

SALMON 2

NÁVOD

VÝKONĚJŠÍ SSO
ANEB
DOKÁŽEME ZAPOJIT HRU?

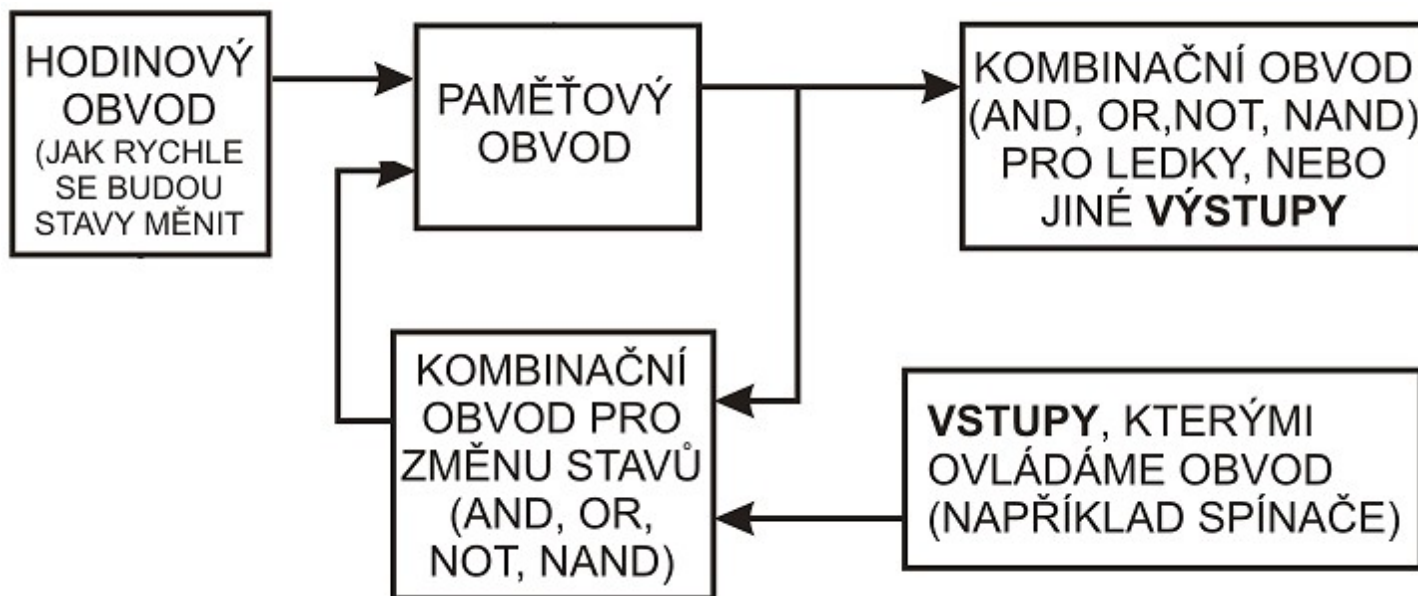
Úvod

V návodu pro modul „Saimon 2 – logické funkce“ budu pokračovat v látce posledního dílu návodu pro modul Saimon 1. Naučíme se dělat výkonnější SSO. Výkonnější budou tak, že budou mít lepší „program“, který je uvnitř kombinačního obvodu pro změnu stavů SSO. Toho dosáhneme díky logickým funkcím na modulu Saimon 2, které tento „program“ tvoří. Ukážu vám, že s takovýmto vybavením už budeme umět zapojit i jednoduchou hru.

Ještě chci uvést nabídku pro ty, kteří se zrovna nechtějí nic učit, že lze přeskočit až na poslední kapitoly. Hry jsou uvedené i se zapojením drátků. Můžete si hry zapojit a zjistit „co obvod dělá“ a k učení se vrátit později.

Kapitola 1 – Počítadlo od 0 to 7

Budeme postupovat po stejných krocích, jako v návodu pro Saimon 1. Nejdříve připomenu, jak vypadá konečný automat a synchronní sekvenční obvod. Konečný automat je pro naše účely systém, který má číslované stavy a ty mění mezi sebou. Obsahuje hodinový obvod, který určuje, jak rychle se stavy mění. Paměťový obvod, jehož hodnoty výstupů znamenají aktuální stav. Kombinační obvod pro změnu vstupů – obsahuje logické funkce, jejichž výstupy jsou připojeny na vstupy paměťového obvodu. Vstupy logických funkcí jsou výstupy paměťového obvodu a nově i spínače. SSO má i výstupní část – ledky nebo číslicovku. Pro tuto zobrazovací část potřebujeme také kombinační obvod, který ji z aktuálních hodnot výstupů paměťového obvodu nastaví (z hodnoty aktuálního stavu SSO). Více viz obrázek 1.1.



Obrázek 1.1 – Blokové schéma SSO

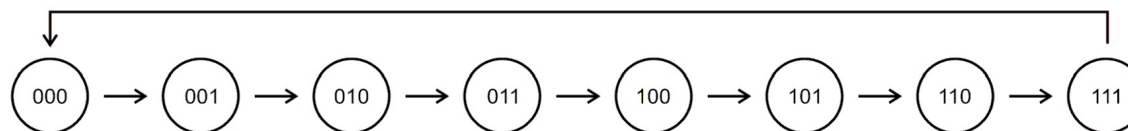
Počet stavů určíme my, i to jak se budou měnit. Posloupnost zobrazujeme v grafu. Potřebujeme vytvořit kombinační obvod pro změnu stavů. Z grafu vyčteme pravdivostní tabulky logických funkcí. Pro ty pak sestavíme obvod. Vytvoříme počítadlo a postupovat budeme po známých krocích.

Krok 1 – Slovní popis funkce obvodu

Vytváříme obvod, který bude na číslicovce postupně zobrazovat čísla 0,1,2,3,4,5,6 a 7. Stavů je tedy celkem 8. Každý stav zobrazí danou číslici.

Krok 2 – nakreslíme si, jak jdou stavy po sobě

Následuje graf přechodů SSO. Máme dvě možnosti – buď stavy očíslováme podle pravidla pro jednoduchý kombinační obvod – binární zápis jejich čísla se bude lišit jen o 1 bit a pak vytvoříme ještě kombinační obvod pro číslicovku, který pro daný stav zobrazí požadovanou číslici. Nebo stavy očíslováme v pořadí čísla, která se budou na číslicovce objevovat, kombinační obvod pro číslicovku vynechat a výstupy z 74175 zapojit rovnou na číslicovku. Já zvolím tuto druhou možnost.



Obrázek 1.2 – Graf přechodů SSO

Krok 3 – z grafu vyčteme tabulku přechodů a pravdivostní tabulky pro kombinační obvod změny stavů

Stavů je celkem 8. Potřebujeme použít paměťové bity A,B,C obvodu 74175. Tabulka bude mít tři vstupy a tři výstupy.

VÝSTUPY 74175			NOVÉVSTUPY 74175		
A	B	C	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Obrázek 1.3 – Tabulka přechodů SSO

Pravdivostní tabulky budou tři. Víme, že každá logická funkce má pouze jeden výstup. Vstupů může mít více, v našem případě budou mít tři vstupy A, B a C. Na tyto vstupy logických funkcí budeme přivádět hodnoty výstupů paměťového obvodu 74175.

VSTUPY			
A	B	C	A
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

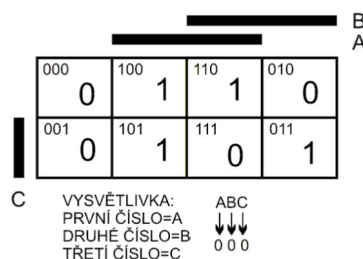
VSTUPY			
A	B	C	B
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

VSTUPY			
A	B	C	C
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Obrázek 1.4 – Pravdivostní tabulky funkcí kombinačního obvodu přechodu stavů SSO

Krok 4 – Sestavit logické funkce podle pravdivostních tabulek

Z pravdivostních tabulek odhadnout logické funkce už by bylo velmi těžké. Pokud by to vůbec šlo. Rozhodně jsme takové ještě neviděli. Obvod bude pravděpodobně složen z více různých logických funkcí neboli **členů**. Abychom určili, z jakých funkcí bude výsledná logická funkce složena, použijeme **Karnaughovy mapy**. Je to jiný zápis pravdivostních tabulek. Také zapisujeme jedničky a nuly do tabulky, ale trochu jinak. Zapisovat budeme **pouze výstup logické funkce**. Místo řádků a sloupců, kam zapisujeme vstupy, použijeme značení okének. **Každé okénko** bude znamenat **jednu kombinaci hodnot vstupů**. Tabulka bude mít **tolik okének, kolik je řádků** v pravdivostní tabulce (tolik je kombinací vstupů). Vysvětlení, jak mapa vypadá, je na obrázku 1.5.



Obrázek 1.5 – Karnaughova mapa pro funkci se třemi vstupy

Jak mapu chápat – každé okénko obsahuje jednu hodnotu výstupu hledané logické funkce. Kam ale která hodnota patří? Všimněte si třech tlustých čar okolo mapy. Jedna je svislá, s písmenkem C. Dvě jsou vodorovné, s písmenky A a B. Písmenka jsou hodnoty vstupů. Tlustá čára C se týká celého spodního řádku a znamená to, že do něj zapisujeme hodnotu výstupu funkce pro takové kombinace hodnot vstupů, kde je hodnota vstupu C rovna 1. Horní řádek tlustou čáru nemá a sem zapisujeme hodnotu výstupu funkce pro takové kombinace hodnot vstupů, kde je hodnota vstupu C rovna 0.

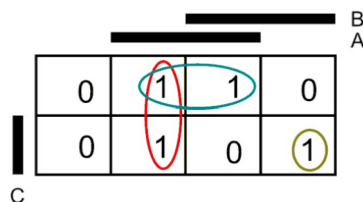
Mapa má čtyři sloupce. Vodorovné tlusté čáry se týkají vstupů A a B. První sloupec nad sebou nemá žádnou tlustou čáru, sem zapisujeme hodnotu výstupu funkce pro takové kombinace hodnot vstupů, kde je hodnota vstupů A i B rovna 0. Druhý sloupec má tlustou čáru pro vstup A, tedy vstup A je v tomto sloupci roven 1. Ve třetím sloupci jsou vstupy A i B rovny 1, protože tlustá čára nad sloupcem je pro vstup A i B, a do čtvrtého sloupce zapisujeme hodnotu výstupu funkce pro takové kombinace hodnot vstupů, kde je hodnota vstupu B rovna 1 a A rovna 0.

Do mapy jsem pro vysvětlení napsal malými číslicemi kombinaci hodnot vstupů, pro lepší pochopení. Ta se tam ale většinou nepíše - kam co zapsat vyplývá z tlustých čar.

Je nutné si uvědomit, že pravdivostní tabulka pro funkci A na obrázku 1.4 (tabulka vlevo) a Karnaughova mapa na obrázku 1.5 je jen **jiný zápis stejné funkce**. Vyjadřuje to ale to samé.

Jak z mapy určit, jak bude vypadat obvod logické funkce?

V mapě musíme zakroužkovat skupiny jedniček. Každá skupina znamená jeden **Člen logické funkce**. Jak, to vysvětlím za chvíli. Nejdříve jaké skupiny jedniček kroužkovat – můžeme zakroužkovat samotnou jedničku, dvě jedničky vedle sebe, nebo čtyři jedničky, osm jedniček. Mapu se zakroužkovanými jedničkami vidíte na obrázku 1.6.



Obrázek 1.6 – Karnaughova mapa se zakroužkovanými skupinami jedniček

Vidíte, že jednička z jedné dvojice jedniček se dá použít i v kroužku druhé dvojice jedniček, jak ukazuje červená (svislá) a modrá (vodorovná) dvojice. Jak z toho vyčíst funkci? Začneme se samotnou jedničkou v zeleném kroužku. Pokud **není** u písmenka vstupu **tlustá čára (A,B,C)**, zapíšeme vstup jako **vstup s čarou**. Kombinace vstupů jedné jedničky je:

Zelený člen = A s čarou - a zároveň B - a zároveň C . „S čarou“ znamená NOT. „A zároveň“ znamená AND.

První člen logické funkce pro vstup A vyčtený z mapy má tvar: (NOT A) AND B AND C. Pro tento člen budeme potřebovat logickou funkci **AND** se **třemi vstupy**. Je umístěna na modulu Saimon 2 vlevo nahoře. Vyčteme další dva členy.

Modrá dvojice = A - a zároveň C s čarou . **Vstup B nezapočítáváme – má různou hodnotu pro každé okénko, které červená elipsa protíná.**

Druhý člen logické funkce pro vstup A vyčtený z mapy má tvar: A AND (NOT C). Na tento člen použijeme funkci AND se dvěma vstupy, integrovaný obvod s touto funkcí je umístěn na modulu Saimon 2 vpravo nahoře.

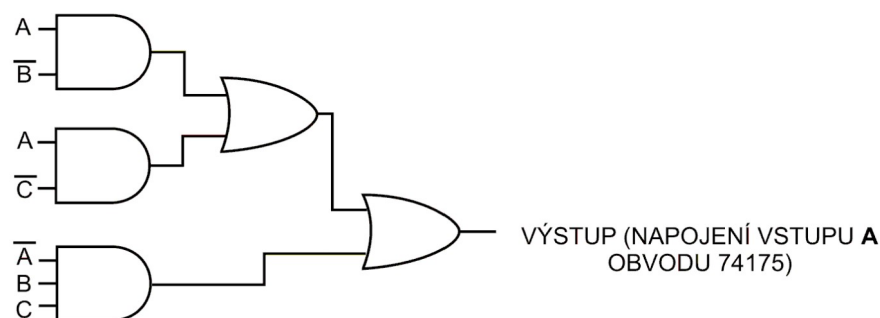
Zbývá červená dvojice. Kombinace vstupů je: A– a zároveň B s čarou . C opět nezapočítáváme, jeho hodnota je různá pro horní řádek (tam je 0) a pro spodní řádek (tam je 1).

Třetí člen logické funkce pro vstup A vyčtený z mapy má tvar: $A \text{ AND } (\text{NOT } B)$. Na tento člen také použijeme funkci AND se dvěma vstupy.

Jak to celé číst: Logická funkce je pravdivá pro tyto případy [$(\text{NOT } A) \text{ AND } B \text{ AND } C$] nebo [$A \text{ AND } (\text{NOT } C)$] nebo [$A \text{ AND } (\text{NOT } B)$]. Co znamená „nebo“ víme také – to je logická funkce OR. Celý zápis logické funkce pro vstup A paměťového obvodu vypadá takto:

Výsledná funkce pro změnu vstupu A má tvar: [$(\text{NOT } A) \text{ AND } B \text{ AND } C$] OR [$A \text{ AND } (\text{NOT } C)$] OR [$A \text{ AND } (\text{NOT } B)$].

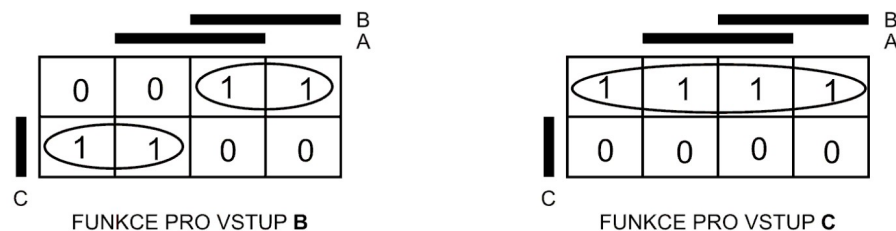
Je to už trochu méně přehledné, tak pozor, ať se v tom neztratíte. Schéma logické funkce pro vstup A obvodu 74175 je na obrázku 1.7.



Obrázek 1.7 – Obvod logické funkce pro změnu vstupu A obvodu 74175

Na obrázku vidíme, že logická funkce pro změnu vstupu A obvodu 74175 není nijak jednoduchá. Rozhodně bychom ji neodhadli, dokonce s pomocí map je to časově náročné. A ještě k použití funkce OR - ideální by bylo použít funkci OR se třemi vstupy. Ta na stavebnici není, protože se OR s více vstupy se dá velice jednoduše složit z funkce OR se dvěma vstupy. Jak to funguje je vidět na obrázku 1.7.

Tak. To není všechno. Zbývá najít logické funkce pro změnu vstupů B a C obvodu 74175. Pro každý potřebujeme mapu. Ty jsou na obrázku 1.8 vyplněné a zakroužkované. Pro procvičení si zkuste mapy nejdříve nakreslit sami a zapsat do nich hodnoty.



Obrázek 1.8 – Karnaughovy mapy pro změnu vstupů B a C

Logická funkce pro vstup B má dvě dvojice jedniček.

První člen logické funkce pro vstup B vyčtený z mapy má tvar: **B AND (NOT C)**.

Druhý člen logické funkce pro vstup B vyčtený z mapy má tvar: **(NOT B) AND C**.

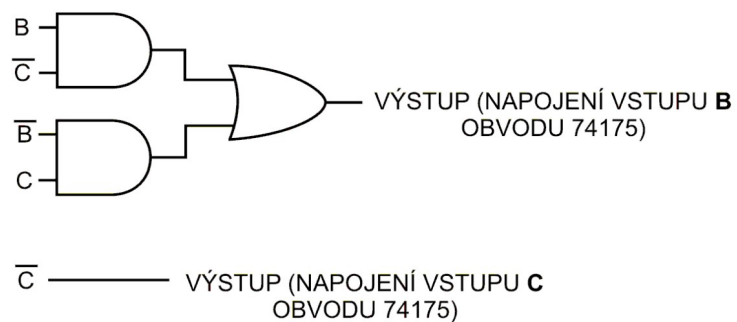
Výsledná funkce pro změnu vstupu B paměťového obvodu 74175 má tvar: [B AND (NOT C)] OR [(NOT B) AND C]

Logická funkce pro vstup C má jednu čtveřici.

První člen logické funkce pro vstup C vyčtený z mapy má tvar: **NOT C**. Více skupin jedniček nemáme, můžeme psát:

Výsledná funkce pro změnu vstupu C paměťového obvodu 74175 má tvar: NOT C

Schéma bloků logických funkcí pro změnu vstupů B a C obvodu 74175 je na obrázku 1.9



Obrázek 1.9 – Obvody logických funkcí pro změnu vstupů B a C obvodu 74175

Logické funkce pro změnu vstupů máme hotové. Zbývají poslední kroky.

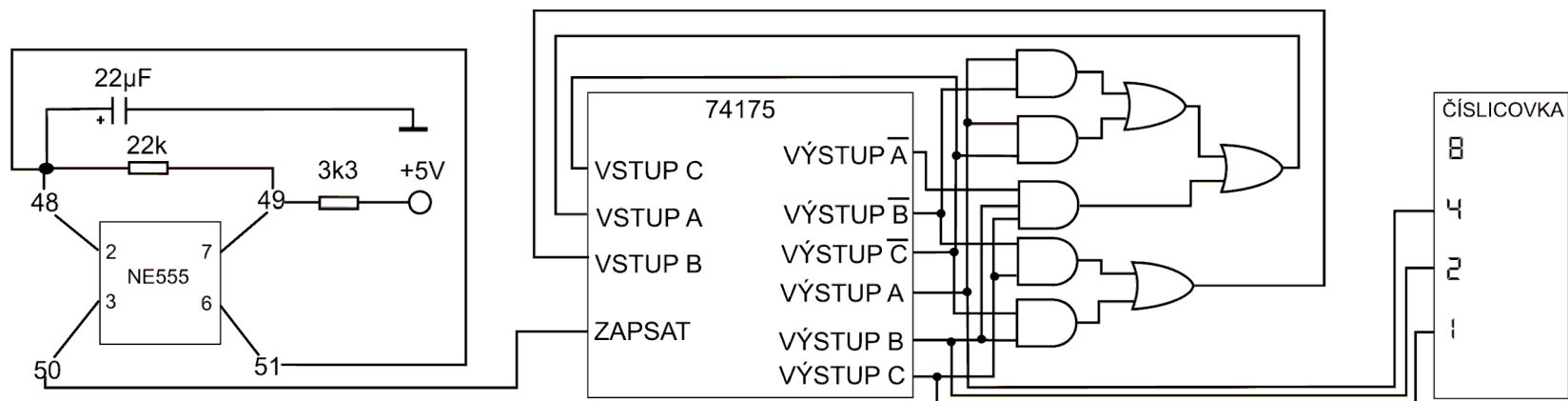
Krok 5 – Napsat pravdivostní tabulky pro logické funkce kombinačního obvodu ovládající výstupní obvod

Není potřeba. Čísla stavů SSO jsou zvoleny tak, abychom mohli výstupy obvodu 74175 napojit přímo na číslicovku.

Krok 6 – Navrhnout kombinační obvod pro výstupní obvod

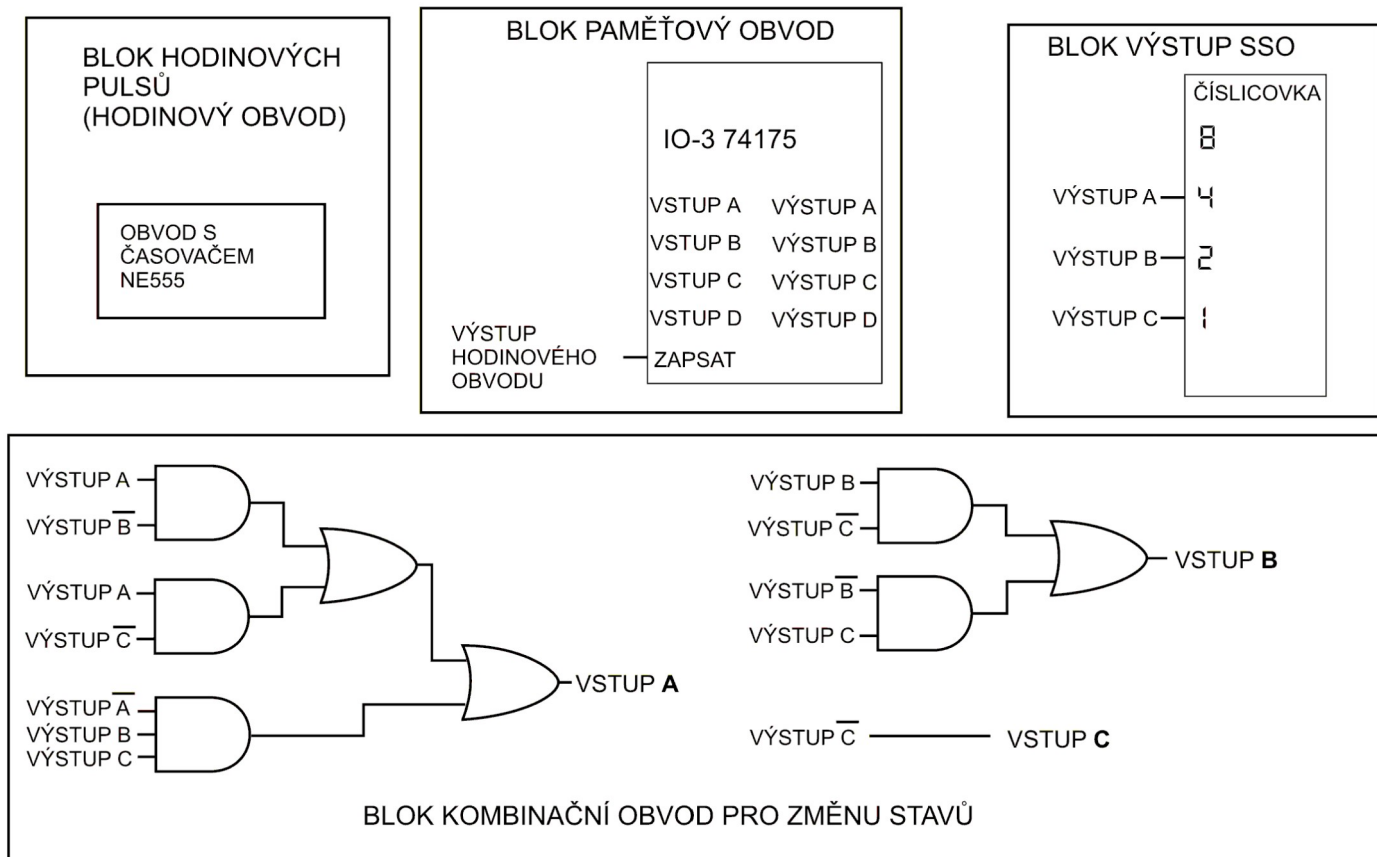
Stejný případ jako v kroku 5.

Zbývá jen nakreslit schéma obvodu. To je na obrázku 1.10.



Obrázek 1.10 – Schéma obvodu počítadla od 0 do 7.

Jak vidíte, schéma obvodu je již dosti velké a začíná být nepřehledné. Navíc některé bloky jsou ve všech SSO stejné. Například hodinový obvod a paměťový obvod. V podstatě vždy jen hledáme, jak bude vypadat kombinační obvod pro změnu stavů SSO a kombinační obvod pro výstupy SSO. Z toho důvodu budu schéma dělit do bloků, které mezi sebou nebudou přímo propojeny čarami ve schématu. Místo toho budou schematické značky v každém bloku, které budou značit, jak bloky mezi sebou propojit. Například pokud bude v kombinačním obvodu pro změnu vstupů napsáno VÝSTUP A, znamená to, že na to místo se má připojit výstup A paměťového obvodu. Blokové schéma na obrázku 1.11 znázorňuje stejný obvod, jako je na obrázku 1.10, jen zapsaný pomocí bloků.















Obrázek 1.11 – Schéma obvodu počítadla od 0 do 7.

V tomto dílu návodu již také nebudu uvádět čísla propojení drátků. Bude na čtenáři, aby si schéma obkreslil, a vždy, když zapojí drátek, tak ho ve schématu škrtl.

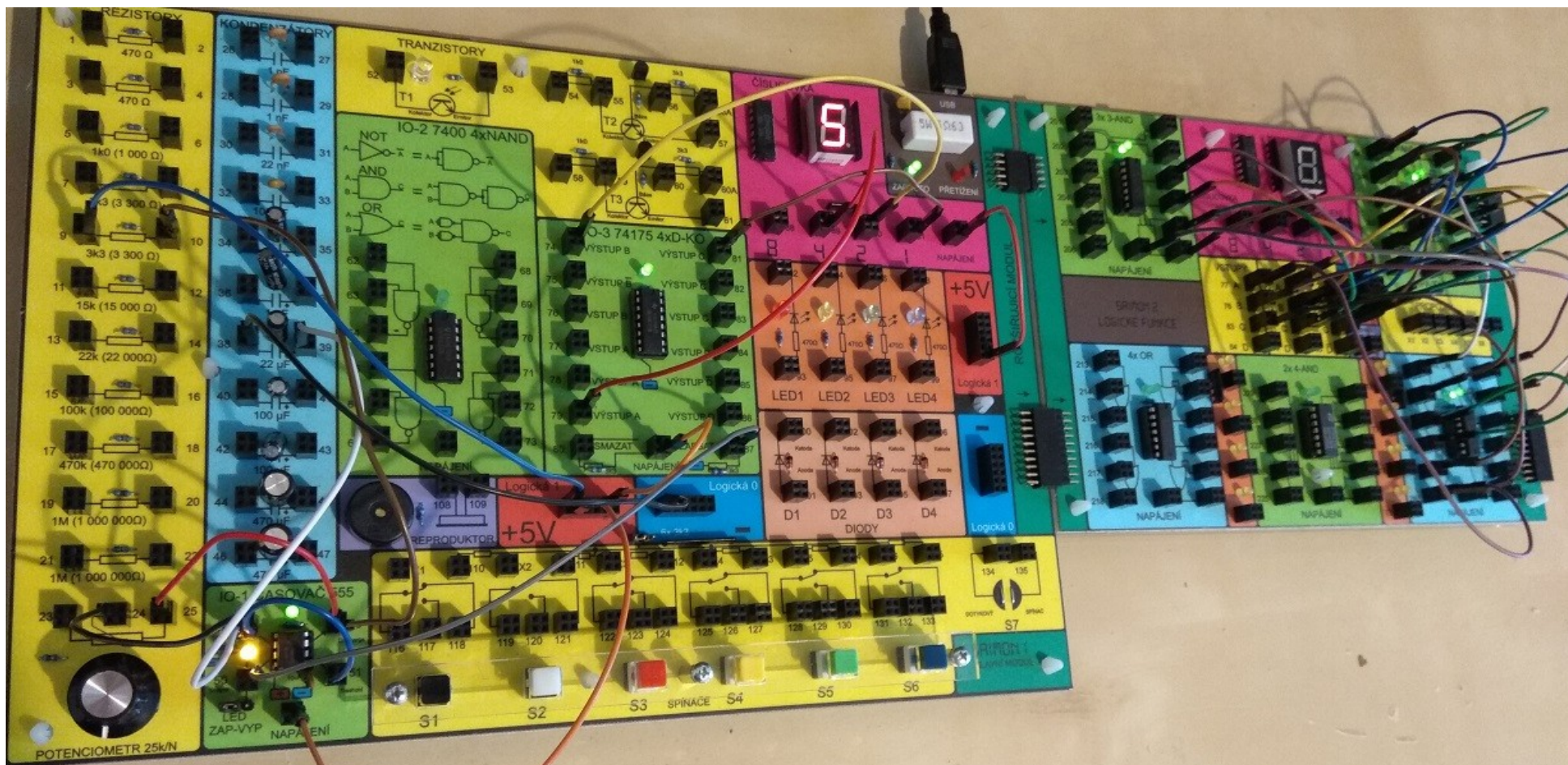
Některé bloky SSO není třeba v podstatě vůbec kreslit. Budou vždy stejné. A je to hodinový obvod a paměťový obvod. Dál se budu soustředit jen na blok kombinační obvod pro změnu stavů a blok s názvem kombinační obvod pro výstupy SSO.

Ještě poznámka - abychom nemuseli vést dlouhé drátky mezi obvody logických funkcí na modulu Saimon 2 a paměťovým obvodem, je na modulu Saimon 2 propojovací pole se vstupy a výstupy paměťového obvodu IO-3 74175, který je umístěn na modulu Saimon 1. Pomocí sběrnice zespolu stavebnice a dvou černých konektorů na pravém kraji Saimon 1 jsou jeho vstupy a výstupy vyvedeny do přidavného modulu. Zdířky jsou označené stejnými čísly, jaká jsou na modulu Saimon 1.

VSTUPY		VÝSTUPY			
77 A		79 A		78 \bar{A}	
76 B		74 B		75 \bar{B}	
83 C		81 C		82 \bar{C}	
84 D		86 D		85 \bar{D}	

Obrázek 1.12 – Propojovací pole na modulu Saimon 2

Fotka zapojení obvodu počítadla od 0 do 7 je na obrázku 1.13. Vidíte, že pokud využijeme sběrnici, tak šetříme drátky a obvod je přehlednější.



Obrázek 1.13 – Zapojení počítadla na stavebnici Saimon. Využití sběrnice místo propojení dlouhými drátky

Kapitola 2 – Knight rider (don't care stavy, použití propojek)

V této kapitole se vrátím ke schématu z minulého dílu. Jmenovalo se „poloviční Knight rider“. V této kapitole zapojíme celý Knight rider. Půjdeme po známých krocích.

Krok 1 – Slovní popis funkce obvodu

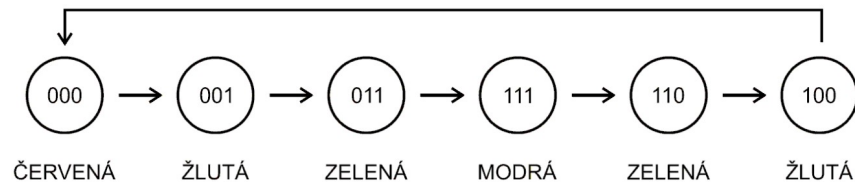
V obvodu budou zapojené 4 ledky. Budou blikat sem a tam – v pořadí červená, žlutá, zelená, modrá, zelená, žlutá a zpět - do stavu červená.

Krok 2 – nakreslíme si, jak jdou stavy po sobě

Očíslování stavů zvolíme opět „výhodné“. To znamená, že se stavy budou lišit jen ve změně jednoho bitu. Kombinační obvod pro změnu stavů bude jednodušší.

Stav 1 = 000, stav 2 = 001, stav 3 = 011, stav 4 = 111, stav 5 = 110, stav 6 = 100

Pro čísla stavů nám vystačí 3 bity, tedy vstupy/výstupy A,B,C obvodu 74175.



Obrázek 2.1 – Graf SSO „Knight rider“

Krok 3 – z grafu vyčteme tabulku přechodů a pravdivostní tabulky pro kombinační obvod změny stavů

Tabulku vytvoříme snadno podle grafu, tak jak už to známe.

VÝSTUPY 74175			NOVÉVSTUPY 74175		
A	B	C	A	B	C
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	1	1	1
1	1	1	1	1	0
1	1	0	1	0	0
1	0	0	0	0	0

Obrázek 2.2 – Tabulka přechodů kombinačního obvodu pro změnu stavů

Rozepíšeme tabulku přechodů na tři pravdivostní tabulky. Každá tabulka bude pro jednu logickou funkci, čili výstup funkce bude zapojený na příslušný vstup paměťového obvodu 74175.

VSTUPY			
A	B	C	A
0	0	0	0
0	0	1	0
0	1	1	1
1	1	1	1
1	1	0	1
1	0	0	0

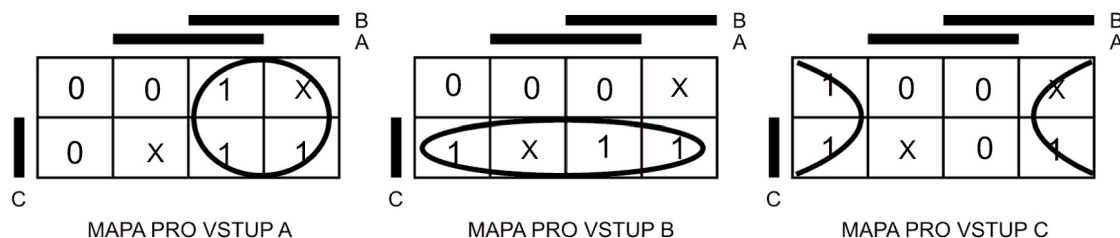
VSTUPY			
A	B	C	B
0	0	0	0
0	0	1	1
0	1	1	1
1	1	1	1
1	1	0	0
1	0	0	0

VSTUPY			
A	B	C	C
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0

Obrázek 2.3 – Pravdivostní tabulky kombinačního obvodu pro jednotlivé funkce

Krok 4 – Sestavit logické funkce kombinačního obvodu pro změnu stavů SSO

Logické funkce se dají v tomto případě odhadnout. Kvůli procvičení Karnaughových map je ale budeme hledat pomocí map. Nakreslíme tři mapy pro logické funkce obvodu změny stavů SSO.



Obrázek 2.4 - Karnaughovy mapy logických funkcí obvodu pro změnu stavů SSO

V mapě jsem do dvou políček nakreslil písmeno X. Jaký to má důvod? SSO se do stavu touto kombinací **nikdy nedostane**. Když se podíváte na tabulku přechodů, nikde není kombinace 010 a 101. Proto nám může být jedno, jakou hodnotu by měl výstup logické funkce v tomto stavu. Takové stavy, na jejichž hodnotě nezáleží, značíme X. Anglicky se nazývají „**Don't care**”. Pro hledání logické funkce to má výhodu – můžeme uvažovat, že v tomto stavu bude mít výstup hodnotu 1. Pak se dá zakroužkovat společně s jedničkami a výsledná funkce může být jednodušší.

Nezapomeňte, že vždy musíme zakroužkovat **všechny jedničky**. Stav **X můžeme a nemusíme** zakroužkovat, podle toho, jak se nám to hodí.

Všimněte si zakroužkování mapy pro vstup C. Sousedí spolu i levý krajní a pravý krajní sloupec, zakroužkovat můžeme i takto.

Funkce pro změnu vstupu A paměťového obvodu 74175 má tvar: vstup A = výstup B

Funkce pro změnu vstupu B paměťového obvodu 74175 má tvar: vstup B = výstup C

Funkce pro změnu vstupu C paměťového obvodu 74175 má tvar: vstup C = výstup A s čarou

Krok 5 – Napsat pravdivostní tabulky pro logické funkce kombinačního obvodu ovládající výstupní obvod (ledky)

Z grafu přechodů vytvoříme pravdivostní tabulky pro logické funkce, které budou ovládat výstupy SSO, což jsou v našem případě ledky.

VSTUPY			ČERVENÁ
A	B	C	
0	0	0	1
0	0	1	0
0	1	1	0
1	1	1	0
1	1	0	0
1	0	0	0

VSTUPY			ŽLUTÁ
A	B	C	
0	0	0	0
0	0	1	1
0	1	1	0
1	1	1	0
1	1	0	0
1	0	0	1

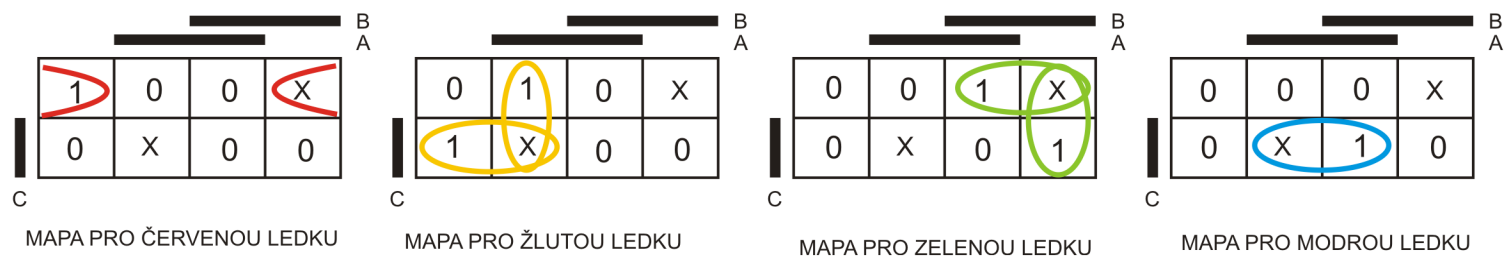
VSTUPY			ZELENÁ
A	B	C	
0	0	0	0
0	0	1	0
0	1	1	1
1	1	1	0
1	1	0	1
1	0	0	0

VSTUPY			MODRÁ
A	B	C	
0	0	0	0
0	0	1	0
0	1	1	0
1	1	1	1
1	1	0	0
1	0	0	0

Obrázek 2.5 – Pravdivostní tabulky kombinačního obvodu pro výstupy (ledky)

Krok 6 – Navrhnout kombinační obvod pro výstupní obvod

Logické funkce ovládající ledky se sice dají odhadnout, ale pomocí Karnaughových map najdeme méně složité funkce.



Obrázek 2.6 - Karnaughovy mapy logických funkcí pro ledky

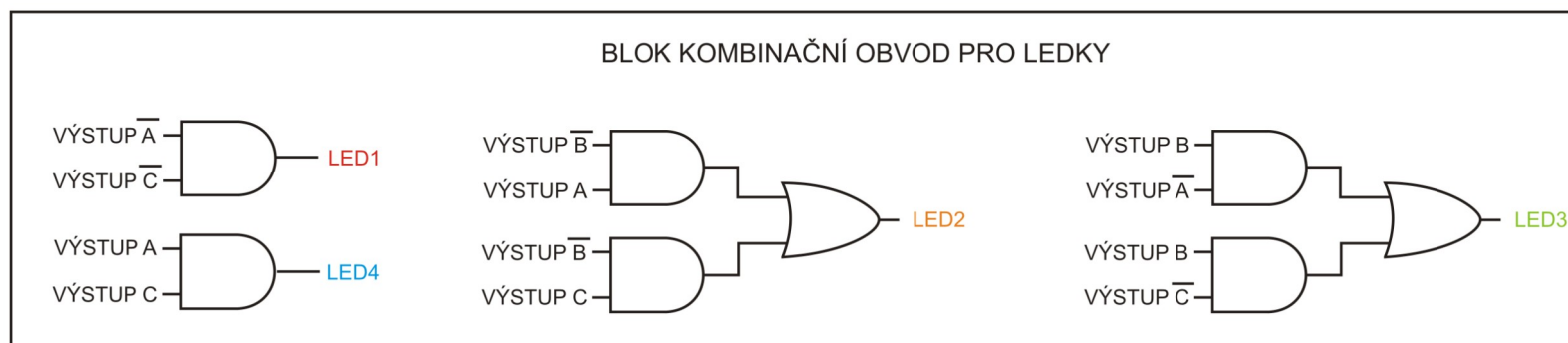
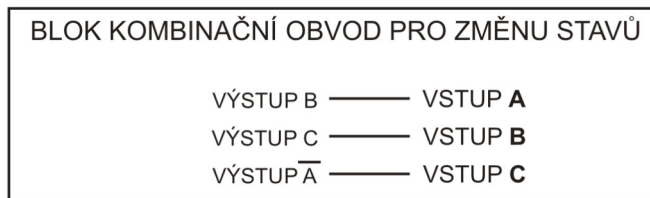
Funkce pro červenou ledku má tvar: LED1 = A s čarou AND C s čarou

Funkce pro žlutou ledku má tvar: LED2 = (A AND B s čarou) OR (B s čarou AND C)

Funkce pro zelenou ledku má tvar: LED3 = (B AND C s čarou) OR (B AND A s čarou)

Funkce pro modrou ledku má tvar: LED4 = A AND C

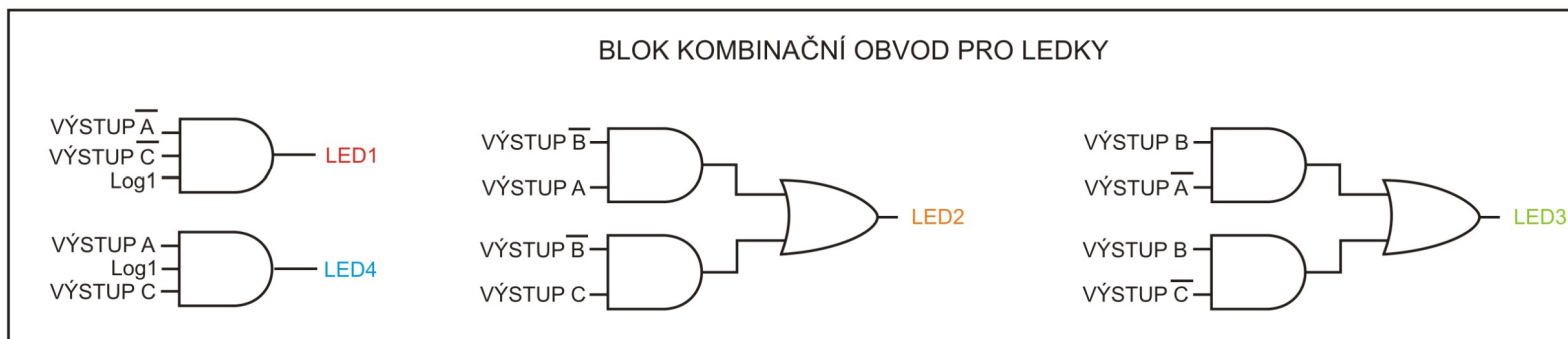
Blokové schéma obvodu je na obrázku 2.7.



Obrázek 2.7 – Blokové schéma obvodu „Knight Rider“

V blokovém schématu už neuvádím bloky, které jsou stále stejné. Hodinový obvod je stále tvořen obvodem NE555, v tomto případě astabilní zapojení. Paměťový obvod je stále obvod 74175.

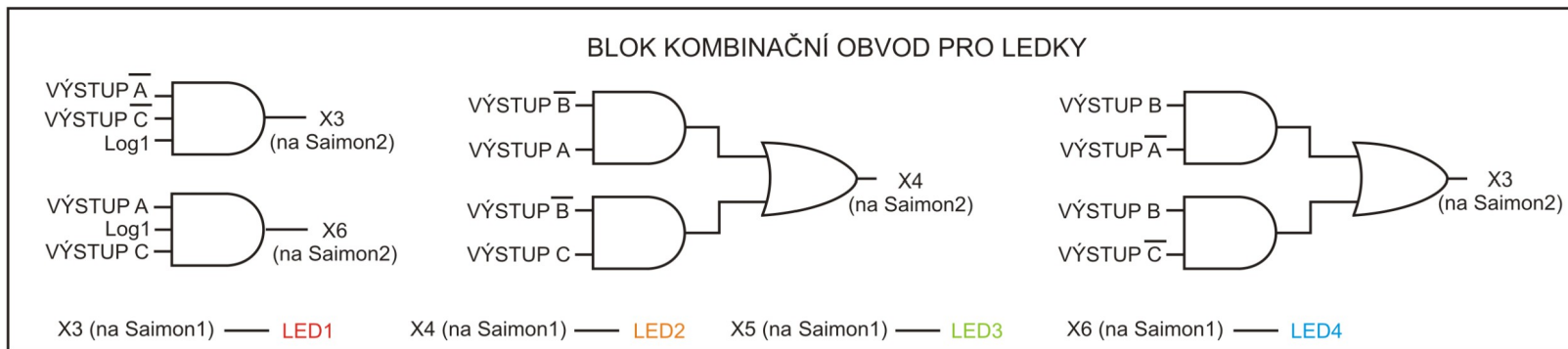
Kombinační obvod pro ledky vyžaduje šest logických funkcí AND se dvěma vstupy. Tolik jich na stavebnici nemáme, ale můžeme si pomoci. Využijeme logickou funkci se AND se třemi vstupy a jeden ze vstupů připojíme na log1. Pak bude fungovat jako logická funkce se dvěma vstupy. Obrázek 2.8. ukazuje, jak na to.



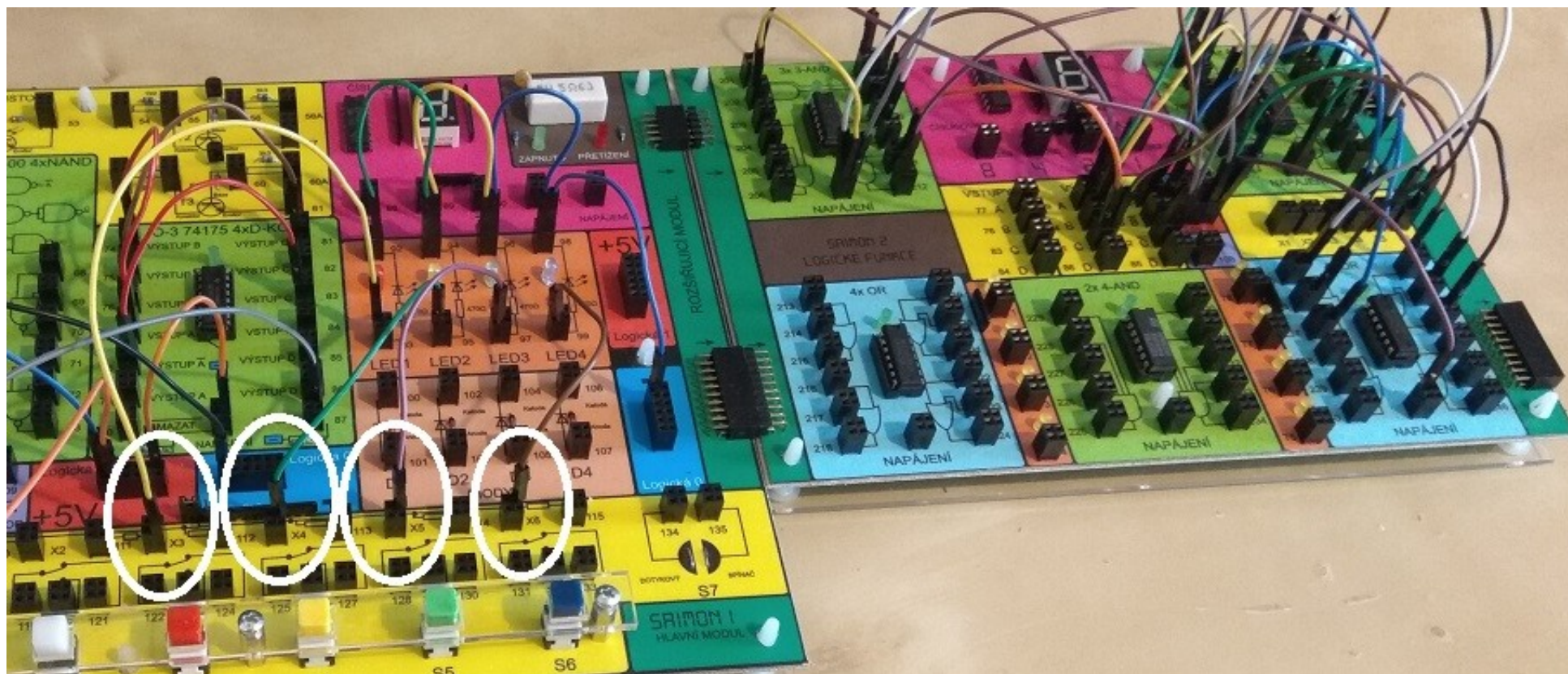
Obrázek 2.8 – Náhrada logické funkce AND se dvěma vstupy pomocí AND se třemi vstupy pro LED1 a LED4

Jak můžete vidět, je jedno, na který vstup logické funkce AND se třemi vstupy přivedeme hodnotu log1. U logické funkce pro LED1 jsem zvolil třetí vstup, u logické funkce pro LED4 jsem zvolil druhý vstup. Všechny vstupy jsou si v tomto případě rovny.

Důležitá poznámka pro snadnější práci – mezi všemi moduly Saimon vedou kromě vstupů a výstupů paměťového obvodu ještě propojovací místa, nazvané **propojky X1,X2,X3,X4,X5 a X6**. Není pak potřeba mezi logickými funkcemi pro ovládání ledek a ledkami zapojovat dlouhé drátky. Výstup logické funkce pro ledku zapojíme do propojky na modulu Saimon 2 a propojku se stejným číslem na modulu Saimon 1 propojíme s anodou ledky. Na obrázku 2.9 můžete vidět spojení kombinačního obvodu pro ledky a ledek pomocí propojek. Fotka zapojení je na obrázku 2.10.



Obrázek 2.9 – Propojení výstupů logických funkcí na Saimon 2 a ledek na Saimon 1 pomocí propojek



Obrázek 2.10 – Foto zapojení obvodu Knight Rider. Výstupy logických funkcí a ledky jsou spojeny pomocí **propojek**.

Kapitola 3 – Hádanka „Dveře do trezoru“ (použití spínačů pro ovládání SSO)

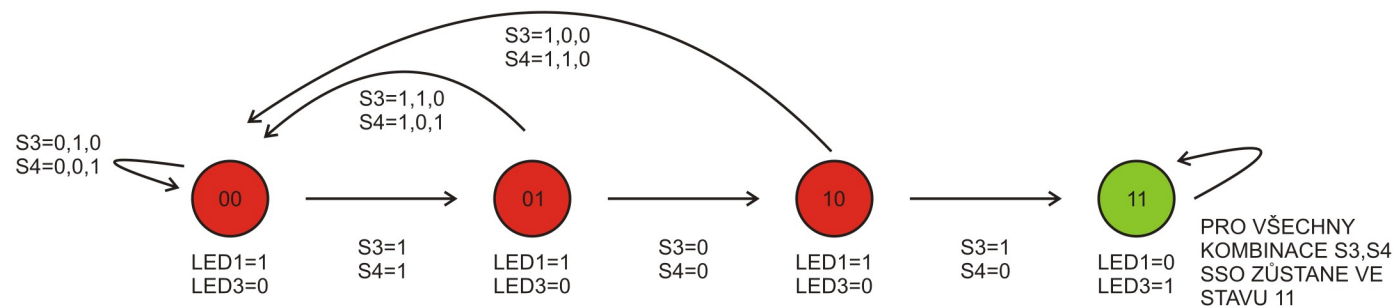
Krok 1 – Slovní popis funkce obvodu

Cílem bude sestavit obvod, který představuje kódovací zařízení pro odemčení dveří od trezoru. Na začátku je trezor zakódovaný. Rozsvícená červená ledka signalizuje, že je zamčený. Otevírá se zadáním tří po sobě jdoucích správných čísel. Pokud jsou čísla zadaná správně, rozsvítí se zelená ledka, což bude signalizovat odemčený trezor.

Stavy SSO budou čtyři, já zvolím číslování 00,01,10 a 11. Do kombinačního obvodu pro změnu vstupů musíme zapojit i spínače, pomocí kterých budeme zadávat kombinaci čísel pro otevření trezoru.

Pro hodinový pulz použijeme tentokrát monostabilní zapojení časovače NE555, protože se potřebujeme přesouvat mezi stavy až ve chvíli, kdy máme nastavenou správnou kombinaci pro otevření trezoru. Číslo pro otevření trezoru budeme zadávat pomocí barevných spínačů S3 a S4. Pulz vygenerujeme stisknutím spínače S1.

Krok 2 – nakreslíme si, jak jdou stavy po sobě



Obrázek 3.1 - Graf stavů SSO „Dveře do trezoru“

Graf ukazuje, jak se bude SSO přesouvat mezi stavy a kdy k tomu dojde. Na začátku je SSO ve stavu 00. Svítí červená led. Pokud nastavíme spínače S3 a S4 do stavu log1, čili stiskneme je, pak stiskneme spínač S1, tak se SSO nastaví do stavu 01. Správná kombinace spínačů S3 a S4 je v grafu na obrázku 3.1 pod vodorovnou šipkou ze stavu 00 do stavu 01. Jakákoliv jiná kombinace způsobí, že automat zůstane ve stavu 00, což znázorňuje ohnutá šipka s kombinacemi S3 a S4:00,10,01. Pokud se zadáním všech správných kombinací dostaneme až do stavu 11, rozsvítí se zelená ledka, červená zhasne a SSO už v tomto stavu musí zůstat. Špatná kombinace hodnot spínačů vrátí SSO do stavu 00.

Krok 3 – z grafu vyčteme tabulku přechodů a pravdivostní tabulky pro kombinační obvod změny stavů

VÝSTUPY 74175		SPÍNAČE		BUDOUCÍ VSTUPY	
A	B	S3	S4	A	B
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0

VÝSTUPY 74175		SPÍNAČE		BUDOUCÍ VSTUPY	
A	B	S3	S4	A	B
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	0	0
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

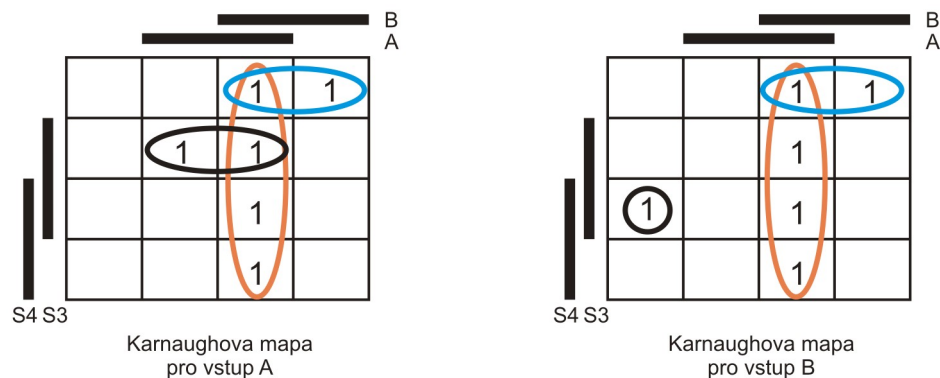
Obrázek 3.2 - Tabulka přechodů

V tabulce je vidět, že nyní jako vstupy kombinačního obvodu pro změnu stavů SSO používáme i spínače. Pouze řádky se správnou kombinací stavů spínačů mají ve sloupci „budoucí vstupy“ nastavenou hodnotu dalšího stavu.

Krok 4 – Sestavit logické funkce podle pravdivostních tabulek

Logické funkce, které budou tvořit kombinační obvod změny stavů SSO, budou dvě. Potřebujeme pouze dva výstupy pro nastavení vstupů A a B obvodu 74175. Obě funkce budou mít 4 vstupy. Dva ze vstupů obou logických funkcí budou výstupy A a B obvodu 74175, další dva vstupy budou připojené na spínače S3 a S4.

Pro logickou funkci se čtyřmi vstupy budeme potřebovat větší Karnaughovu mapu, kterou jsme ještě nepoužívali. Bude mít 16 políček, do kterých zapíšu hodnotu výstupu logické funkce podle hodnot vstupů daných pro políčka. Karnaughovy mapy vyplním rovnou – podle tabulky přechodů.



Obrázek 3.3 – Karnaughovy mapy pro 4 vstupy

V mapě už nebudu psát nuly, protože nás stejně zajímají pouze jedničky (nebo X stavy). Zakroužkováním skupin jedniček získáme počet a tvar členů logických funkcí. Vidíme, že obě mapy obsahují tři skupiny jedniček, obě funkce budou mít tři členy.

Funkce pro změnu vstupu A paměťového obvodu 74175 má tvar: vstup A = (S3 s čarou AND S4 s čarou AND B) OR (A AND B) OR (S3 AND S4 s čarou AND A)

Funkce pro změnu vstupu B paměťového obvodu 74175 má tvar: vstup B = (S3 s čarou AND S4 s čarou AND B) OR (A AND B) OR (S3 AND S4 AND A s čarou AND B s čarou)

Krok 5 – Napsat pravdivostní tabulky pro logické funkce kombinačního obvodu ovládající výstupní obvod

Výstupní funkce budou dvě. Jedna pro zelenou ledku a druhá pro červenou ledku. Tyto funkce budou mít pouze dva vstupy, protože ledky svítí pouze podle stavu SSO, který nezáleží na stavu spínačů.

VÝSTUPY 74175			
A	B	LED1	
0	0	1	
1	0	1	
0	1	1	
1	1	0	

VÝSTUPY 74175			
A	B	LED3	
0	0	0	
1	0	0	
0	1	0	
1	1	1	

Obrázek 3.4 – Pravdivostní tabulky funkcí pro ovládání ledek

Krok 6 – Navrhnout kombinační obvod pro výstupní obvod

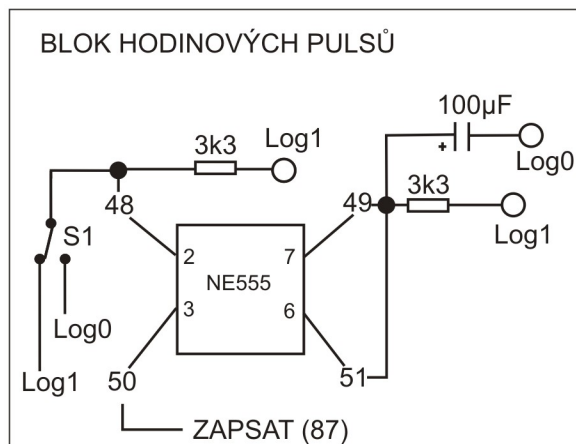
Z tabulek na obrázku 3.4 můžeme tvar funkce odhadnout, není třeba psát Karnaughovy mapy. Je vidět, že ledka LED3 svítí jen v případě, že A i B je rovno log1. Takovou pravdivostní tabulku už známe z druhého dílu návodu pro Saimon 1 - je uvedena na stránce 5, obrázek 2.2. Představuje funkci AND.

Pravdivostní tabulku pro LED1 známe také s druhého dílu návodu pro Saimon 1. Je na stránce 11, obrázek 2.10. Představuje logickou funkci NAND. Tedy:

Funkce pro červenou ledku LED1 má tvar: $LED1 = A \text{ NAND } B$

Funkce pro zelenou ledku LED3 má tvar: $LED3 = A \text{ AND } B$

A nakonec nakreslíme blokové schéma pro celý obvod. Blok hodinových pulsů bude tvořit monostabilní zapojení časovače NE555.



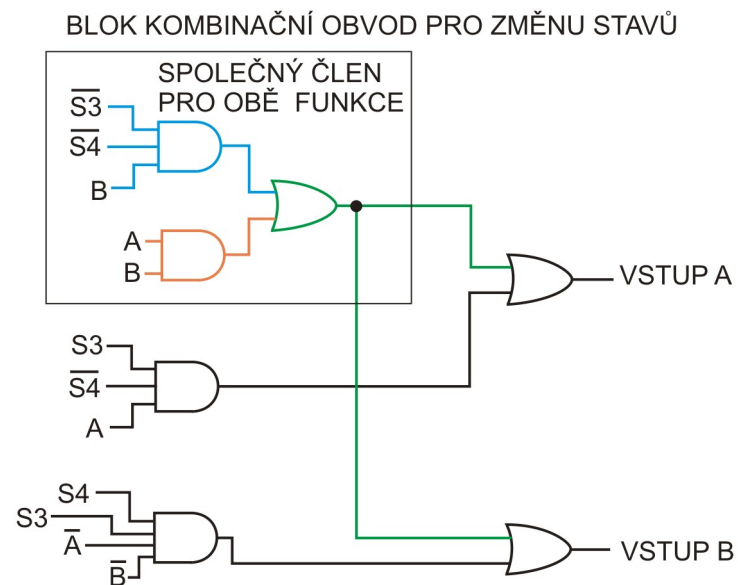
Obrázek 3.5 – Blok hodinových pulsů

Kombinační obvod pro změnu stavů je tvořen následujícími funkcemi

Funkce pro změnu vstupu A paměťového obvodu 74175 má tvar: vstup A = (S3 s čarou AND S4 s čarou AND B) OR (A AND B) OR (S3 AND S4 s čarou AND A)

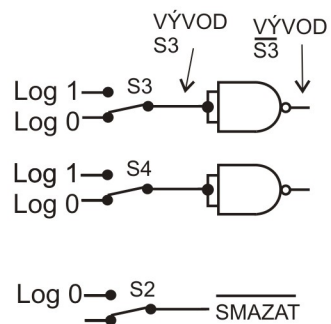
Funkce pro změnu vstupu B paměťového obvodu 74175 má tvar: vstup B = (S3 s čarou AND S4 s čarou AND B) OR (A AND B) OR (S3 AND S4 AND A s čarou AND B s čarou)

V této kapitole vám ukážu, jak ušetřit některé členy funkcí, když tyto členy obsahuje více funkcí. Skupiny jedniček (tedy členy logické funkce) jsem záměrně obarvil. Můžeme vidět, že obě funkce mají modře označený člen **S3 s čarou AND S4 s čarou AND B**. Také obě obsahují oranžový člen **A AND B**. Dokonce mají obě složený člen: **(S3 s čarou AND S4 s čarou AND B) OR (A AND B)**. Tento člen OR označím zeleně. Nemusíme tyto členy pro každou funkci zapojovat zvlášť. Stačí je zapojit jednou a výstup z těchto členů přivést na OR příslušné funkce. Více řekne obrázek 3.6, kde je nakreslené celé schéma bloku. V obdélníku je znázorněný člen, který je stejný pro obě funkce. Všimněte si na obrázku propojení z jeho výstupu do vstupů obou členů OR. Výstup z prvního z těchto členů OR vede na **vstup A** a výstup z druhého vede na **vstup B** obvodu 74175.



Obrázek 3.6 – Schéma bloku kombinačního obvodu pro změnu stavů

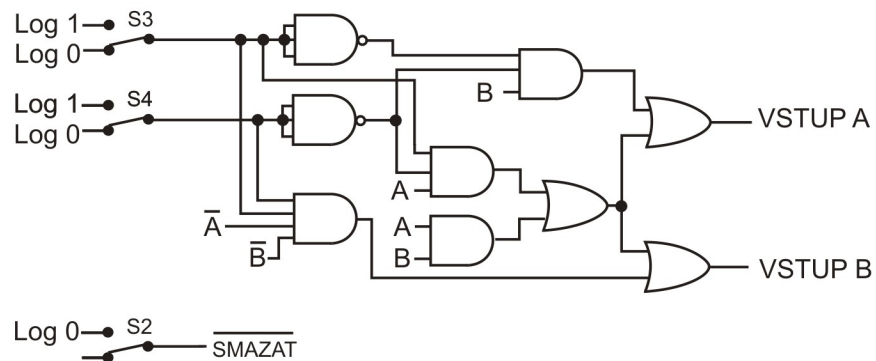
Ještě zbývá otázka, jak získat S3 s čarou a S4 s čarou. Je to jednoduché. Výstupy ze spínačů přivedeme na funkci NOT, kterou vytvoříme pomocí funkce NAND. Vysvětlení je na obrázku 3.7. Ještě jsem přidal spínač S2 zapojený tak, aby uměl SSO resetovat. Je to užitečné pro to, když už je trezor otevřený a my chceme začít znovu, nebo v případě, že po zapnutí obvodu nejsou výstupy 74175 nastaveny na hodnotu 00, což se může také stát.



Obrázek 3.7 – Zapojení negace vývodu spínačů a resetu

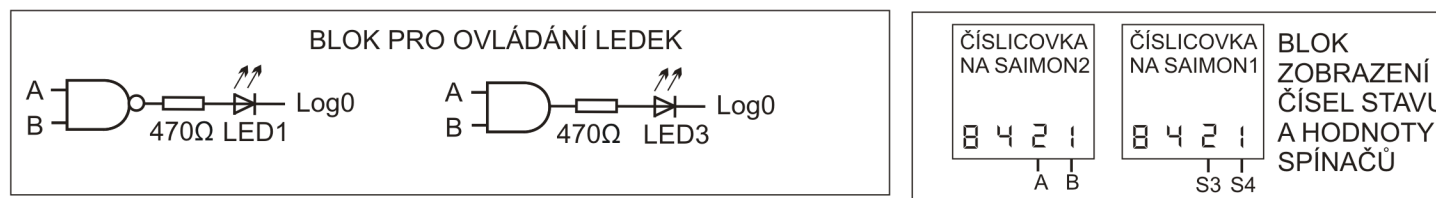
Celé schéma kombinačního obvodu pro změnu stavů může vypadat třeba tak, jak je znázorněno na obrázku 3.8. Mě osobně jeho rozdělení na části připadá přehlednější.

BLOK KOMBINAČNÍ OBVOD PRO ZMĚNU STAVŮ



Obrázek 3.8 – Schéma kombinačního obvodu pro změnu stavů

Zbývá ještě nakreslit schéma bloku ovládaní ledek. Funkce již známe z kroku 6, stačí jen nakreslit obvod, který je na obrázku 3.9. Do schéma jsem ještě přidal možnost zapojení číslicovek pro zobrazení stavu, ve kterém se SSO nachází a také pro zobrazení čísla, které zadáváme pomocí spínačů.



Obrázek 3.9 – Bloky pro ovládaní ledek a číslicovek

Číslicovka na modulu Saimon 1 bude zobrazovat číslo zadávané pomocí spínačů a číslicovka na modulu Saimon 2 bude zobrazovat stav SSO.

Pro bystré hlavy: Funkci $A \text{ AND } B$, která rozsvěcí zelenou LED3, už máme zapojenou v kombinačním obvodu pro změnu stavů (oranžový člen). Nemusíme tento člen zapojovat ještě jednou, stačí výstup z oranžového členu připojit na anodu ledky LED3.

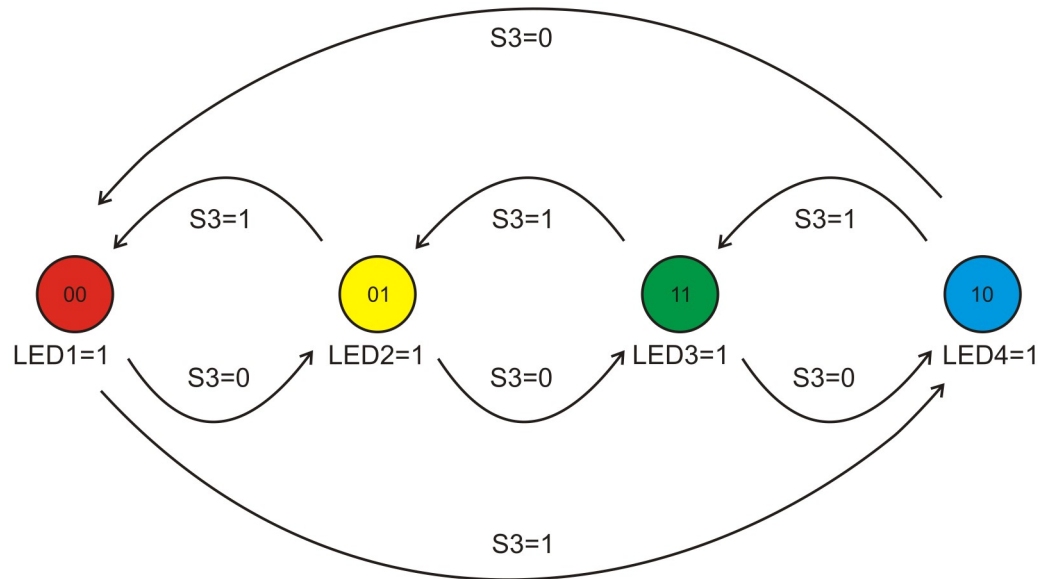
Kapitola 4 – Reverzní „Knight Rider“

Ted', když už umíme do obvodu zapojit spínače tak, aby jejich hodnoty ovlivňovaly následující stav SSO, můžeme použít spínač přímo k ovládní obvodu. Například můžeme použít spínač do už známého blikajícího obvodu poloviční Knight Rider.

Krok 1 – Slovní popis funkce obvodu

Do známého obvodu přidáme spínač. Výstup tohoto spínače bude přiveden jako jeden ze vstupů kombinačního obvodu pro změnu stavů SSO. Pokud spínač nebude stisknutý, bude obvod blikat stejně, jako již známý poloviční Knight Rider. Pokud spínač stiskneme, bude blikat v opačném směru. Čili při hodnotě spínače log0 se stavy budou měnit v pořadí 00-01-11-10 a zpět do 00. Pokud bude hodnota spínače rovna log1, budou se stavy SSO měnit v pořadí 00-10-11-01 a zpět do 00. Více je vidět na grafu.

Krok 2 – nakreslíme si, jak jdou stavy po sobě



Obrázek 4.1 – Graf spínačem ovládaného SSO „poloviční Knight Rider“

Krok 3 – z grafu vyčteme tabulku přechodů a pravdivostní tabulky pro kombinační obvod změny stavů

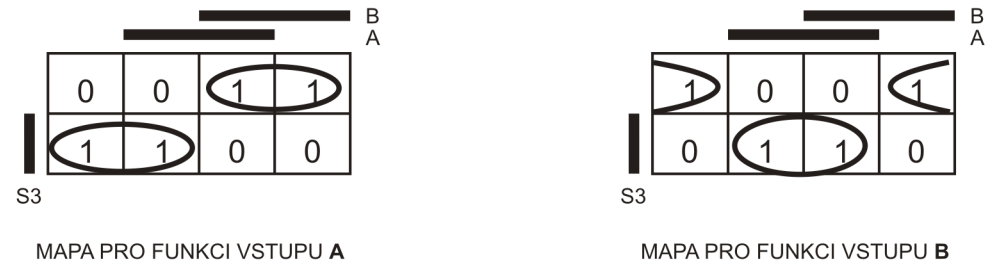
Do tabulky přechodů jsem zakreslil spínač S3 jako jeden ze vstupů. Všimněte si následujících stavů pro stisknutý spínač.

VÝSTUPY 74175		SPÍNAČ		BUDOUCÍ VSTUPY	
A	B	S3		A	B
0	0	0		0	1
0	1	0		1	1
1	1	0		1	0
1	0	0		0	0
0	0	1		1	0
0	1	1		0	0
1	1	1		0	1
1	0	1		1	1

Obrázek 4.2 – Tabulka přechodů pro reverzní Knighr Rider

Krok 4 – Sestavit logické funkce podle pravdivostních tabulek

Pravdivostní tabulky není třeba kreslit, když je tabulka přechodů hezky čitelná. Nakreslím rovnou Karnaughovy mapy a sestavím logické funkce pro kombinační obvod změny stavů SSO. Karnaughovy mapy jsou na obrázku 4.3.

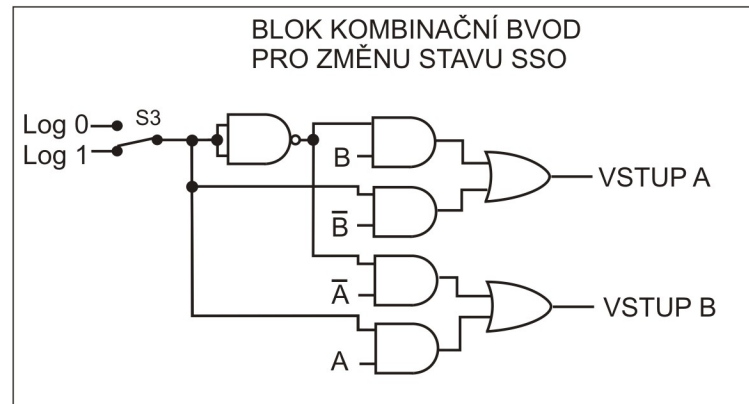


Obrázek 4.3 – Karnaughovy mapy pro určení funkcí kombinačního obvodu změny stavů SSO

Funkce pro změnu vstupu A paměťového obvodu 74175 má tvar: vstup A = (S3 s čarou AND B) OR (S3 AND B s čarou)

Funkce pro změnu vstupu B paměťového obvodu 74175 má tvar: vstup B = (S3 s čarou AND A s čarou) OR (S3 AND A)

Pro každou s funkcí nám stačí dvakrát člen AND se dvěma vstupy a jeden člen OR. Blok schéma kombinačního obvodu pro změnu vstupů je na obrázku 4.4.



Obrázek 4.4 – Blokové schéma kombinačního obvodu pro změnu stavu SSO

Krok 5 – Napsat pravdivostní tabulky pro logické funkce kombinačního obvodu ovládající výstupní obvod

Každá ledka svítí pouze v jednom stavu SSO. Záleží tedy pouze na výstupech A a B paměťového obvodu 74175, nikoliv na stavu spínače.

VÝSTUPY 74175		
A	B	LED1
0	0	1
1	0	0
1	1	0
0	1	0

VÝSTUPY 74175		
A	B	LED2
0	0	0
1	0	1
1	1	0
0	1	0

VÝSTUPY 74175		
A	B	LED3
0	0	0
1	0	0
1	1	1
0	1	0

VÝSTUPY 74175		
A	B	LED4
0	0	0
1	0	0
1	1	0
0	1	1

Obrázek 4.5 – Pravdivostní tabulky pro ovládání ledek

Krok 6 – Navrhnout kombinační obvod pro výstupní obvod

Funkce pro rozsvěcení ledek jsou jednoduché. Každá ledka svítí pouze v jednom stavu. Kombinační obvod bude vždy tvořit jedna funkce AND. Jako vstupy zvolíme vždy správnou kombinaci A a B, dle řádku v tabulce, ve kterém je log1 příslušné ledky.

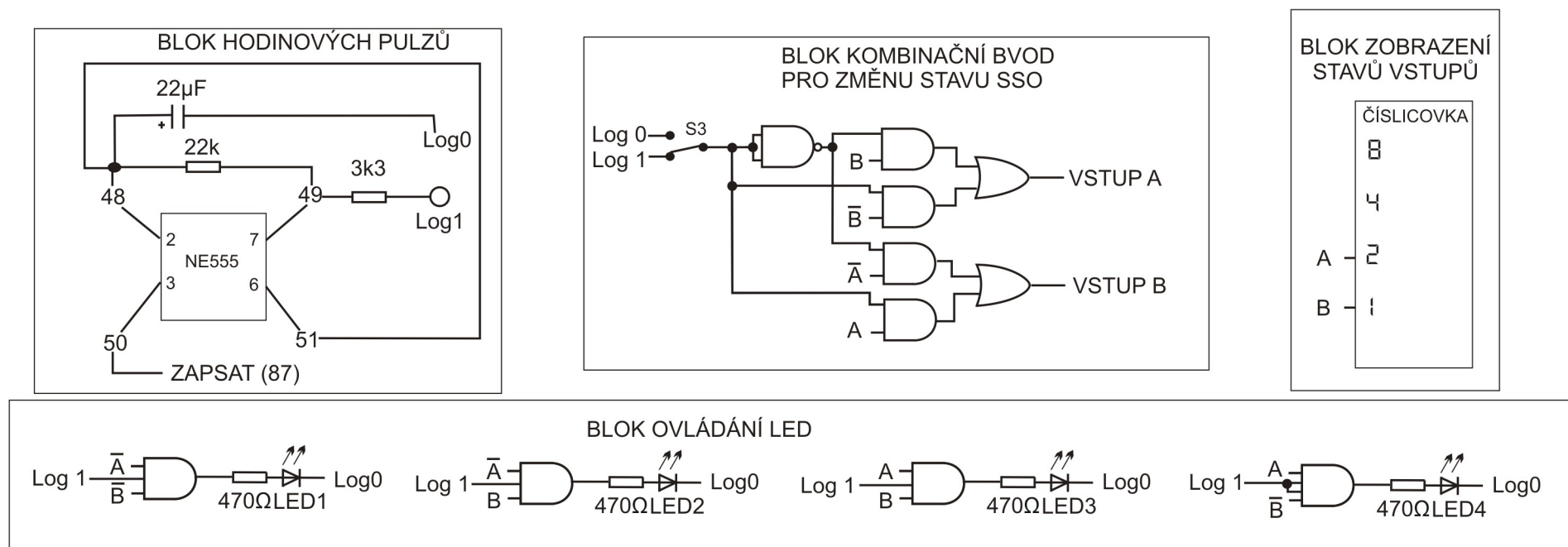
Funkce pro červenou ledku LED1 má tvar: $LED1 = A \text{ s čarou AND } B \text{ s čarou}$

Funkce pro žlutou ledku LED2 má tvar: $LED2 = A \text{ s čarou AND } B$

Funkce pro zelenou ledku LED3 má tvar: $LED3 = A \text{ s čarou AND } B \text{ s čarou}$

Funkce pro modrou ledku LED4 má tvar: $LED4 = A \text{ AND } B \text{ s čarou}$

Pro ovládání ledek potřebujeme čtyři členy AND se dvěma vstupy. Bohužel na modulu Saimon 2 máme takové členy pouze čtyři a ty jsme již použili pro zapojení kombinačního obvodu pro změnu stavů SSO. Musíme si pomoci a z kapitoly 2 už víme jak. Použijeme členy AND s více vstupy. Všechny nepoužité vstupy zapojíme na log1. Ve schématu jsem zvolil třikrát AND se třemi vstupy a jednou AND se čtyřmi vstupy. Napojení nevyužitých vstupů na log1 znázorňuje schéma na obrázku 4.6.



Obrázek 4.6 – Blokové schéma obvodu „reverzní Knight Rider“

Ještě poznámka: V bloku hodinových pulzů můžeme místo rezistoru 22k zapojit potenciometr a měnit tak rychlost blikání obvodu.

Kapitola 5 - Světelný had (testovací ledky)

Už několik kapitol známe „výhodnou“ posloupnost stavů SSO. Pro SSO se dvěma stavy A a B to byla $AB=00,01,11,10$. Kombinace dvou vstupů nám umožňuje získat čtyři různé stavy ($2 \times 2=4$).

V kapitole 2 jsem použil výhodnou posloupnost pro SSO se třemi vstupy A, B a C. Bylo to $ABC=000,001,011,111,110,100$. Kombinace tří vstupů by nám měla umožnit získat osm různých stavů ($2 \times 2 \times 2=8$). Toto výhodné zapojení jich však umožní použít pouze 6. Některé stavy totiž nejsou vhodné pro toto použití.

Zapojení kombinačního obvodu pro změnu stavu SSO pak bylo velice jednoduché. Nepotřebovali žádné složité logické funkce. Ukážeme si, jaké je výhodné zapojení, když budeme chtít použít všechny vstupy, které máme, tedy A, B, C, D. Výhodné je pořadí 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000 a pak zase od začátku. Tabulka přechodů je na obrázku 4.1.

VÝSTUPY					VSTUPY			
A	B	C	D		A	B	C	D
0	0	0	0		0	0	0	1
0	0	0	1		0	0	1	1
0	0	1	1		0	1	1	1
0	1	1	1		1	1	1	1
1	1	1	1		1	1	1	0
1	1	1	0		1	1	0	0
1	1	0	0		1	0	0	0
1	0	0	0		0	0	0	0

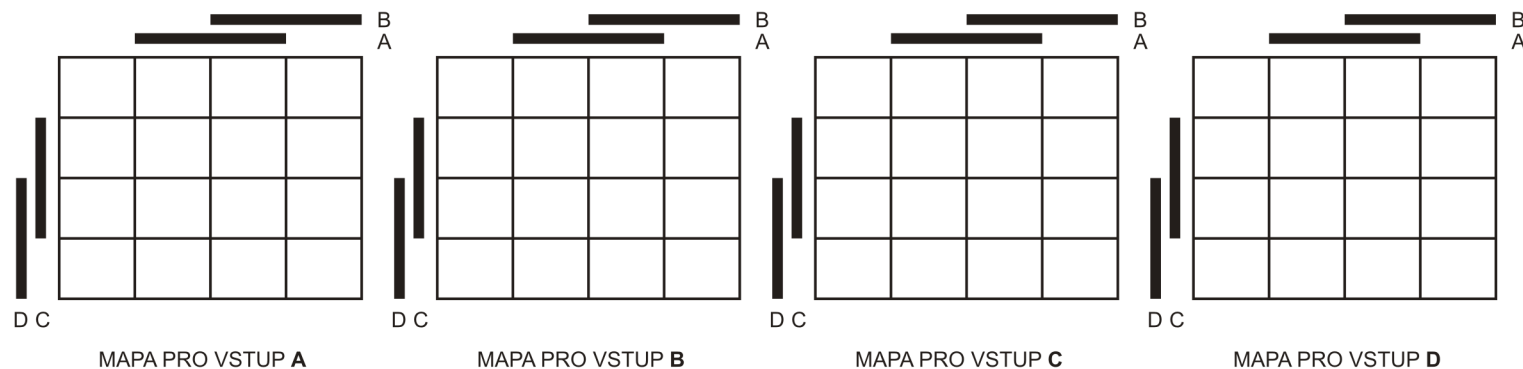
Obrázek 5.1: Tabulka přechodů SSO

V této kapitole záměrně vynechávám kreslení grafu. Nechám ho na čtenáři jako cvičení.

Cvičení 1: Nakreslete pomocí tabulky přechodů z obrázku 5.1 graf přechodů SSO.

Další krok bude opět cvičení. Z tabulky přechodů je zřejmé, že hledáme čtyři logické funkce. Potřebujeme měnit čtyři vstupy obvodu 74175. Každá funkce má vždy jen jeden výstup, takže může měnit jen jeden vstup obvodu 74175.

Cvičení 2 : Pro každou funkci vyplňte Karnaughovu mapu a nalezněte odpovídající funkci.



Obrázek 5.2 – Prázdné Karnaughovy mapy pro vyplnění

Po vyplnění map, zakroužkování skupin jedniček a vyčtení funkcí by vám musí vyjít toto:

Funkce pro změnu vstupu A paměťového obvodu 74175 má tvar: vstup A=B

Funkce pro změnu vstupu B paměťového obvodu 74175 má tvar: vstup B=C

Funkce pro změnu vstupu C paměťového obvodu 74175 má tvar: vstup C=D

Funkce pro změnu vstupu D paměťového obvodu 74175 má tvar: vstup D=A s čarou

Funkce i mapy jsou tedy velmi jednoduché, proto jsem zvolil tuto úlohu jako cvičení. Ovládání ledek bude také velmi jednoduché. Každou ledku zapojíme na jeden výstup obvodu 74175.

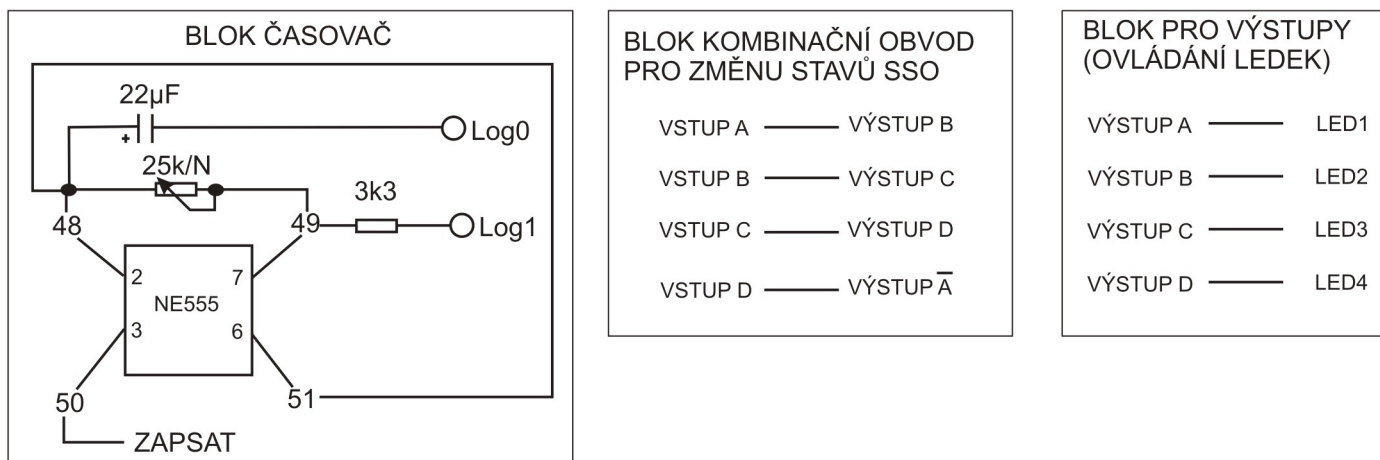
LED1=Výstup A

LED2=Výstup B

LED3=Výstup C

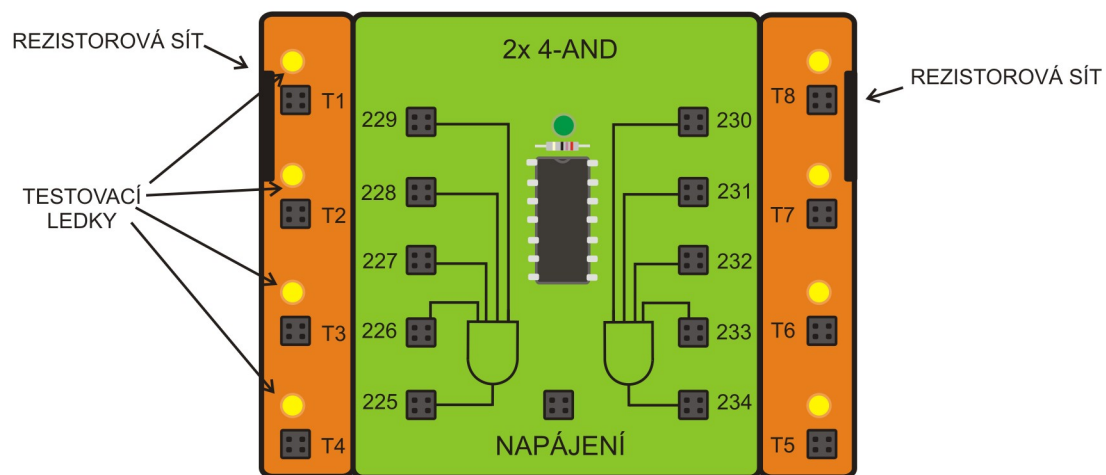
LED4=Výstup D

Zbývá jen nakreslit blokové schéma. Obvod je velmi jednoduchý. Potenciometrem můžete měnit rychlost „jízdy hada“



Obrázek 5.3 - Blokové schéma obvodu „světelný had“

Obvod můžeme ještě vylepšit zapojením dalších ledek. Na modulu Saimon 2 jsou ve dvou oranžových polích **Testovací ledky**. Jsou značené T1,T2,T3,T4,T5,T6,T7 a T8. Katody těchto ledek jsou zapojené přes rezistor na log0 zespolu stavebnice. Tyto rezistory, které omezují protékající proud, nejsou na první pohled vidět. Jsou schované v černé podélné součástce, která má název „**Rezistorová síť**“. Uvnitř této součástky jsou čtyři rezistory, pro každou ledku jeden. U druhého oranžového pole je po straně druhá taková rezistorová síť, pro druhou skupinu ledek. Jak pole ledek vypadá, je nakresleno na obrázku 5.4.



Obrázek 5.4 – Testovací ledky na modulu Saimon 2

Zapojení drátků obvodu „světelný had“ (počet zapojených drátků je 28)

Zapojení bloku hodinových pulsů: 48 - 24 , 49 - 23 , 24 - 25 , 48 - 51 , 48 - 39 , 38 - log0 , 49 - 9 , 10 - log1, 50 - 87

Zapojení bloku změn stavů SSO: 77 - 74 , 76 - 81 , 83 - 86 , 84 - 78

Zapojení bloku ovládání ledek: 85 - T5 , 82 - T6 , 75 - T7 , 78 - T8 , 79 - T1 , 74 - T2 , 81 - T3 , 86 - T4, 79 - 93, 74 - 95, 81 - 97, 86 - 99, 92 - 94, 94 - 96, 96 - 98, 98 - log0

Ostatní zapojení: Zapojit napájení všech potřebných IO

Kapitola 6 - Hra Videostop

Zapojení „světelný had“ z předchozí kapitoly můžeme upravit a vytvořit zapojení, které jsem nazval hra **Videostop**. Videostop je hra, ve které se na dvou displejích zobrazují v časových intervalech různé znaky. Cílem je stisknout spínač ve chvíli, kdy jsou znaky na obou displejích stejné. Hra se hraje buď se dvěma, nebo třemi displeji. V našem případě budeme tvořit hru pro dva displeje.

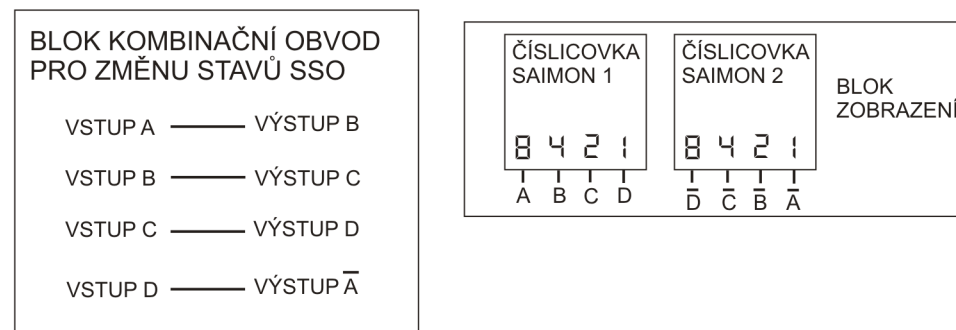
Krok 1 – Slovní popis funkce obvodu

Obvod bude zobrazovat na dvou číslicovkách postupně různá čísla. Bude existovat možnost, že se zobrazí stejná čísla na obou číslicovkách. Cílem hry bude v takovém okamžiku stisknout spínač. Pokud hráč stiskne spínač S2 ve chvíli, kdy jsou na číslicovkách stejné symboly, rozsvítí se zelená ledka - nalezena shoda. V opačném případě se rozsvítí červená ledka. Obvod bude „tikat“ při každé změně stavu.

Jelikož bez použití dalších integrovaných obvodů nemáme možnost vytvořit náhodné číslo (vyžadovalo by to generátor náhodných čísel) a také nemáme možnost porovnání, zda jsou čísla stejná (vyžadovalo by to bitový komparátor), musíme si poradit jinak. První část Videostopu je vytvořit světelného hada z předchozí úlohy. Výstupy obvodu 74175 však nepřivedeme na ledky, ale na číslicovku. Čísla, která se budou zobrazovat, nepůjdou po sobě v pořadí, na které jsme zvyklí (tedy například 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F), ale pořadí se bude zdát nesouvislé.

Stavy SSO jdou po sobě v tomto pořadí: 0000 – 0001 – 0011 – 0111 – 1111 – 1110 – 1100 – 1000 a zpět do 0000. Pokud zapojíme A – 88, B – 89, C – 90, D – 91, budou znaky zobrazené na číslicovce v pořadí 0, 1, 3, 7, F, E, C, 8.

V tomto okamžiku máme první polovinu obvodu – zobrazení znaků na prvním displeji v nesouvislém pořadí. Pro druhou část potřebujeme, aby se na druhé číslicovce měnily jiné znaky. Dále potřebujeme, aby existovala možnost, že některé znaky budou na obou číslicovkách shodné. Mě napadla možnost zapojit na druhou číslicovku komplementární výstupy obvodu 74175 a to v opačném pořadí. Schéma je na obrázku 6.1



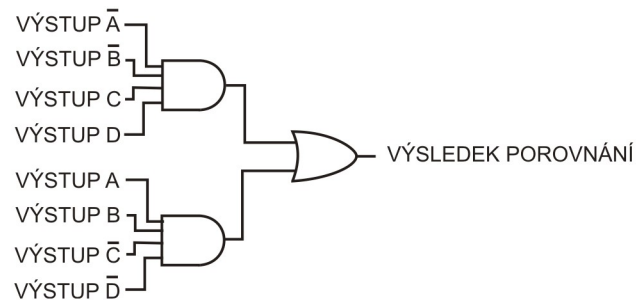
Obrázek 6.1 – Zapojení obou číslicovek a kombinačního obvodu pro změnu stavů SSO

Znaky, které budou shodné, jsou znak odpovídající stavu 0011, což je znak „3“ a znak odpovídající stavu 1100, což je znak „C“.

Dále potřebujeme obvod, který oba stavy porovná. Potřebujeme logickou funkci, která nabývá hodnoty log1 ve stavu A=0 a B=0 a C=1 a D=1, **nebo** ve stavu, kdy A=1 a B=1 a C=0 a D=0. Schválně jsem tučně označil „a“ a „nebo“. Víme, jakou logickou funkci tyto slova představují. Uměli byste logickou funkci sestavit? Při použití Karnaughových map jistě ano. Mělo by to ale dobře jít i při této slovní reprezentaci, kdy víme, že „a“ představuje funkci AND, a to v našem případě se čtyřmi vstupy. Ty budeme potřebovat dvě. Slovo „nebo“ představuje funkci OR. Kdo chce, ať si nakreslí Karnaughovy mapy, já uvedu rovnou tvar funkce.

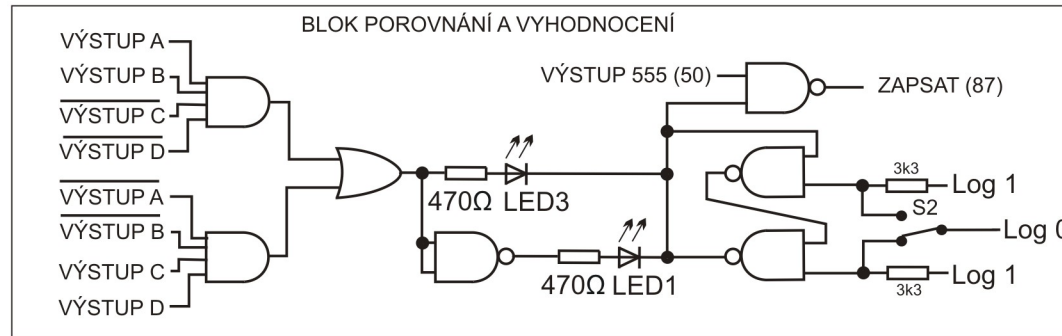
Funkce pro výsledek porovnání má tvar: Výsledek=(A s čarou AND B s čarou AND C AND D) OR (A AND B AND C s čarou AND D s čarou).

Zapojení je na obrázku 6.2.



Obrázek 6.2 – Logická funkce pro určení stavu zobrazení stejných symbolů na číslicovkách

Máme výsledek porovnání. Potřebujeme tuto funkci zapojit do obvodu tak, aby po stisknutí spínače S2 se obvod zastavil a začal signalizovat, zda jsme se trefili, nebo ne. Po stisku spínače se musí přerušit propojení s obvodem hodinových pulzů. V kapitole 5 druhého dílu návodu pro Saimon 1 jsme si ukazovali, co se stane, když jako zdroj hodinových pulzů použijeme obyčejný spínač. To stále platí, takže nelze pouze spínačem oddělit obvod hodinových pulzů od paměťového obvodu. Výsledkem by byly nevypočitatelně zrychlené změny stavu při stisknutí i uvolnění spínače (spínač „jiskří“). Použijeme obvod RS, který je v druhém díle návodu pro Saimon 1 na obrázku 5.2. Schéma našeho komparátoru (obvodu pro porovnání) je na obrázku 6.3.



Obrázek 6.3: Obvod pro porovnání a vyhodnocení stejných znaků

Nestisknutý spínač nastavuje spodní výstup RS obvodu na hodnotu log1. Tento výstup se kombinuje pomocí logické funkce NAND se signálem hodinových pulzů. Pokud má RS hodnotu log1, tak sčítání RS a hodinových pulzů funkcí NAND znamená změnu hodnoty výstupu funkce NAND.

Pokud spínač S2 stiskneme, RS bude mít hodnotu log0 a funkce NAND bude mít na jednom ze vstupů vždy log0, čili její výstup se nebude moci měnit a hra se zastaví. Zároveň se bude moci rozsvítit jedna z ledek, protože katody jsou přivedené právě na výstup RS, kde je nyní hodnota log0. Zelená se rozsvítí v případě, že komparátor má hodnotu log1 (trefa). V opačném případě se pomocí funkce NOT rozsvítí červená ledka.

Zapojení obvodu „Hra Videostop“: (počet zapojených drátků je 48)

Zapojení bloku hodinových pulzů : log0 - 40, 41 - 48, 48 - 51, 48 - 24, 25 - 49, 49 - 5, 6 - log1, 100 - 48, 101 - 49.

Zapojení bloku změn stavů: 77 - 74 , 76 - 81 , 83 - 86 , 84 - 78

Ostatní zapojení: Zapojit napájení všech potřebných IO

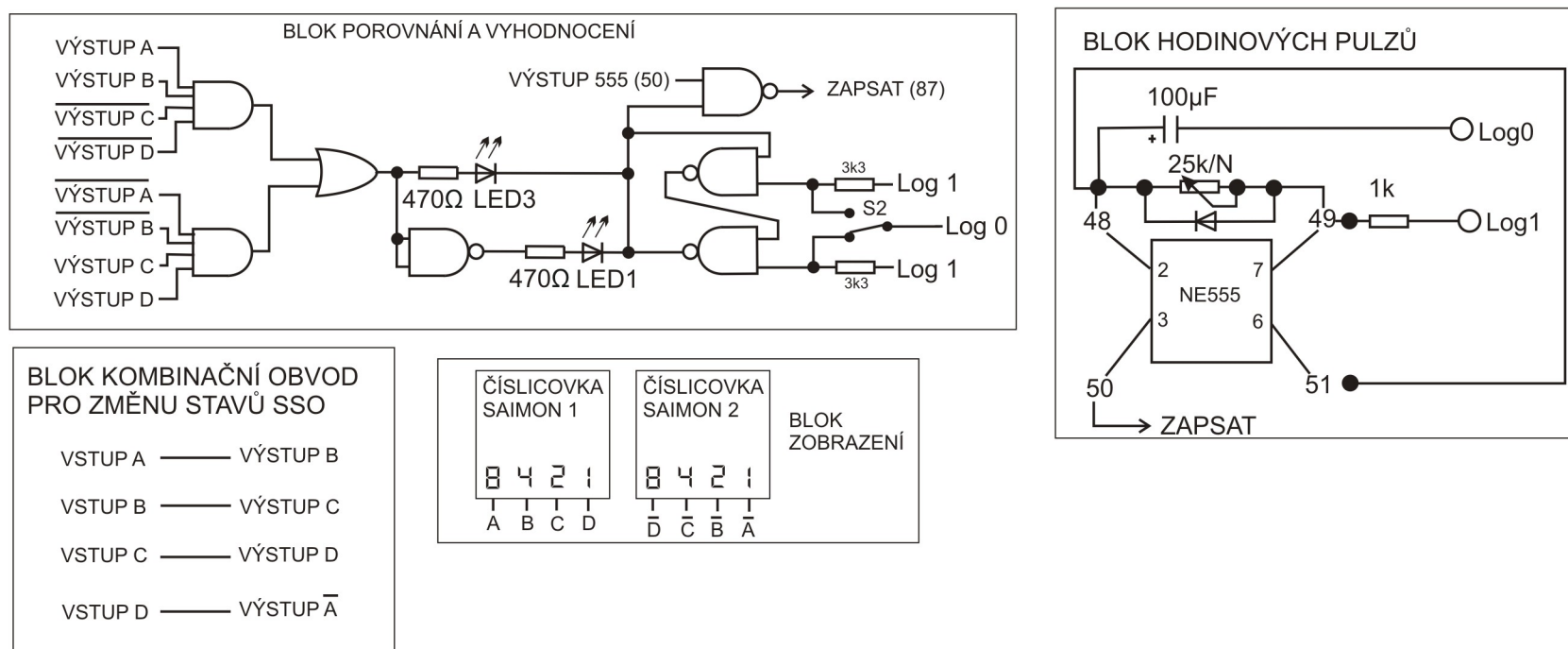
Zapojení zobrazení: 79 - 88, 74 - 89, 81 - 90, 86 - 91, 85 - 259, 82 - 260, 75 - 261, 78 - 262

Zapojení bloku porovnání: 85 - 226, 82 - 227, 79 - 229, 74 - 228, 75 - 231, 78 - 230, 81 - 232, 86 - 233, 225 - 238, 234 - 239, 240 - X6, X6 - 97, 97 - 65, 65 - 66, 67 - 93, 92 - 96, 92 - 70, 70 - 62, 64 - 68, 70 - 71, 50 - 72, 73 - 87, 121 - 69, 69 - 8, 7 - 9, 9 - log1, 63 - 10, 120 - 63, 119 - log0

Zapojení bloku zvuk: 109 - log0, 108 - 72

Hru Videostop můžeme pozměnit tím, že přepojíme drátky u číslicovek. Pokud prohodíme drátky vedoucí do zdířek 89 a 90 a zároveň 260 a 261, tak budou obě číslicovky zobrazovat trochu jiné znaky. Shoda bude tentokrát na znacích „A“ a „5“.

Na obrázku 6.4 je nakreslené celé blokové schéma obvodu. Všimněte si bloku hodinových pulzů. Použil jsem kondenzátor s kapacitou $100\mu\text{F}$, rezistor 1k a paralelně s potenciometrem je zapojena dioda. Díky ní je kratší doba, kdy svítí ledka u časovače NE555 a delší doba, kdy nesvítí. Přitom délka pulzu zůstává stejná.



Obrázek 6.4 – Blokové schéma hry Videostop

Kapitola 7 - Hra Pyrotechnik

Krok 1 – Slovní popis funkce obvodu

Pyrotechnik je osoba, která má za úkol zneškodnit časovanou bombu. Bomba tiká, odpočítává čas na počítadle a pyrotechnik musí najít správnou kombinaci tlačítek, aby bombu deaktivoval.

Cíl hry bude deaktivovat časovanou bombu dříve, než vybuchne. Bomba bude obsahovat časovač, který bude odpočítávat. Jakmile odpočet dosáhne nuly, bomba vybuchne. Úkolem pyrotechnika bude zadat správnou kombinaci, která deaktivuje odpočítávání a bomba tím bude zneškodněna.

Hru bude tvořit SSO, který bude představovat obvod pro odpočet. Pokud odpočet dosáhne čísla „0“, rozsvítí se červená ledka, signalizující vybuchlou bombu. Zadáním správné kombinace pomocí spínačů se odpočet zastaví a rozsvítí se zelená ledka. To bude znamenat, že je bomba deaktivována a vyřešili jsme hru. Během odpočtu bude obvod tikat a blikat žlutá ledka.

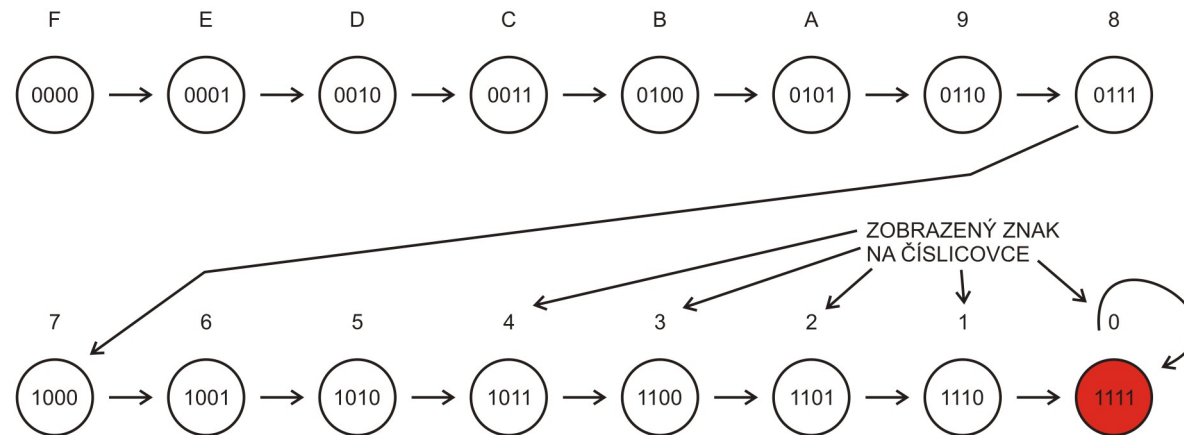
Kolik stavů budeme potřebovat a jak půjdou po sobě? V první kapitole jsme vytvořili SSO, který měl 8 stavů. Každý stav zobrazoval číslo na číslicovce. Číslo se měnilo v pořadí od 0 do 7. Čísel umíme zobrazit i více. Pamatujete si kolik jich je celkem včetně písmen?

Je to 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Zkusíme navrhnout SSO, který by zobrazoval čísla od „F“ do „0“. Ve stavu 0000 se rozsvítí červená ledka. Správná kombinace spínačů přeruší přívod hodinových pulsů a rozsvítí zelenou ledku.

První krok je navrhnout počítadlo od 0 do F. Musíme ho udělat takto, začít od 0, protože obvod 74175 bude po mít po zapnutí a smazání nastaveny výstupy na log0. Pro zobrazení znaků na číslicovce vytvoříme kombinační obvod pro ovládání číslicovky, který ve stavu 0000 zobrazí „F“, ve stavu 0001 zobrazí „E“ a tak dále.

Pokud bomba vybuchne, je pro další hru třeba ji restartovat. K tomu zapojíme spínač S1, který vynuluje obvod 74175.

Krok 2 – nakreslíme si, jak jdou stavy po sobě



Obrázek 7.1 – Graf přechodů počítadla bomby

SSO bude procházet stavy podle grafu na obrázku 7.1. Jakmile dosáhne stavu 1111, musí v něm již zůstat a také se musí rozsvítit červená ledka. Nad každým stavem je v grafu nakreslený znak, který má být zobrazen na číslicovce. Z grafu vyčteme tabulku přechodů.

Krok 3 – z grafu vyčteme tabulku přechodů a pravdivostní tabulky pro kombinační obvod změny stavů

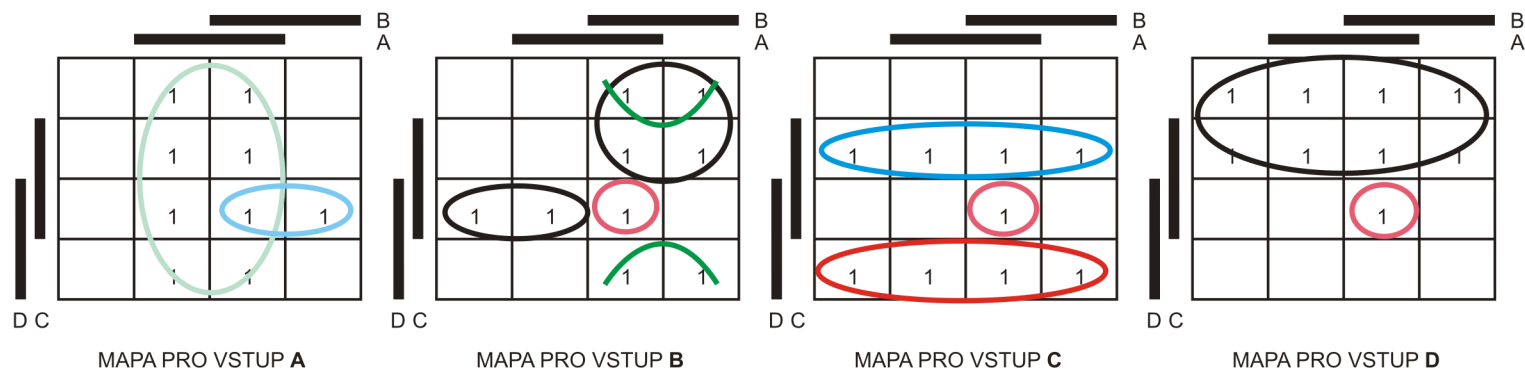
VÝSTUPY 74175				BUDOUCÍ (NÁSLEDUJÍCÍ VSTUPY STAV SSO)				
(SOUČASNÝ STAV SSO)	A	B	C	D	A	B	C	D
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	1
0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	0	1	1
0	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	1	1
0	1	1	1	1	0	0	0	0

VÝSTUPY 74175				BUDOUCÍ (NÁSLEDUJÍCÍ VSTUPY STAV SSO)				
(SOUČASNÝ STAV SSO)	A	B	C	D	A	B	C	D
1	0	0	0	1	0	0	0	1
1	0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	1	1
1	0	1	1	1	1	1	0	0
1	1	0	0	1	1	1	0	1
1	1	0	1	1	1	1	1	0
1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Obrázek 7.2 – Tabulka přechodů stavů SSO

Pravdivostní tabulky pro jednotlivé funkce opět vynechám, protože Karnaughovy mapy lze vyplnit přímo z tabulky přechodů. Do map budu zapisovat pouze jedničky, nuly vynechám pro větší přehlednost.

Krok 4 – Sestavit logické funkce podle pravdivostních tabulek



Obrázek 7.3 – Karnaughovy mapy pro funkce kombinačního obvodu změn stavů SSO

Je vidět, že počítaadlo je složitý obvod. Funkce budou mít hodně členů, naštěstí se některé budou opět opakovat. Zakroužkoval jsem barevně jednotlivé skupiny jedniček, tedy členy logických funkcí. Stejně barevně je označím v definici níže. Díky barvám bude lépe vidět, které funkce mají stejné členy (tedy bude stačit takový člen zapojit pouze jednou).

Funkce pro změnu vstupu A paměťového obvodu 74175 má tvar:
vstup A = A OR (B AND C AND D)

Funkce pro změnu vstupu B paměťového obvodu 74175 má tvar:
vstup B = (A AND B AND C AND D) OR (B AND D s čarou) OR (B s čarou AND C AND D) OR (B AND C s čarou)

Funkce pro změnu vstupu C paměťového obvodu 74175 má tvar:
vstup C = (A AND B AND C AND D) OR (C AND D s čarou) OR (C s čarou AND D)

Funkce pro změnu vstupu D paměťového obvodu 74175 má tvar:
vstup D = (A AND B AND C AND D) OR D s čarou

Způsobů, jak zakroužkovat skupiny jedniček tak, aby byly zakroužkovány všechny jedničky alespoň jednou, tak těch naleznete určitě více. Některé budou výhodnější, některé méně výhodné. Tomu se říká **problém pokrytí**. Nalézt nejvýhodnější pokrytí je velmi složitý výpočetní proces. Já jsem na tohle téma psal diplomovou práci. Kdo si bude zkoušet vyplňovat mapy sám, zřejmě nalezne jiné skupiny jedniček a tím pádem funkce budou mít trochu jiné členy.

Jedna jednička je zakroužkovaná samotná. Je to člen **A AND B AND C AND D**. Tato jednička se vyskytuje ve třech mapách a ve všech by šla zvolit větší skupina jedniček, a to dvou jedniček. Důvod, proč jsem zakroužkoval pouze tuto jedničku, je ten, že pro funkci časovače potřebujeme zjistit, že dosáhl stavu 1111, který bude zobrazovat znak „0“. A to bude znamenat, že bomba vybuchla. Tento člen bude mít hodnotu log1 právě v okamžiku, kdy SSO bude ve stavu 1111. Na výstup tohoto členu napojíme červenou ledku, která bude symbolizovat vybuchlou bombu.

Krok 5 – Napsat pravdivostní tabulky pro logické funkce kombinačního obvodu ovládající výstupní obvod

Do tabulky napíšu pro každý stav SSO znak, který chceme zobrazit na číslicovce. Které znaky to jsou je již nakresleno na obrázku 7.1.

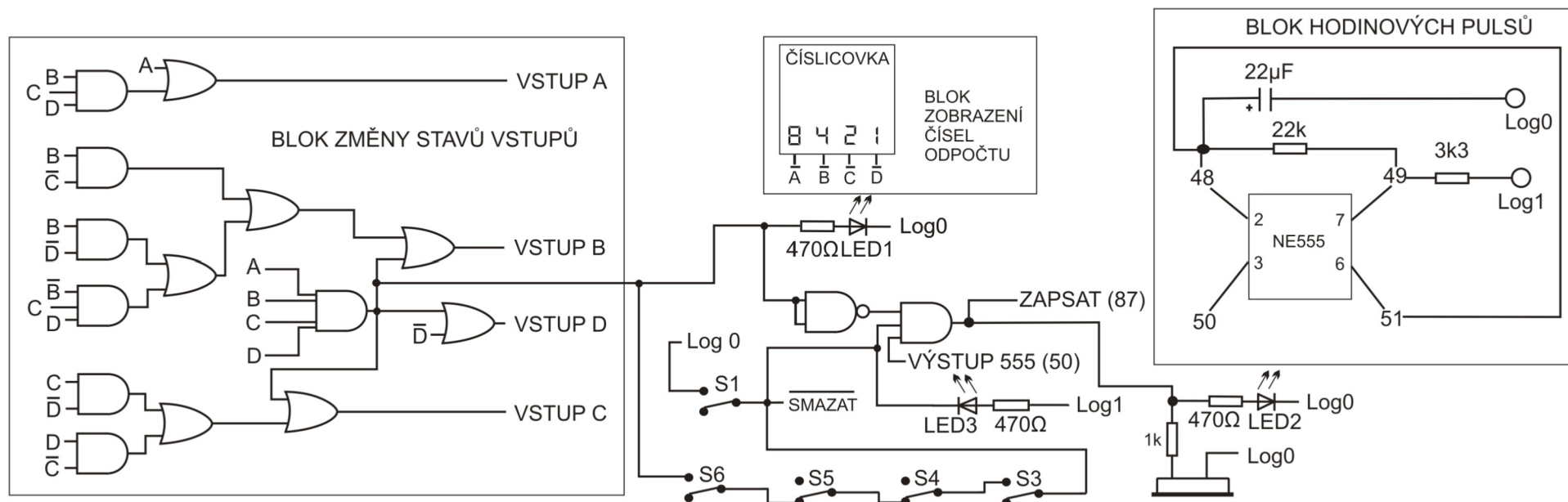
VÝSTUPY 74175				ZOBRAZENÝ ZNAK			
A	B	C	D	8	4	2	1
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	0

VÝSTUPY 74175				ZOBRAZENÝ ZNAK			
A	B	C	D	8	4	2	1
1	0	0	0	0	1	1	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0

Obrázek 7.4 – Pravdivostní tabulka pro zobrazení znaku

Krok 6 – Navrhnout kombinační obvod pro výstupní obvod

I když bychom mohli pomocí tabulky vyplnit čtyři Karnaughovy mapy a určit čtyři funkce, tak tentokrát si ušetříme čas. Podívejte se na tabulku na obrázku 7.4 a zjistíte, že zobrazený znak má vždy opačné logické hodnoty jednotlivých bitů, než má stav SSO. Například stav 0000 má zobrazit znak 1111. Stav 0110 má zobrazit znak 1001. A to platí pro každý znak. Stačí tedy na číslicovku přivést komplementární výstupy obvodu 74175 a počítadlo bude počítat od F do 0. Kdo by přesto vyplnil Karnaughovy mapy, vyjde mu přesně to samé. Zbývá jen nakreslit blokové schéma obvodu.



Obrázek 7.5: Blokové schéma hry Pyrotechnik

Jak funguje blok změny stavů vstupů je asi jasné, stejně jako blok zobrazení, časovač a zvuk. Zbylé zapojení funguje následovně. Červená LED1 je napojena na výstup funkce AND, která říká, že bomba vybuchla (počítadlo je na nule a výstupy ABCD=1111). Výstup pokračuje dál přes funkci NAND do funkce AND se třemi vstupy. Ta má za úkol rozsvěcet žlutou LED2, která svítí (bliká) pouze pokud nesvítí červená LED1, není bomba zneškodněná kombinací tlačítek a pokud výstup z časovače je roven log1 (při hodinovém pulzu). Když bliká žlutá LED2, tak bomba tiká, zapojení pokračuje do reproduktoru. Mezi reproduktor a hodinový pulz jsem zapojil rezistor 1k, protože bez tohoto rezistoru protékal reproduktorem veliký proud, který zkresloval hodinový pulz a odpočítávání pak přeskakovalo některé číslice.

Spínačem S1 resetujeme obvod 74175 a tím i hru. Kombinace tlačítek je zapojená tak, že jejich stiskem se musí spojit obvod, který přivede log0 na zdířku 80 (SMAZAT), rozsvítí zelenou LED3 a zabrání blikání LED2 přes zmíněný AND se třemi vstupy. Log 0 je možné přivést jen pokud bomba nevybuchla, proto je tlačítko S6 napojené na výstup funkce AND se čtyřmi vstupy (který vede i na LED1). U použitého zapojení je vidět, že musíme zmáčknout S3 a S6, aby se obvod propojil a bomba deaktivovala. Spínače můžete zapojit i jinak, než je ve schématu a vytvořit si svou kombinaci pro deaktivaci bomby.

Zapojení drátků obvodu „Hra pyrotechnik“ (počet zapojených drátků je 72)

Zapojení bloku změn stavů SSO: 209-74, 210-81, 211-86, 212-219, 220-79, 221-77, 79-229, 74-228, 81-227, 86-226, 74-251, 85-250, 251-247, 82-248, 75-201, 210-207, 211-202, 225-222, 249-223, 224-213, 208-214, 215-216, 252-217, 76-218, 225-239, 85-238, 84-240, 239-245, 81-253, 250-254, 248-256, 86-257, 258-242, 255-241, 243-244, 83-246

Zapojení bloku hodinových pulsů: log0 - 38, 39 - 48, 48 - 51, 48 - 13, 14 - 49, 14 - 7, 8 - log1, 50-X1, X1-204

Zapojení ovládání: 225-X6, X6-93, 92-log0, 97-log1, 80-96, 122-116, 117-log0, 92-94, 96-203, 87-206, 95-87, 93-68, 68-69, 70-X2, X2-205

Zapojení zobrazení: 78-88, 75-89, 82-90, 85-91

Zapojení kombinace tlačítek: 132-93, 131-130, 128-127, 125-123, 122-80

Zapojení bloku zvuk: 108-5, 6-87, 109-log0.

Pro hru pyrotechnik je použito 72 drátků. Při zapojování dávejte dobrý pozor, protože v takovém množství drátků se špatně hledají chyby. Propojky X vždy používejte na tom modulu Saimon, na kterém je zdířka, se kterou je propojka spojená. Například 225-X6 znamená propojku X6 na modulu Saimon 2. X6-93 znamená použití propojky X6 na modulu Saimon 1.

Obsah

Úvod.....	2
Kapitola 1 – Počítadlo od 0 to 7.....	3
Kapitola 2 – Knight rider (don't care stavy,použití propojek).....	16
Kapitola 3 – Hádanka „Dveře do trezoru“ (použití spínačů pro ovládnání SSO).....	26
Kapitola 4 – Reverzní „Knight Rider“.....	34
Kapitola 5 - Světelný had (testovací ledky).....	39
Kapitola 6 - Hra Videostop.....	44
Kapitola 7 - Hra Pyrotechnik.....	48
Slovník pojmů.....	56

Slovník pojmů

(přidržením ctrl + klik myši na tučná slovíčka se v návodu přesunete na místo jeho prvního popisu)

Karnaughova mapa	Tabulky, pomocí kterých získáme členy logické funkce a funkci sestavíme
Člen logické funkce	Skupina jedniček v Karnaughově mapě. V obvodu je člen tvořen logickou funkcí AND nebo NAND. Jednotlivé členy se sčítají funkcí OR. Skupinu tvoří 1,2,4 nebo 8 jedniček.
Don't care	Stav výstupu logické funkce, kdy nezáleží na jeho hodnotě, protože kombinace vstupů pro tento stav nikdy nenastane. Můžeme uvažovat jeho hodnotu log1. Hledaný člen funkce bude jednodušší. Tento stav značíme X.
propojky X	Abychom nemuseli propojovat vzdálené zdířky mezi moduly dlouhými drátky, jsou moduly mezi sebou propojeny zespolu. Tyto propojení jsou značené X1,X2,X3,X4,X5,X6 a jsou na každém modulu vyvedeny.
Testovací ledky	Skupina 2x čtyř ledek na modulu Saimon 2, označených T1 až T8.
Rezistorová síť	Součástka obsahující více rezistorů. Rezistory mají buď jeden vývod společný (síť typu A), nebo má každý rezistor dva vývody odděleně od ostatních rezistorů (síť typu B)