

## Vzorové riešenie 3. zadania

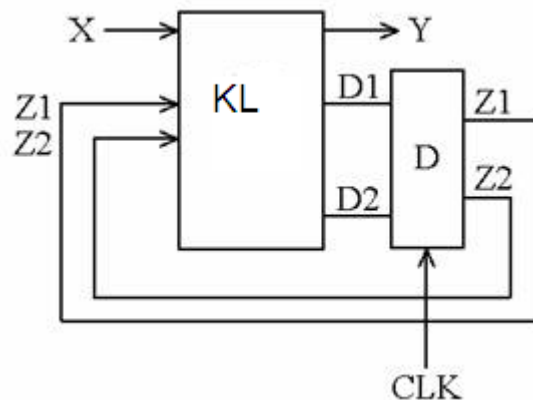
### SYNTÉZA SEKVENČNÝCH LOGICKÝCH OBVODOV

Navrhnete synchronný sekvenčný obvod so vstupom  $x$  a výstupom  $y$  s nasledujúcim správaním: na výstupe  $Y$  bude 1 vždy vtedy, ak sa (zo začiatočného stavu) vo vstupnej postupnosti vyskytne postupnosť **100101** (postupnosti sa môžu prekryvať, v tomto prípade 1010101 je možné chápať ako dve postupnosti). Vlastné riešenie overte progr. prostriedkami ESPRESSO a LogiSim (príp. LOG alebo FitBoard).

Úlohy:

- 1) V pamäťovej časti použite minimálny počet preklápacích obvodov **JK-PO**.
- 2) Navrhnuté B-funkcie v tvare MDNF overte programom pre ESPRESSO. Pri návrhu B-funkcií kladte dôraz na skupinovú minimalizáciu funkcií.
- 3) Optimálne riešenie (treba zhodnotiť, ktoré riešenie je lepšie a prečo) vytvorte obvod s členmi NAND (výhradne NAND, t.j. ani žiadne NOT).
- 4) Výslednú schému nakreslite v simulátore LogiSim (príp. LOG alebo FitBoard) a overte simuláciou.
- 5) Riešenie vyhodnoťte (zhodnotenie zadania, postup riešenia, vyjadrenie sa k počtu logických členov).

*Upozornenie toto je príklad pre 5 číselnú postupnosť, vy máte 6 číselnú postupnosť.  
Vytvárame obvod, ktorý má nasledovnú všeobecnú schému:*



- $X$  – vstupná premenná, môže ich byť viacero.
- $Y$  – výstupná premenná, tiež ich môže byť viacero.
- $Z1, Z2, \dots$  – stavové premenné, pomocou ktorých sú kódované jednotlivé stavy.
- $D1, D2, \dots$  – budiace funkcie.
- $KL$  – kombinačná logika (zjednodušene povedané, toto ste robili na 2. zadaní).
- $D$  – preklápacie obvody. Pamäťová časť obvodu, vďaka nej obvod vie v akom stave sa nachádza.
- $CLK$  – hodinový signál, ktorý synchronizuje preklápacie obvody (a “posúva” obvod do nasledujúceho stavu).

## Riešenie

Zadaná postupnosť: **100101**

**Poznámka:**

A) Výstupná funkcia pri automate typu Moore je závislá len od aktuálneho stavu. Pri vytváraní prechodovej tabuľky pre automat je, v prípade typu Moore, len jeden stĺpec pre každý výstup (pretože hodnota závisí len od aktuálneho stavu). Pri automate typu Mealy má každá výstupná premenná taký počet prislúchajúcich stĺpcov, ktorý pokryje všetky možnosti hodnôt vstupných premenných (pretože výstupná hodnota závisí od aktuálneho stavu a hodnôt vstupných premenných).

B) Výstupná funkcia automatu typu Mealy závisí od aktuálneho stavu a od hodnoty vstupných premenných. Pri zakresľovaní prechodového grafu sa pri automate Mealy zvykne uvádzať aj hodnota výstupu na hranách (viď. ukážka).

Dôsledkom je, že automat typu Moore poskytuje výsledok o jeden stav (takt) neskôr ako automat typu Mealy. Prechod do ďalšieho stavu je aj pre automat typu Moore a Mealy závislý od aktuálneho stavu a od hodnoty vstupných premenných.

**Zadanie treba realizovať pomocou Moore alebo Mealy (výber je na vás).**

Prechodová tabuľka pre automat typu Moore

stav	Nový stav		Y	Čo je splnené?
	x=0	x=1		
S1	S1	S2	0	Nič
S2	S3	S2	0	"1"
S3	S4	S2	0	"10"
S4	S1	S5	0	"100"
S5	S6	S2	0	"1001"
S6	S4	S7	0	"10010"
S7	S1	S2	1	"100101"

**Poznámka:** Vysvetlenie prechodov medzi jednotlivými stavmi:

S0: začíname postupnosť a zatiaľ nemáme vôbec nič splnené. Ak nám príde iný vstup ako čakáme, tak stále nič nemáme a preto zostávame v tomto stave (S0). Ak nám príde prvá hodnota postupnosti („1“), tak sa posúvame do ďalšieho stavu (S1).

S1: V tomto stave máme prvú hodnotu postupnosti, v našom prípade „1“ (ešte budeme potrebovať 0101). Ak bude na vstupe x=0, tak by sme získali postupnosť „10“, čo je podčasť, ktorú potrebujeme, takže postupujeme do ďalšieho stavu (S2). Ak príde hodnota 1, tak nie je to hodnota, ktorá je potrebná pre správnu postupnosť (teraz máme „11“). Teda je potrebné pozrieť sa, do ktorého stavu sa môžeme vrátiť (je potrebné hľadať stav čo najbližšie k aktuálnemu, vrátane aktuálneho). V stave S1 vieme, že máme spracovanú jednu hodnotu postupnosti („1“), tak si necháme len poslednú „1“. Vidíme, že je totožná s tým, čo máme mať v stave S1, preto doplníme v S1, kde x=1 S1 (opäť budeme mať načítanú prvú hodnotu postupnosti a čakáme na ďalšie hodnoty).

S2: V tomto stave už máme verifikovanú postupnosť „10“. Ak je x=1, tak by sme mali postupnosť „101“, čo je to čo hľadáme zo vstupnej postupnosti, teda ideme do stavu S3. Ak nám príde x=0, tak by sme mali „100“, čo je pre nás nepoužiteľné, z hľadiska kontroly postupnosti. Keď si vezmeme len „00“ (najnovšie dve hodnoty, tak si overíme, či môžeme ísť do predchádzajúceho

stavu), tak vidíme, že je to rozdielne s tým čo máme v S2 („10“), teda nepôjdeme do tohto stavu. Keď opäť skrátime načítanú postupnosť na „0“ (poslednú načítanú hodnotu), keď ju porovnáme s tým, čo vieme v stave S1 („1“), opäť sa nám nezhodujú hodnoty a presúvame sa už do stavu S0 (t.j. žiadnu podčasť načítanej postupnosti nevieme použiť a musíme začať od začiatku).

S3: V tomto stave máme „101“, ak je  $x=0$ , tak získame postupnosť „1010“ a môžeme sa presunúť do ďalšieho stavu S4. Ak máme  $x=1$ , tak by sme získali postupnosť „1011“, čo nie je správne. Opäť skrátime postupnosť o jej najstaršiu hodnotu a získame „011“, to sa nerovná hodnote v S3 („101“). Znovu skrátime postupnosť na „11“, čo sa znovu nerovná hodnote S2 („10“). Nezostáva nám nič iné a skrátime postupnosť na „1“, čo vidíme, že je totožné s S1 („1“), takže sa môžeme presunúť do tohto stavu.

S4: V tomto stave máme „1010“, ak je  $x=1$ , tak získame postupnosť „10101“ a môžeme sa presunúť do posledného stavu S5 (v ňom máme načítanú celú postupnosť). Ak je  $x=0$ , tak získame „10100“. Ako v predchádzajúcich stavoch začneme odoberať najstaršie hodnoty a porovnávať so stavmi, kde sa vieme presunúť:  $0100 \rightarrow S4$ ,  $100 \rightarrow S3$ ,  $00 \rightarrow S2$ ,  $0 \rightarrow S1$ , teda nevieme použiť žiadnu časť postupnosti a musíme sa presunúť na začiatok (S0).

Pre automat Mealy nastáva v tomto stave zmena, keďže máme v S4=“1010“ a  $x=1$ , tak máme splnenú postupnosť a je pre nás zbytočné používať stav, v ktorom by sme si pamätali dlhšiu postupnosť (keďže túto už máme celú načítanú). Pozeráme preto, do ktorého stavu sa vieme vrátiť, opäť pomocou skracovania postupnosti:  $0101 \rightarrow S4$ ,  $101 = S3$ , teda presúvať sa budeme do S3.

S5: potrebný len pre Moore. Postupnosť máme načítanú a zisťujeme, čo teraz. Ak je  $x=1$ , tak máme „101011“ a skracujeme:  $01011 \rightarrow S5$ ,  $1011 \rightarrow S4$ ,  $011 \rightarrow S3$ ,  $11 \rightarrow S2$ ,  $1 = S1$ , teda v tomto prípade pôjdeme do S1. Ak je  $x=0$ , tak získame „101010“ a skracujeme:  $01010 \rightarrow S5$ ,  $1010 = S4$ , teda pôjdeme do stavu S4. Opäť stav S6 by bol zbytočný, lebo si nepotrebujeme pamätať tak dlhú postupnosť.

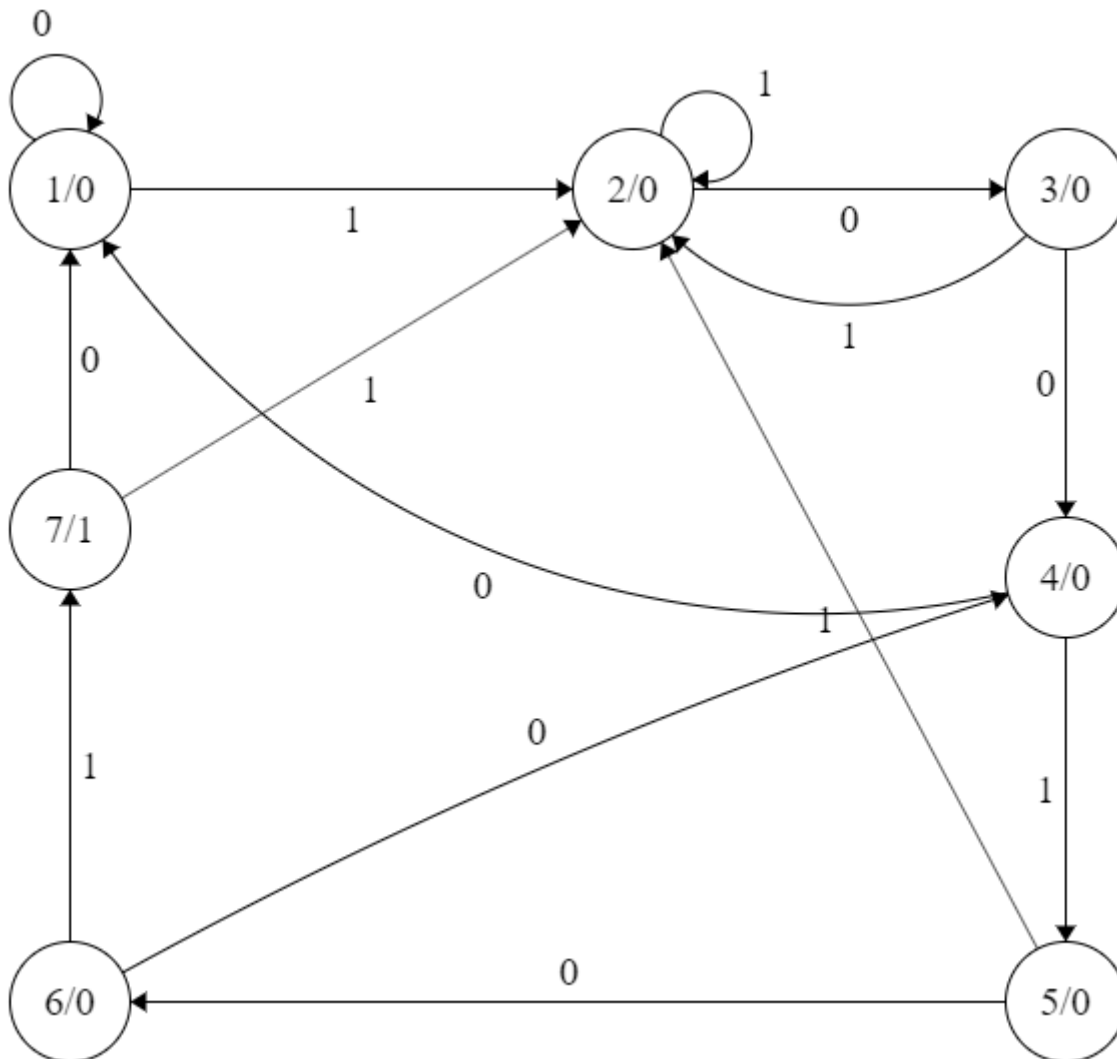
Doplnenie výstupnej hodnoty:

Moore: z definície vyplýva, že výstupné hodnoty závisia len od aktuálneho stavu. Pozrieme sa, v ktorom stave splníme celú zadanú postupnosť (S5=10101) a v ňom upravíme výstupnú hodnotu na „1“, všade inde doplníme hodnotu „0“.

Mealy: z definície vyplýva, že výstupné hodnoty závisia len od aktuálneho stavu a hodnoty vstupných premenných. Preto pozeráme kombináciu postupnosť (ktorú v danom stave už máme spracovanú) a hodnotu  $x$ . Vidíme, že celá postupnosť je načítaná v stave S4 a  $x=1$ , teda pre  $Y$  v S4 a  $x=1$  zapíšeme „1“, keďže v žiadnom inom prípade nie je splnená postupnosť, tak inde doplníme hodnotu „0“.

Zostrojíme prechodový graf stavového automat typu Moore/Mealy

Prechodový graf typu Moore (hodnota hrany reprezentuje hodnotu vstupnej premennej):



### Kódovanie stavov

*Poznámka: Zvolenie kódovania stavov má vplyv na veľkosť kombinačnej logiky. Stavy kódujeme pomocou stavových premenných. Platí, že  $\text{počet\_stavov} \leq 2^{\text{počet\_stavových\_premenných}}$ . V tomto prípade máme 5 stavov, teda potrebujeme 3 stavové premenné. Na prednáške budete mať vysvetlený optimálny spôsob kódovania stavov, môžete ho použiť. Tu je zvolené náhodné kódovanie, ktoré nie je podľa tých odporúčaní.*

		z3	
		z2	
z1	S0	S2	S3
	S4	X	X
	S1		
	X		

Stav	z <sub>1</sub> z <sub>2</sub> z <sub>3</sub>
S1	000
S2	001
S3	010
S4	100
S5	011
S6	101
S7	110

Prechodová tabuľka pre automat Moore po dosadení zakódovaných stavov

stav	Nový stav		Y	
	x=0	x=1	x=0	x=1
000	000	001	0	0
001	010	001	0	0
010	100	001	0	0
100	000	011	0	0
011	101	001	0	0
101	100	110	0	0
110	000	001	0	1

### Budiace funkcie pre D preklápacie obvody (D-PO) a výstupná funkcia

*Poznámka: Budiace funkcie nám pomáhajú určiť do akého stavu sa presunieme. Keďže nasledujúci stav (aj pre Mealy aj pre Moore) závisí od aktuálneho stavu (stav určujú stavové premenné) a vstupnej premennej, tak tieto hodnoty budeme potrebovať do Karnaughových máp jednotlivých budiacich funkcií.*

*Postup vyplňania pre zlučenie Karnaughovu mapu D1, D2 a D3: Nájdeme pozíciu pre stav S0 (z<sub>1</sub>=0, z<sub>2</sub>=0, z<sub>3</sub>=0), nájdeme dve políčka, ktoré sa líšia len podľa hodnoty vstupu (premennej x). Po pozretí do prechodovej tabuľky zistíme, že keď je x=0, tak z S0 ideme do S0 (000), teda do políčka v Karnaughovej mape, kde je S0 a x=0 napíšeme „000“ (S0 – stavu kam ideme), podobným spôsobom doplníme do druhého políčka „001“ (S1). Podobným spôsobom doplníme hodnoty pre funkciu Y, len si všimame stĺpce v prechodovej tabuľke, ktoré prislúchajú k funkcii Y.*

		z <sub>3</sub>			
		z <sub>2</sub>			
x	z <sub>1</sub>	000	100	101	010
	0	000	000	XXX	100
	1	011	001	XXX	110
	1	001	001	001	001

D1,D2,D3

		z <sub>3</sub>			
		z <sub>2</sub>			
x	z <sub>1</sub>	0	1	1	0
	0	0	0	X	1
	0	0	0	X	1
	0	0	0	0	0

**D1**

		<u>z2</u>		<u>z3</u>
X	z1	0	0	0
		0	0	X
		1	0	X
		0	0	0

**D2**

		<u>z2</u>		<u>z3</u>
X	z1	0	0	1
		0	0	X
		1	1	X
		1	1	1

**D3**

		<u>z2</u>		<u>z3</u>
X	z1	0	0	0
		0	1	X
		0	1	X
		0	0	0

$$Y = Z1 * Z2 = (Z1 \uparrow Z2) \uparrow$$

### Budiace funkcie pre JK preklápacie obvody (JK-PO)

Pri zostavení budiacich funkcií sa vychádza z nižšie uvedenej tabuľky. Ku každej stavovej premennej z D preklápacích obvodov korešponduje stavová premenná pre JK preklápacie obvody (z z D-PO, Z z JK-PO).

Vysvetlenie dopĺňania hodnôt do Karnaughovej mapy (KM): pre budiacu funkciu J1 prislúcha stavová premenná Z1 (z1 z D-PO). Tam, kde je KM pokrytá Z1=1, tam doplníme hodnotu „X“. V miestach, kde je KM pokrytá Z1=0, tak tam prepíšeme hodnotu z KM pre D1. Pre budiacu funkciu K1 postupujeme analogicky podľa tabuľky, t.j. Z1=0, tam doplníme do KM hodnotu „X“. Pozor (!), pri Z1=1 prepisujeme z KM pre D1 invertovanú hodnotu (ak bola v KM pre D1 „0“, tak do KM pre K1 napíšeme „1“). Podobným spôsobom postupujeme s budiacimi funkciami J2, K2, J3, K3, ale už si všímame Z2, resp. Z3 (sú prepojené číselnými indexami).

z->Z	J	K
0->0	0	X
0->1	1	X
1-> <u>0</u>	X	<u>1</u>
1-> <u>1</u>	X	<u>0</u>

		Z2		Z3
X	Z1	0	1	0
		X	X	X
		X	X	X
		0	0	0

$J1 = \overline{X} * Z2 = ((X \uparrow) \uparrow Z2) \uparrow$

		Z2		Z3
X	Z1	X	X	X
		1	1	0
		1	1	0
		X	X	X

$K1 = Z1 * \overline{Z3} = (Z1 \uparrow (Z3 \uparrow)) \uparrow$

		Z2		Z3
X	Z1	0	X	X
		0	X	0
		1	X	1
		0	X	0

$J2 = X * Z1 + \overline{X} * \overline{Z1} * Z3 = (X \uparrow Z1) \uparrow ((\uparrow X) \uparrow (Z1 \uparrow) \uparrow Z3))$

		Z2		Z3
X	Z1	X	1	X
		X	1	X
		X	1	X
		X	1	X

$K2 = 1$

		Z2		Z3
X	Z1	0	0	X
		0	0	X
		1	1	X
		1	1	X

$J3 = X * \overline{Z3} = (X \uparrow (Z3 \uparrow)) \uparrow$

			Z2	Z3	
		X	X	0	1
	Z1	X	X	X	1
X		X	X	X	1
		X	X	0	0

$$K3 = Z3 * Z1 + \bar{X} * \bar{Z1} * \bar{Z2} = (Z3 \uparrow Z1) \uparrow ((X \uparrow) \uparrow (Z1 \uparrow) \uparrow (Z2 \uparrow))$$

## Espresso

Poznámka: rovnakým spôsobom ako pri 2. zadaní (vstup a výstup vyzerá rovnako ako pri 2. zadaní). Riešenia sú totožné. Môže byť aj: sú ekvivalentné (obvod je rovnako veľký, má rovnaký počet vstupov)/ mnou navrhnuté riešenie je lepšie pretože .../mnou navrhnuté riešenie bolo horšie pretože...

espresso

j1 = (!x&z2);

k1 = (!z3);

j2 = (!x&!z1&z3) | (x&z1);

k2 = (!x&z2) | (x);

j3 = (x);

k3 = (!x&!z2) | (x&z1);

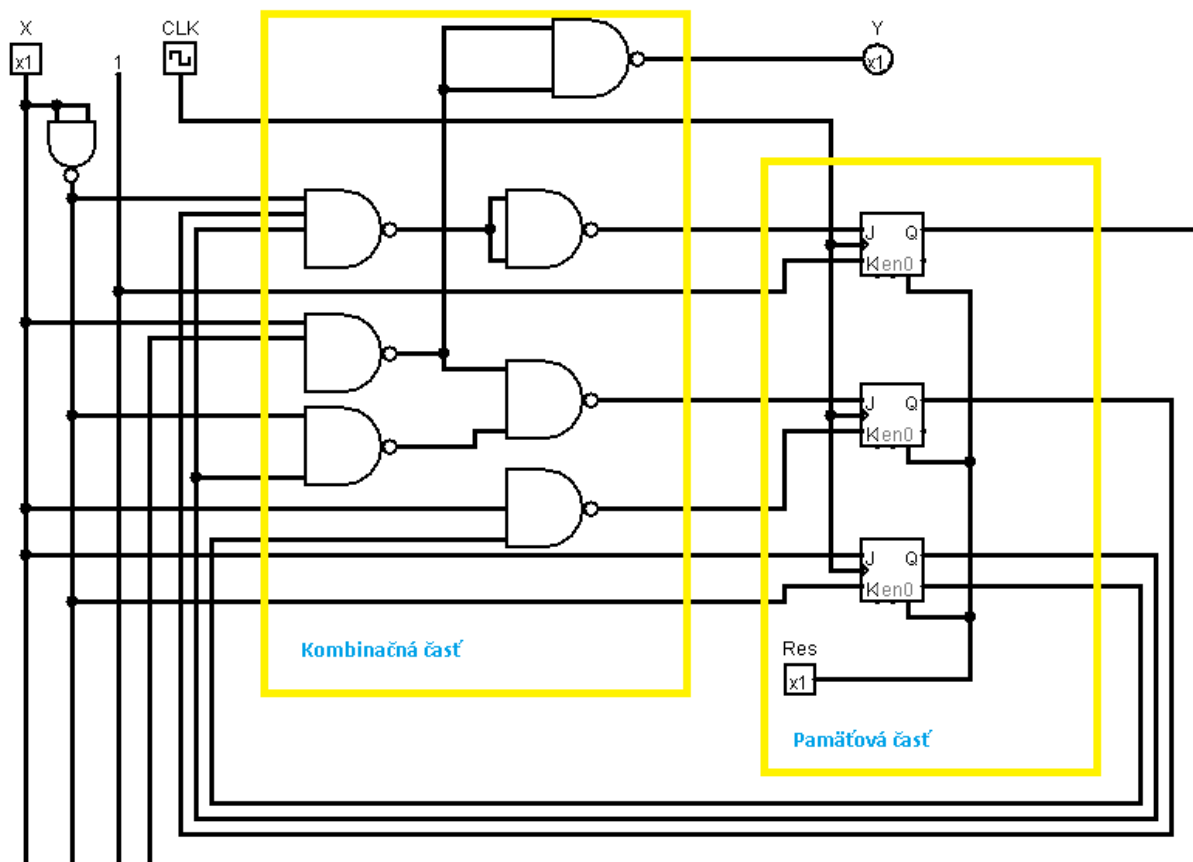
Mnou navrhnuté riešenie bolo horšie pretože pozíva viac logických funkcií aj logických vstupov.

Vyjadrenie k počtu logických členov obvodu: 14 členov NAND (Pozor na rozvetvenia, napr. X.Z1, zároveň nie je potrebné počítat' NAND na negovanie stavových premenných, keďže aj ich negovanú hodnotu nám generuje preklápací obvod). a 3 preklápacie obvody JK (každá stavová premenná potrebuje jeden).

Vyjadrenie k počtu vstupov do logických členov obvodu: 30 (17 v kombinačnej časti a 12 v pamäťovej časti). Je potrebné pripočítat' aj hodinový a resetovací signál na každý preklápací obvod JK (t.j.  $2 \cdot 3 = 6$  v tomto prípade).

Schéma:





*Poznámka: Otestujte si túto schému pre správne zobrazenie postupnosti 10101. Prepínané stavy vidíte v JK preklápacích obvodoch, keď si ich spojíte, tak získate stav, v ktorom sa nachádzate. Pre prejdienie celej periódy je potrebné dvakrát kliknúť na CLK (aby ste urobili aj nábežnú aj dobežnú hranu hodinového signálu).*

## Zhodnotenie

*Stručne popísať zadanie úlohy postup riešenia a spôsob overenia riešenia, vyjadrenie k funkčnosti obvodu.*

Dostal som za úlohu zostrojiť obvod kde y bude 1 len v prípade že užívateľ zadá postupnosť 100101. Inak svieti 1. Pomocou karnafovych map som si vyjadril obvody ktoré som následne vyskúsal v logisime tie fungovali ako mali. S toho som usúdil že moj obvod je správny.

Na miesto odovzdania sa odovzdáva

- dokument (vo formáte docx alebo pdf),
- vstupný súbor pre Espresso (postačuje napísať v hlavnej dokumentácii),
- výstupný súbor z Espresso (postačuje napísať v hlavnej dokumentácii),

- súbor s obvodom pre overenie riešenia simuláciou.

### Upozornenie

Odovzdaný dokument musí obsahovať **len** nasledovné informácie:

- identifikáciu autora riešenia,
- nadpis,
- text zadania,
- zadanú postupnosť,
- prechodovú tabuľku a graf typu Mealy alebo Moore,
- Karnaughove mapy pre budiace funkcie D-PO, Y,
- Karnaughove mapy pre budiace funkcie JK-PO,
- DNF formu pre budiace funkcie JK-PO a Y,
- celý postup úpravy na Shefferovú funkciu s použitím  $\uparrow$  (NAND),
- zhodnotenie.