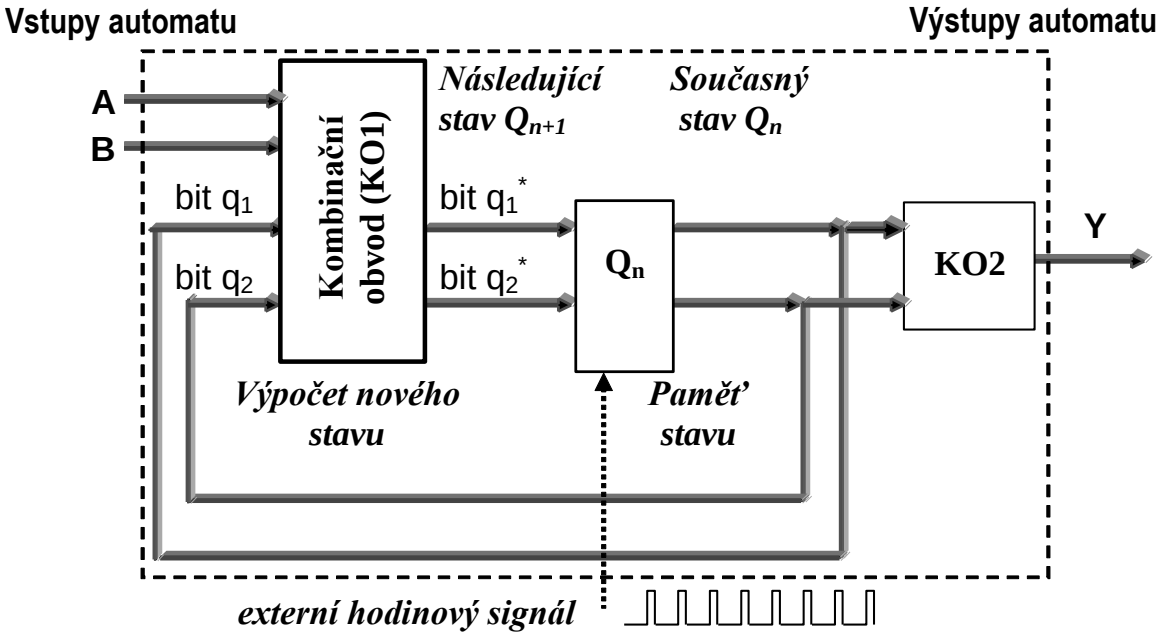
Návrh čítače jako automatu

Obsah

1. Synchronní a asynchronní automat

Automat je zařízení, jehož výstup závisí na okamžitém vstupu a 0 až *n* předchozích vstupech. Jestliže výstup nezávisí na předchozích vstupech jedná se o kombinační síť. Pro uchování vnitřních stavů potřebuje automat paměť. Paměť může být realizována z klopných obvodů RS, D, T, a JK. Protože RS je asynchronní klopný obvod, získáme realizací paměti z RS obvodů asynchronní automat. V ostatních případech se bude jednat o synchronní automat.

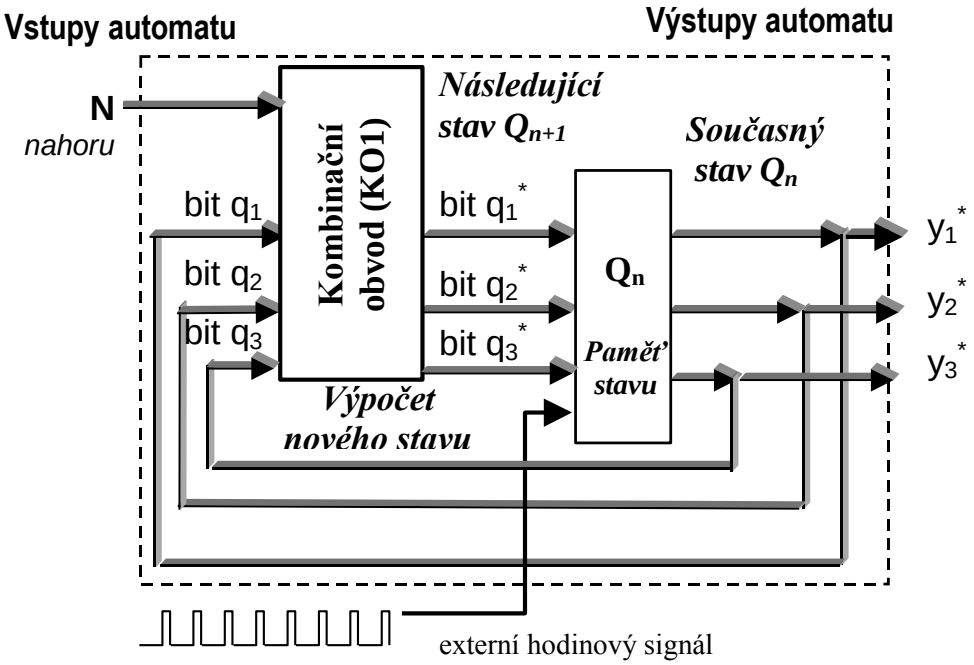
Synchronní automat má kromě vstupů ještě přiveden hodinový signál. Rozdíl mezi *synchronním* a *asynchronním* automatem je v tom, že u synchronního automatu jsou změny vnitřního stavu synchronizovány s hodinami. Výhoda asynchronního automatu je v tom, že je rychlejší. U synchronního automatu zase nedochází k dynamickým hazardům a proto jsou všechny dnešní mikroprocesory synchronní automaty.



Paměť stavu realizovaná pomocí asynchronních obvodů RS má maximálně rychlou odezvu, ale žádá si fundamentální režim činnosti KO1 (na obrázku).

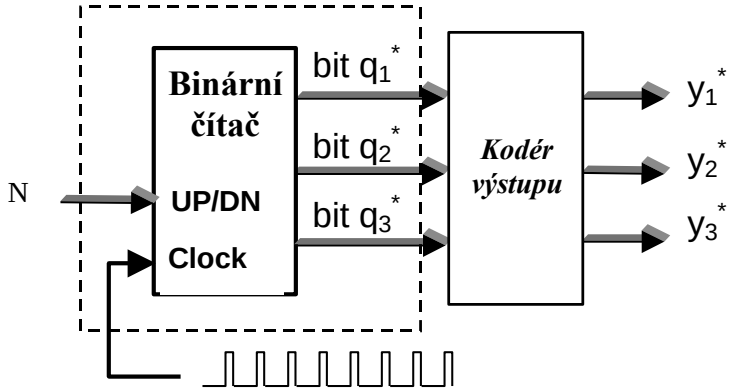
Paměť stavu založená na synchronních klopných obvodech JK či D vyžaduje externí hodinový signál pro periodické vzorkování výstupu KO1, zpravidla o vyšší frekvenci (řádu kHz až MHz). Zpomalí se tím ale reakce automatu na změnu vstupních signálů. **Proč?**

1.a. ***Výstupy automatu mohou být přímo bity paměti stavu***



1.b. ***Mezi vnitřními stavy a výstupem může být nějaký kombinační obvod***

(použijeme například, když chceme zobrazovat výstup na display – obvody mezi výstupem automatu a jednotlivými segmenty jsou vlastně výstupní kombinační obvod)



Lze navrhnout pomocí Karnaughových map, ty zavedl Maurice Karnaugh z Bellových laboratoří v roce 1950), ale překódování výstupu není vždy vhodné kvůli možným hazardům.

1.c. Pravdivostní tabulky klopných obvodů

Asynchronní:

R-S

S	R	Q_t
0	0	Q_{t-1}
0	1	0
1	0	1
1	1	x

$\overline{R}$ - $\overline{S}$ (lze jej sestavit ze 2 hradel NAND)

$\overline{S}$	$\overline{R}$	Q_t
0	0	x
0	1	1
1	0	0
1	1	Q_{t-1}

Synchronní:

J-K

J	K	Q_t
0	0	Q_{t-1}
0	1	0
1	0	1
1	1	$\overline{Q}_{t-1}$

D

D	Q_t
0	0
1	1

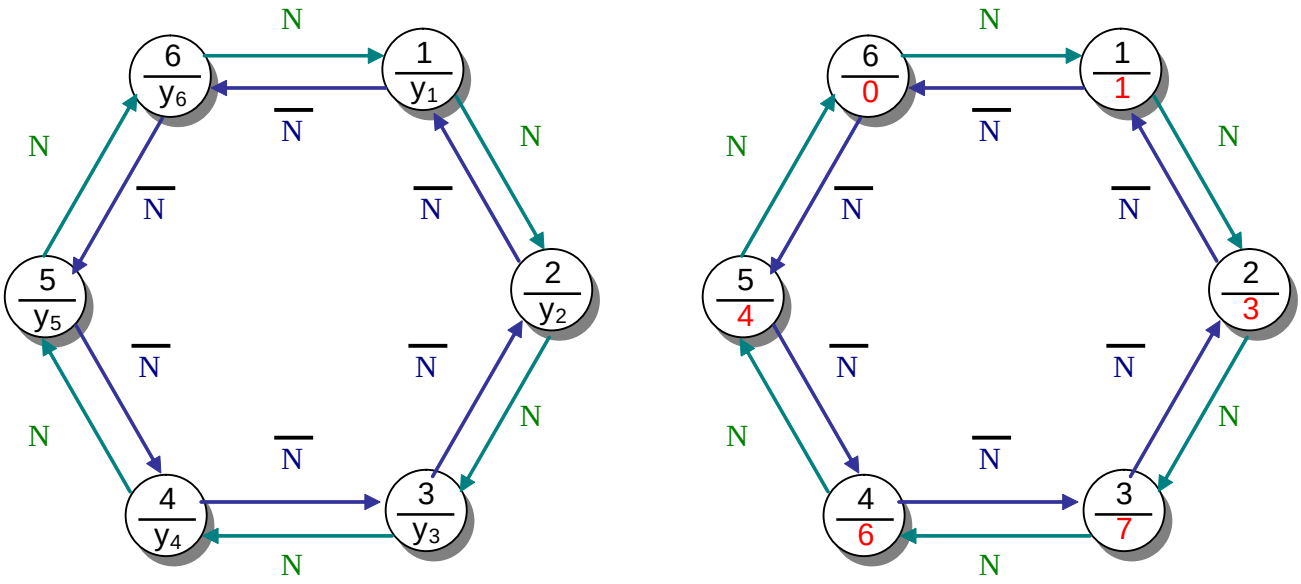
T

T	Q_t
0	1
1	0

2. *Návrh čítače jako automatu pomocí obdodů D*

Čítač je speciální případ jednoduchého synchronního nestabilního automatu, který s každým hodinovým pulsem přechází do dalšího stavu.

Přechodový diagram - orientovaný graf



Přechodová tabulka je jenom jiná forma popisu.

To, jak se změní stav automatu, záleží na hodnotě vstupu *N* v okamžiku příchodu hodinového pulsu.

Současný stav	Následující stav:		Generovaný výstup	
	když N=1	když N=0	Symbol	Hodnota
1	**2**	**6**	y_1	**1**
2	**3**	**1**	y_2	**3**
3	**4**	**2**	y_3	**7**
4	**5**	**3**	y_4	**6**
5	**6**	**4**	y_5	4
6	**1**	**5**	y_6	**0**

Vyjdeme z přechodové tabulky: Sloupce *generovaný výstup* a *výstupní hodnoty* představují kódování (reprezentaci) výstupu.

Současný stav	Následující stav		Generovaný výstup	Výstupní hodnoty
	Když je N=1	Když je N=0		
1	**2**	**6**	y_1	*001*
2	**3**	**1**	y_2	*011*
3	**4**	**2**	y_3	*111*
4	**5**	**3**	y_4	*110*
5	**6**	**4**	y_5	*100*
6	**1**	**5**	y_6	*000*

Pořadová čísla stavů nahradíme jejich binárními kódy – Sloupce *Současný stav* a *kód* představují kódování vnitřních stavů.

Současný stav	Kód $q_3q_2q_1$	Následující stav $q_3q_2q_1$		Hodnota výstupu
		Když je N=1	Když je N=0	
1	001	011	000	*001*
2	011	111	001	*011*
3	111	110	011	*111*
4	110	100	111	*110*
5	100	000	110	*100*
6	000	001	100	*000*

Pro zakódování šesti stavů potřebujeme tři paměťové proměnné – klopné obvody D. Rozepíšeme pravdivostní tabulky pro jejich vstupy $d_3d_2d_1$.

d_3^*

$q_3q_2q_1$	N=1	N=0
001	***0***	***0***
011	***1***	***0***
111	***1***	***0***
110	***1***	***1***
100	***0***	***1***
000	***0***	***1***

d_2^*

$q_3q_2q_1$	N=1	N=0
001	***1***	***0***
011	***1***	***0***
111	***1***	***1***
110	***0***	***1***
100	***0***	***1***
000	***0***	***0***

d_1^*

$q_3q_2q_1$	N=1	N=0
001	***1***	***0***
011	***1***	***1***
111	***0***	***1***
110	***0***	***1***
100	***0***	***0***
000	***1***	***0***

Navrhneme kombinační síť pro $d_3d_2d_1$, čili napíšeme Karnaughovy mapy a sestavíme rovnice.

d_3^*

N \ q_1q_2	00	01	11	10
q_3	1	X	0	0
$\bar{q}_3$	1	1	0	X
$\bar{q}_3$	0	1	1	X
q_3	0	X	1	0

d_2^*

N \ q_1q_2	00	01	11	10
q_3	0	X	0	0
$\bar{q}_3$	1	1	1	X
$\bar{q}_3$	0	0	1	X
q_3	0	X	1	1

d_1^*

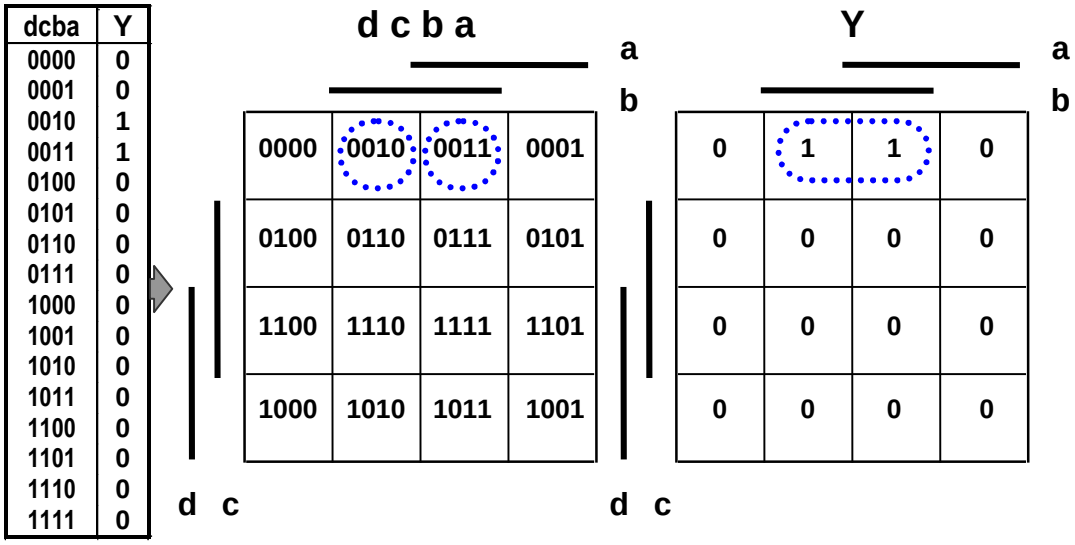
N \ q_1q_2	00	01	11	10
q_3	0	X	1	0
$\bar{q}_3$	0	1	1	X
$\bar{q}_3$	0	0	0	X
q_3	1	X	1	1

$$d_3 = \bar{q}_1.\bar{N} + q_2.N \qquad d_2 = q_1.N + q_3.\bar{N} \qquad d_1 = q_2.\bar{N} + \bar{q}_3.N$$

2.a. *Opakování: Karnaughovy mapy*

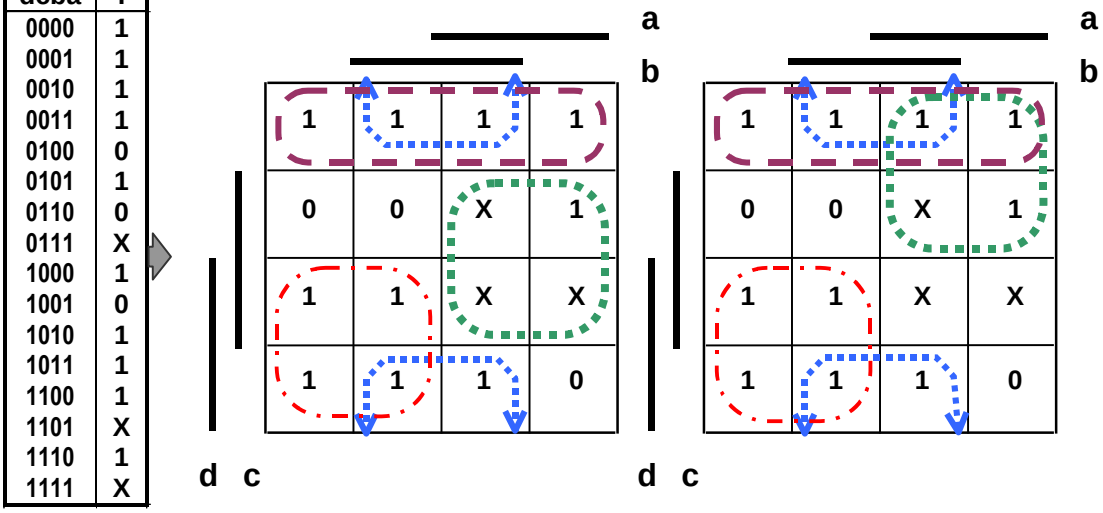
Princip zápisu do mapy a minimalizace:

dcba	Y
0000	0
0001	0
0010	1
0011	1
0100	0
0101	0
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0



$$F = \bar{d}.\bar{c}.b.a + \bar{d}.\bar{c}.b.\bar{a} = \bar{d}.\bar{c}.b.(a + \bar{a}) = \bar{d}.\bar{c}.b$$

dcba	Y
0000	1
0001	1
0010	1
0011	1
0100	0
0101	1
0110	0
0111	X
1000	1
1001	0
1010	1
1011	1
1100	1
1101	X
1110	1
1111	X



$$F_{levá}(x) = \bar{c}.\bar{d} + b.\bar{c} + a.c + \bar{a}.d \qquad F_{pravá}(x) = \bar{c}.\bar{d} + b.\bar{c} + a.\bar{d} + \bar{a}.d$$

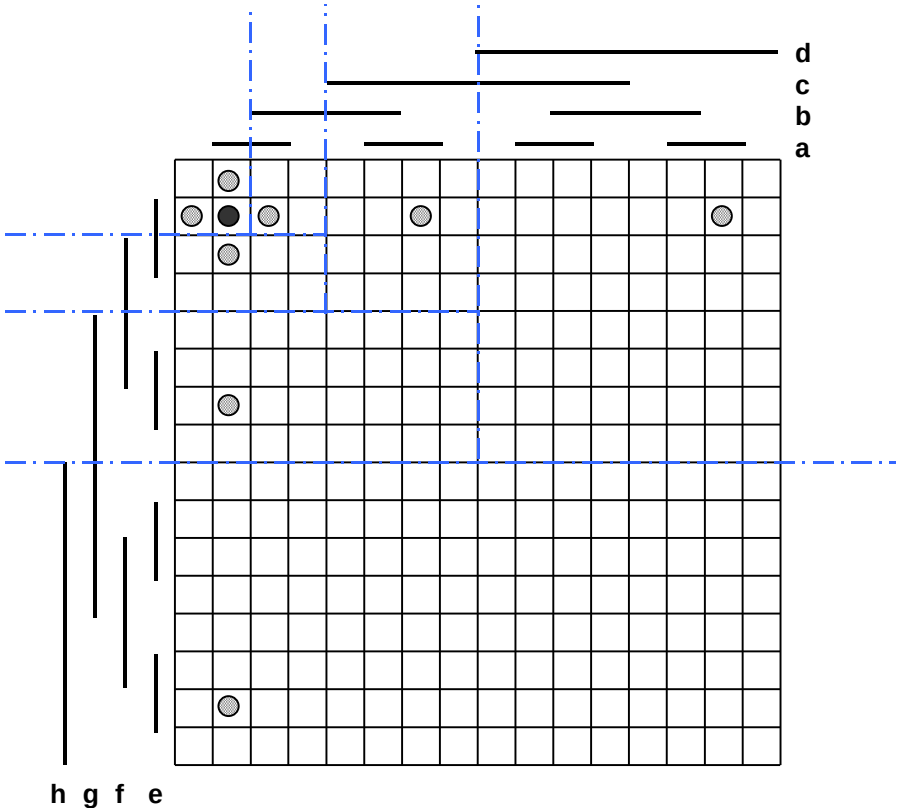
$$F_{nuly}(x) = (a + \bar{c} + d).(\bar{a} + b + \bar{d})$$

$$F_{levá}(x) = F_{pravá}(x) = F_{nuly}(x);\ [\forall x; F_{definice}(x) = 0 \vee 1]$$

Různé možnosti minimálního pokrytí nabývají shodných hodnot pro výstupy definované do 0 a 1. Mohou se ale lišit v bodech X a rovněž způsobem fyzické realizace. Neurčitý výstup X (možnost volby 0 anebo 1) se vztahuje výhradně k okamžiku návrhu a po něm má fixní hodnotu 0 anebo 1.

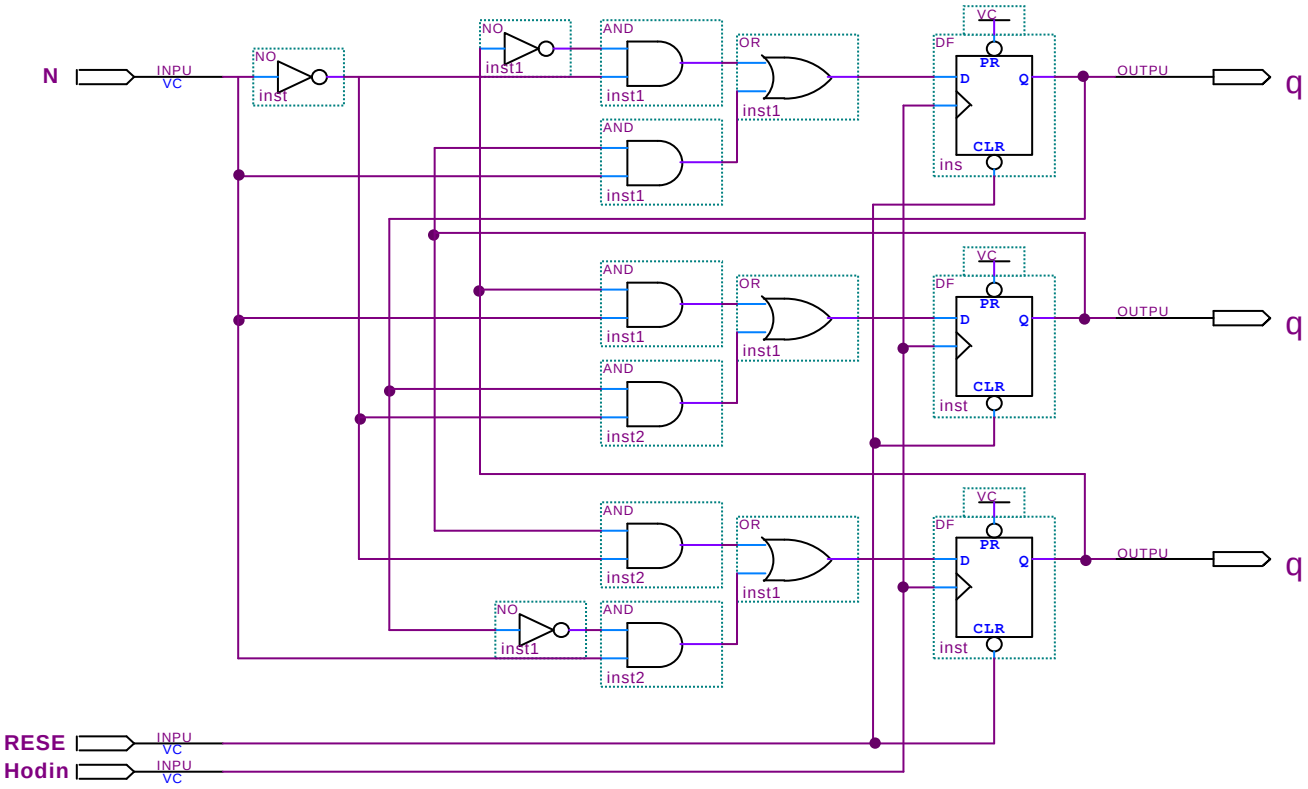
Karnaughova mapa pro 8 proměnných

→ jeden prvek má 8 sousedů

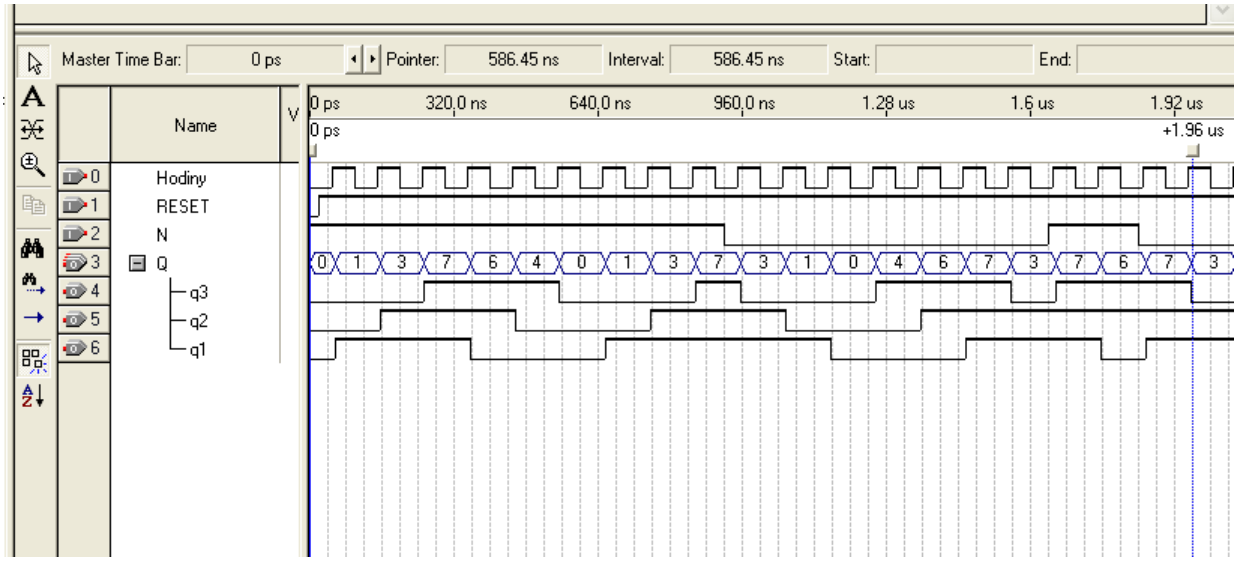


Dělení mapy na menší mapy a sousedé jednoho prvku.

2.b. Schéma v prostředí Quartus



2.c. Emulace v prostředí Quartus

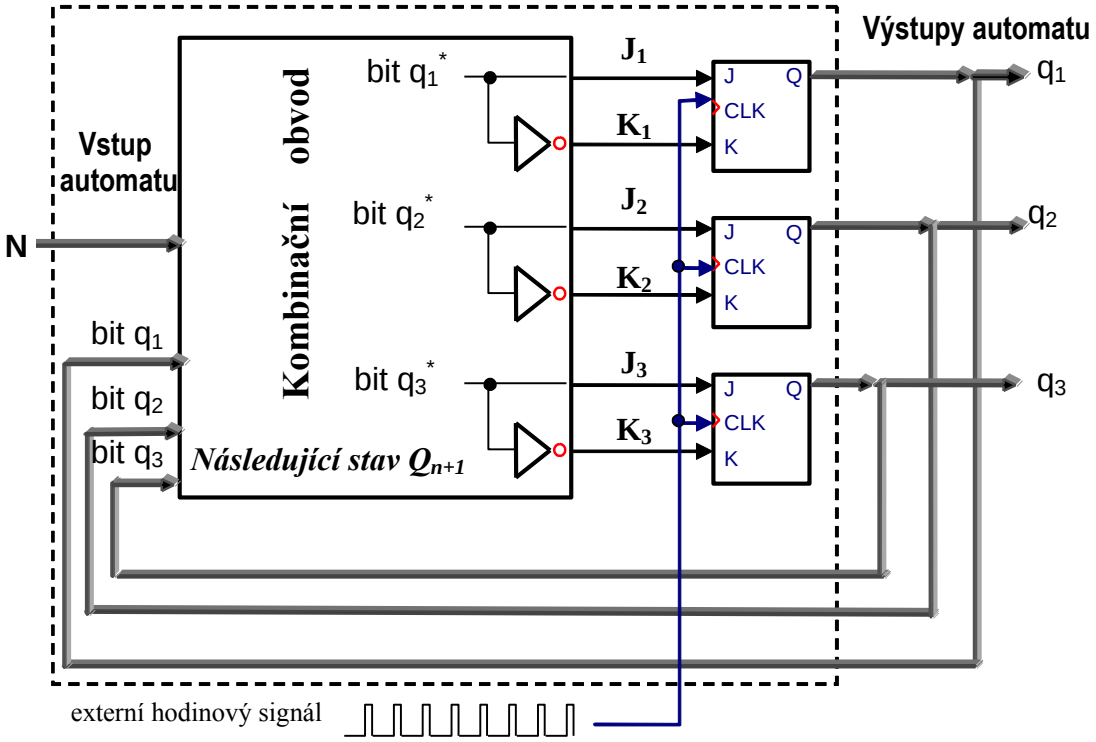


Simulace není rozhodně samospasitelná! Nefunguje-li, pak zpravidla nefunguje ani zapojený obvod, avšak **opačná implikace rozhodně neplatí.** Mnohé emulátory nepoznají některé záludnosti.

2.d. Návrh pro J-K klopné obvody

Předchozí rovnice platily pouze pro D klopný obvod, který má jediný datový vstup, ale J-K nebo S-R klopné obvody se ovládají dvěmi vstupy. V této úloze vytvoříme J-K vstupy negacemi. Synchronizace hodinovým signálem odstraňuje také vliv možných hazardů (řekněte proč!), takže není potřeba brát je v úvahu při návrhu logických funkcí.

$$D_3 = \begin{matrix} J_3 = \overline{q}_1.\overline{N} + q_2.N \\ K_3 = \overline{J}_3 \end{matrix} \qquad D_2 = \begin{matrix} J_2 = q_1.N + q_3.\overline{N} \\ K_2 = \overline{J}_2 \end{matrix} \qquad D_1 = \begin{matrix} J_1 = q_2.\overline{N} + \overline{q}_3.N \\ K_1 = \overline{J}_1 \end{matrix}$$



Existuje i jiný způsob, návrh pomocí pokrytí tlustých 0 a 1, který dává (někdy) úspornější řešení, jde však již o složitější řešení, které má v moderních FPGA obvodech minoritní postavení, jelikož tam se používají obvody typu D..

3. Překódování binárního výstupu

Předpokládejme, že máme externí binární signál, bity *c,b,a*, udávající polohu nějakého přepínače, a chceme jeho výstup překódovat na $y_3y_2y_1$ podle tabulky:

Stav	c b a	$y_3y_2y_1$
1	000	001
2	001	011
3	010	111
4	011	110
5	100	100
6	101	000

→

$q_3q_2q_1$	y_3
000	0
001	0
010	1
011	1
100	1
101	0

$q_3q_2q_1$	y_2
000	0
001	1
010	1
011	1
100	0
101	0

$q_3q_2q_1$	y_1
000	1
001	1
010	1
011	0
100	0
101	0

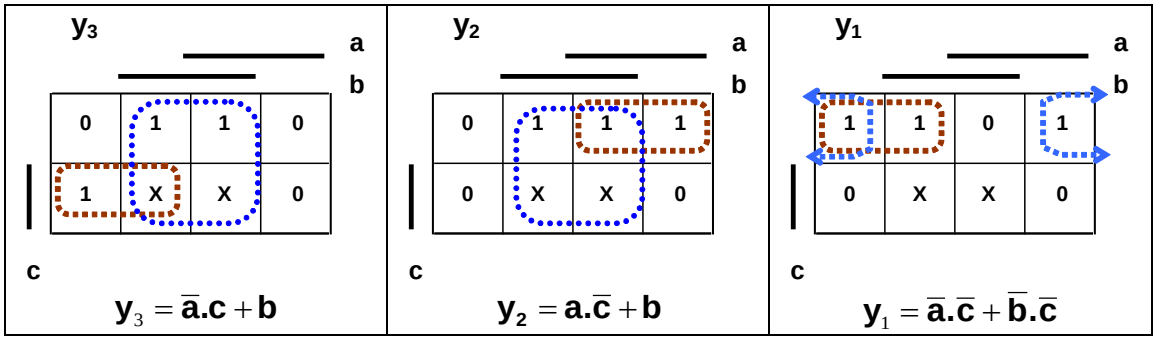


Schéma zapojení pro průmyslový automat LOGO – řešíme jako pouhý kombinační obvod