 Q3	Y2



Mealy

A':

Q\X	X1 X2	X1	X2
Q1	Q3 Q1	Y2	Y3
Q2	Q1 Q2	Y3	Y1
Q3	Q2 Q3	Y1	Y2

Závěr

- Dotazy do chatu Teams – Prosemináře a konzultace (příprava na on-line konzultace v pondělí 8. 3. od 16:15 a od 18:00)
- Příklady v nahraném prosemináři a ve skriptu (courses, sekce Skripta)
- Možné otázky/příklady do testů najdete na courses v sekci Přednášky u jednotlivých přednášek