

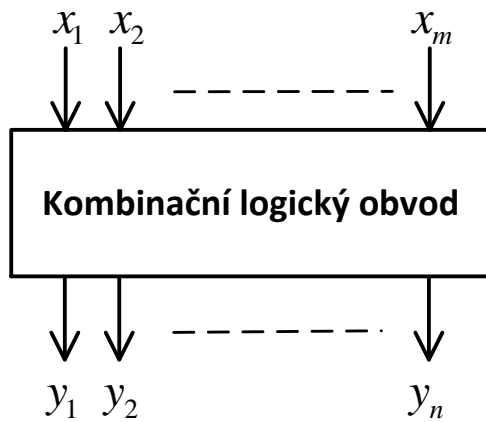
Kombinační a sekvenční logické obvody

Kombinační log. obvody , logické hazardy, sekvenční obvody,
klopné obvody RS, JK, D, T

Ing. Pavel Lafata, Ph.D.
lafatpav@fel.cvut.cz

Kombinační vs. sekvenční logické obvody – definice, příklady kombinačních obvodů

▪ Kombinační logický obvod (KLO)



$$y_1 = f_1(x_1, x_2, \dots, x_m)$$

$$y_2 = f_2(x_1, x_2, \dots, x_m)$$

$$\vdots$$

$$y_n = f_n(x_1, x_2, \dots, x_m)$$

$$\mathbf{Y} = \mathbf{F}(\mathbf{X})$$

x_1, x_2, \dots, x_m – vstupní hodnoty

množina vstupních stavů – \mathbf{X}

y_1, y_2, \dots, y_n – výstupní hodnoty

množina výstupních stavů – \mathbf{Y}

f_1, f_2, \dots, f_n – výstupní funkce

množina výstupních funkcí – \mathbf{F}

- obecně logický obvod složený z log. hradel a dalších elementárních prvků, jehož **výstupy y_1, y_2, \dots, y_n jsou jednoznačně určeny kombinací vstupních hodnot x_1, x_2, \dots, x_m**
- **obvod neobsahuje žádný paměťový člen a žádnou zpětnou vazbu!**
- **výstup obvodu y** – je v každém okamžiku jednoznačně určen pouze vstupní kombinací!
- **pro m vstupů a n výstupů** – můžeme sestavit pro popis obvodu 2^n Booleových rovnic
- **ideální kombinační logický obvod** – jakákoliv změna vstupní hodnoty vyvolá okamžitě změnu výstupní hodnoty obvodu – do další změny na vstupu je pak výstup konstantní
- **reálný (nedokonalý) kombinační logický obvod** – vlivem doby zpoždění jednotlivých hradel obvodu se změna vstupu KLO projeví na jeho výstupu se zpožděním
- **doba zpoždění** = časová prodleva mezi okamžikem změny na vstupu do nastavení správného výstupu obvodu (viz dále doba zpoždění hradel)
- nenulová doba zpoždění logických hradel – **vznik logických hazardů v obvodech**

Kombinační vs. sekvenční logické obvody – KLO, příklady kombinačních obvodů

▪ **Kombinační logické obvody**

- pro jejich popis a realizaci používáme zejména:
 1. **algebraické vyjádření** – popis funkce obvodu s využitím zákonů Booleovy algebry
 2. **pravdivostní tabulka** – tabulkové vyjmenování výstupů obvodu pro vstupní kombinace
 3. **schéma obvodu** – grafické vyjádření zapojení KLO pomocí elementárních logických hradel a obvodových prvků

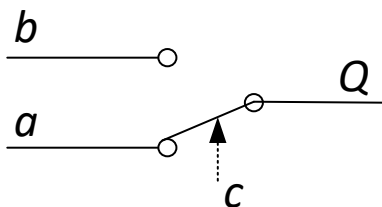
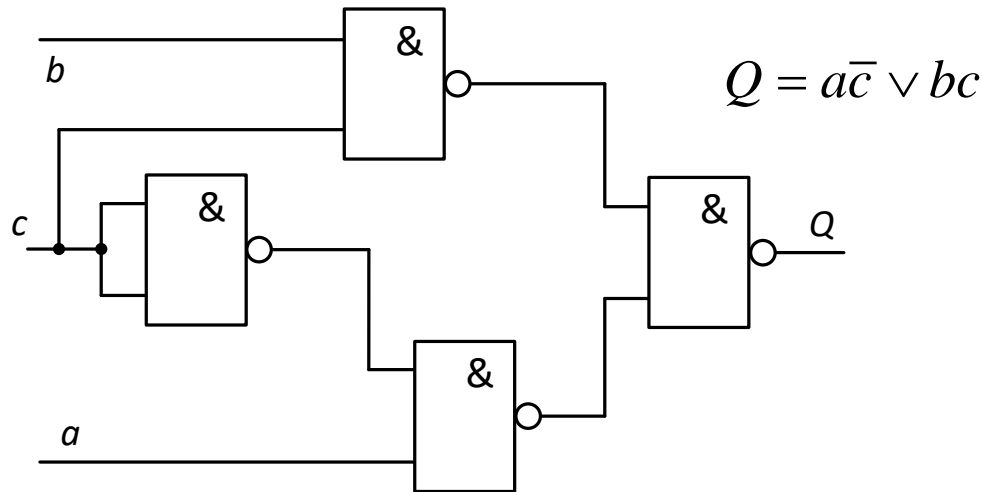
- **typické příklady KLO:**
 - jednoduché logické operace
 - komparátory, paritní kodéry a dekodéry, přepínače
 - multiplexory a demultiplexory
 - převodníky kódů
 - binární kodéry a dekodéry, prioritní kodéry, dekodéry segmentových displejů apod.
 - sčítačky, odčítačky, násobičky, děličky

Kombinační vs. sekvenční logické obvody – KLO, příklady kombinačních obvodů

▪ Multiplexor (mux)

- obecně, kombinační obvod, který přepíná jeden z n vstupů do jednoho společného výstupu na základě stavu řídicího vstupu
- zjednodušeně – programovatelný přepínač, který lze přepnout mezi několika různými vstupy pomocí řídicího mechanismu
- **příklad** – nejjednodušší 2vstupový multiplexor, 2 na 1
- 2 datové vstupy – a , b , 1 řídicí vstup – c , 1 výstup – Q , popis funkce:
pokud je $c = 0$ -> výstup je přepnut $Q = a$, pokud je $c = 1$ -> výstup je přepnut $Q = b$

c	b	a	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



- **demultiplexor (demux)** má přesně opačnou funkci – řízené přepínání jednoho datového vstupu do n výstupů
- **v jazyce VHDL**, multiplexor jednoduše vytvoříme pomocí **podmíněného přiřazování** – řada možností, viz další přednášky

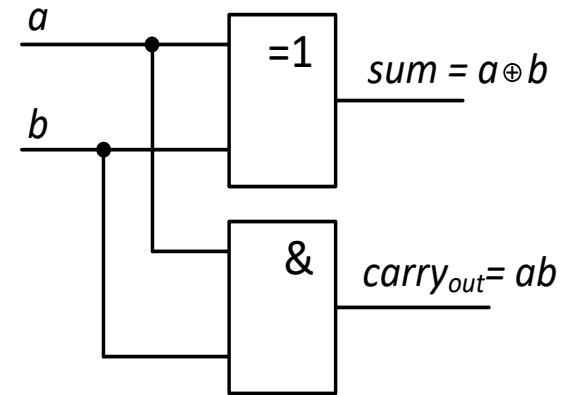
Kombinační vs. sekvenční logické obvody – KLO, příklady kombinačních obvodů

- **Sčítačka (binární sčítačka)** – provádí operaci sčítání dvou čísel
- **neúplná (poloviční) 1bitová sčítačka** -> a, b jsou 1bitové vstupy, sum je 1bitový výstup, $carry-out$ je 1bitový výstup přenosu: $b + a = carry_{out} + sum$

b	a	$carry-out$	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$sum = a\bar{b} \vee \bar{a}b = a \oplus b$$

$$carry_{out} = a \cdot b$$



- **úplná 1bitová sčítačka** -> a, b jsou 1bitové vstupy, $carry-in$ je 1bitový vstup přenosu, sum je 1bitový výstup, $carry-out$ je 1bitový výstup přenosu: $carry_{in} + b + a = carry_{out} + sum$

$carry-in$	b	a	$carry-out$	sum
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

- **carry-in** je vstupní přenos z nižšího řádu, **carry-out** je výstupní přenos do vyššího řádu – indikuje přetečení výsledku do vyššího řádového místa
- sčítačka **bez vstupu přenosu** $carry-in$ -> **poloviční sčítačka** **se vstupem přenosu** $carry-in$ -> **úplná sčítačka**

- **n-bitová sčítačka** – kaskádní zřetězení 1bitových sčítaček

Kombinační vs. sekvenční logické obvody – KLO, příklady kombinačních obvodů

▪ Sčítačka

- realizace úplné sčítačky

	a		b
sum			
$carry_{in}$	0	1	0
	1	0	1

	a		b
$carry_{out}$			
$carry_{in}$	0	0	1
	0	1	1

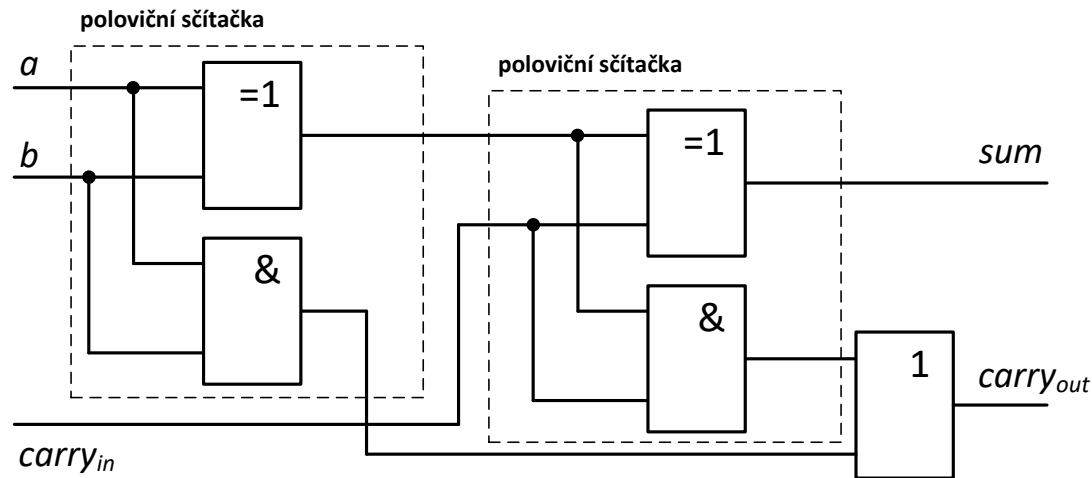
$$sum = \bar{a}\bar{b}c_{in} \vee a\bar{b}\bar{c}_{in} \vee abc_{in} \vee \bar{a}b\bar{c}_{in} = a \oplus b \oplus c_{in}$$

$$carry_{out} = ab \vee ac_{in} \vee bc_{in} = ab \vee a \cdot (b \vee \bar{b}) \cdot c_{in} \vee (a \vee \bar{a}) \cdot bc_{in} = ab \vee c_{in} \cdot (a \oplus b)$$

- díky této úpravě vyjádření **může být úplná sčítačka realizována pomocí 2 polovičnicků!**
- výstup přenosu $carry-out$ celé úplné sčítačky může být realizován jako logický součet přenosu z první poloviny sčítačky a logického součinu vstupu přenosu $carry-in$ a součtu sum z první poloviny sčítačky

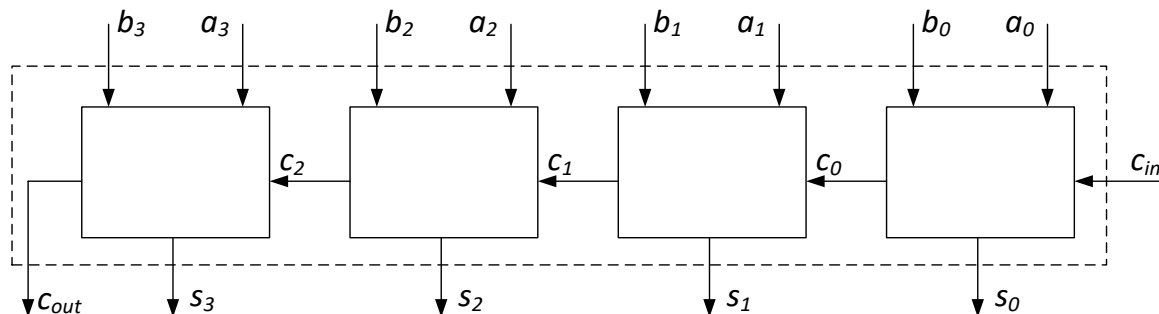
Kombinační vs. sekvenční logické obvody – KLO, příklady kombinačních obvodů

▪ Sčítáčka – realizace úplné sčítáčky pomocí 2 polovičních



- **paralelní sčítáčka s postupným přenosem (ripple carry)** – kaskádním řazením n 1bitových úplných sčítaček snadno vytvoříme n -bitovou úplnou sčítáčku
- **nevýhoda** – n x zvýšení zpoždění přenosu, pro velká čísla (např. 32bitová) pomalé

4bitová paralelní sčítáčka s postupným přenosem



- **paralelní sčítáčka s predikcí přenosu (carry look-ahead)** – opět paralelní sčítáčka, ale přenos (příznak přenosu) je počítán ve zvláštním kombinačním obvodu – na základě příznaku pak rychlejší nastavení při samotném sčítání

Kombinační vs. sekvenční logické obvody – hazardy v kombinačních obvodech

▪ Logické hazardy v kombinačních logických obvodech

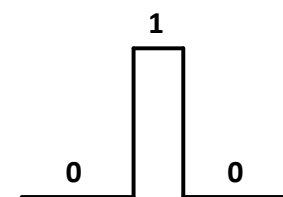
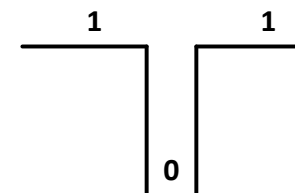
- **logický hazard** = neočekávaná hodnota na výstupu logického obvodu, která neodpovídá jeho výstupní funkci (dle algebraického popisu, pravdivostní tabulky...)
- **vznik logických hazardů v obvodech je zapříčiněn zejména nenulovou dobou zpoždění**
- **doba zpoždění logického hradla** = časová prodleva, která uběhne od okamžiku, kdy signál na vstupu hradla dosáhne určené rozhodovací úrovně (log. 1 či 0) a okamžiku, kdy se na výstupu hradla objeví správná a ustálená odpovídající hodnota
- **kritická je celková doba šíření jednotlivých vstupů obvodu na jeho celkový výstup**
- **logické hazardy představují problém pro KLO i SLO:**
 1. výstup KLO obsahující logický hazard je připojen na vstup jiného KLO – logický hazard se tak z jednoho KLO šíří do dalšího a vytváří neočekávané výstupy
 2. výstup KLO je připojen na vstup synchronního SLO – v tomto případě záleží na poměru doby trvání hazardu a periody hodinového signálu Clk SLO obvodu
- obecně, logické hazardy můžeme dělit dle různých kritérií a typů, zaměříme se na:
 1. **Statické hazardy**
 - statický hazard v logické 1
 - statický hazard v logické 0
 2. **Dynamické hazardy**
- dále rozlišujeme hazardy funkční a strukturální, se změnou v jedné proměnné a ve více proměnných atd.

Kombinační vs. sekvenční logické obvody – hazardy v kombinačních obvodech

1. Statické hazardy

- statický hazard představuje situaci, kdy **kombinace vstupních proměnných** při přechodu z jednoho do druhého sousedního stavu **vyvolá na výstupu obvodu neočekávaný přechodný (dočasný) stav** s opačnou hodnotou než doposud, ačkoliv výstup obvodu měl zůstat nezměněn
- statický hazard může vzniknout zejména v obvodu **kdy dva a více implikantů dané výstupní funkce obvodu obsahují tutéž proměnnou v různých tvarech**, přímo a negaci
- **statický hazard v logické 1** – hazard, kdy se na výstupu obvodu objeví krátce hodnota logická 0, ačkoliv by výstup měl setrvávat v hodnotě logická 1
- **statický hazard v logické 1** se může objevit jen v obvodu realizovaném **součinovými hradly (AND, NAND)** při realizaci pomocí **disjunktčního tvaru funkce**
- **statický hazard v logické 0** – obdobně, hazard, kdy se na výstupu obvodu objeví krátce hodnota logická 1, ačkoliv by výstup měl setrvávat v hodnotě logická 0
- **statický hazard v logické 0** se může objevit jen v obvodu realizovaném **součtovými hradly (OR, NOR)** při realizaci pomocí **konjunktčního tvaru funkce**
- **odstranění statického hazardu:**
 1. vložení zpožďovacích členů – vyrovnaní zpoždění v obvodu – nepoužívat
 2. strukturální úprava obvodu – preferované
 3. použití synchronizačního signálu – pak jde ale o SLO

Statický hazard v logické 1

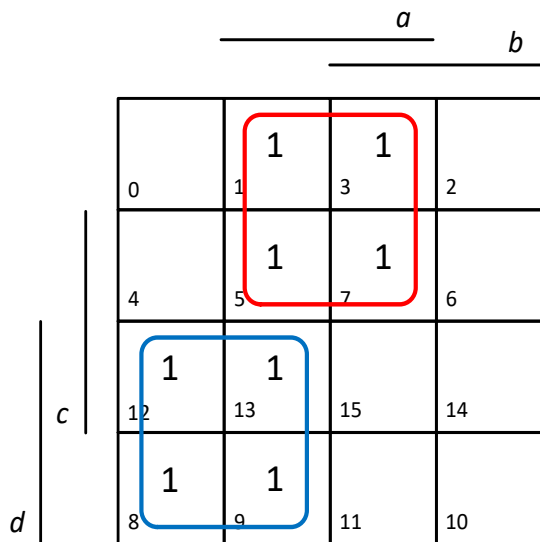


Statický hazard v logické 0

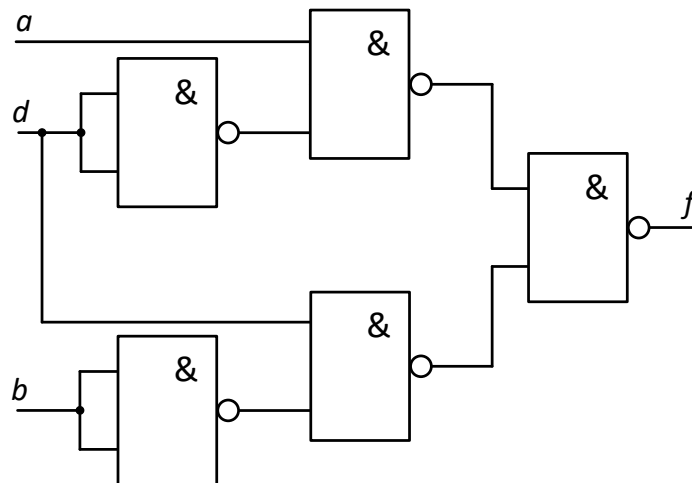
Kombinační vs. sekvenční logické obvody – hazardy v kombinačních obvodech

1. Statické hazardy

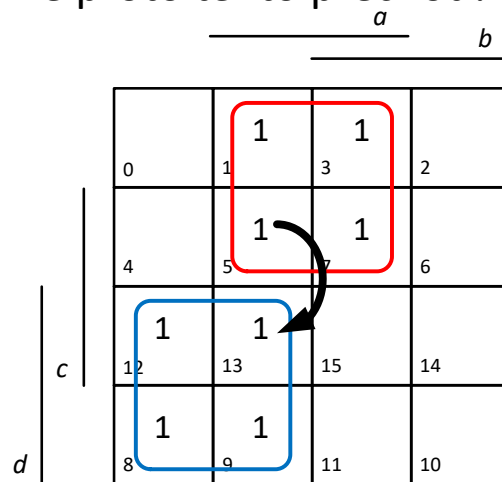
- příklad statického hazardu v log. 1: realizujte MNDF funkce $f = 1, 3, 5, 7, 8, 9, 12, 13$



$$f = a\bar{d} \vee \bar{b}d = \overline{\overline{a}\bar{d}} \cdot \overline{\overline{\bar{b}}\bar{d}}$$



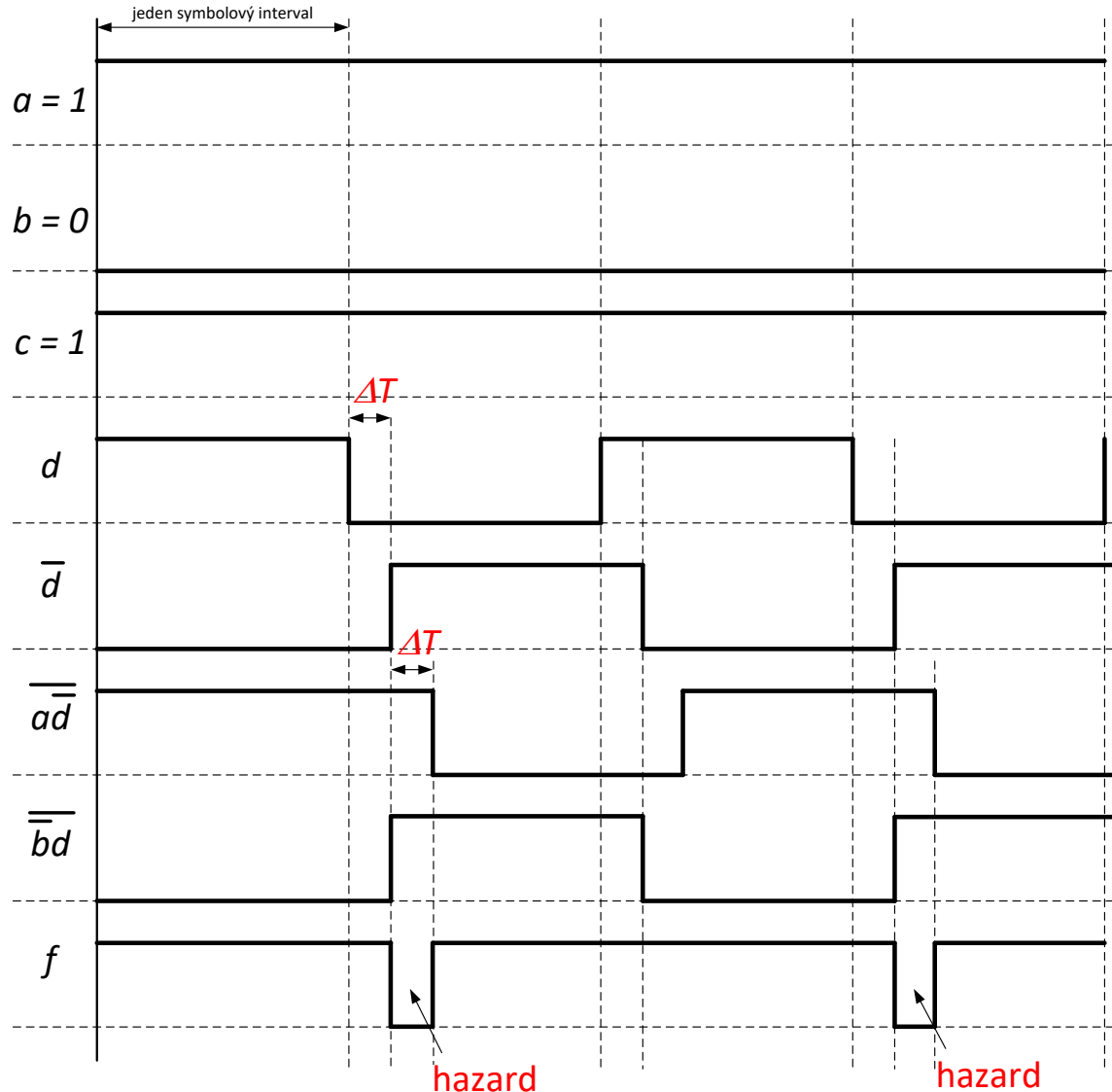
- evidentně **proměnná d** je použita 2x a šíří se ze vstupu na výstup obvodu dvěma cestami – **negovanou a přímou** -> vyšetřeme proto tento přechod!
- přechod mezi indexy 5 – 13:
5: $a = 1, b = 0, c = 1, d = 0$
13: $a = 1, b = 0, c = 1, d = 1$
- přechod mezi indexy 5 – 13:
 d – se mění, a, b, c zůstávají konstantní
- pro oba indexy 5, 13, je hodnota funkce **$f = 1$**



Kombinační vs. sekvenční logické obvody – hazardy v kombinačních obvodech

1. Statické hazardy

- pro vyšetření obvodu sestavíme **časový diagram** šíření daného přechodu obvodem
- uvažujeme konstantní dobu zpoždění všech hradel v obvodu – ΔT



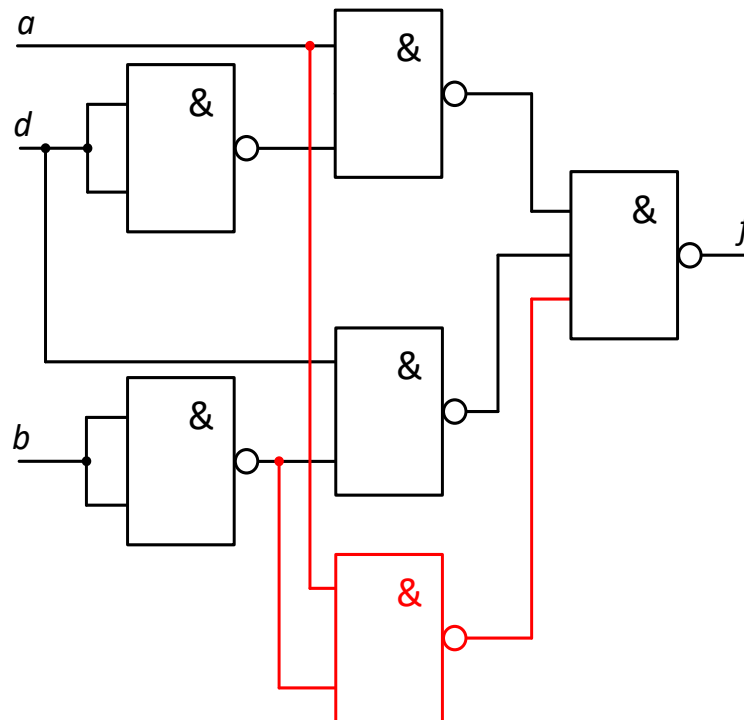
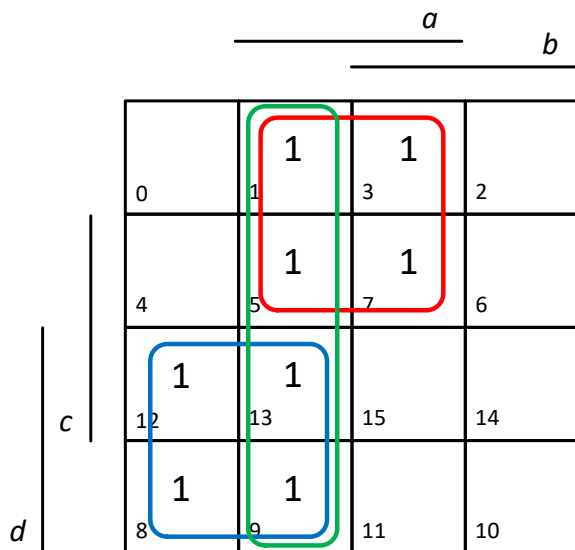
- negace \bar{d} představuje zpoždění o $1 \times \Delta T$
- proto na krátký časový okamžik mohou obě d a \bar{d} nabývat logické 0, díky tomu se může logická 0 objevit i na výstupu obvodu f !
- to způsobí vznik statického hazardu v logické 1 funkce f

(výstup z posledního hradla je opět zpožděn o ΔT , ale pro lepší čitelnost nezakresleno)

Kombinační vs. sekvenční logické obvody – hazardy v kombinačních obvodech

1. Statické hazardy

- jak tento problém vyřešit?
- musíme překrýt indexy 5 a 13 **dodatečným** (redundantním) **implikantem**, který nebude obsahovat proměnnou **d** -> zelená smyčka v mapě



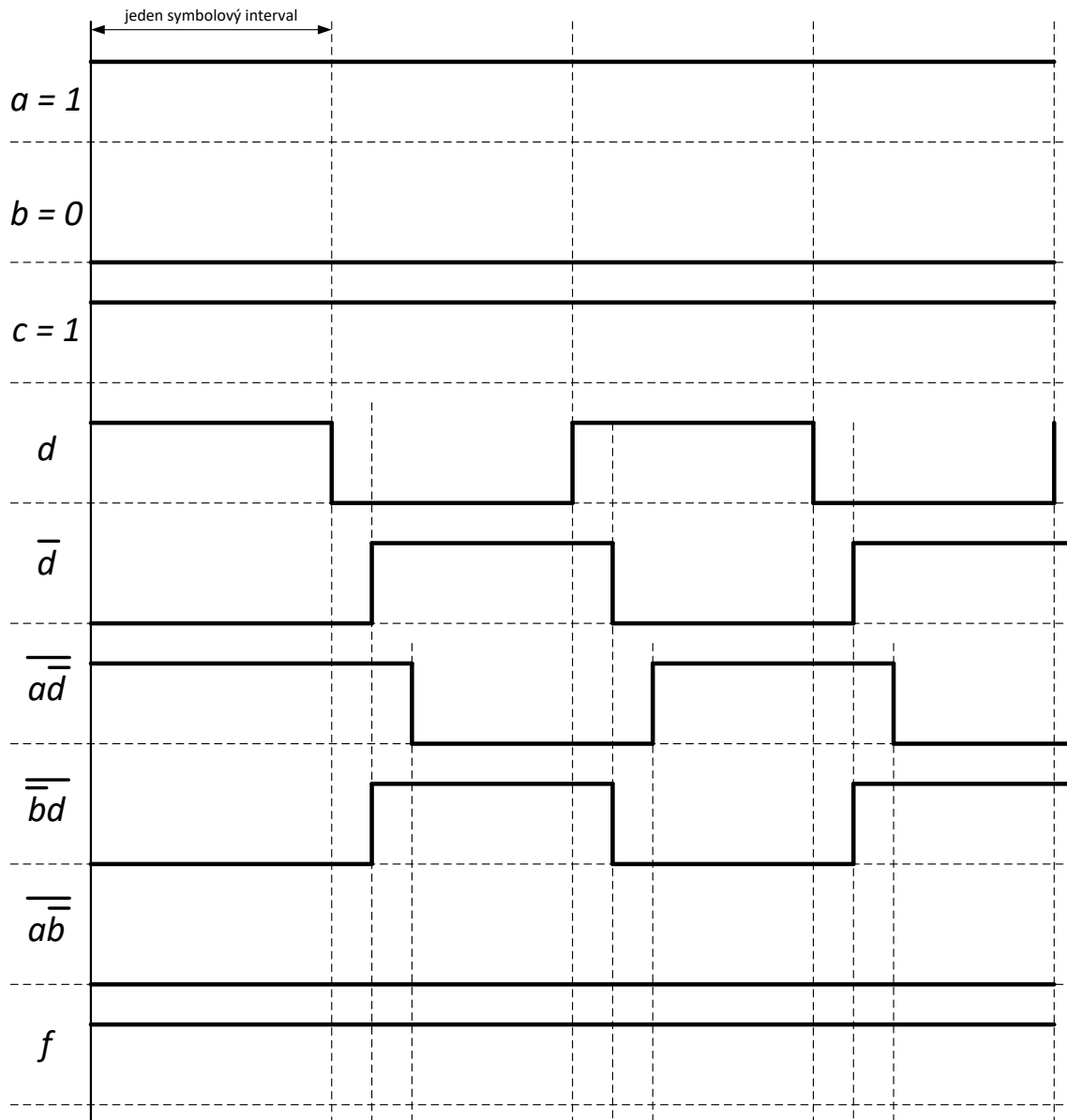
$$f = a\bar{d} \vee \bar{a}b \vee a\bar{b} = \overline{\overline{a\bar{d}} \cdot \overline{\bar{a}b} \cdot \overline{a\bar{b}}}$$

- pomocí **dodatečného prostého implikantu** $a\bar{b}$ jsme pokryli najednou oba indexy 5, 13 bez použití proměnné **d**
- díky tomu jsme doplnili do realizace obvodu **jedno dodatečné hradlo NAND** a museli jsme použít **3vstupé hradlo NAND** na výstupu obvodu (místo původního 2vstupého)
- pro ověření vytvoříme nový časový diagram upraveného obvodu

Kombinační vs. sekvenční logické obvody – hazardy v kombinačních obvodech

1. Statické hazardy

- časový diagram upraveného obvodu

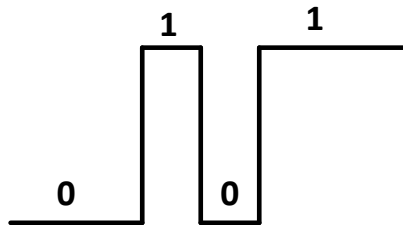


- hodnota na výstupu hradla $\overline{\overline{a\overline{b}}}$ je konstantně logická 0 při přechodu mezi indexy 5, 13
- díky tomu, **alespoň jeden ze tří vstupů 3vstupého hradla NAND na výstupu obvodu je vždy logická 0**, tedy na jeho výstupu bude vždy logická 1 (výstup f)
- díky tomu jsme **odstranili původní statický hazard v logické 1**

Kombinační vs. sekvenční logické obvody – dynamické hazardy

2. Dynamické hazardy

- je definován jako **několikanásobný přechod mezi logickou 0 a logickou 1 před ustálením výstupu obvodu** – např. **0101, 1010** apod.
- vzniká v obvodech **složených ze součinných a součtových hradel**
- dynamický hazard vznikne v důsledku existence nejméně trojice různých cest stejné proměnné v obvodu a kdy každá z těchto cest vykazuje odlišné zpoždění
- schéma obvodu kreslíme vhodně tak, že směrem od vstupu k výstupu obvodu uspořádáme hradla pod sebe dle jejich pořadí – **počet hradel v cestě = stupeň obvodu**
- díky tomu můžeme jednoduše odhadnout rozdílné zpoždění jednotlivých cest
- dynamické hazardy **je obtížnější odhalit a odstranit** než statické – **nejprve eliminujeme postupně všechny statické hazardy a tím eliminujeme i dynamický hazard**
- eliminace dynamického hazardu obvykle spočívá v úplné modifikaci zapojení obvodu, nebo použití taktovacího vstupu (převod na SLO)

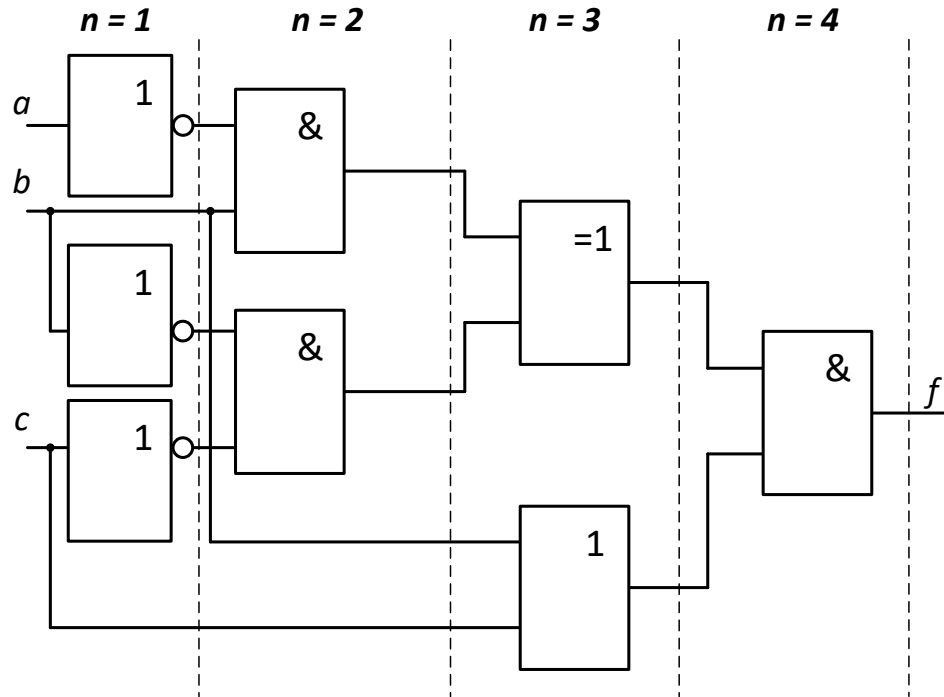


Dynamický hazard

Kombinační vs. sekvenční logické obvody – dynamické hazardy

2. Dynamické hazardy

- **příklad** – vyšetřete funkci následujícího obvodu, n je počet stupňů

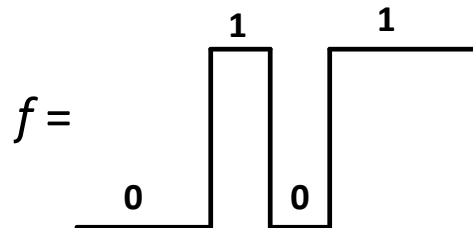
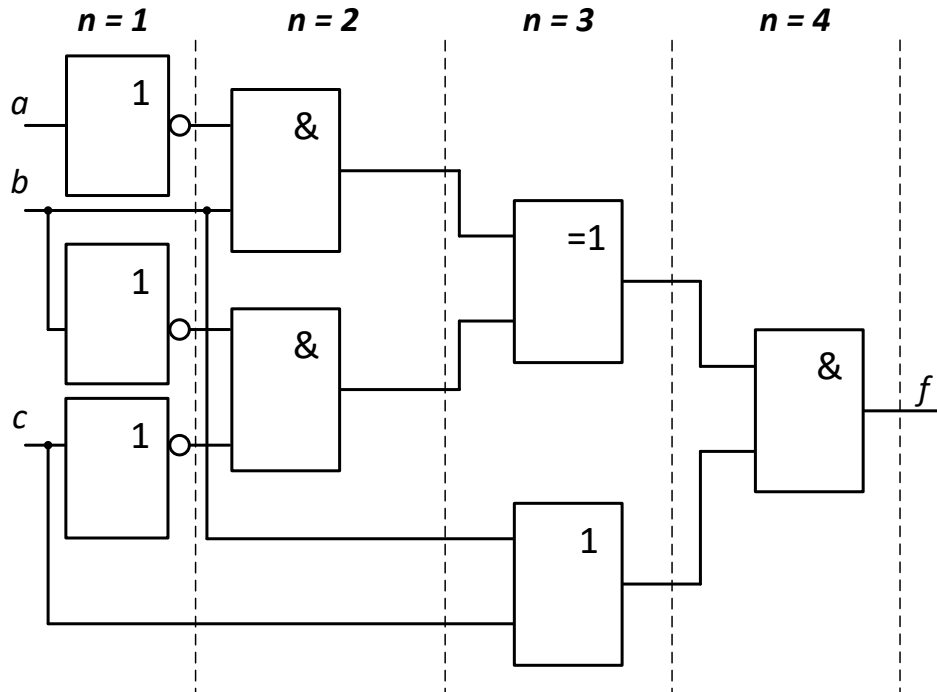


- proměnná **b** se šíří všemi třemi větvemi obvodu na jeho výstup:
horní větví – skrz **3 logická hradla**
prostřední větví – skrz **4 logická hradla**
spodní větví – skrz **2 logická hradla**
- měli bychom tedy analyzovat výstup obvodu při změně hodnoty proměnné b , zatímco proměnné a , c zůstávají stejné, například přechod mezi: **000** -> **010**

Kombinační vs. sekvenční logické obvody – dynamické hazardy

2. Dynamické hazardy

- **příklad** – uvažujeme konstantní dobu zpoždění všech hradel ΔT



Dynamický hazard

- **výchozí stav obvodu:**

$$a = 0, b = 0, c = 0$$

$$\text{výstup } f = 0$$

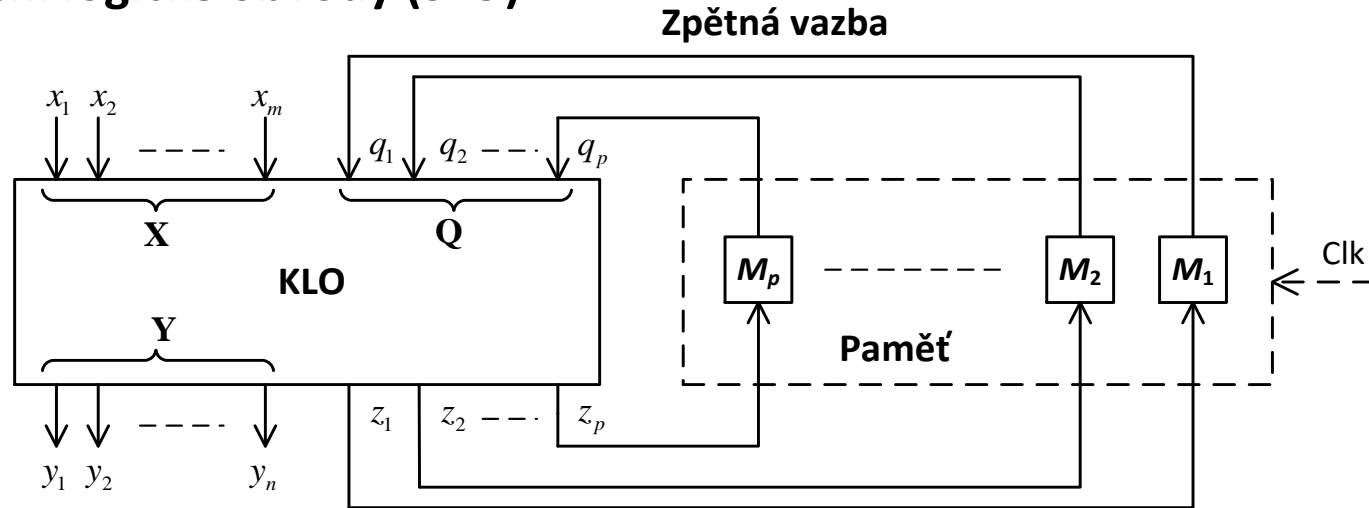
- **nyní změníme hodnoty vstupů:**

$$a = 0, b = 1, c = 0$$

1. spodní větev obvodu má nejmenší zpoždění = $2 \times \Delta T$ (2 hradla), tedy celkový výstup obvodu po uplynutí **$2 \times \Delta T$** je **$f = 1$**
2. další v pořadí je horní větev se zpožděním = $3 \times \Delta T$ (3 hradla), výstup obvodu po uplynutí **$3 \times \Delta T$** je **$f = 0$**
3. konečně, po uplynutí $4 \times \Delta T$ se dostane proměnná b na výstup obvodu skrz prostřední větev, tedy po **$4 \times \Delta T$** bude výstup obvodu **$f = 1$**

Sekvenční logické obvody

▪ Sekvenční logické obvody (SLO)



- **sekvenční log. obvod (SLO) = kombinační log. obvod (KLO) + paměť + zpětná vazba**
- každý SLO je plně určen:
 1. konečná množina vstupních stavů: $\mathbf{X} = \{x_1, x_2, \dots, x_M\}$
 2. konečná množina výstupních stavů: $\mathbf{Y} = \{y_1, y_2, \dots, y_N\}$
 3. konečná množina vnitřních stavů: $\mathbf{Q} = \{q_1, q_2, \dots, q_p\}$
 4. konečná množina přechodových funkcí \mathbf{G} : $\mathbf{Q}^{t+1} = \mathbf{G}(\mathbf{X}^t; \mathbf{Q}^t) = g(x_1^t, x_2^t, \dots, x_m^t; q_1^t, q_2^t, \dots, q_p^t)$
 5. konečná množina výstupních funkcí \mathbf{F} :
 - a. **obvod typu Mealy** $\mathbf{Y}^{t+1} = \mathbf{F}(\mathbf{X}^t; \mathbf{Q}^t) = f(x_1^t, x_2^t, \dots, x_m^t; q_1^t, q_2^t, \dots, q_p^t)$
 - b. **obvod typu Moore** $\mathbf{Y}^{t+1} = \mathbf{F}(\mathbf{Q}^t) = f(q_1^t, q_2^t, \dots, q_p^t)$

Sekvenční logické obvody

▪ Sekvenční logické obvody (SLO) – Mealy vs. Moore – synchronní režim

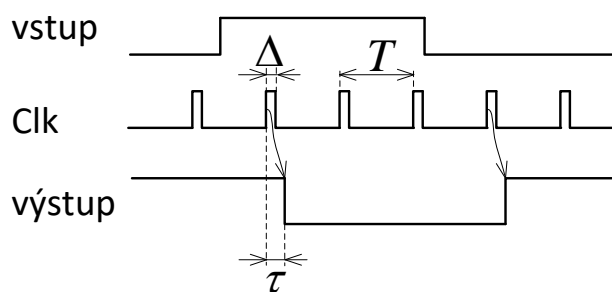
- **typ Mealy** – výstup obvodu závisí na aktuální **hodnotě vstupů a vnitřních stavů**:

$$\mathbf{Y}^{t+1} = \mathbf{F}(\mathbf{X}^t; \mathbf{Q}^t) = f(x_1^t, x_2^t, \dots, x_m^t; q_1^t, q_2^t, \dots, q_p^t)$$

- **typ Moore** – výstup obvodu závisí pouze na aktuální **hodnotě vnitřních stavů**:

$$\mathbf{Y}^{t+1} = \mathbf{F}(\mathbf{Q}^t) = f(q_1^t, q_2^t, \dots, q_p^t)$$

- **čas t** = okamžik těsně před příchodem hrany hodinového signálu Clk
- **čas t+1** = okamžik těsně po uskutečnění přechodu do nového stavu
- u **synchronního obvodu** se vstupní hodnota \mathbf{X}^t vyhodnocuje **pouze během doby trvání hodinového impulsu** Clk Δ (v ideálním obvodu na jeho nekonečně strmou hranu)
- ke **změně vnitřního stavu** obvodu dochází **během doby T** mezi dvěma po sobě následujícími impulsy hodinového signálu Clk



kde:

- Δ je doba trvání jednoho hodinového pulzu Clk
- T je jeho perioda
- τ představuje zpoždění výstupu obvodu

- v případě **asynchronního obvodu** – **změna vstupní hodnoty přímo způsobí změnu výstupní hodnoty** se zpožděním τ
- další změna vstupních hodnot tak musí přijít až po ukončení přechodu do nového vnitřního stavu – více se asynchronními obvody zabývat nebudeme

▪ Sekvenční logické obvody (SLO) – popis obvodů

1. algebraický popis SLO

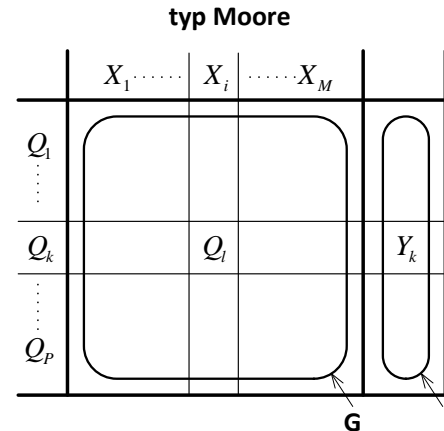
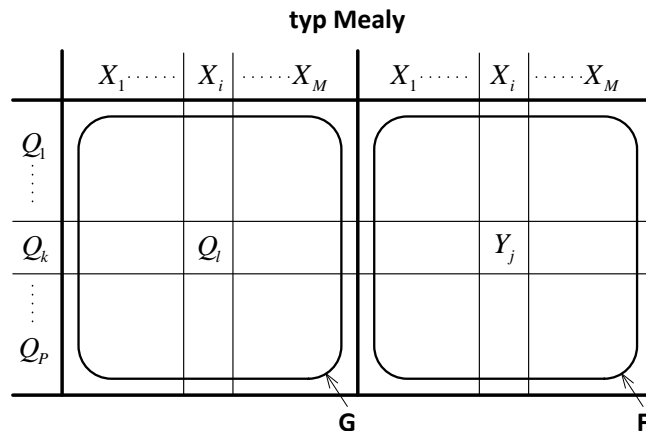
- popis pomocí algebraického vyjádření množiny přechodových a výstupních funkcí

$$\mathbf{Q}^{t+1} = \mathbf{G}(\mathbf{X}^t; \mathbf{Q}^t) = g(x_1^t, x_2^t, \dots, x_m^t; q_1^t, q_2^t, \dots, q_p^t)$$

$$\mathbf{Y}^{t+1} = \mathbf{F}(\mathbf{X}^t; \mathbf{Q}^t) = f(x_1^t, x_2^t, \dots, x_m^t; q_1^t, q_2^t, \dots, q_p^t) \quad \text{typ Mealy}$$

$$\mathbf{Y}^{t+1} = \mathbf{F}(\mathbf{Q}^t) = f(q_1^t, q_2^t, \dots, q_p^t) \quad \text{typ Moore}$$

2. tabulkami přechodů a výstupů

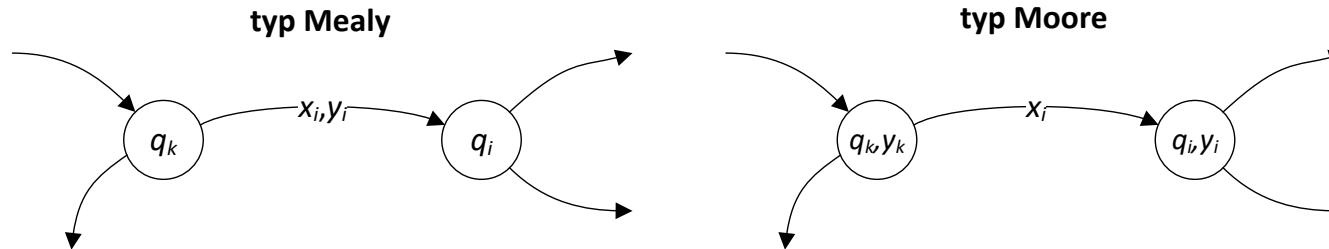


- P** řádků = **P** vnitřních stavů obvodu
- typ Mealy:
- 2x M** sloupců vstupních stavů = **1x M** sloupců pro přechody + **1x M** sloupců pro výstupy
- Y = M x P** výstupních stavů
- typ Moore:
- 1x M** sloupců vstupních stavů + **1** = **1x M** sloupců pro přechody + **1** sloupec pro výstupy
- Y = 1 x P** výstupních stavů

▪ Sekvenční logické obvody (SLO) – popis obvodů

3. orientovaný graf přechodů (přechodový diagram, stavový diagram)

- grafické vyjádření přechodů a výstupů SLO



- **P vnitřních stavů** = **P uzlů grafu**, přechody = orientované hrany grafu
- **N vstupů obvodu** = **$P+N$ orientovaných hran grafu**
- **typ Mealy** – hrany grafu = vstupní stavy, výstupní stavy obvodu
- **typ Moore** – hrany grafu = vstupní stavy, uzly grafu = výstupní stavy

4. popis chování obvodu v jazyce VHDL, Verilog...

- v pozdější přednášce

5. popis vývojovým diagramem

- standardizované grafické značky, nejčastěji jazyk UML

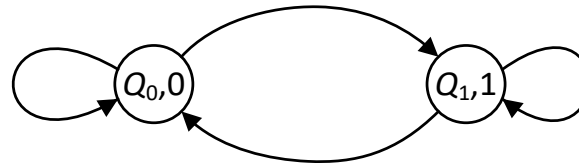
6. časovým diagramem, slovním popisem

Klopné obvody, sekvenční obvody – hladinové a hranové obvody, transformace

▪ Klopné obvody, hladinové vs. hranové obvody

- dle počtu stabilních stavů dělíme klopné obvody (KO)
- **astabilní KO (AKO)** – nemají žádný stabilní stav, zdroje periodického signálu
- **monostabilní KO (MKO)** – jeden stabilní stav, generátory pulzů požadované délky
- **bistabilní KO (BKO)** – dva stabilní stavy, mohou plnit funkci paměťového členu

• obecný přechodový diagram BKO



• 4 základní transformace v BKO

Název transformace	Označení transformace	Přechodový graf
Paměťová	M	<pre>graph LR; 0((0)) --> 0; 1((1)) --> 1;</pre>
Jedničková	1	<pre>graph LR; 0((0)) --> 1((1)); 1 --> 1;</pre>
Nulová	0	<pre>graph LR; 1((1)) --> 0((0)); 0 --> 0;</pre>
Klopná	K	<pre>graph LR; 0((0)) --> 1((1)); 1 --> 0;</pre>

• 4 základní BKO:

- **typ RS** – realizuje transformace M, 0 a 1
- **typ JK** – realizuje všechny čtyři transformace
- **typ D** – realizuje transformace 0 a 1
- **typ T** – realizuje transformace M a K

Klopné obvody, sekvenční obvody – hladinové a hranové obvody, transformace

▪ Klopné obvody, hladinové vs. hranové obvody

- **latch vs. flip-flop ve VHDL**
- jak vytvořit hranově řízený obvod ve VHDL (flip-flop)?
... pokud je clock ve stavu log. 1 (vzestupná hrana) a současně detekována hrana, pak...
- jak vytvořit hladinově řízený obvod ve VHDL (latch)?
... pokud je clock ve stavu log. 1, pak...
- ve VHDL se často omylem stane, že vyrobíme nechtěný a nežádoucí latch, jak třeba?
- například zapomeneme určit co se má stát při všech možných vstupních kombinacích:

```
pokud A = B, pak  
    C = 1;  
jinak  
    C = 0;  
konec;
```

```
pokud A = B, pak  
    C = 1;  
konec;
```

ve VHDL lze zapsat oboje, ale co se stane, pokud **A ≠ B**?

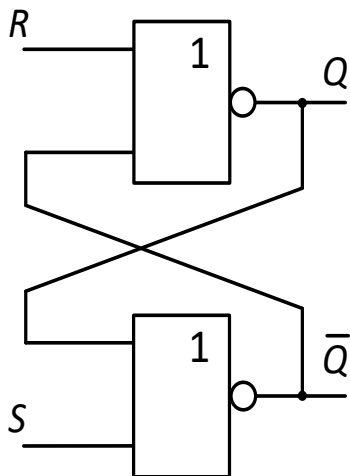
- v prvním případě $C = 0$
- ale co ve druhém? syntezátor neví, když neví, domyslí si, že se nemá stát nic, takže výsledek bude $C = C$ – tím vytvoříme latch (paměť) pro zapamatování stavu C!

Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

▪ Klopný obvod RS (reset-set latch)

- 2 vstupy: S – Set, R – Reset, 2 výstupy: Q – přímý, \bar{Q} – negovaný
- asynchronní – hradla NOR, NAND, synchronní – řízený hladinou Clk

RS asynchronní – realizace NOR – aktivován logickou 1

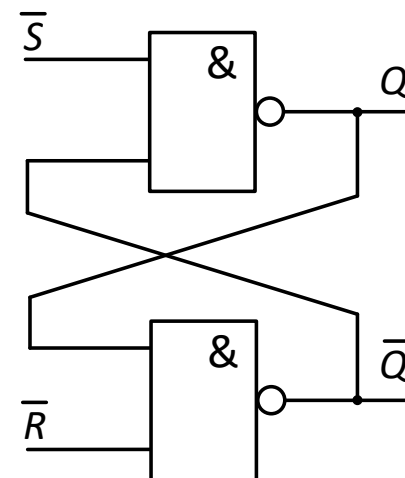


1. pokud S = logická 1 – Q = logická 1, \bar{Q} = logická 0 -> jedničková transf.
2. pak S nastavíme na logickou 0 – výstupy se nezmění -> paměťová tr.
3. pak R nastavíme na logickou 1 – Q = logická 0, \bar{Q} = logická 1 -> nulová tr.
4. pak R vrátíme na logickou 0 – výstupy se nezmění -> paměťová tr.
5. pokud S i R nastavíme na logickou 1 – výstup není určen – **zakázaný stav!**

R	S	Q^{t+1}	transformace
0	0	Q^t	M
0	1	1	1
1	0	0	0
1	1	zakázaný stav	

RS asynchronní – realizace NAND – aktivován logickou 0

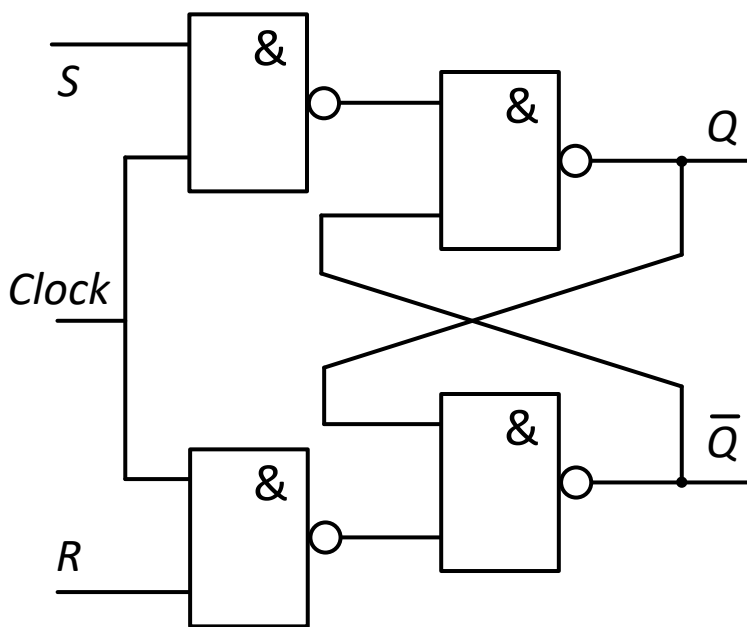
- výměnou hradel NOR za NAND obdržíme rovněž klopný obvod RS, ale:
- aktivovaný logickou 0 (inverzní funkce)



Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

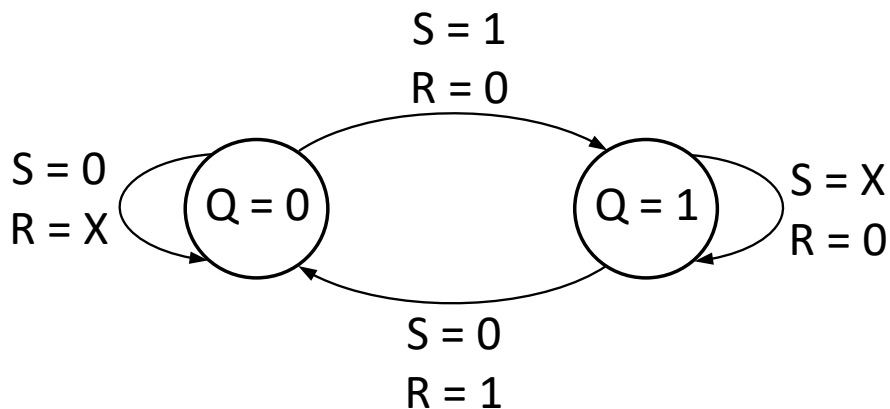
▪ RS synchronní s hladinovým i hranovým vstupem Clock

- synchronní úprava RS klopného obvodu

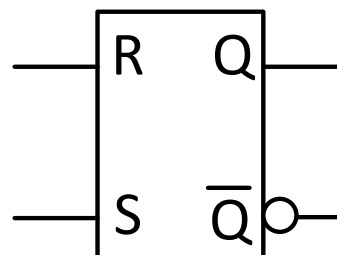


<i>Clock</i>	<i>R</i>	<i>S</i>	<i>Q^t</i>	<i>Q^{t+1}</i>	<i>Transformace</i>
↑	0	0	0	0	Paměťová (M)
↑	0	0	1	1	
↑	0	1	0	1	Jedničková (1)
↑	0	1	1	1	
↑	1	0	0	0	Nulová (0)
↑	1	0	1	0	
↑	1	1	0	!	zakázaný stav
↑	1	1	1	!	
jiný	x	x	x	Q ^t	-

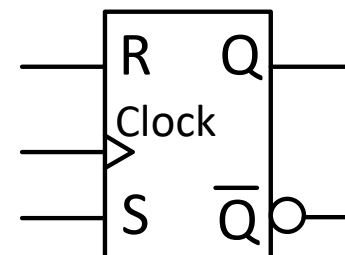
- přechodový diagram, značky



RS asynchronní



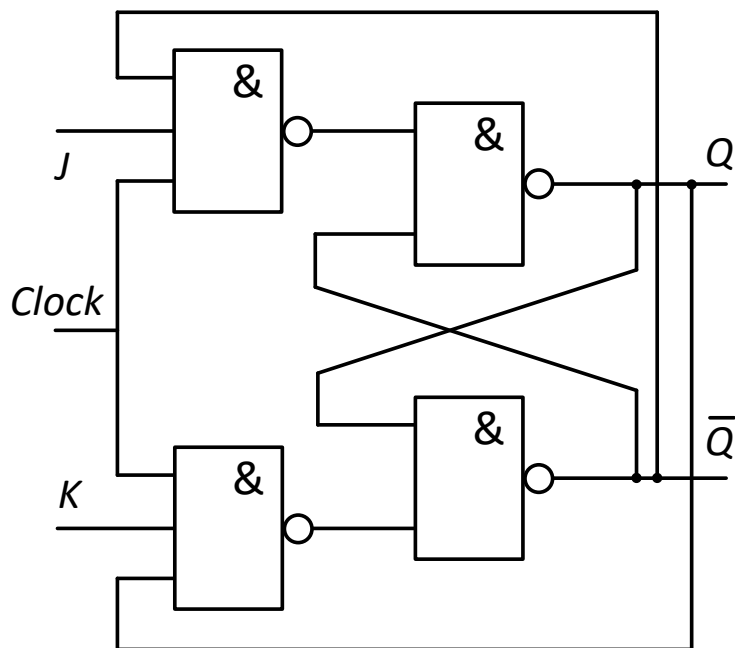
RS synchronní



Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

▪ Klopný obvod JK – synchronní s hladinovým i hranovým vstupem Clock

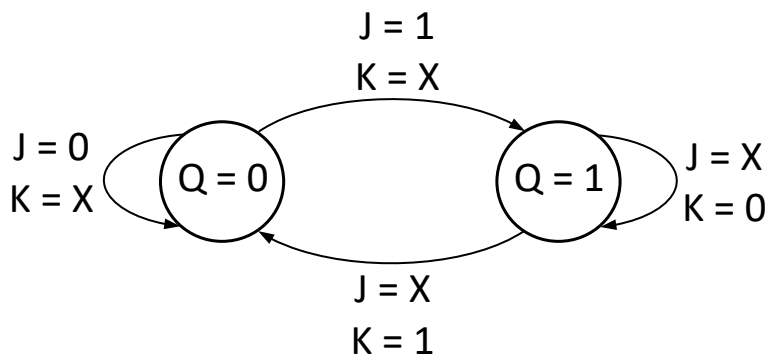
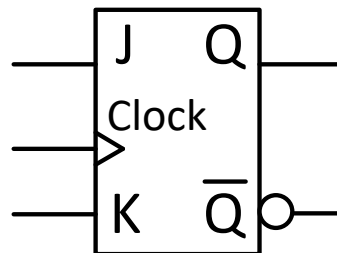
- realizuje **všechny 4 transformace**, pojmenován po svém objeviteli (Jack Kilby)
- **pro odstranění zakázaného stavu RS obvodu** – doplníme **2 zpětné vazby** -> JK obvod



přechodový diagram

<i>Clock</i>	<i>J</i>	<i>K</i>	Q^t	Q^{t+1}	<i>Transformace</i>
↑	0	0	0	0	Paměťová (M)
↑	0	0	1	1	
↑	0	1	0	0	Nulová (0)
↑	0	1	1	0	
↑	1	0	0	1	Jedničková (1)
↑	1	0	1	1	
↑	1	1	0	1	Klopná (K)
↑	1	1	1	0	
jiný	x	x	x	Q^t	-

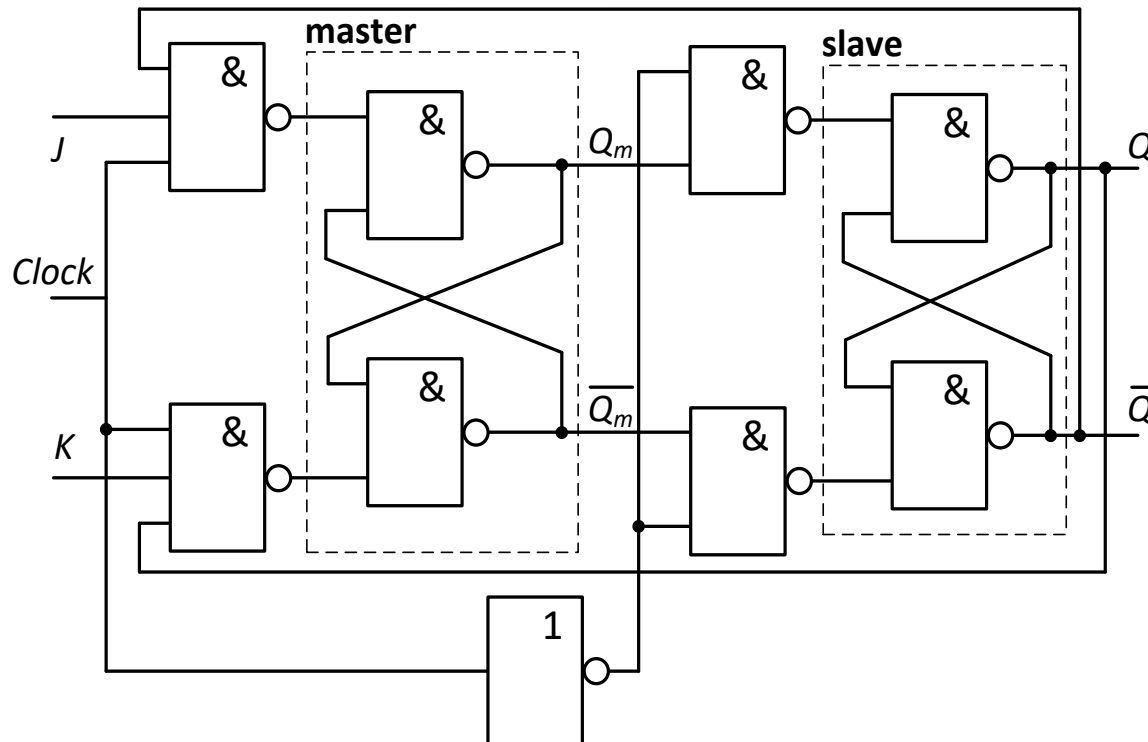
značka JK klopný obvod



Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

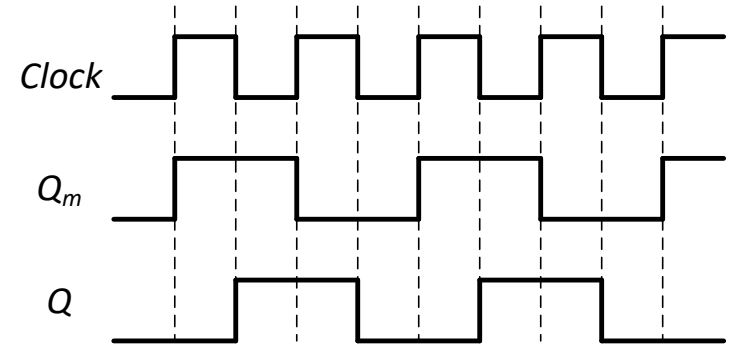
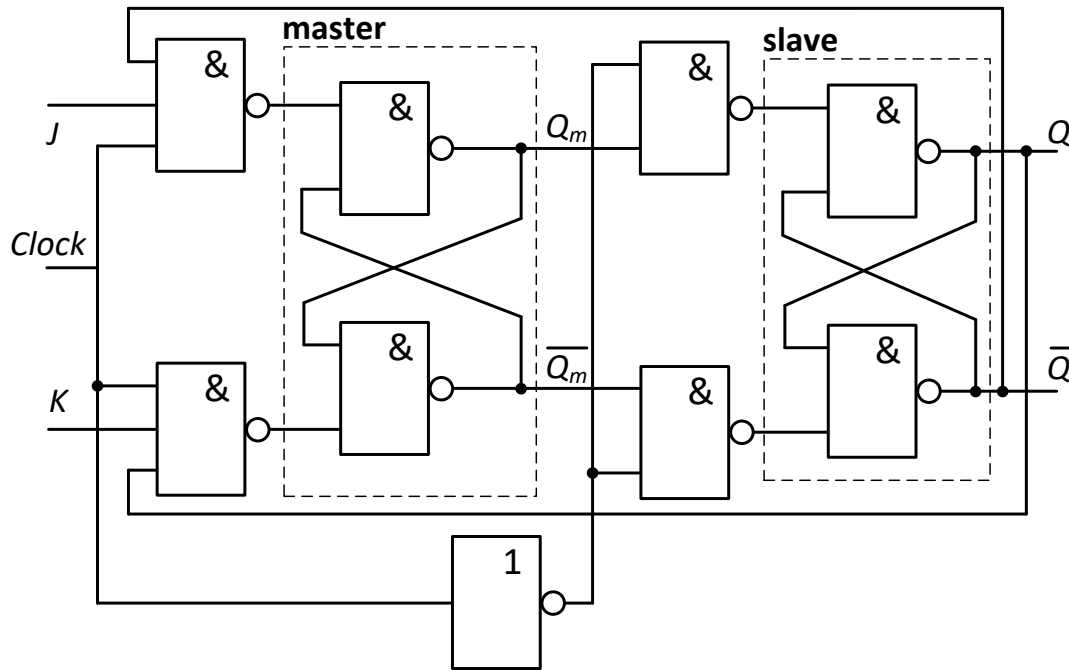
▪ Klopný obvod JK

- problém základního JK – **možnost vícenásobného kmitání** (tzv. **race-hazard**)
- problém způsoben **dvojicí zpětných vazeb** -> díky zpětným vazbám se změna výstupu obvodu dostane zpět na jeho vstup -> pokud stále trvá hrana (či hladina) vstupu Clock, způsobí to další překlopení obvodu a tak dále... -> ve výsledku se výstup obvodu může nekontrolovatelně rozkmitat mezi logickou 1 a 0
- první řešení – zvýšit frekvenci hodinového signálu Clock (zkrátit aktivní dobu obvodu) – ne vždy možné s ohledem na aplikaci
- druhé řešení – tzv. **master-slave zapojení**



Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

▪ Klopný obvod JK v master-slave zapojení

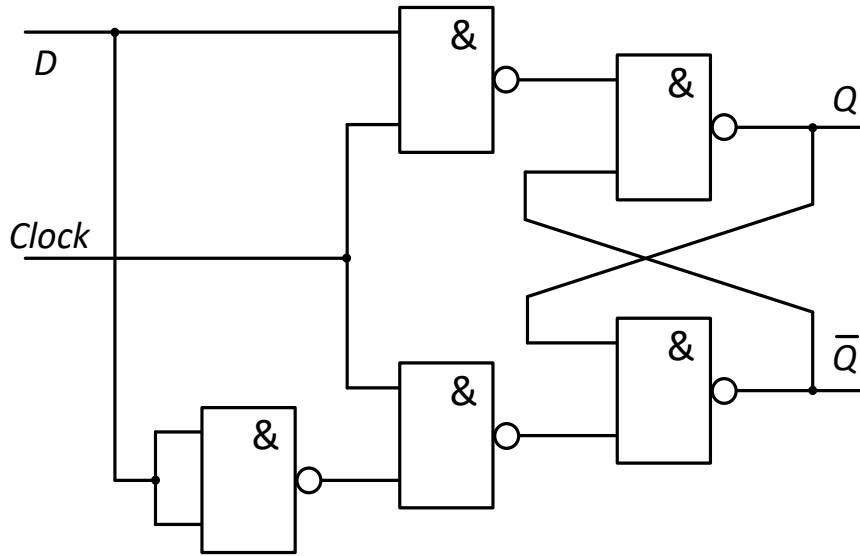


- **část master** – reaguje na **vzestupnou hranu Clock**, **část slave** – **sestupná hranu Clock**
- **při vzestupné hraně Clock:**
 - část master aktivní, na výstupech Q_m a $\overline{Q_m}$ se objeví příslušný výstup
 - vstup části slave však blokován (negace Clock) – slave část neaktivní
 - zpětné vazby zablokovány (slave část neaktivní)
- **při sestupné hraně Clock:**
 - část master neaktivní – na výstupu zůstávají předchozí stavy Q_m a $\overline{Q_m}$
 - část slave aktivní – zareaguje na vstupy Q_m a $\overline{Q_m}$
 - zpětné vazby sice přenesou změnu zpět na vstup části master – ale ten je neaktivní

Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

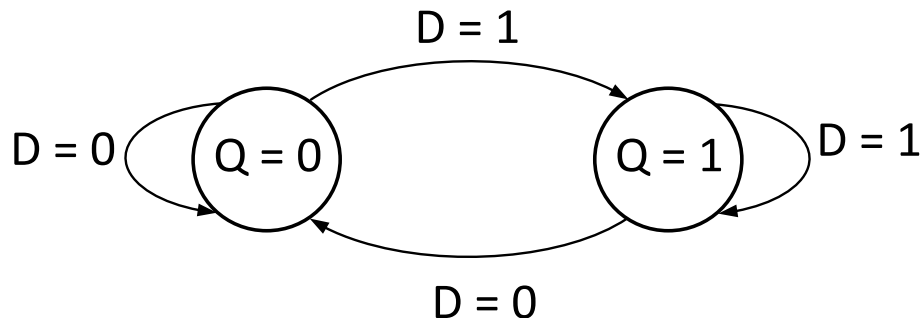
▪ Klopný obvod D

- synchronní klopný obvod RS, kdy spojíme vstupy $R = \overline{S}$, jen jedničková a nulová transf.
- 1 vstup D – data (delay), obvod D – paměťový člen s omezenou dobou záznamu

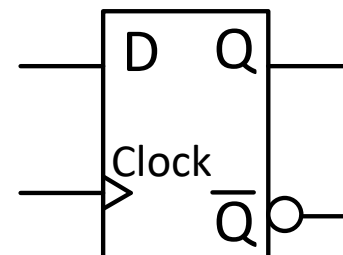


<i>Clock</i>	<i>D</i>	<i>Q^t</i>	<i>Q^{t+1}</i>	<i>Transformace</i>
↑	0	0	0	Nulová (0)
↑	0	1	0	
↑	1	0	1	Jedničková (1)
↑	1	1	1	
jiny	x	x	Q^t	-

- obvod D = kopíruje hodnotu ze vstupu D na výstup Q (a negovanou hodnotu na výstup \overline{Q}) s každou hranou (hladinou) hodinového vstupu Clock
- dočasná paměť – hodnota ze vstupu D se uchová jen do následujícího pulzu Clock



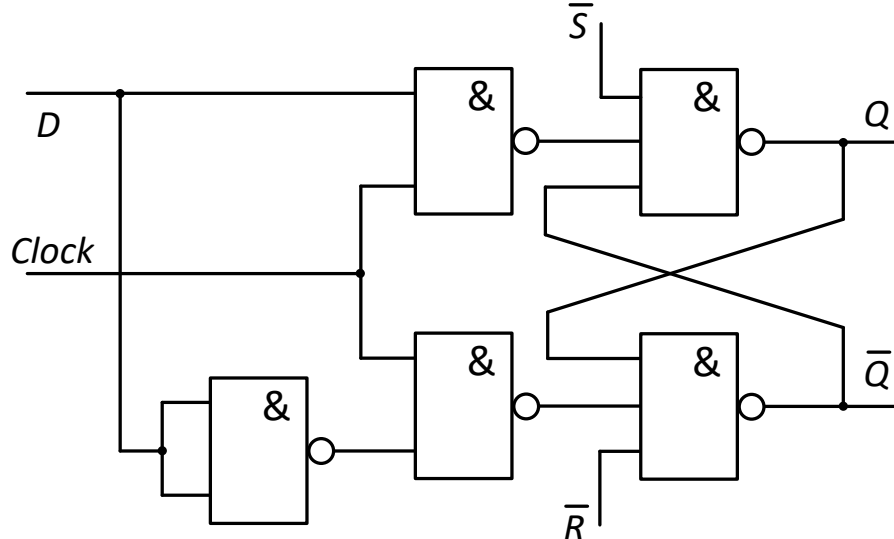
značka klopný obvod D



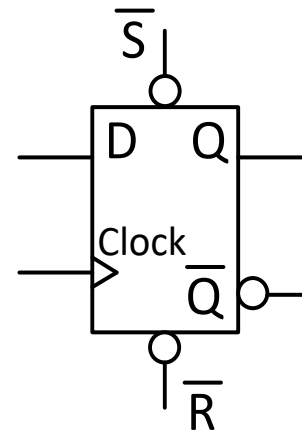
Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

▪ Klopný obvod D

- **problém hladinového obvodu D** – pokud je doba trvání hladiny Clock > doba trvání jedné hodnoty na vstupu D, tzn. pokud během aktivní doby se změní hodnota vstupu D vícekrát, změní se vícekrát i výstup a po skončení zůstane uložena jen poslední hodnota ze vstupu (tzv. ripple-through)
- řešení – **hranový klopný obvod D** – použitím dodatečných hradel (obvykle NAND) a využitím jejich doby zpoždění zkrátíme trvání Clock – jen krátké pulsy (hrany)
- klopný obvod D s **asynchronními vstupy R (clear) a S (preset)**
 - dva asynchronní vstupy \bar{S} , \bar{R} jako pozůstatek původního klopného obvodu RS



klopný obvod D se vstupy S a R

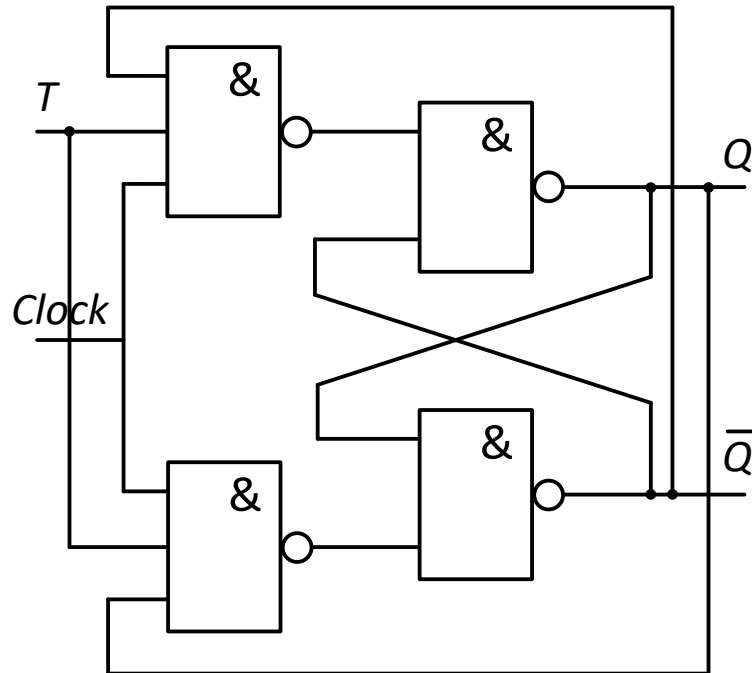


- **vstup D je synchronní** – je synchronizován pomocí hodinového vstupu Clock
- **vstupy S a R jsou asynchronní** – oba jsou **aktivní v logické 0** a nejsou synchronizovány hodinovým vstupem Clock! – **jejich funkce převáží (má prioritu) nad vstupem D!** – proč? podívejte se do pravdivostní tabulky hradla NAND

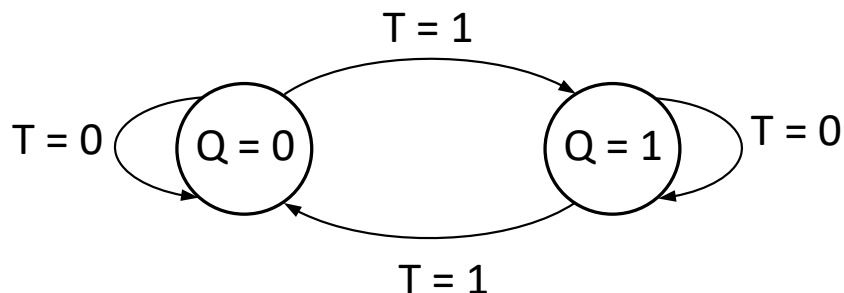
Klopné obvody, sekvenční obvody – RS, JK, D a T klopné obvody

▪ Klopný obvod T

- vychází z klopného obvodu JK, pouze spojíme dohromady oba vstupy $J = K$
- 1 vstup – T , jen transformace paměťová a klopná, **T = toggle** (překlápění)
- klopný obvod T buď jen zaznamenává poslední hodnotu, nebo ji překlápí



<i>Clock</i>	<i>T</i>	<i>Q^t</i>	<i>Q^{t+1}</i>	<i>Transformace</i>
↑	0	0	0	Paměťová (M)
↑	0	1	1	
↑	1	0	1	Klopná (K)
↑	1	1	0	
jiný	x	x	Q^t	-



klopný obvod T značka

