

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

Příprava komponent pro e-kurz Příklady ČPS
BAKALÁŘSKÁ PRÁCE

Martin Dolejš
Informatika se zaměřením na vzdělání

Vedoucí práce: Ing. Petr MICHALÍK, Ph.D.

Plzeň, 2015

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

Plzeň, 23. června 2015

.....
Martin Dolejš

Poděkování

Chtěl bych tímto poděkovat vedoucímu bakalářské práce Ing. Petru Michalíkovi, Ph.D. za odborné vedení, podnětné připomínky a konzultace.

Originál zadání práce (student obdrží od svého vedoucího práce).

OBSAH

Úvod	1
1. GRAFICKÉ PRVKY	3
1.1 ANIMACE	3
1.2 OBRÁZKY	3
2. PROGRAMY PRO TVORBU KOMPONENT	4
2.1 PROGRAM ADOBE CAPTIVATE	4
2.2 PROGRAM MULTISIM	5
2.3 PROGRAM PROAUTHOR	6
3. TVORBA KOMPONENT	8
3.1 KOMPONENTY E-KURZU PRO CVIČENÍ 1 - ZÁKLADNÍ LOGICKÉ FUNKCE	10
3.1.1 Zadání	10
3.1.2 Vypracování simulace v Multisimu	11
3.1.3 Ukázka obrázku	11
3.1.4 Ukázka animace	12
3.2 KOMPONENTY E-KURZU PRO CVIČENÍ 2 - ROVNICE A FUNKCE	12
3.2.1 Zadání	12
3.2.1 Vypracování simulace v Multisimu	13
3.2.2 Ukázka obrázku	14
3.2.1 Ukázka animace	15
3.2.2 Otázky autotestu pro cvičení 1 a 2	15
3.3 KOMPONENTY E-KURZU PRO CVIČENÍ 3 - ÚPLNÝ DISJUNKTNÍ A KONJUNKTNÍ TVAR	16
3.3.1 Zadání	17
3.3.2 Vypracování simulace v Multisimu	17
3.3.3 Ukázka animace	18
3.3.4 Ukázka obrázku	19
3.3.5 Otázky autotestu pro cvičení 3	19
3.4 KOMPONENTY E-KURZU PRO CVIČENÍ 4 - MINIMALIZACE DISJUNKTNÍHO A KONJUNKTNÍHO TVARU	20
3.4.1 Zadání	20
3.4.2 Vypracování simulace v Multisimu	21
3.4.3 Ukázka obrázku	21
3.4.4 Ukázka animace	22
3.4.5 Otázky autotestu pro cvičení 4	22
3.5 KOMPONENTY E-KURZU PRO CVIČENÍ 5 - HOMOGENNÍ STRUKTURY LOGICKÉ FUNKCE	23
3.5.1 Zadání	24
3.5.2 Vypracování simulace v Multisimu	24
3.5.3 Ukázka obrázku	25
3.5.4 Ukázka animace	25
3.5.5 Otázky autotestu pro cvičení 5	26
3.6 KOMPONENTY E-KURZU PRO CVIČENÍ 6 - INTEGROVANÉ OBVODY	27
3.6.1 Zadání	28
3.6.2 Vypracování simulace v Multisimu	28
3.6.3 Ukázka animace	29
3.6.4 Ukázka obrázku	30
3.6.5 Otázky autotestu pro cvičení 6	30
3.7 KOMPONENTY E-KURZU PRO CVIČENÍ 7 - NEURČITÉ STAVY A MINIMALIZACE	31

3.7.1	Zadání	32
3.7.2	Vypracování simulace v Multisimu	32
3.7.3	Ukázka obrázku.....	32
3.7.4	Ukázka animace.....	33
3.7.5	Otázky autotestu pro cvičení 7.....	33
3.8	KOMPONENTY E-KURZU PRO CVIČENÍ 8 - VYUŽITÍ DEKODÉRŮ K REALIZACI LOGICKÝCH FUNKCÍ.....	34
3.8.1	Zadání	34
3.8.2	Vypracování simulace v Multisimu	35
3.8.3	Ukázka obrázku.....	35
3.8.4	Ukázka animace.....	36
3.8.5	Otázky autotestu pro cvičení 8.....	36
3.9	KOMPONENTY E-KURZU PRO CVIČENÍ 9 - VYUŽITÍ MULTIPLEXERU K REALIZACI LOGICKÝCH FUNKCÍ	37
3.9.1	Zadání	38
3.9.2	Vypracování simulace v Multisimu	38
3.9.3	Ukázka animace.....	39
3.9.4	Ukázka obrázku.....	40
3.10	KOMPONENTY E-KURZU PRO CVIČENÍ 10 - REALIZACE LOGICKÉ FUNKCE POMOCÍ MULTIPLEXERU S TŘÍ- STAVOVÝMI VÝSTUPY	41
3.10.1	Zadání	41
3.10.2	Ukázka obrázku.....	41
3.10.1	Vypracování simulace v Multisimu	42
3.10.2	Ukázka animace.....	42
3.10.3	Otázky autotestu pro cvičení 9 a 10	43
3.11	KOMPONENTY E-KURZU PRO CVIČENÍ 11 - REALIZACE LOGICKÉ FUNKCE POMOCÍ POLOVODIČOVÉ PAMĚTI	44
3.11.1	Zadání	45
3.11.2	Vypracování simulace v Multisimu	46
3.11.3	Ukázka animace.....	47
3.11.1	Ukázka obrázku.....	48
3.11.2	Otázky autotestu pro cvičení 11.....	48
	ZÁVĚR.....	50
	RESUMÉ.....	52
	SEZNAM OBRÁZKŮ A TABULEK	53
	SEZNAM POUŽITÉ LITERATURY	54
	PŘÍLOHY.....	55

Úvod

Tato bakalářská práce se bude zabývat tvorbou komponent pro internetový výukový e-kurz. Využívání e-kurzů a samostatné internetové výuky v dnešní době stále roste a je velmi populární nejen mezi studenty a učiteli, ale i v soukromých a státních sektorech na vzdělávání a zaučování zaměstnanců. Hlavním důvodem oblíbenosti u studentů na školách je, že si mohou zvolit své tempo výuky a mohou si zvolit čas, kdy si kurz spustí, protože většina kurzů jsou přístupné neomezeně dlouho dobu. Pro učitele je náročné kurz vytvořit, ale je to jistě dobrá investice do budoucnosti, protože jakmile kurz vytvoří, mohou ho použít nesčetněkrát a ušetří si spoustu času. Případné změny, nebo úpravy lze provést prakticky ihned a okamžitě se projeví u studentů. Možnosti automatických oprav jsou další součástí, která šetří čas. Mnoho e-kurzů je pouze textových, bez jakýchkoliv grafických prvků, ale pro lepší pochopení, znázornění i zapamatování je doporučováno používat animace a obrázky. A právě animace a obrázky budou studenta provázet jednotlivými cvičeními v konečném výsledku této práce.

Toto téma mě zaujalo zejména z důvodu, že rád pracuji s úpravou videa a jinými grafickými prvky. Animace jsou v dnešní době relativně rychlým a užitečným pomocníkem ve výuce na školách, zejména z důvodu demonstrování jednotlivých principů a systémů. Pomocí obrázků, animací a doprovodných textů, které budou výsledkem této bakalářské práce by studenti měli bez problémů samostatně zvládnout a pochopit princip jednotlivých cvičení.

Veškeré grafické komponenty budou vytvářeny v profesionálním programu Adobe Captivate verze 8. Tento program byl vytvořen k tvorbě e-learningových kurzů a je tedy vhodným softwarem pro tuto práci. Program disponuje množstvím různých funkcí, od základních až po profesionální. Zde byly použity zejména základní funkce, jako je snímání obrazovky, tvorba popisků a zvýrazňovačů, které zaručují lepší přehlednost a orientaci v animacích.

Zapojení jednotlivých obvodů bude demonstrováno v programu Multisim, který je základním stavebním kamenem těchto cvičení. Program dokáže simulovat chování téměř každé elektronické součástky a i celých elektronických obvodů. Poslouží nám tedy výborně pro naše účely. Byla zvolena verze 10, která je shodná s verzí na školních

počítačích. Díky tomu by měla být prostředí téměř totožná a pomoci studentům v orientaci.

Na závěr bude použit program ProAuthor, který je vyvinutý Západočeskou univerzitou a zabývá se tvorbou výukových kurzů. Tento program se na katedře výpočetní a didaktické techniky používá hojně, takže prostředí výukového kurzu bude známé a věřím, že všichni studenti budou umět rychle a intuitivně pracovat s kurzem.

Hlavním účelem této bakalářské práce bude vytvořit animační videa příkladů pro jednotlivá cvičení do e-kurzu k předmětu Číslicové prvky a systémy. Dalšími nezbytnými komponentami bude vytvoření zadání, ve kterém budou napsány nezbytné požadavky ke splnění semináře. Ke každému cvičení dále vytvořit, obrázky, které budou pomáhat při návrhu řešení. A jako poslední tvorba auto-testů, respektive vymyšlení otázek a odpovědí, které budou použity v programu ProAuthor. Cílem této práce je připravit podklady pro tvorbu výukového kurzu, který bude sloužit pro efektivnější výuku semináře předmětu Číslicové prvky a systémy.

1. GRAFICKÉ PRVKY

1.1 ANIMACE

Animace jsou nezbytnou součástí každého cvičení. Je to proces vytváření na první pohled pohybujících se objektů, ve skutečnosti jde ale o jednotlivé snímky, promítané rychle za sebou. Vlivem setrvačnosti lidského oka tvoří dojem pohybu. Ekvivalentem slova animace je oživení. Byl zvolen prvek animace, protože názorné řešení problému mnohem lépe pomůže studentům pochopit danou problematiku.

Všechny animace byly tvořeny v rozlišení XGA - 1024x768 a exportovány do formátu SWF, který je vhodný pro zobrazování na webových stránkách. Vzhledem k tomu, že výsledný e-kurz bude umístěn na webových stránkách katedry je použití tohoto formátu téměř samozřejmostí. Dalším důvodem použití tohoto formátu je implementace v programu ProAuthor. Formát swf byl vytvořený v roce 1996, nyní patří s výhradním právem firmě Adobe. V dřívějších verzích byl tento formát určen pouze pro zobrazování vektorových obrázků. Jak se ale formát vyvíjel, tak byly přidávány různé implementace a dnes umí mnoho dalších funkcí, jako je například zobrazování videí, přehrávání hudebních souborů a zejména užitečné interakce s uživatelem. Formát je určen především pro multimedia a vektorovou grafiku. Právě ve formátu swf budou všechny vytvořené animace.

Na začátku každé animace se zobrazí název daného cvičení. Animace se na začátku zastaví a nespustí se, dokud student neklikne na tlačítko „Play“. Hlavním smyslem všech animací, je vysvětlit zapojení jednotlivých příkladů, funkce jednotlivých součástí a pochopení probírané látky. Veškeré animace se zdrojovými soubory jsou na přiloženém CD.

1.2 OBRÁZKY

Obrázky zobrazují jednotlivé součástky, příkladné zapojení a pro použití v ProAuthoru i některé rovnice a tabulky. Veškeré obrázky byly vytvářeny pomocí klávesy sejmutí obrazovky (PrintScreen) a následné ořezání v malování nebo v programu Microsoft Word. Všechny vytvořené obrázky jsou ve formátu png a dostupné na přiloženém CD.

2. PROGRAMY PRO TVORBU KOMPONENT

Jednotlivé komponenty byly vytvořeny za pomoci následujících tří programů. První z nich je program Adobe Captivate, který byl použit na zachytávání dění na obrazovce, tvorbu animací a následně jejich úpravu. Druhý je simulační program Multisim, ve kterém se vytváří veškerá zapojení a který je stěžejní pro seminář předmětu ČPS a poslední je vývojářský program ProAuthor pro tvorbu e-kurzu vytvořený a spravovaný Západočeskou univerzitou.

2.1 PROGRAM ADOBE CAPTIVATE

Pro tvorbu jednotlivých animací byl vybrán program od společnosti Adobe. Program Adobe Captivate 8 představuje vytváření, zpracovávání a údržbu profesionálního e-learningu. Není nutné umět programovat, protože prostředí je velice intuitivní. Je to výborný nástroj pro tvorbu výukových e-kurzů. V oboru zachytávání obrazovky a tvorby výukových videí bychom asi těžko hledali vhodnější nástroj.

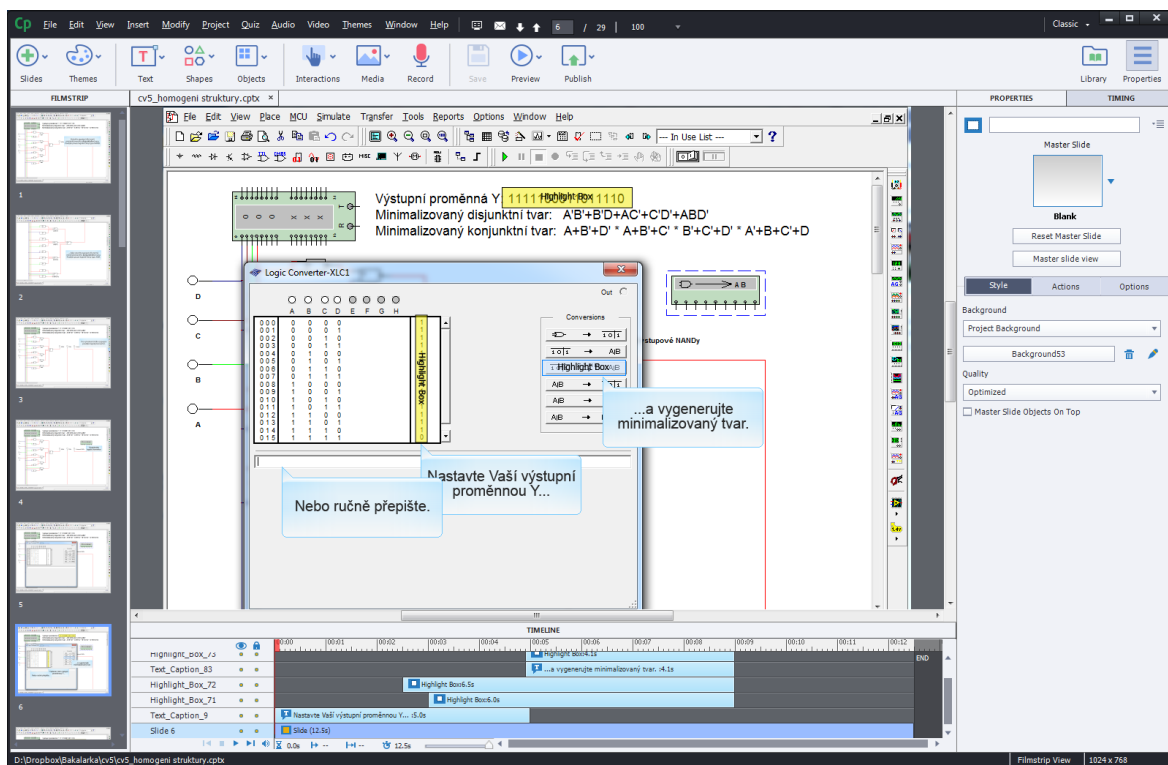
Pro tvorbu animací má program Adobe Captivate několik přednastavených režimů. Použitý režim Software Simulation zachytává veškeré dění ve snímané oblasti. Animace byly tvořeny v módu demonstrace. Tento mód ovšem zachytává úplně každou interakci, nebo pohyb myši, takže výsledná animace je velmi dlouhá a nepřehledná. Důsledkem toho byla každá animace v programu upravena tak, aby byla přehledná, pokud možno co nejkratší a přitom dostatečně vysvětlila problematiku každého příkladu.

Tvorbu každé animace doprovázelo dodržování několika zásad. Každá animace se na začátku zastaví a spustí se až poté, co student klikne na tlačítko Play. U každé animace je před spuštěním název cvičení, ke kterému se vztahuje. Jednotlivé použité obrázky pro zobrazení názvu, byly vytvořeny v programu Adobe Captivate a poté byly jen exportovány do formátu png. V konkrétních částích byly použity zvýrazňovače, aby se student mohl lépe orientovat v tom, na co se má zrovna zaměřit. Všechny zvýrazňovače mají žlutou barvu s průhledností 50% a černý okraj, jsou tedy dostatečně odlišené od pozadí a rychle upoutají pozornost. Pro zadávání textů byly použity informační bubliny. Před každým zobrazením bubliny je prodleva 0.2 sekundy, která

zajišťuje, že upoutá pozornost očí. Pro jednotný vzhled přehrávače byl u všech animací použit Captivate default.



Obrázek 1 - vzhled přehrávače - Captivate default

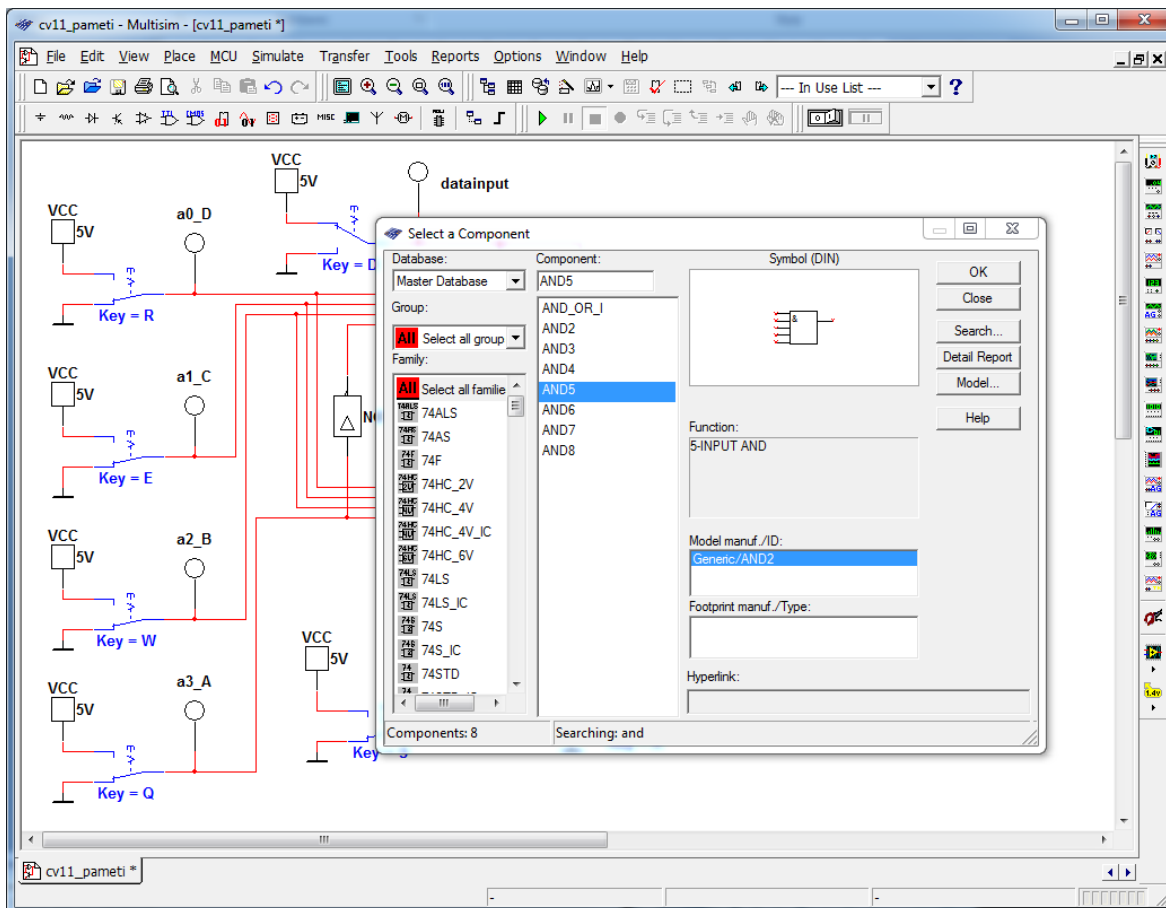


Obrázek 2 - prostředí programu Adobe Captivate 8

2.2 PROGRAM MULTISIM

Jedná se o simulační elektronický program vyvinutý firmou National Instruments. Dokáže simulovat jednoduché i složité elektrické obvody a zobrazit je až ve 3D. Pro účely této práce se ovšem bude používat pouze integrovaná knihovna součástek a zobrazování ve 2D. V základní knihovně jsou součástky s přesně danými parametry, které se průmyslově vyrábějí a součástky u kterých můžeme parametry nastavovat a měnit. Tento program se používá nejen ve výuce, ale také v průmyslu při výrobě složitých elektronických obvodů. Kromě základních elektronických součástek jsou zde i složité součástky, integrované obvody a měřicí přístroje, které lze vložit na pracovní plochu a pracovat s nimi jako v reálném prostředí. Znalosti programu Multisim jsou nezbytnou součástí pro absolvování předmětu. Kvůli zvyklostem na katedře byla zvolena starší verze

programu, konkrétně verze 10, která je instalována na školních počítačích a prostředí, odkazy a veškeré náležitosti se shodují. Veškeré úkony a úlohy budou prováděny právě v tomto programu. U každého cvičení je podrobně popsáno jak postupovat při tvorbě jednotlivých zapojení.



Obrázek 3 - prostředí programu Multisim

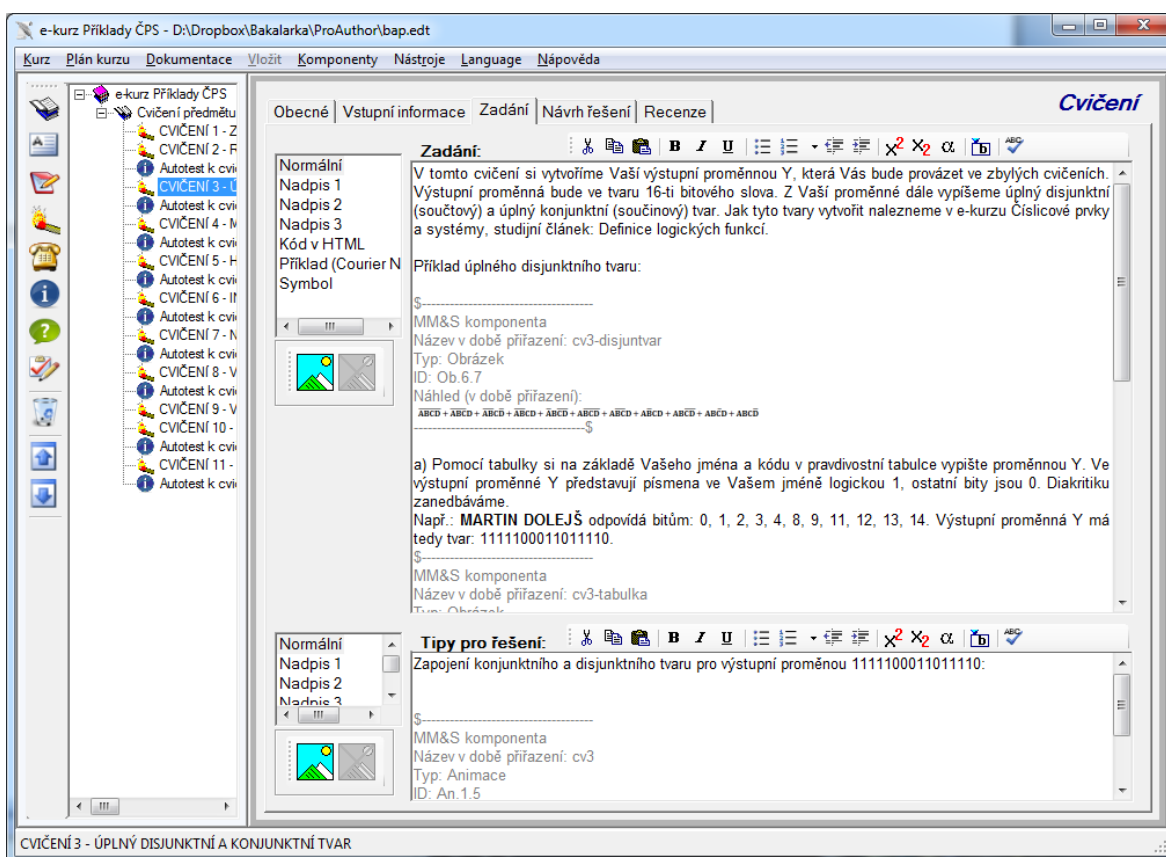
2.3 PROGRAM PROAUTHOR

Tento program je vytvořený Západočeskou univerzitou a jsou v něm vytvářeny téměř veškeré e-kurzy na katedře výpočetní a didaktické techniky. Vzhledem k těmto okolnostem je tento program vhodnou volbou. Byla použita nejnovější verze, která je k dostání na stránkách univerzity¹. Základní demoverze má omezený počet aktivit, které lze vytvořit. Po konzultaci s vývojáři programu mi bylo vyhověno a zasláno licenční číslo určené pro studenty. Pro správné zobrazování a fungování e-kurzu je požadováno mít prohlížeč Internet Explorer alespoň verze 8 a v prohlížeči nainstalovaný Flash Player

¹ <http://athena.zcu.cz/ProAuthor/>

alespoň verze 10. Použití ProAuthoru zaručuje dodržování standardů, se kterými jsou již studenti a zaměstnanci na katedře seznámeni.

V ProAuthoru byl zvolen typ aktivity cvičení. Jednotlivé komponenty byly rozděleny do cvičení, dle toho jak jsou uvedeny v harmonogramu předmětu. Každé cvičení začíná zadáním, v tipech pro řešení je vložena animace s postupem a vysvětlením fungování a na konci je návrh řešení, který je reprezentován jako obrázek zapojení. Dále jsou vkládány tabulky, součástky nebo schémata subobvodů. Dále byly vytvořeny sady otázek ke každému cvičení a vloženy do programu jako autotesty, aby si mohli studenti zkontrolovat naučené znalosti. Některá cvičení se zabírají stejnou problematikou, tak mají tyto cvičení jeden souhrnný autotest. U autotestů je možnost odpovědi přesnou odpovědí (text správně), výběrem správné odpovědi, výběrem více správných odpovědí, ano/ne. U některých otázek, kde to má smysl je vysvětleno proč tomu tak je. Veškeré otázky, obrázky a animace jsou v knihovně vytvořeného kurzu, který je na přiloženém CD.



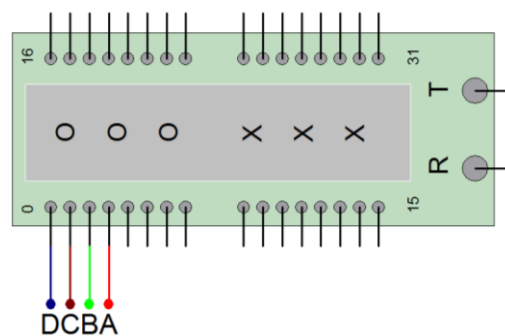
Obrázek 4 - prostředí programu ProAuthor

3. TVORBA KOMPONENT

Všechny vytvořené komponenty jako jsou obrázky, sady otázek pro autotesty, tabulky a animace jsou na příloženém CD, zde v práci jsou zobrazené jen některé z nich, případně jen ukázky. V každé podkapitole je vytvořeno několik komponent, které odpovídají požadavkům a náplni předmětu Číslicové prvky a systémy. Všechna zadání byla po dohodě s vedoucím práce převzata ze seminářů předmětu. Informace byly čerpány především ze seminářů předmětu ČPS, který jsem již absolvoval a z e-kurzu Číslicové prvky a systémy. E-kurz předpokládá základní znalosti práce v Multisimu a znalosti již získané v předchozím studiu z předmětu Analogové prvky a systémy.

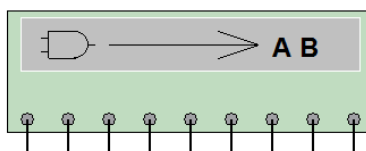
Komponenty obrázků zobrazují, jak by zhruba mělo finální zapojení vypadat a zobrazují některé rovnice a schémata pro lepší pochopení problematiky.

Vyjma prvního a posledního obsahují všechna cvičení součástku generátor datových slov. Tato součástka dokáže postupně generovat datová slova až pro velikost 32 bitového slova. Pro účely v této práci ovšem postačí využití pouze prvních čtyř bitů. Práci při vypracování simulace cvičení usnadňuje tím, že binárně generuje nultý až patnáctý bit. U generátoru lze nastavit, jestli se generování bude provádět automaticky při určité frekvenci (volba Cycle), nebo jestli bude vyžadováno každé další generované slovo kliknutím uživatele (volba Step). Volba závisí čistě na uživateli. Ve všech animacích je označen nejvíce významový bit písmenem A a červenou barvou, bit B zelenou barvou, bit C hnědou barvou a nejméně významový bit D modrou barvou. Součástku (Word Generator) nalezneme na paletě vpravo.



Obrázek 5 - Generátor datových slov

Další novou součástkou, kterou využijeme v pátém cvičení je logický konvertor. Logický konvertor je až pro 8 vstupních proměnných. Lze nastavit výstupní proměnou o délce 255 bitů a následně z ní generovat konjunktní tvar. Zde využijeme pouze 4 vstupní proměnné, vzhledem k šesnásobitové výstupní proměnné Y. Vygenerovaný tvar se přepíše do řádku pod tabulkou. Lze sem taky tvar přepsat ručně a poté vygenerovat výstupní proměnou. Z konjunktního tvaru poté pouhým kliknutím na jedno z tlačítek, konvertor vygeneruje celé zapojení. Součástku (Logic Converter) nalezneme na paletě vpravo.



Obrázek 6 - Logický konvertor

Dále se používají základní součástky jako logické členy, zem, zdroj napětí, barevné diody, přepínače a integrované obvody. Ke každé součástce je definovaná nápověda. V případě, že si nejsme jisti pravdivostní tabulkou nebo nevíme co připojit na některé vstupy a výstupy, tak můžeme najít součástku v nápovědě a veškeré informace, včetně pravdivostních tabulek se dozvíme.

Před začátkem nahrávání bylo každé zapojení přeskládáno, aby bylo přehlednější. Pro ještě lepší přehlednost bylo v nastavení Multisimu v Sheet Properties vypnuto zobrazování názvu hodnot, čísel spojů, mřížky a okrajů. Zobrazování bitů je barevně odlišeno.

V následujících kapitolách budou rozebrána jednotlivá cvičení, vysvětleno, co se po studentech žádá, jak vytvořit jednotlivá zapojení a na co nezapomenout. V každé komponentě je krátký úvod, ve kterém je popsáno, o co v kapitole půjde. Zadání, ve kterém jsou požadavky a podmínky pro splnění daného cvičení. Dále je popsáno, jak postupovat při vytváření konkrétního zapojení. Následující podkapitola představuje ukázkou animace, která je součástí e-kurzu v ProAuthoru. Další součástí je vybraný obrázek

použitý u konkrétní kapitoly a na konci každé kapitoly jsou vypsány sady otázek pro autotesty, které jsou implementovány do e-kurzu v ProAuthoru. Všechny komponenty pracují s výstupní proměnnou 1111100011011110.

3.1 KOMPONENTY E-KURZU PRO CVIČENÍ 1 - ZÁKLADNÍ LOGICKÉ FUNKCE

V prvním cvičení si procvičíme základní logické funkce. Od vedoucího semináře dostaneme zadanou jednu z logických funkcí (OR, AND, NOR, NAND, IOR, XOR) a pomocí zadané funkce doplníme pravdivost logických tabulek. Cílem prvního cvičení je zopakovat si základní logické funkce, vytvořit jednoduché zapojení a ověřit pravdivost podle tabulek.

3.1.1 ZADÁNÍ

a) Pro zadanou logickou funkci napište pravdivostní tabulky pro 2, 3 a 4 vstupní proměnné. Nejvíce významový bit (MSB) je vždy ve sloupci A.

bit	A	B	Y
0	0	0	
1	0	1	
2	1	0	
3	1	1	

bit	A	B	C	Y
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

bit	A	B	C	D	Y
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

Pravdivostní tabulky - pro dvě, tři a čtyři vstupní proměnné

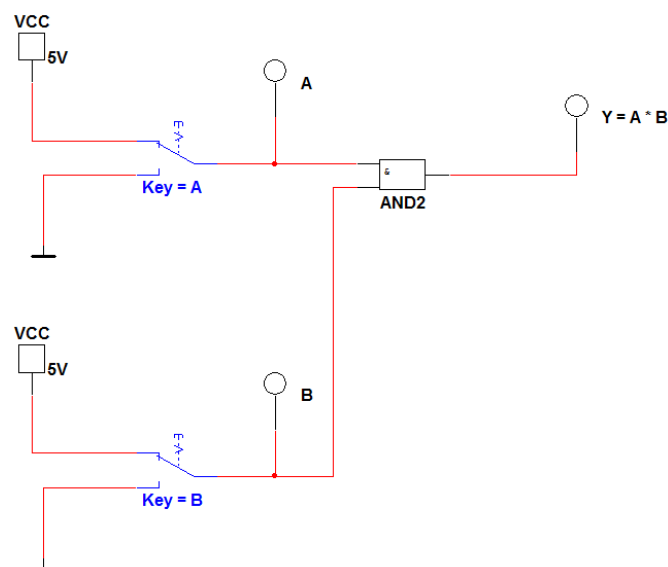
b) Pro zadanou logickou funkci ověřte simulací její chování pro 2, 3 a 4 vstupní proměnné.

3.1.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

Pro toto zapojení budeme potřebovat součástku, která nám bude rozlišovat dva stavy. Pro zadávání datových slov použijeme součástku přepínače, která se může nacházet ve dvou polohách. V poloze uzemnění, které nám představuje logickou nulu a v druhé poloze ve které je vstup připojen na zdroj napětí, který nám představuje logickou jedničku. Počet přepínačů je přímo úměrný počtu vstupních proměnných. U přepínače lze nastavit klávesu, která při stisku změní polohu přepínače. Pro jednotlivá zadání potřebujeme tedy zapojení se dvěma, třemi a čtyřmi přepínači. Pro lepší přehlednost připojíme na výstup přepínačů diody, které nám signalizují, jestli je nebo není v obvodu napětí (logická 1 při rozsvícení, logická 0 při zhasnutí). Vstupy součástky, která představuje jednu z logických funkcí (v tomto případě AND) připojíme na výstupy přepínačů. Na výstup logické součástky AND opět připojíme diodu.

Nyní se výstup celého obvodu bude měnit v závislosti na poloze přepínačů, tedy na kombinacích dle pravdivostní tabulky pro dvě vstupní proměnné logické funkce. Stejným principem vytvoříme i zbylé další dvě zapojení, pouze se změnami součástek logické funkce a přidáním nutného počtu přepínačů.

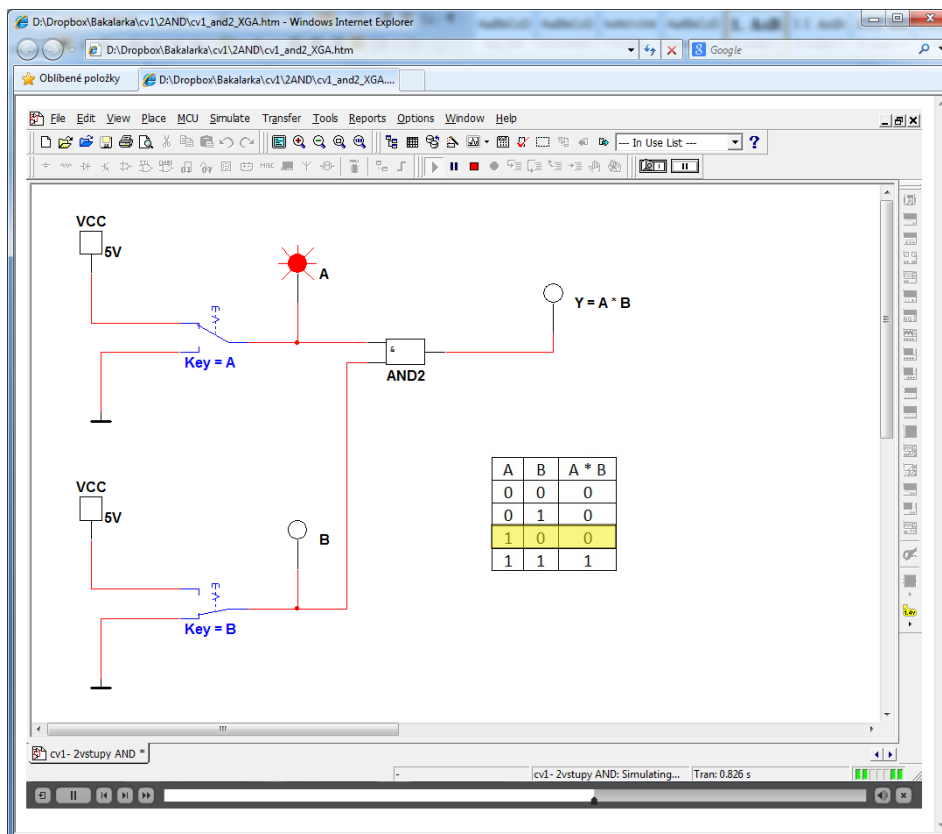
3.1.3 UKÁZKA OBRÁZKU



Obrázek 7 - Cvičení 1 - návrh řešení

3.1.4 UKÁZKA ANIMACE

V této animaci demonstrujeme fungování logického členu dvouvstupový AND. Pomocí přepínačů volíme vstupní kombinaci a kontrolujeme výstupy dle pravdivostní tabulky.



Obrázek 8 - Ukázka animace 1. cvičení

3.2 KOMPONENTY E-KURZU PRO CVIČENÍ 2 - ROVNICE A FUNKCE

V tomto cvičení si procvičíme logické funkce z minulé hodiny a naučíme se využívat generátor datových slov, který nám bude pomáhat i po zbytek semináře.

3.2.1 ZADÁNÍ

a) Je dána rovnice:
$$\overline{(\overline{B} + A \times C + B \times \overline{D})} \times C = B \times (\overline{A} \times \overline{C}) \times (\overline{B} + D) + \overline{C}$$

- Zapojte v simulačním programu levou a pravou stranu rovnice.
- Ověřte chování simulací v simulačním programu.
- Využijte generátor datových slov.

b) Je dána funkce: $f = AB + ABC + ABCD$

- Předložte vedoucímu semináře funkci k dodefinování negací.
- Napište pravdivostní tabulku dodefinované funkce.
- Ověřte chování v simulačním programu.

bit	A	B	C	D	AB	ABC	ABCD	f
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

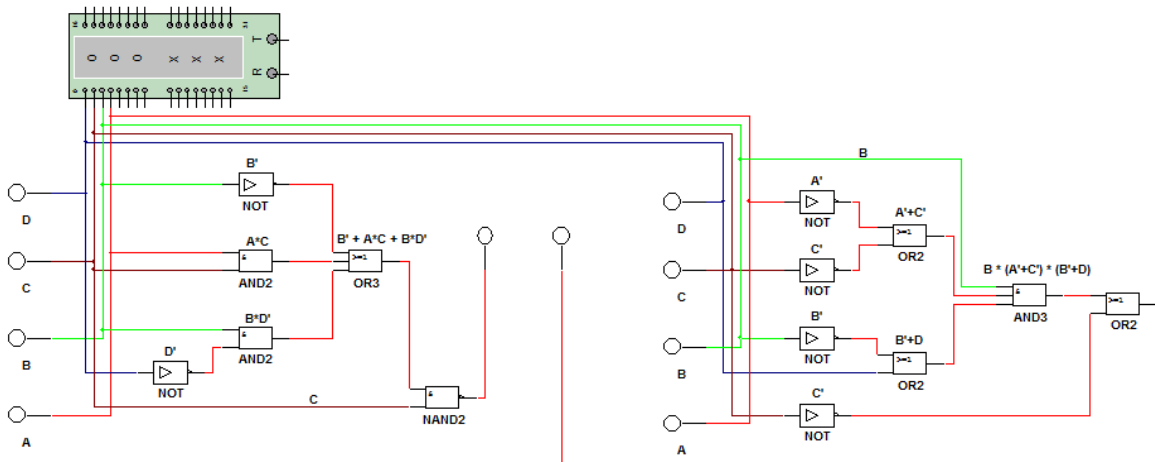
Pravdivostní tabulka 1 - pro $f = AB + ABC + ABCD$

3.2.1 VYPRACOVÁNÍ SIMULACE V MULTISIMU

V tomto cvičení použijeme generátor datových slov, který nám nahrazuje funkci přepínačů. Generátor najdeme na paletě vpravo. Generátor má tu funkci, že nám může automaticky nastavovat jednotlivá datová slova. V tomto případě, kdy máme 4 vstupní proměnné se jedná o 16 kombinací. Budeme tedy využívat pouze 4 z celkových 32 výstupů. Jednotlivé výstupy generátoru představují bity datového slova na určité pozici. Nejméně významový bit je nejvíce vlevo (je označen 0), který reprezentuje bit D, poté směrem doprava jsou zbylé tři bity, v pořadí C, B, A (nejvíce významový bit). Pro lepší přehlednost jsou jednotlivé bity barevně odlišené. V případě že si nejsme jisti na který výstup připojit jednotlivé bity, můžeme se podívat do nápovědy, kde jsou jednotlivé části popsány. Pro každou stranu rovnice můžeme použít samostatný generátor, ale lze všechny zapojení připojit k jednomu generátoru. Nejprve si vezmeme levou část rovnice.

Vidíme, že první bit je negované B, potřebujeme tedy součástku NOT, která nám zneguje výstup. Výstup generátoru na pozici B, připojíme na vstup součástky NOT. Výstup součástky NOT tedy bude negované B. Další je součin dvou bitů A a C, ke kterému nám poslouží součástka dvouvstupový AND, opět z generátoru slov připojíme příslušně bity na vstupy dvouvstupového ANDu. Poslední součástí rovnice v závorce je součin bitů B a negovaného D. Postup bude stejný jako u součinu A a C s rozdílem, že výstup z generátoru slov bitu D nejdříve připojíme na součástku NOT a až výstup součástky NOT připojíme na jeden ze vstupů dvouvstupového ANDu. Tyto tři výstupy musíme dle rovnice sečíst. To provedeme tak, že jednotlivé výstupy připojíme na vstupy třívstupové součástky OR. Teď tedy máme zapojenou závorku rovnice, posledním krokem je součin závorky s bitem C a znegování celkového výstupu. V tom nám pomůže součástka dvouvstupový NAND, která nám vynásobí výstup rovnice v závorce s bitem C a následně ho zneguje. Tímto máme vyřešenou levou stranu rovnice. Stejným principem za použití vhodných součástek zapojíme i druhou stranu rovnice. Kontrolu provádíme pomocí generátoru datových slov. Máme více možností, rozklikneme generátor a buď klikneme na tlačítko Step a kontrolujeme slovo po slovu opakovaným klikáním, nebo vybereme Cycle a zmenšíme frekvenci, abychom stíhali kontrolovat výstupy. Pokud se výstupy obou rovnic rovnají a zároveň jsou i správně dle pravdivostní tabulky, je zapojení správně.

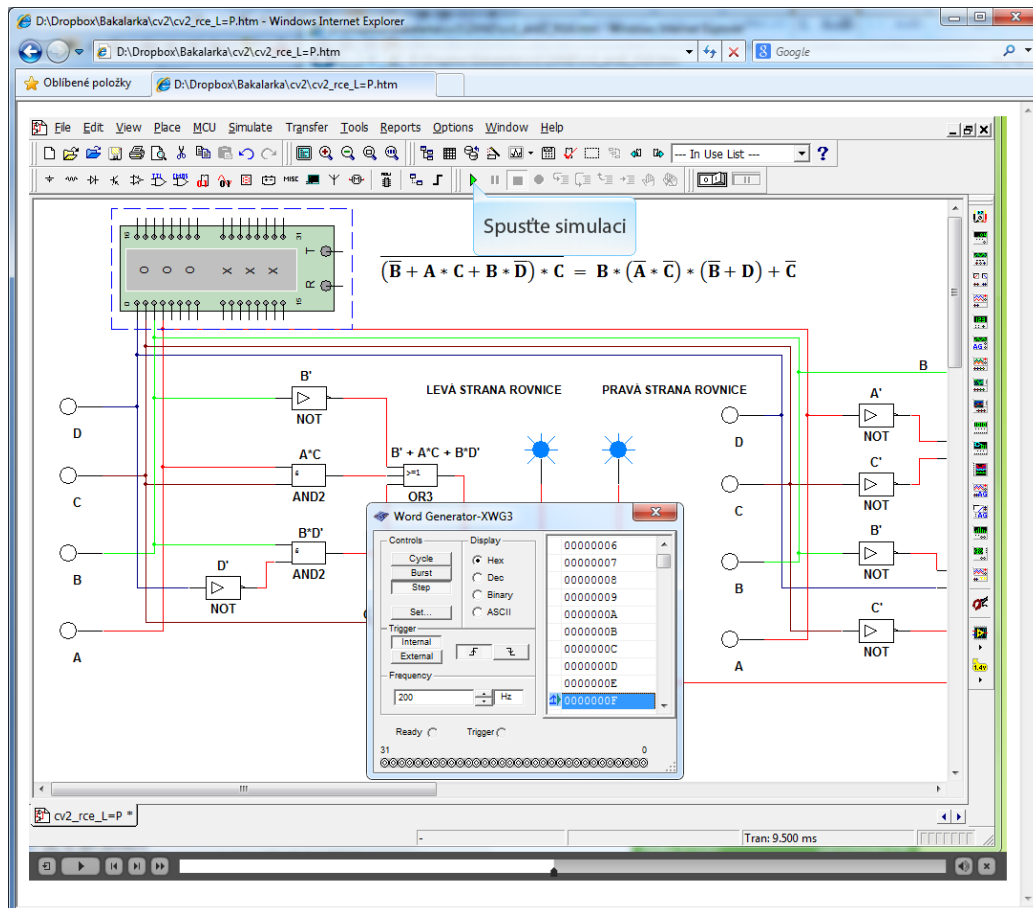
3.2.2 UKÁZKA OBRÁZKU



Obrázek 9 - Cvičení 2 - návrh řešení

3.2.1 UKÁZKA ANIMACE

Tato animace vysvětluje zapojování jednotlivých součástek dle zadané rovnice. Jak používat a kde najít generátor datových slov a jak s ním pracovat při kontrole výstupních proměnných a ověřování rovnosti obou stran rovnice.



Obrázek 10 - Ukázka animace 2. cvičení

3.2.2 OTÁZKY AUTOTESTU PRO CVIČENÍ 1 A 2

- 1) Základní logické operace jsou:
 - a) logický součet, logický součin, ekvivalence
 - b) logický součet, logický součin, negace
 - c) logický součin, negace, ekvivalence
- 2) Která logická funkce představuje negovaný součet:
 - a) IOR

- b) NOR
 - c) XOR
- 3) Která z uvedených logických funkcí neexistuje:
- a) XOR
 - b) NAND
 - c) XAND
- 4) Která vstupní kombinace odpovídá 4. bitu:
- a) 011
 - b) 101
 - c) 100
- Vysvětlení: $1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 4$
- 5) Pro kolik vstupních proměnných máme výstupní osmibitové slovo:
- a) 3 (Volná odpověď)
- Vysvětlení: Jeden bit má dva stavy, minimální počet vstupů pro dosažení osmi kombinací je tedy tři vstupy. $2^3 = 8$
- 6) Zkratka MSB znamená:
- a) Most Significant Bit - označení nejvýznamnějšího bitu
 - b) Multisim Simulation Bit - označení simulačního bitu
 - c) Most Significant Byte - označení nejvýznamnějšího bytu
- 7) Lze nahradit generátor datových slov přepínači?
- a) ano
- Správné odpovědi: 1b, 2b, 3c, 4c, 6a

3.3 KOMPONENTY E-KURZU PRO CVIČENÍ 3 - ÚPLNÝ DISJUNKTNÍ A KONJUNKTNÍ TVAR

V tomto cvičení si vytvoříme výstupní proměnnou Y, která vás bude provázet ve zbylých cvičeních. Výstupní proměnná bude ve tvaru 16 bitového slova. Z vytvořené výstupní proměnné dále vypíšeme úplný disjunktivní (součtový) a úplný konjunktivní

(součinový) tvar. Jak tyto tvary vytvořit nalezneme v e-kurzu Číslicové prvky a systémy, studijní článek: Definice logických funkcí². Příklad úplného disjunktčního tvaru:

$$\overline{A}BCD + A\overline{B}CD + \overline{A}BC\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD$$

3.3.1 ZADÁNÍ

a) Pomocí tabulky si na základě Vašeho jména a kódu v pravdivostní tabulce vypište proměnnou Y. Ve výstupní proměnné Y představují písmena ve Vašem jméně logickou 1, ostatní bity jsou 0. Diakritiku zanedbáváme.

Např.: **MARTIN DOLEJŠ** odpovídá bitům:

0, 1, 2, 3, 4, 8, 9, 11, 12, 13, 14.

Výstupní proměnná Y má tedy tvar:

1111100011011110.

- Vypište disjunktční a konjunktční tvar Vaší proměnné.
- Zrealizujte oba tvary v simulačním programu.
- Ověřte chování v simulačním programu.

bit	A	B	C	D	kód	Y
0	0	0	0	0	AQ	1
1	0	0	0	1	BR	1
2	0	0	1	0	CS	1
3	0	0	1	1	DT	1
4	0	1	0	0	EU	1
5	0	1	0	1	FV	0
6	0	1	1	0	GW	0
7	0	1	1	1	HX	0
8	1	0	0	0	IY	1
9	1	0	0	1	JZ	1
10	1	0	1	0	K	0
11	1	0	1	1	L	1
12	1	1	0	0	M	1
13	1	1	0	1	N	1
14	1	1	1	0	O	1
15	1	1	1	1	P	0

Pravdivostní tabulka 2 - Tvorba výstupní proměnné Y

3.3.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

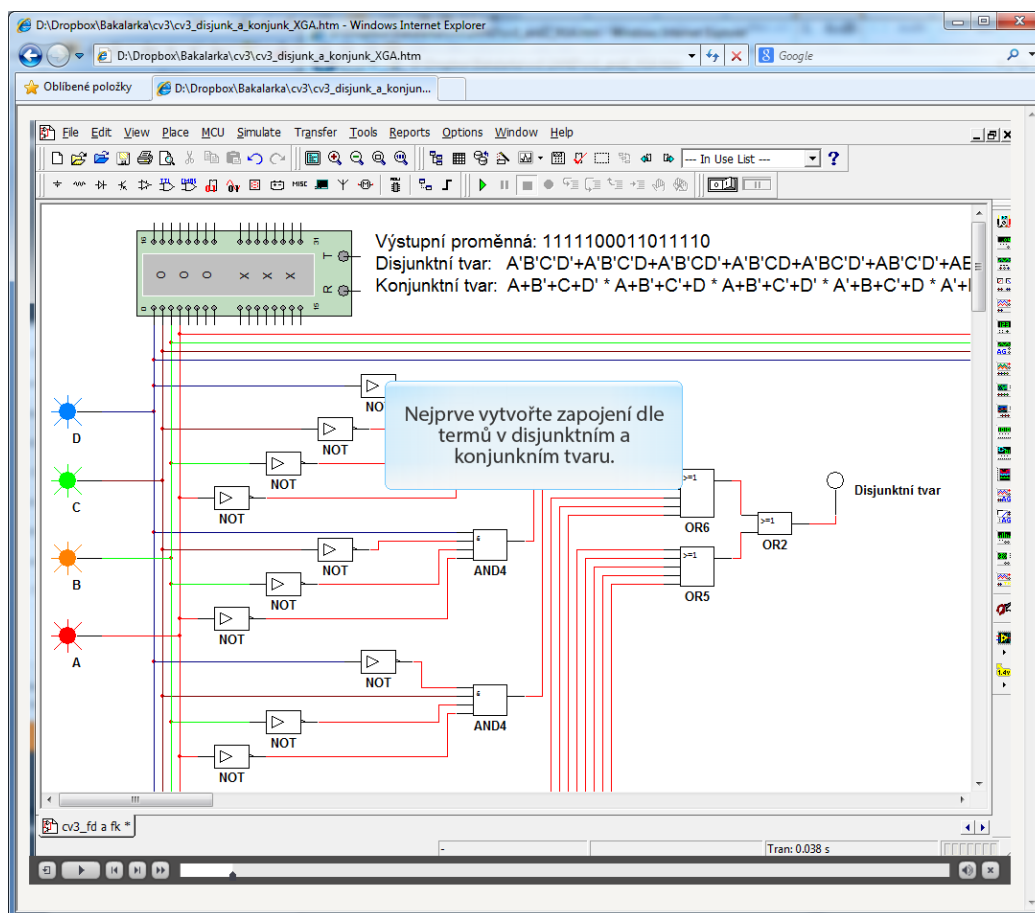
Zapojení v tomto cvičení jsme si rozdělili na disjunktční a konjunktční obvod. Pro zadávání vstupních proměnných opět použijeme generátor datových slov. Všechny mintermy v disjunktčním (součinovém) tvaru mají čtyři proměnné, takže můžeme u všech mintermů použít součástku čtyřvstupový AND. Jednotlivé výstupy z generátoru datových slov, připojíme na vstupy čtyřvstupových ANDů, které odpovídají jednotlivým bitům. Pokud je proměnná negovaná nejdříve předřadíme do zapojení součástku NOT. Po zapojení všech mintermů, přivedeme výstupy ANDů do součástky OR, která nám zajistí

² <http://www.kvd.zcu.cz/cz/materialy/cps/cps/cps/HTML/9/default.htm>

logický součet. V tomto případě, kdy úplný disjunktivní tvar má jedenáct mintermů, bychom měli použít součástku jedenáctivstupový OR, ale takováto součástka není v Multisimu implementována, tak musíme použít jeden šestivstupový a jeden pětivstupový OR a jejich výstupy zapojit do dvou vstupového ORu. Na výstup opět pro kontrolu připojíme diodu. U konjunkčního (součtového) tvaru použijeme pro všechny maxtermy čtyřvstupový OR a výstupy těchto ORů připojíme na pětivstupový AND, protože vzorový úplný konjunkční tvar má právě pět maxtermů. Výstupy jednotlivých tvarů se musí shodovat s výstupní proměnou Y.

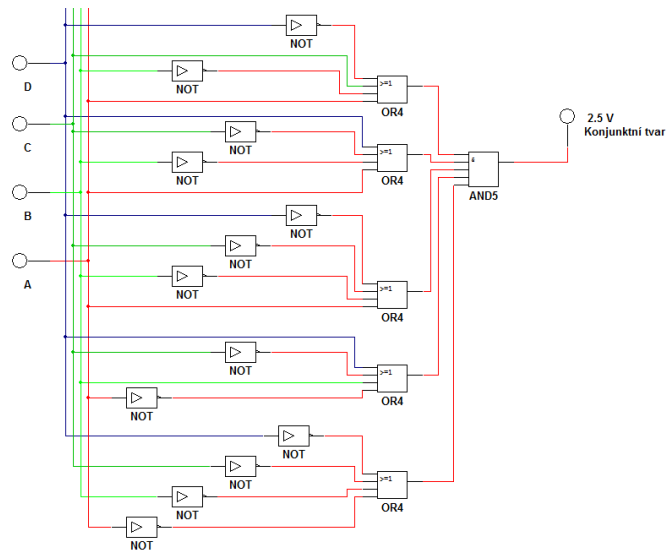
3.3.3 UKÁZKA ANIMACE

Tato animace demonstruje, jak vytvořit zapojení dle konkrétního úplného disjunktivního tvaru a úplného konjunkčního tvaru. Zapojení provádíme postupně dle jednotlivých mintermů a maxtermů. Na konci animace kontroluji, zda se rovnají výstupy obou tvarů.



Obrázek 11 - Ukázka animace 3. cvičení

3.3.4 UKÁZKA OBRÁZKU



Obrázek 12 - Cvičení 3 - návrh řešení

3.3.5 OTÁZKY AUTOTESTU PRO CVIČENÍ 3

- 1) Konjunktivní tvar znamená:
 - a) součtový
 - b) součinnový
 - c) rozdílový
- 2) Předpis "Vypíšeme řádky pravdivostní tabulky s hodnotou funkce log 1 jako součiny jednotlivých proměnných a výsledný zápis sestavíme jako součet těchto součinů" platí pro:
 - a) zápis funkce v disjunktivním tvaru
 - b) zápis funkce v konjunktivním tvaru
 - c) zápis funkce v negovaném tvaru
- 3) Pro sestavení mintermů v konjunktivním tvaru pro 16bitové datové slovo potřebujeme hlavně součástky typu:
 - a) 4 vstupový AND
 - b) 4 vstupový OR
 - c) 4 vstupový NAND

Vysvětlení: Konjunktvní tvar je součet součinů, pro sestavení mintermů tedy potřebujeme součástku, která provádí logický součin, tedy AND.

4) Kolik bitů musí mít Vaše výstupní proměnná Y?

a) 16 (text správně)

Vysvětlení: Máme zadány čtyři vstupní proměnné, $4^2 = 16$.

5) Generátor datových slov nahrazuje:

a) operační zesilovač

b) klopné obvody

c) funkci přepínačů

6) Vyberte správné tvrzení:

a) konjunktvní tvar je součet součinů

b) disjunktvní tvar je součet součinů

c) konjunktvní tvar je součin součtů

Správné odpovědi: 1b, 2a, 3b, 5c, 6b

3.4 KOMPONENTY E-KURZU PRO CVIČENÍ 4 - MINIMALIZACE DISJUNKTNÍHO A KONJUNKTNÍHO TVARU

V tomto cvičení minimalizujeme Váš disjunktvní a konjunktvní tvar vytvořený v předchozím cvičení. Minimalizaci provedeme pomocí zákonů Booleovy algebry, nebo pomocí Karnaughovy mapy. Jak provést minimalizaci nalezneme v e-kurzu Číslicové prvky a systémy, studijní článek: Zjednodušování zápisu logické funkce³.

Příklad minimalizovaného konjunktvního tvaru:

$$(A + \bar{B} + \bar{D}) \times (A + \bar{B} + \bar{C}) \times (\bar{B} + \bar{C} + \bar{D}) \times (\bar{A} + B + \bar{C} + D)$$

3.4.1 ZADÁNÍ

- Stanovte všechny minimalizované disjunktvní a konjunktvní tvary.
- Nechte zkontrolovat vedoucím semináře minimalizaci (smyčky).

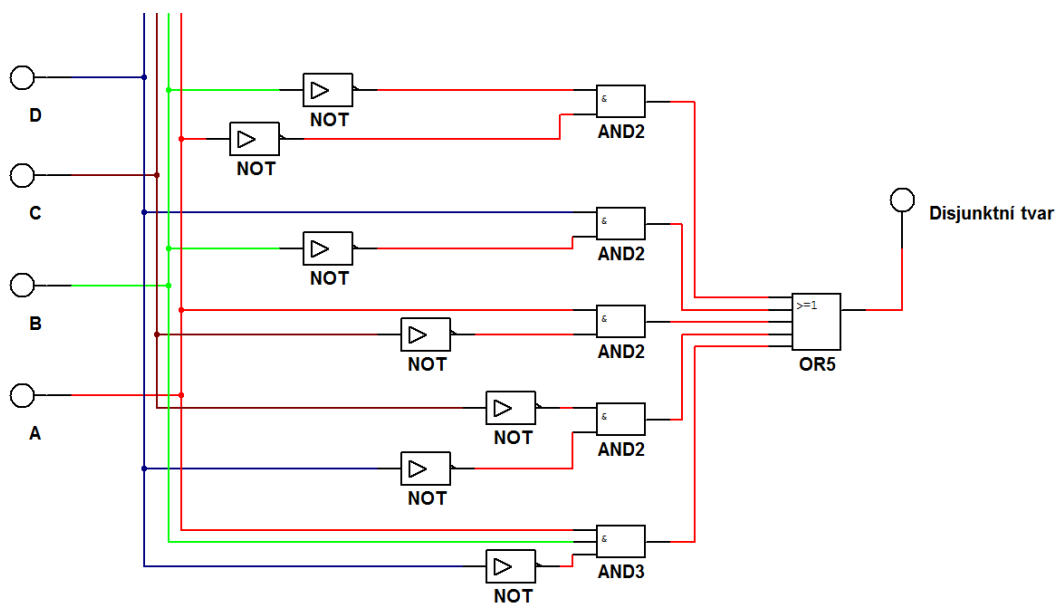
³ <http://www.kvd.zcu.cz/cz/materialy/cps/cps/cps/HTML/31/default.htm>

- Ověřte chování v simulačním programu.
- Dokažte pomocí zákonů Booleovy algebry, že se minimalizovaný disjunktivní a minimalizovaný konjunktivní tvar rovnají.

3.4.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

Základem čtvrtého cvičení je minimalizace disjunktivního a konjunktivního tvaru. Zvolíme jeden ze způsobů minimalizace, například pomocí Karnaughovy mapy. Minimalizovaný konjunktivní tvar vypadá takto: $(A + \bar{B} + \bar{D}) \times (A + \bar{B} + \bar{C}) \times (\bar{B} + \bar{C} + \bar{D}) \times (\bar{A} + B + \bar{C} + D)$ a minimalizovaný disjunktivní tvar takto: $\bar{A}\bar{B} + \bar{B}D + A\bar{C} + \bar{C}\bar{D} + AB\bar{D}$. Poté co máme vypsané oba tvary, vložíme generátor datových slov a na jeho výstupy připojíme jednotlivé bity obou tvarů. Použité součástky jsou negace, součástka AND a součástka OR. Počet vstupů u jednotlivých součástek se odvíjí od počtu členů v rovnici. Výstupy obou tvarů zkontrolujeme porovnáním světelných diod s výstupní proměnou Y.

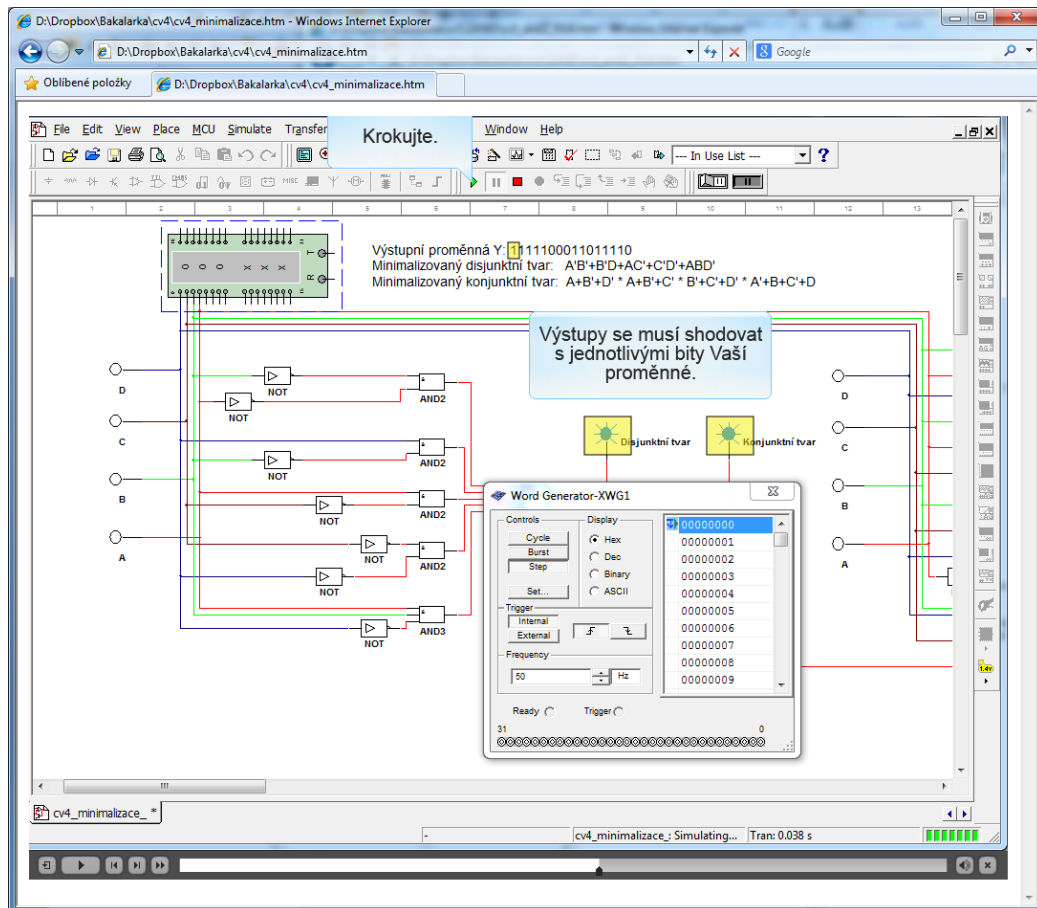
3.4.3 UKÁZKA OBRÁZKU



Obrázek 13 - Cvičení 4 - návrh řešení

3.4.4 UKÁZKA ANIMACE

Animace demonstruje zapojení minimalizovaného disjunktčního a minimalizovaného konjunktčního tvaru pro výstupní proměnnou 111100011011110. Ke konci animace ověřujeme pomocí generátoru slov, že se výstupy obou tvarů rovnají.



Obrázek 14 - Ukázka animace 4. cvičení

3.4.5 OTÁZKY AUTOTESTU PRO CVIČENÍ 4

- 1) Zjednodušení zápisu minimalizace se provádí pomocí:
 - a) pouze Karnaughovy mapy
 - b) pouze zákonů Booleovy algebry
 - c) obou dvou výše zmíněných způsobů
- 2) Pro sestavení zapojení mintermů v disjunktčním tvaru potřebujeme hlavně součástku:
 - a) AND
 - b) OR

c) NAND

Vysvětlení: Potřebujeme součástku AND, protože minterm se skládá se součinu jednotlivých bitů.

3) Vyberte pravdivé tvrzení:

- a) smyčky se snažíme tvořit co nejmenší
- b) velikost smyček je vždy mocnina dvou
- c) Karnaughova mapa má vždy stejný počet smyček

4) Mezi zákony Booleovy algebry nepatří:

- a) zákon převrácené absorpce
- b) zákon o agresivnosti logické 0 a logické 1
- c) De Morganův zákon

5) Která rovnice nevyjadřuje De Morganův zákon:

- a) $\overline{x + y} = \bar{x} \times \bar{y}$
- b) $\overline{x \times y} = \bar{x} + \bar{y}$
- c) $\overline{x + y} = \bar{x} + \bar{y}$

6) Výstupy konjunkčního a disjunkčního tvaru se musí rovnat:

- a) ano

Vysvětlení: Po aplikaci zákonů Booleovy algebry se z jednoho tvaru stane druhý.

Správné odpovědi: 1c, 2a, 3b, 4a, 5c,

3.5 KOMPONENTY E-KURZU PRO CVIČENÍ 5 - HOMOGENNÍ STRUKTURY LOGICKÉ FUNKCE

V tomto cvičení se naučíme vytvářet a zapojovat homogenní (stejnorodé) struktury logické funkce. Použijeme jeden minimalizovaný tvar a vytvoříme zapojení s použitím pouze součástek typu NOR. Poté s využitím pouze součástek typu NAND a s využitím logického konvertoru vygenerujeme zapojení pouze se součástkami dvouvstupový NAND. Jak upravit zápis logické funkce do homogenní struktury nalezneme v e-kurzu Číslicové prvky a systémy, studijní článek: Zjednodušování zápisu logické funkce.

3.5.1 ZADÁNÍ

Zrealizujte v simulačním programu jeden minimální tvar pomocí logických členů typu:

- Samé logické členy typu NAND.
- Samé logické členy typu NOR.
- Samé dvouvstupové logické členy typu NAND s využitím návrhu pomocí logického konvertoru.
- Ověřte chování v simulačním programu.

3.5.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

K realizaci zapojení s logickými členy typu NAND vybereme například minimalizovaný disjunktivní tvar. Logickou funkci přepíšeme do homogenní struktury aplikováním zákona dvojité negace a následnou úpravou vnitřní negace deMorganovým zákonem, tím docílíme, že se ve funkci budou nacházet pouze logické součiny. Výsledná rovnice po aplikování obou zákonů vypadá takto: $\overline{\overline{A} \times \overline{B}} \times \overline{\overline{C} \times \overline{D}} \times \overline{\overline{A} \times \overline{B} \times \overline{C} \times \overline{D}}$. Dle zadání použijeme u prvního zapojení pouze součástky typu NAND. Počet vstupů u jednotlivých součástek se odvozuje od počtu jednotlivých bitů v datových slovech. Jelikož musíme používat pouze součástky typu NAND, tak musíme nahradit negaci (součástku NOT) spojením obou vstupů NANDu do jednoho a tím získáme plnohodnotnou náhradu za součástku NOT. Pokud bude vstupní signál logickou jedničkou, bude jednička na obou vstupech a tak se negovaný výstup bude rovnat nule, v opačném případě kdy bude na vstupu nula, bude díky spojení nula na obou vstupech a tím pádem na výstupu bude logická jednička. Jednotlivé výstupy připojíme na vstupy dalšího dvouvstupového NANDu. Všechny ostatní termy vyřešíme stejným způsobem, vyjma posledního termu, který je třívstupový a tedy musíme použít třívstupový NAND. Výstupy celé funkce připojíme na pětivstupový NAND a na výstup diodu pro kontrolu.

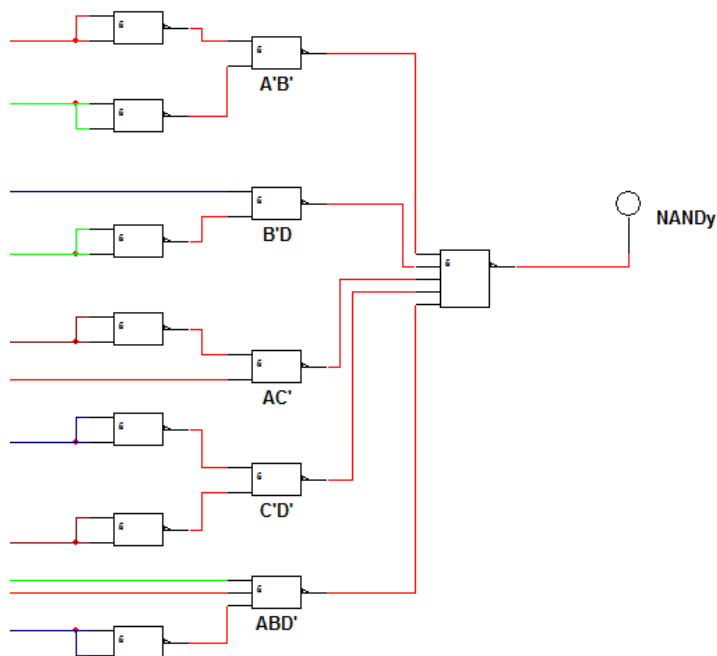
U druhého zapojení s logickými členy typu NOR pracujeme s minimalizovaným konjunktivním tvarem. Homogenní funkci vytvoříme stejným principem. Použijeme zákon dvojité negace a následně na vnitřní negaci aplikujeme deMorganův zákon. Výsledná rovnice vypadá takto: $\overline{\overline{A + B + D}} + \overline{\overline{A + B + C}} + \overline{\overline{B + C + D}} + \overline{\overline{A + B + C + D}}$. Náhradu negace

u součástky NOR vytvoříme opět stejným způsobem jako u předchozího zapojení, tedy spojením vstupů do jednoho. Vstupy jednotlivých termů připojíme do NORů, kteří mají stejný počet vstupů jako je počet bitů jednotlivých termů. Nakonec výstupy těchto NORů připojíme na vstupy čtyřvstupového NORu, který má na výstupu připojenou diodu.

Třetí zapojení je vytvořené automaticky pomocí logického konvertoru. Logický konvertor nalezneme na pravé paletě. Po rozkliknutí konvertoru přepíšeme minimalizovaný disjunktivní tvar do textového okna a klikneme na tlačítko konvertovat do zapojení ze samých dvouvstupových NANDů ($A|B \rightarrow \text{NAND}$). Vstupy vygenerovaného zapojení připojíme na příslušné bity generátoru datových slov a na výstup připojíme diodu.

Spustíme generátor datových slov a kontrolujeme výstup, respektive diody všech tří zapojení. Výstupy signalizované diodami musí souhlasit s příslušnými stavy na jednotlivých bitech výstupní proměnné Y.

3.5.3 UKÁZKA OBRÁZKU

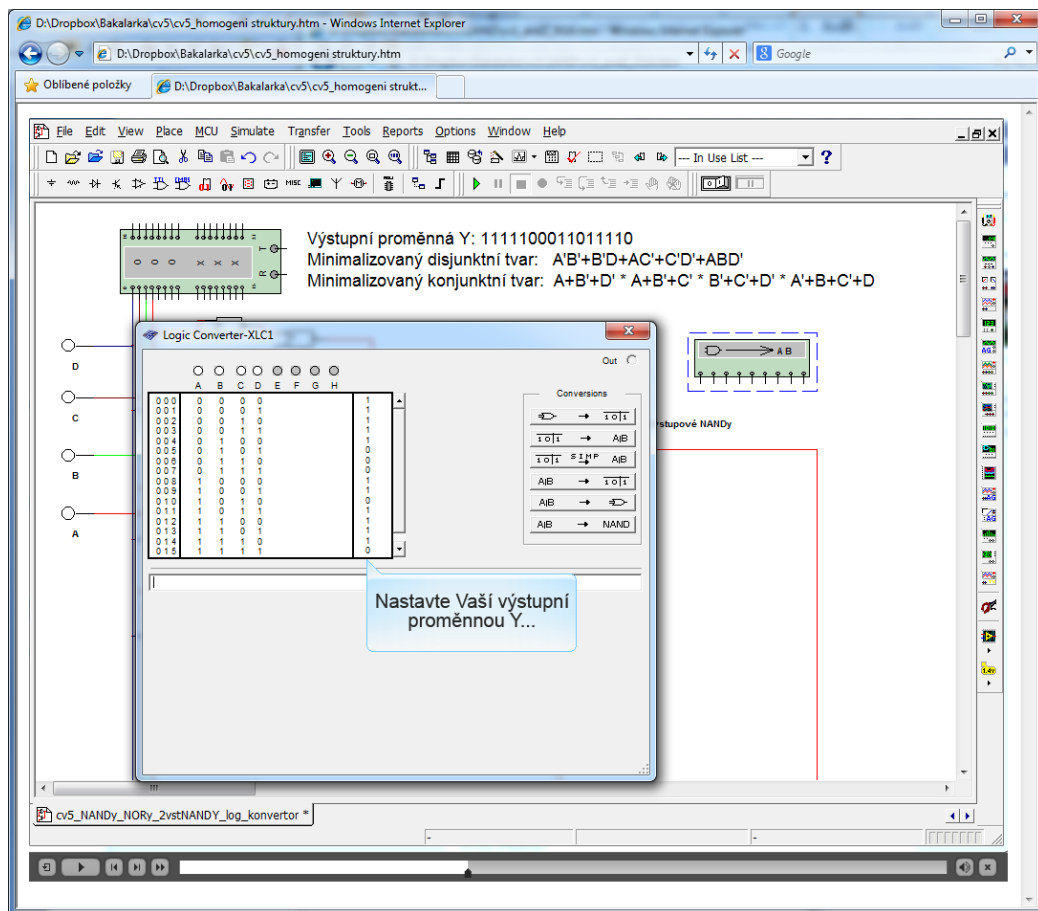


Obrázek 15 - Cvičení 5 - návrh řešení

3.5.4 UKÁZKA ANIMACE

V animaci pátého cvičení vytváříme homogenní strukturu typu samé NANDy pro minimalizovaný disjunktivní tvar, samé NORy pro minimalizovaný konjunktivní tvar

a samé dvouvstupové NANDy dle minimalizovaného disjunktčního tvaru. Dále demonstrujeme možnosti a využití nové součástky logický konvertor.



Obrázek 16 - Ukázka animace 5. cvičení

3.5.5 OTÁZKY AUTOTESTU PRO CVIČENÍ 5

1) Hlavní příčina zavedení homogenních struktur je:

- a) levnější výroba
- b) vyšší rychlosti
- c) nižší vstupní napětí

Vysvětlení: Při výrobě se používají součástky stejného typu a tím se šetří peníze.

2) Postup při úpravě logické funkce do homogenní struktury je:

- a) pouze aplikování de Morganova zákona
- b) aplikování zákonů dvojité negace a následně de Morganova zákona
- c) aplikování zákona dvojité negace a následně sestavení Karnaughovy mapy

3) Aplikováním dvojité negace na funkci $f = (a + b + c) \times (a + b + \bar{c})$ dostaneme:

a) $f = (a + b + c) + (a + b + \bar{c})$

b) $f = \overline{(a + b + c) \times (a + b + \bar{c})}$

c) $f = \overline{(a + b + c) + (a + b + \bar{c})}$

4) Homogenní struktura znamená použití:

a) součástek stejného typu

b) součástek různého typu

c) součástek pouze typu NAND a NOR

Vysvětlení: Homogenní = stejnorodá

5) V cvičení nepoužíváme součástky typu:

a) NAND

b) AND

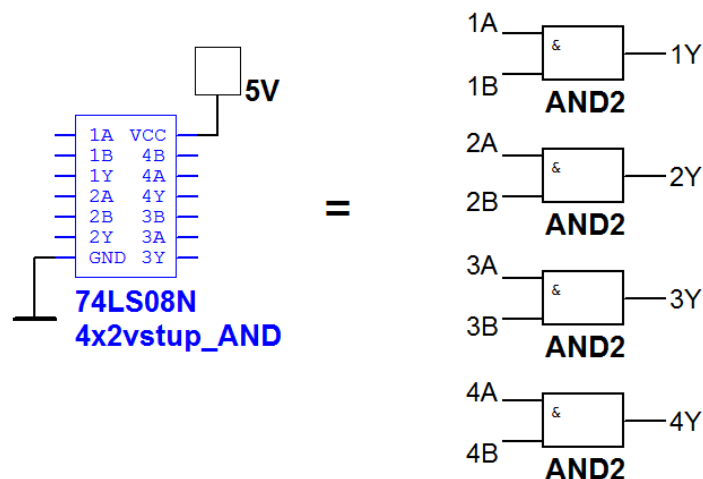
c) NOR

d) OR

Správné odpovědi: 1a, 2b, 3c, 4a, 5bd

3.6 KOMPONENTY E-KURZU PRO CVIČENÍ 6 - INTEGROVANÉ OBVODY

V tomto cvičení si nahradíme základní logické členy integrovanými obvody. Integrovaný obvod je pouzdro, ve kterém je několik základních na sobě nezávislých součástek. Pro lepší pochopení se podíváme na obrázek. Jeden minimalizovaný tvar zapojíme pomocí integrovaných obvodů typu TTL a CMOS. Integrované obvody nalezneme na horní paletě.



Obrázek 17 - Vnitřní struktura integrovaných obvodů

3.6.1 ZADÁNÍ

Zrealizujte v simulačním programu jeden minimalizovaný tvar pomocí integrovaných obvodů typu:

- TTL.
- CMOS.
- Ověřte chování v simulačním programu.

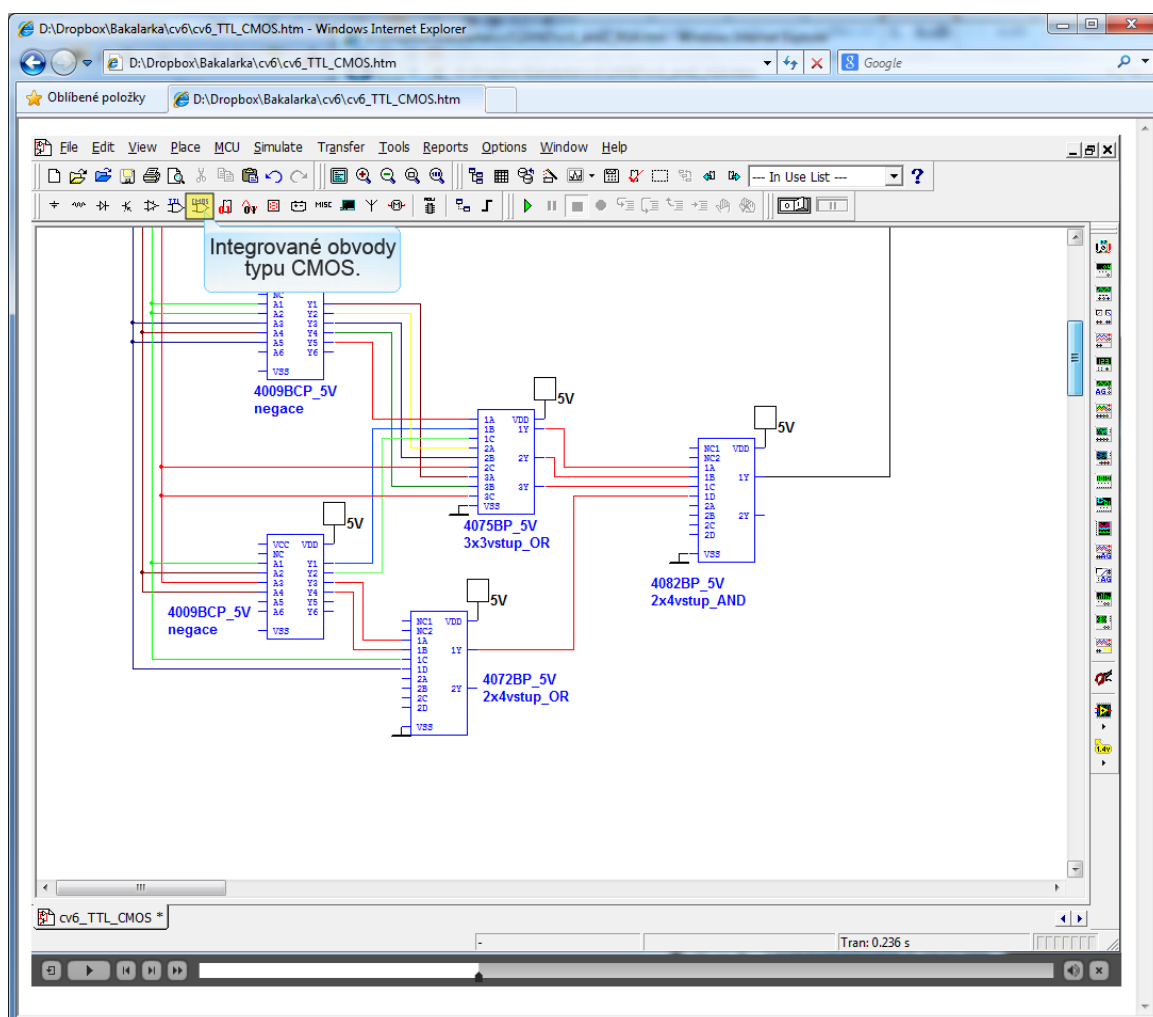
3.6.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

Zde si musíme nejdříve uvědomit, na jakém principu integrované obvody fungují. Každý integrovaný obvod, který zde použijeme, reprezentuje několik základních stejných na sobě nezávislých součástek, vložených do jediné. Důležité je nezávislých, takže jeden z výstupů integrovaného obvodu můžu připojit na jiný vstup toho samého obvodu. Pro minimalizovaný konjunktní tvar potřebujeme integrovaný obvod TTL typu negace, dvouvstupový AND, třívstupový AND a dvouvstupový OR. Pro minimalizovaný disjunktní tvar potřebujeme integrovaný obvod CMOS typu negace, třívstupový OR, čtyřvstupový OR a čtyřvstupový AND. Počet součástek v integrovaném obvodu se může lišit. Každý použitý integrovaný obvod je specifický svou logickou funkcí a počtem vstupů, které jsou popsány v popisku součástky. Například: QUAD 2-INPUT AND značí, že integrovaný obvod nahradí čtyři dvouvstupové ANDy. Jednotlivé součástky jsou označeny číslicí a následně písmenem. Například pokud u obvodu vidíme 1A, 1B a 1C víme, že se jedná o třívstupovou

součástku. Stejné číslo nám značí, že jde o jednu součástku a následná písmena jednotlivé vstupy. Každá součástka musí mít výstup a ten se u integrovaného obvodu značí číslicí a následně písmenem Y. V tomto případě by tato třívstupová součástka měla označení výstupu 1Y. Každý integrovaný obvod musí být připojen k napětí (VCC u TTL, VDD u CMOS) a uzemněn (GND u TTL, VSS u CMOS). Výstupy generátoru datových slov připojíme na vstupy jednotlivých integrovaných obvodů. Výstupy obou zapojení opět zkontrolujeme generátorem datových slov a porovnáme s výstupní proměnou Y.

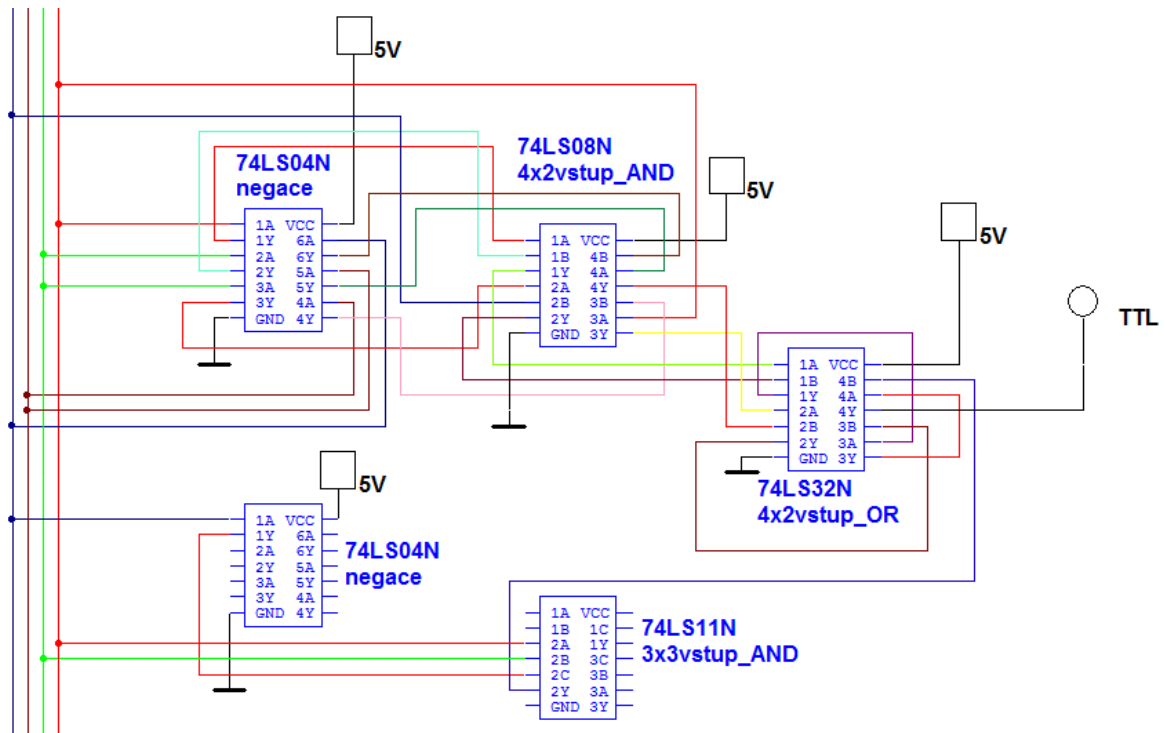
3.6.3 UKÁZKA ANIMACE

Tato animace vysvětluje jak nahradit logické členy integrovanými obvody. Pro minimální disjunktivní tvar použijeme integrované obvody typu TTL a pro minimální disjunktivní tvar integrované obvody typu CMOS.



Obrázek 18 - Ukázka animace 6. cvičení

3.6.4 UKÁZKA OBRÁZKU



Obrázek 19 - Cvičení 6 - návrh řešení

3.6.5 OTÁZKY AUTOTESTU PRO CVIČENÍ 6

1) Pomocí integrovaných obvodů typu TTL můžeme vytvořit:

- pouze konjunktní tvar
- pouze disjunktní tvar
- konjunktní i disjunktní tvar

Vysvětlení: Integrované obvody nám nahrazují základní součástky, takže lze vytvořit oba tvary.

2) Označení integrovaných obvodů typu TTL vždy začíná číslem 71

- ne

3) Integrovaný obvod obsahuje:

- jeden logický člen s více výstupy
- více logických na sobě nezávislých členů
- více logických vzájemně ovlivňujících se členů

4) Integrované obvody se nevyrábí technologií:

- a) TTL
 - b) Bi-CMOS
 - c) Bi-TTL
- 5) Která technologie obsahuje unipolární tranzistory:
- a) TTL
 - b) CMOS
 - c) ECL
- 6) Lze nahradit integrovaný obvod logickými členy?
- a) ano

Správné odpovědi: 1c, 2b, 3c, 4c, 5b

3.7 KOMPONENTY E-KURZU PRO CVIČENÍ 7 - NEURČITÉ STAVY A MINIMALIZACE

V tomto cvičení začleníme neurčité stavy do výstupní proměnné. To provedeme tak, že neurčitými stavy nahradíme bity, které odpovídají Vaším číslům ve studijním čísle. Neurčitý člen značíme písmenem X.

Např.: **P11B0459B** odpovídá bitům: 0, 1, 4, 5, 9. Výstupní proměnná Y_n má tedy tvar: **XX11XX001X011110**.

bit	A	B	C	D	Y	Y_n
0	0	0	0	0	1	X
1	0	0	0	1	1	X
2	0	0	1	0	1	1
3	0	0	1	1	1	1
4	0	1	0	0	1	X
5	0	1	0	1	0	X
6	0	1	1	0	0	0
7	0	1	1	1	0	0
8	1	0	0	0	1	1
9	1	0	0	1	1	X
10	1	0	1	0	0	0
11	1	0	1	1	1	1
12	1	1	0	0	1	1
13	1	1	0	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	0	0

Pravdivostní tabulka 3 - Tvorba výstupní proměnné Y_n s neurčitými stavy

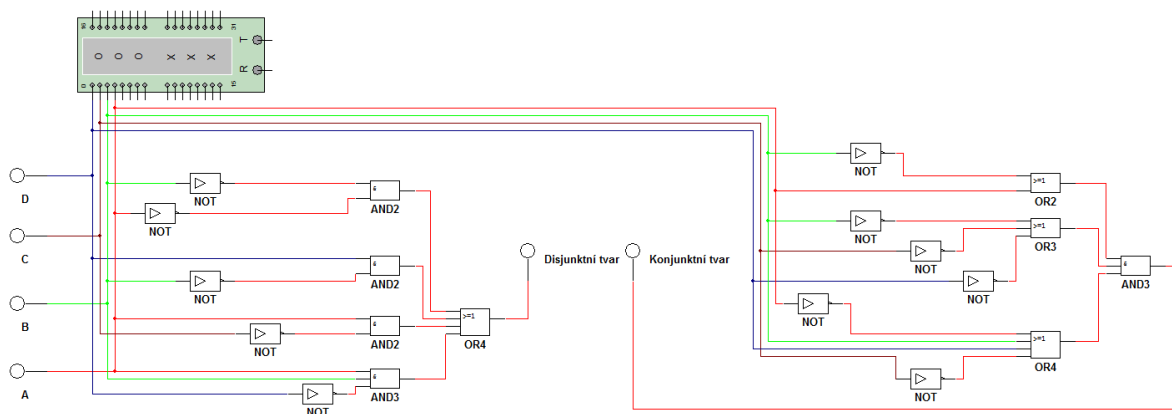
3.7.1 ZADÁNÍ

- Vytvořte novou logickou funkci s neurčitými stavy.
- Stanovte minimální disjunktivní a minimální konjunktivní tvar.
- Nechte zkontrolovat vedoucím semináře minimalizaci (smyčky).
- Ověřte chování v simulačním programu.

3.7.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

Dle zadání nahradíme některé bity neurčitými stavy. Nahrazení nám vytvořilo novou výstupní proměnou XX11XX001X011110, takže z nové výstupní proměnné musíme vytvořit oba minimalizované tvary. Lze použít minimalizace pomocí Karnaughovy mapy nebo Booleovy algebry. Neurčité stavy mohou nabývat logické jedničky i logické nuly, volíme tedy stavy tak, aby smyčky vycházely co největší. Zde použijeme minimalizaci pomocí Karnaughovy mapy. Funkce minimálního disjunktivního tvaru s neurčitými stavy pro novou výstupní proměnnou vypadá takto: $\bar{A}\bar{B} + \bar{B}D + A\bar{C} + AB\bar{D}$ a minimální konjunktivní funkce s neurčitými stavy $(A + \bar{B}) \times (\bar{B} + \bar{C} + \bar{D}) \times (\bar{A} + B + \bar{C} + D)$. Pomocí těchto tvarů za pomoci vhodných součástek (NOT, OR, AND a diod na výstupech) vytvoříme zapojení, které je připojené ke generátoru datových slov. Při kontrole této simulace se musí shodovat pouze ty bity, které nejsou neurčité.

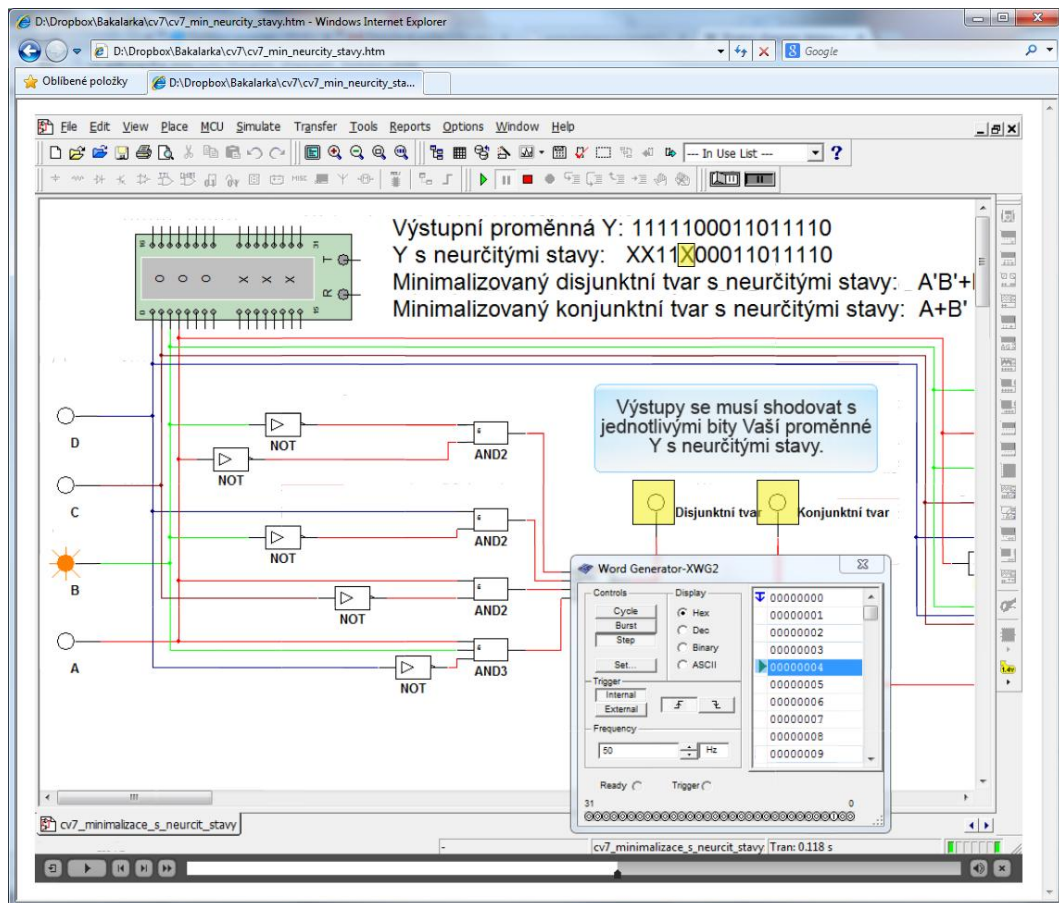
3.7.3 UKÁZKA OBRÁZKU



Obrázek 20 - Cvičení 7 - návrh řešení

3.7.4 UKÁZKA ANIMACE

Tato animaci zobrazuje zapojení minimalizovaného disjunktčního a konjunktčního tvaru nově vytvořené výstupní proměnné s neurčitými stavy.



Obrázek 21 - Ukázka animace 7. cvičení

3.7.5 OTÁZKY AUTOTESTU PRO CVIČENÍ 7

- 1) Funkce v základním součtovém tvaru je:
 - a) minterm
 - b) maxterm
 - c) plusterm
- 2) Neurčitý stav značíme:
 - a) X (text správně)
- 3) Vyberte správné tvrzení:

- a) každá funkce musí mít neurčitý stav
 - b) neurčitý stav nemá určenou hodnotu výstupní logické funkce
 - c) neurčité stavy nahrazují negaci
- 4) Minimalizaci funkce s neurčitými stavy provedeme pomocí:
- a) pouze Karnaughovy mapy
 - b) pouze zákonů Booleovy algebry
 - c) obou dvou výše zmíněných způsobů

Správné odpovědi: 1a, 3b, 4c,

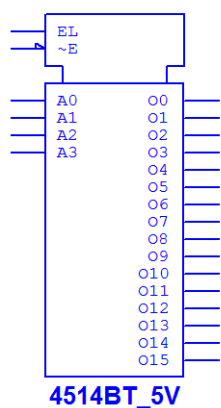
3.8 KOMPONENTY E-KURZU PRO CVIČENÍ 8 - VYUŽITÍ DEKODÉRŮ K REALIZACI LOGICKÝCH FUNKCÍ

V tomto cvičení si prakticky vyzkoušíme, jak pomocí dekodéru realizujete výstupní proměnnou. Použijeme dva typy. Dekodér s přímými výstupy (technologie CMOS) a dekodér s negovanými výstupy (technologie TTL). Více informací o dekodérech nalezneme v e-kurzu Číslíkové prvky a systémy, studijní článek: Dekodéry⁴.

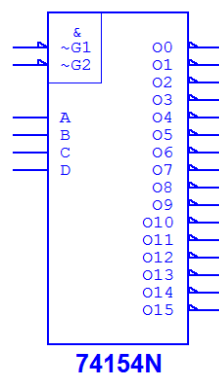
3.8.1 ZADÁNÍ

a) Realizujte logické funkce s využitím dekodérů:

- S přímými výstupy (IO např. 4514), BIN / 1 z 16.
- S negovanými výstupy (IO např. 74154), BIN / 0 z 16.
- Nakreslete blokově oba návrhy na papír.



Obrázek 22 - Dekodér 1/16



Obrázek 23 - Dekodér 0/16

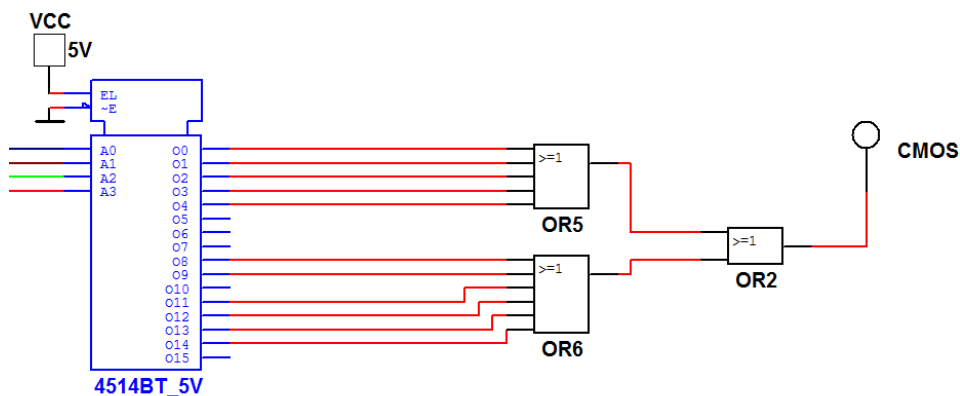
⁴ <http://www.kvd.zcu.cz/cz/materialy/cps/cps/cps/HTML/28/default.htm>

3.8.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

Dekodér převádí vstupní kód na jiný. Potřebujeme 16-ti bitový dekodér typu TTL s negovanými výstupy a dekodér typu CMOS s přímými výstupy. Dekodéry nalezneme na horní paletě, v záložce Place TTL a Place CMOS. Pro jistotu se podíváme u obou dekodéru na informace, kvůli zjištění, které označení je pro nejvíce významový bit a které pro nejméně významový bit. Po rychlém prozkoumání zjistíme, že u TTL dekodéru je LSB na vstupu s označením A a MSB na vstupu s označením D. Musíme tedy výstupy generátoru datových slov správně připojit. Takže nejméně významový bit, který jsme doposud označovali jako D musíme připojit na vstup A a naopak. Dále bychom měli na výstupy TTL dekodéru připojit součástku jedenáctivstupový NAND (Výstupy dekodéru jsou negované), tato součástka, ale není v programu implementována, a tak použijeme stejnou alternativu jako ve cvičení 3. Počet vstupů se odvíjí od počtu logických jedniček ve výstupní proměnné. K součástce NAND ale připojíme pouze ty bity, které ve výstupní proměnné odpovídají hodnotě logické jedničky.

U CMOS dekodéru zjistíme, že MSB je označen vstupem A3 a LSB vstupem A0. Správně připojíme k výstupům generátoru datových slov. Výstupy CMOS dekodérů zapojíme stejným principem jako u TTL dekodéru, pouze s tím rozdílem, že místo součástky NAND musíme použít součástku OR. Protože výstupy dekodéru odpovídají řádkům pravdivostní tabulky s hodnotou logické nuly. Na výstupy připojíme diody a generováním datových slov zjišťujeme, zdali máme zapojení správně a jestli se výstupy shodují s výstupní proměnnou Y.

3.8.3 UKÁZKA OBRÁZKU



Obrázek 24 - Cvičení 2 - návrh řešení

3) Pokud má výstupní proměnná 5 bitů kolika vstupový dekodér potřebujete:

- a) 2 vstupový
- b) 3 vstupový
- c) 4 vstupový

Vysvětlení: Dvouvstupový nám umožní pouze 4 kombinace (potřebujeme pro 5 bitů), čtyřvstupový 16 kombinací, takže nejvhodnější je třívstupový, který nám umožní 8 kombinací.

4) Musí mít vstupní a výstupní datové slovo dekodéru stejnou velikost:

- a) ne

Vysvětlení: Vyrábějí se různé dekodéry, které převádí jinak dlouhé slova na jiné, takže to není podmínkou.

5) Který druh kódu u dekodérů datového slova se nepoužívá:

- a) BCD 0 z n
- b) sedmissegmentovka
- c) BFD 1 z n

Vysvětlení: BFD není zkratka žádného dekodéru.

6) Dekodér s negovanými vstupy se značí:

- a) 1 z 0
- b) 0 z n
- c) n z n

Správné odpovědi: 1a, 3b, 5c, 6b

3.9 KOMPONENTY E-KURZU PRO CVIČENÍ 9 - VYUŽITÍ MULTIPLEXERU K REALIZACI LOGICKÝCH FUNKCÍ

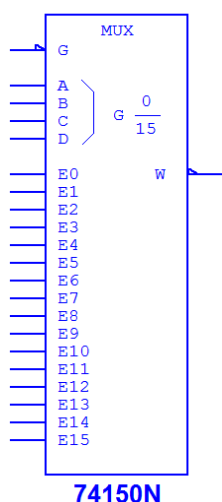
V tomto cvičení využijeme multiplexery k realizaci výstupní proměnné. Vyzkoušíme dva způsoby. Klasickou metodu (multiplexer s šestnácti datovými vstupy) a úspornou metodu (multiplexer s osmi datovými vstupy). Popis jednotlivých metod

nalezneme v e-kurzu Číslicové prvky a systémy, studijní článek: Multiplexery a demultiplexery⁵.

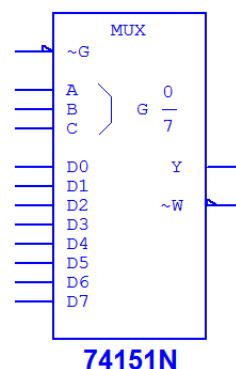
3.9.1 ZADÁNÍ

a) Realizujte logické funkce s využitím multiplexerů:

- Klasická metoda (IO např. 74150), BIN / 1 z 16, (pozor na negovaný výstup).
- Úsporná metoda (IO např. 74151), BIN / 1 z 8.
- Nakreslete blokově oba návrhy na papír.
- Ověřte chování v simulačním programu.



Obrázek 26 - Multiplexer 1/16



Obrázek 27 - Multiplexer 1/8

3.9.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

Multiplexer přepíná logickou proměnou ze svých vstupů na jeden výstup. Multiplexery nalezneme na horní paletě v záložce TTL. Pro účely zadání použijeme jeden šestnactivstupový, který je zapojený klasickou metodou a jeden osmivstupový, který je zapojen úspornou metodou. U šestnactivstupového multiplexeru se nejprve opět podíváme do informací, kde zjistíme, že MSB je na vstupu označeném písmenem D a LSB je vstupu označeném písmenem A. Následně připojíme adresové vstupy na příslušné bity generátoru datových slov. Zapojení klasickou metodou spočívá v tom, že na vstupy multiplexeru (E0 - E15) připojíme napětí (logickou jedničku) na ty vstupy, které odpovídají

⁵ <http://www.kvd.zcu.cz/cz/materialy/cps/cps/cps/HTML/29/default.htm>

logické jedničky ve výstupní proměnné Y. Logické nuly ve výstupní proměnné Y uzemníme. Pokud tedy máme výstupní proměnnou 11110001101110. Bity na pozici 5, 6, 7, 10 a 15 uzemníme, ostatní připojíme na zdroj napětí. Důležité je ještě nezapomenout na to, že výstup tohoto multiplexeru je negovaný, takže, aby jsme dostali správné výstupní hodnoty musíme na výstup připojit součástku NOT.

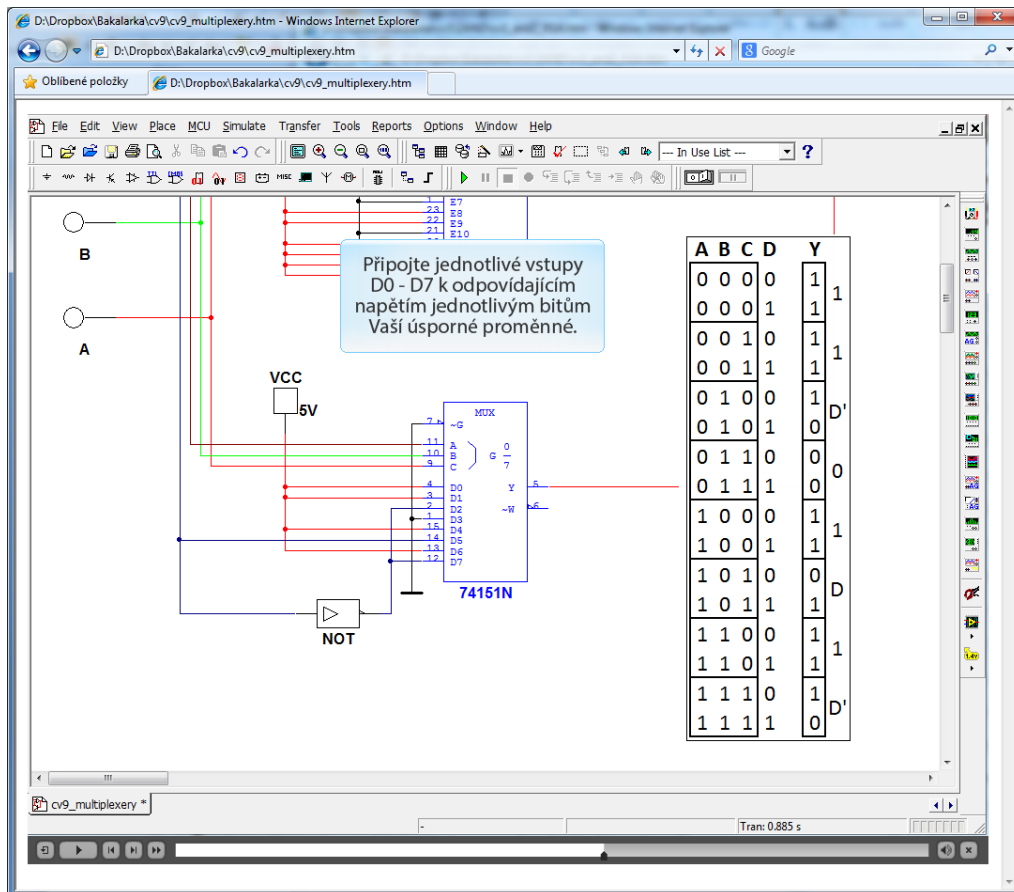
Na zapojení s úspornou metodou nám postačí pouze osmivstupový multiplexer. Vytvoříme si pravdivostní tabulku, kterou rozdělíme po dvou řádcích. V každém dvouřádku se mění pouze hodnota proměnné D. Hodnota může nabývat čtyř stavů, logické jedničky, logické nuly, hodnotě D, nebo hodnotě negovaného D. Jednotlivé varianty poté jen porovnáme s výstupní proměnnou Y a správně připojíme. Nakonec na adresovací vstupy připojíme hodnoty poměňých A, B, C, které rozlišují jednotlivé dvouřádky.

bit	A	B	C	D	Y	Y _n
0	0	0	0	0	1	1
1	0	0	0	1	1	
2	0	0	1	0	1	1
3	0	0	1	1	1	
4	0	1	0	0	1	\bar{D}
5	0	1	0	1	0	
6	0	1	1	0	0	0
7	0	1	1	1	0	
8	1	0	0	0	1	1
9	1	0	0	1	1	
10	1	0	1	0	0	D
11	1	0	1	1	1	
12	1	1	0	0	1	1
13	1	1	0	1	1	
14	1	1	1	0	1	\bar{D}
15	1	1	1	1	0	

Pravdivostní tabulka 4 - Úsporná varianta multiplexeru 1/8

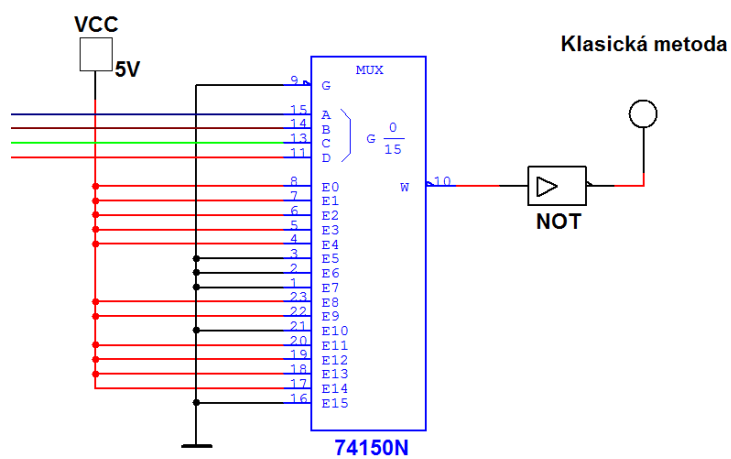
3.9.3 UKÁZKA ANIMACE

Tato animaci demonstruje využití multiplexeru k realizaci funkce pomocí klasické a úsporné metody.



Obrázek 28 - Ukázka animace 9. cvičení

3.9.4 UKÁZKA OBRÁZKU



Obrázek 29 - Cvičení 9 - návrh řešení

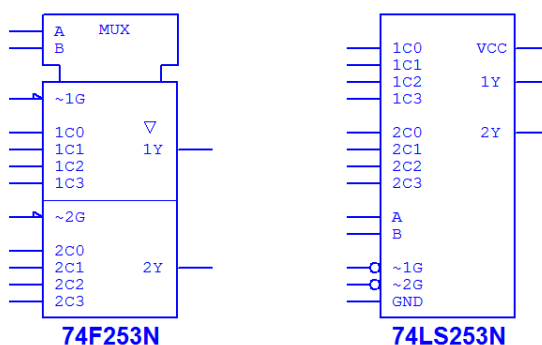
3.10 KOMPONENTY E-KURZU PRO CVIČENÍ 10 - REALIZACE LOGICKÉ FUNKCE POMOCÍ MULTIPLEXERU S TŘÍ-STAVOVÝMI VÝSTUPY

V tomto cvičení využijeme multiplexer s tří-stavovými výstupy. Potřebné informace nalezneme v e-kurzu Číslkové prvky a systémy, studijní článek: Multiplexery a demultiplexery⁶.

3.10.1 ZADÁNÍ

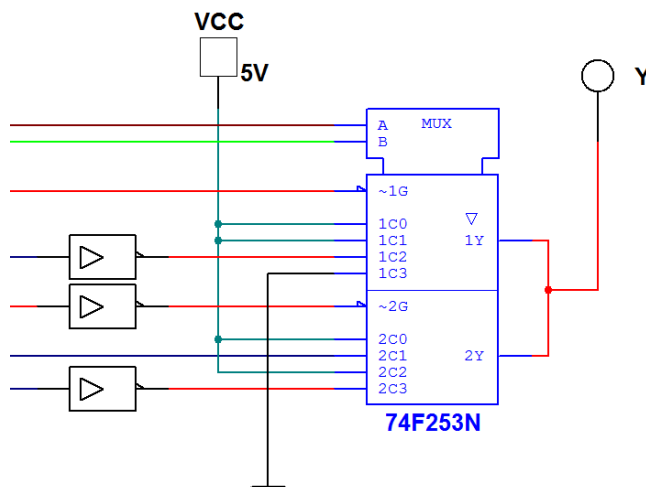
a) Realizujte logické funkce s využitím multiplexerů s tří-stavovými výstupy

- Využijte např. IO 74253.
- Nakreslete blokově oba návrhy na papír.
- Ověřte chování v simulačním programu.



Obrázek 30 - Multiplexery s tří-stavovými výstupy

3.10.2 UKÁZKA OBRÁZKU



Obrázek 31 - Cvičení 10 - návrh řešení

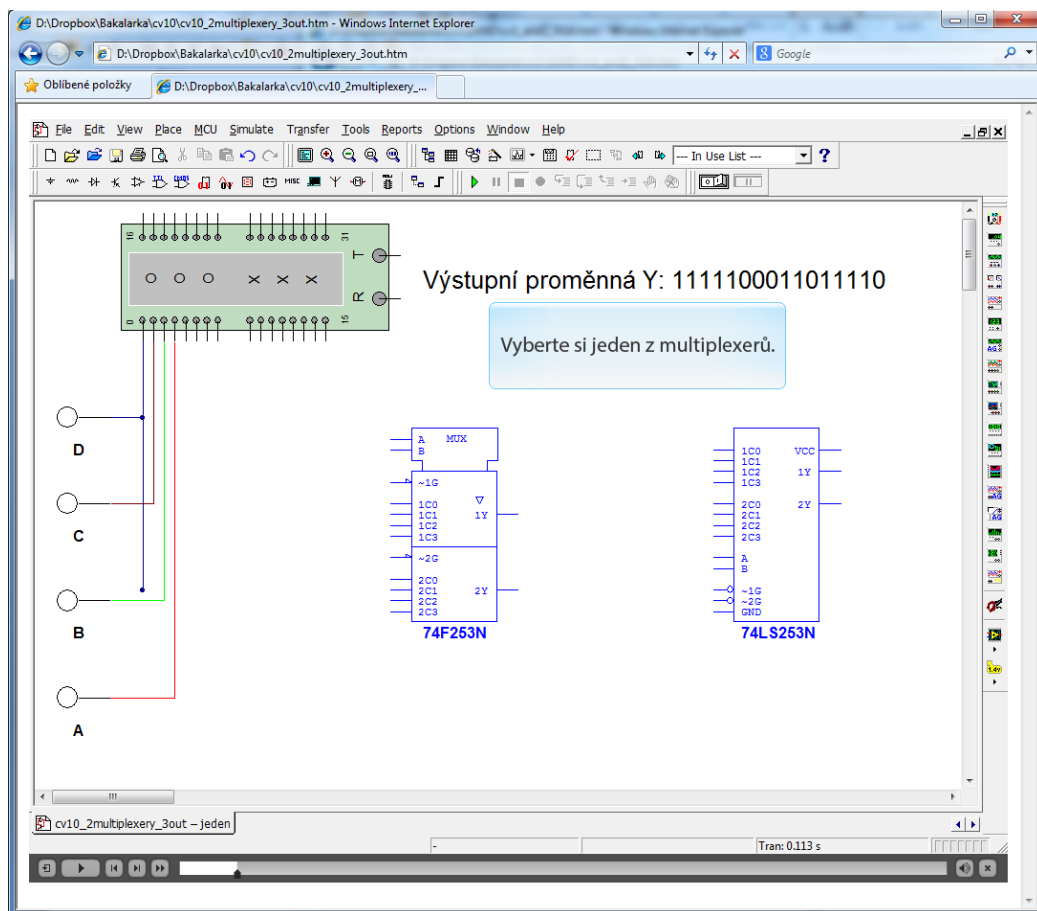
⁶ <http://www.kvd.zcu.cz/cz/materialy/cps/cps/cps/HTML/29/default.htm>

3.10.1 VYPRACOVÁNÍ SIMULACE V MULTISIMU

Použijeme multiplexer s označením 74F253N. Vycházíme z pravdivostní tabulky 4 (úsporné varianty multiplexeru). Podíváme se do nápovědy k součástce, kde se dozvíme, které vstupy co znamenají a kam je připojit. Vstup A je LSB připojíme tedy na výstup bitu C, vstup B na výstup bitu B a řídicí vstupy 1G a 2G připojíme k výstupu A a k negovanému A. Vstupy pro data jsou označeny C0 - C3. Výstupní proměnná úsporné varianty je ve tvaru $11\bar{D}01D1\bar{D}$ první čtyři bity připojíme k příslušným datovým vstupům označených 1Cx a zbylé čtyři bity ke vstupům označených 2Cx. Například vstup 1C0 odpovídá prvnímu bitu, logické jedničce, přivedeme tedy zdroj napětí. Vstup 2C3 odpovídá osmému bitu, negovanému D, připojíme na výstup nejprve negaci a až výstup negace na vstup 2C3. Na výstup připojíme diodu pro kontrolu.

3.10.2 UKÁZKA ANIMACE

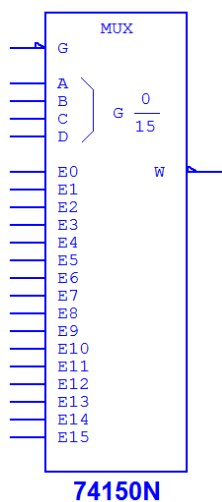
V této animaci realizujeme funkci pomocí multiplexeru s tří-stavovými výstupy.



Obrázek 32 - Ukázka animace 10. cvičení

3.10.3 OTÁZKY AUTOTESTU PRO CVIČENÍ 9 A 10

- 1) Multiplexer funguje na principu:
 - a) přepínače
 - b) klopného obvodu
 - c) komparátoru
- 2) Demultiplexer má:
 - a) jeden adresní vstup a mnoho vstupů
 - b) jeden vstup, mnoho výstupů, adresní vstupy
 - c) mnoho adresovacích vstupů a mnoho výstupů
- 3) Kolik adresových vstupů má multiplexer 74150N
 - a) 4 (text správně)



Vysvětlení: A, B, C, D jsou adresové vstupy.

- 4) Vyberte neexistující metodu:
 - a) klasická
 - b) úsporná
 - c) záložní

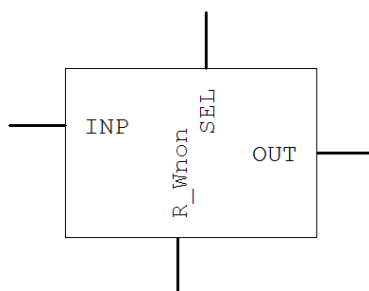
Správné odpovědi: 1a, 2b, 4c

3.11 KOMPONENTY E-KURZU PRO CVIČENÍ 11 - REALIZACE LOGICKÉ FUNKCE POMOCÍ POLOVODIČOVÉ PAMĚTI

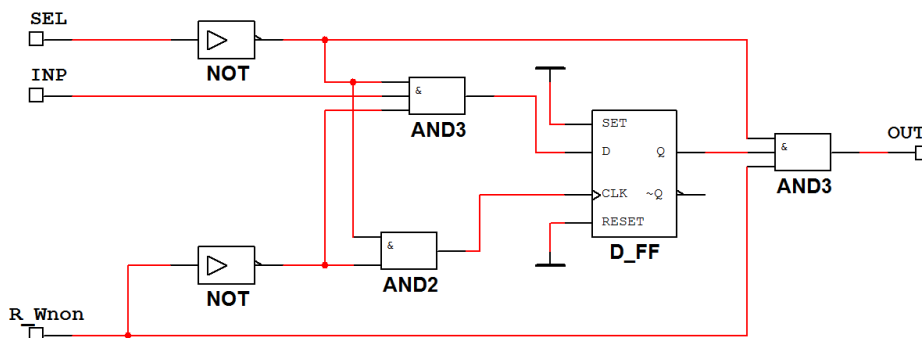
V tomto cvičení vytvoříme šestnácti bitovou polovodičovou paměť. Součástka polovodičová paměť není v Multisimu implementována. Musíme ji tedy vytvořit za pomoci subobvodů. Budeme potřebovat subobvod paměťové buňky a subobvod osmibitové polovodičové paměti.

1) Paměťová buňka

Paměťová buňka musí obsahovat vstup (INPut), výstup (OUTput), výběr (SElect) a zápis/čtení (ReadWriteNON). Do subobvodu umístíme součástku klopný obvod a připojíme ostatní součástky dle vzoru.



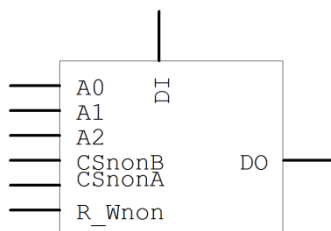
Obrázek 33 - Schéma paměťové buňky



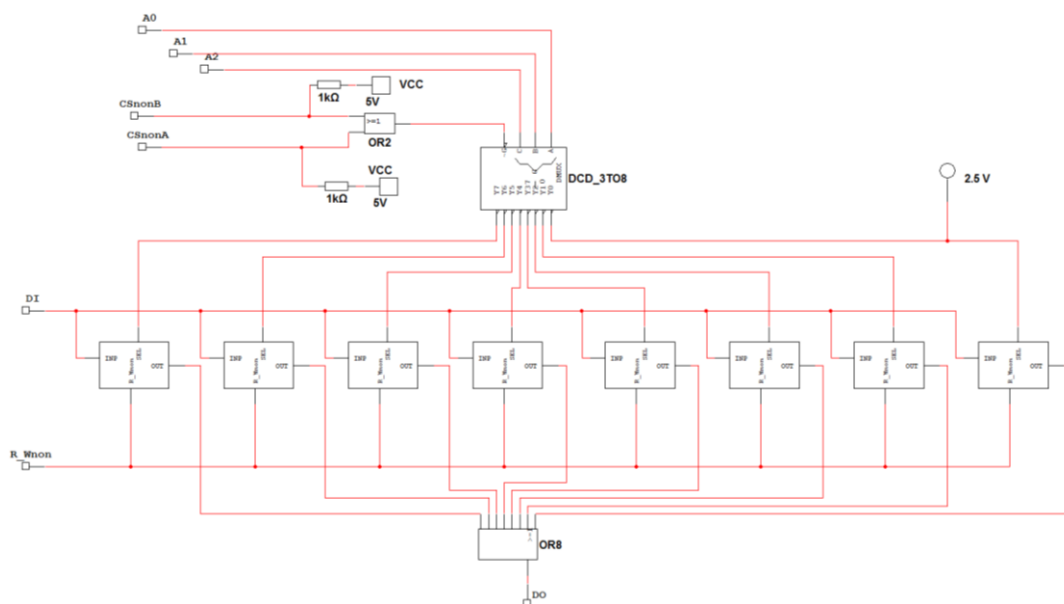
Obrázek 34 - Subobvod paměťové buňky

2) Osmibitová polovodičová paměť

Paměť musí obsahovat vstup pro data (Data Input), výstup pro data (DataOutput), adresní vstupy (A0, A1, A2), výběr (CS) a zápis/čtení (ReadWriteNON). Do subobvodu umístíme součástku dekodér 3/8, 8 paměťových buněk a připojíme ostatní součástky dle vzoru.



Obrázek 35 - Schéma osmibitové polovodičové paměti



Obrázek 36 - Subobvod osmibitové polovodičové paměti

3.11.1 ZADÁNÍ

a) Realizujte logické funkce pomocí polovodičové paměti.

- Nakreslete blokové schéma.

3.11.2 VYPRACOVÁNÍ SIMULACE V MULTISIMU

V tomto cvičení musíme vytvořit dva subobvody, subobvod je vlastně námi vytvořená součástka. Subobvod vytvoříme tak, že klikneme pravým tlačítkem myši na pracovní plochu Multisimu a zvolíme Place Schematic → New Subcircuit. Pojmenujeme ho a vložíme na plochu. Poté dvojklikneme na obvod a dáme Edit HB/SC. Otevře se nám nová pracovní plocha, ovšem veškeré změny, které zde provedeme, se nám projeví v součástce jako takové.

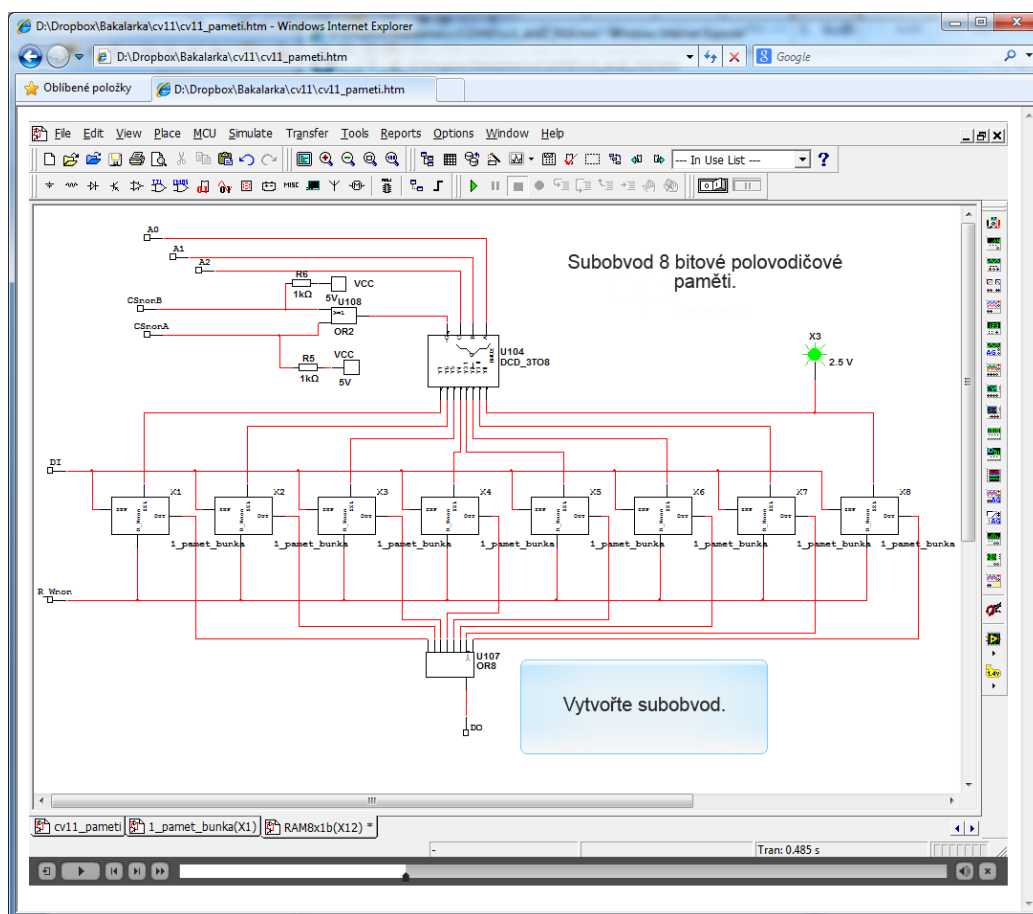
Vytvoříme zapojení dle schématu, součástku klopného obvodu najdeme na horní paletě pod názvem Misc Digital, v záložce TIL pod názvem D_FF. Ke vstupům a výstupům musíme připojit konektory. Součástku konektoru nalezneme, když pravým tlačítkem klikneme na pracovní plochu a vybereme Place Schematic → HB/SC Conector. Doporučuje se si jednotlivé vstupy a výstupy popsat. Tato paměťová buňka bude součástí obvodu osmibitové polovodičové paměti. Takže si vytvoříme další subobvod a vložíme do něho osm těchto buněk a zapojíme je dle vzoru. Součástku dekodéru nalezneme na horní paletě pod názvem Misc Digital, v záložce TIL pod zkratkou DCD_3TO8. Tímto máme vytvořený subobvod, který nám představuje osmibitovou polovodičovou paměť. K realizaci logické funkce musíme mít šestnáctibitovou polovodičovou paměť, takže subobvod paměti zduplikujeme a zapojíme zbylé součástky (přepínače, NOT, OR, diody) dle vzoru. K přepínačům nastavíme klávesové zkratky pro rychlejší práci s obvodem. Klávesa A určuje, jestli se bude do paměti zapisovat (log. 1) nebo se z ní bude číst (log. 0). Klávesa D určuje, jestli se zapíše logická nula nebo logická jednička. Vstupní datové slovo se nastaví kombinací stisku kláves Q, W, E, R, kde Q je MSB a R je LSB. Takže pokud chceme na pozici nultého bitu zapsat logickou jedničku, přepneme přepínač A do polohy logické jedničky (zápis), přepneme klávesu D do polohy zapnuto (log 1) a necháme všechny přepínače v poloze uzemněno (log. 0, kombinace 0000 odpovídá nultému bitu) a klávesou S přepneme do polohy logické jedničky a následovně zpět. Tímto způsobem jsme na nultý bit zapsali logickou jedničku. Pokud chceme zapsat logickou nulu na pozici osmého bitu, přepneme přepínač Q do pozice zapnuto (log. 1) a ostatní přepínače necháme vypnuté (log. 0). Tím dostaneme kombinaci 1000, která nám odpovídá osmému bitu. Přepneme klávesu D do pozice uzemněno, abychom zapsali logickou nulu. Pomocí

přepnutí přepínače S do pozice napětí a zpět uzemněno uložíme do paměti logickou nulu na pozici osmého bitu. Tímto principem zapíšeme všechny bity výstupní proměnné.

Pro čtení z paměti musíme přepnout polohu přepínače A do polohy uzemněno (čtení), dále už jen nastavujeme vstupní kombinaci. Pokud chceme přečíst bit na pozici patnáct (binárně 1111), přepneme všechny přepínače do polohy logické jedničky a na výstupu se nám zobrazí zapsaný bit na této pozici, v tomto případě to je nula. Výstupy jednotlivých bitů se musí shodovat s výstupní proměnou Y.

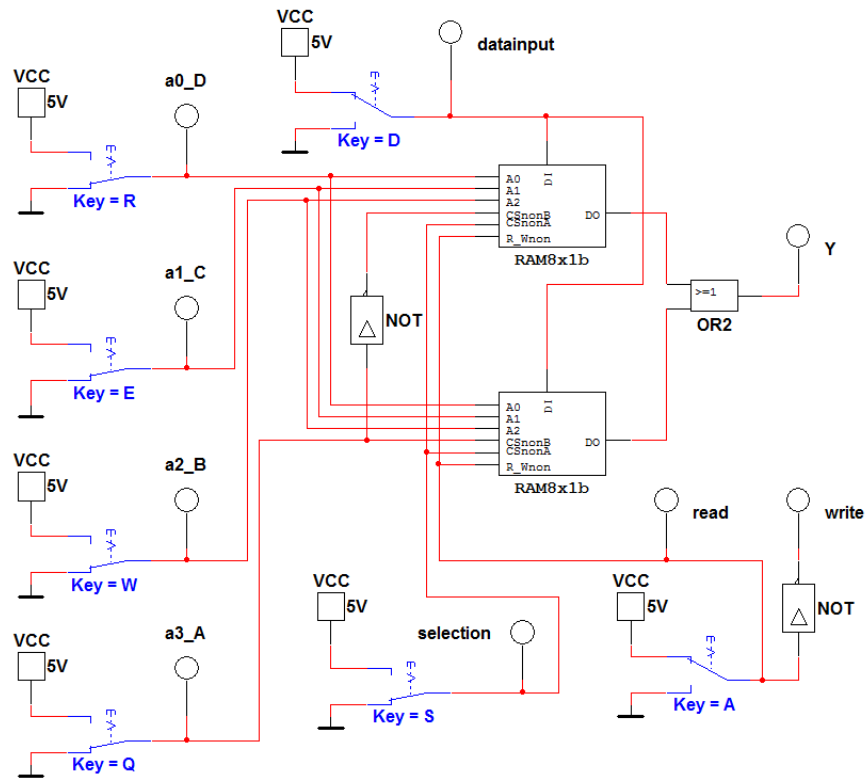
3.11.3 UKÁZKA ANIMACE

V této animaci posledního cvičení demonstrujeme jak vytvořit jednotlivé subobvody nezbytné pro vytvoření polovodičové paměti. První subobvod je jednobitová paměťová buňka a druhý subobvod je osmibitová polovodičová paměť. Dále je zobrazeno jak pomocí přepínačů provádět zápis a čtení jednotlivých bitů z nebo do paměti.



Obrázek 37 - Ukázka animace 11. cvičení

3.11.1 UKÁZKA OBRÁZKU



Obrázek 38- Cvičení 11 - návrh řešení

3.11.2 OTÁZKY AUTOTESTU PRO CVIČENÍ 11

- 1) K uložení jednoho bitu nám postačí:
 - a) jedna dvoubitová polovodičová paměť
 - b) jedna paměťová buňka
 - c) osm paměťových buněk
- 2) Polovodičová paměť typu RAM pracuje na principu:
 - a) sekvenční přístup k datům
 - b) libovolný přístup k datům
 - c) paralelní přístup k datům
- 3) Přepínač SELECT slouží k:
 - a) smazání bitu z paměti
 - b) uložení bitu do paměti
 - c) přepsání bitu v paměti

- 4) Vyberte správné tvrzení::
- a) do polovodičové paměti se může pouze zapisovat
 - b) z polovodičové paměti se může pouze číst
 - c) možnosti zápisu/čtení jsou ovlivněny typem paměti
- 5) Jaká je zkratka statické nevolativní paměti:
- a) ROM (text správně)
- 6) Hlavní parametry u polovodičových pamětí jsou:
- a) vybavovací doba
 - b) kapacita
 - c) výrobní technologie
 - d) cena za bit
 - e) přenosová rychlost

Správné odpovědi: 1b, 2b, 3b, 4c, 6abde

ZÁVĚR

Cílem této bakalářské práce bylo vytvořit komponenty, které se použijí při tvorbě samostatného e-kurzu Příklady číslicových prvků a systémů. Nejdůležitější komponentami celé bakalářské práce jsou zajisté animace, které mají provázet studenta konkrétní problematikou a mají mu pomoci při řešení problému. Tyto animace jsem vytvářel se vším úsilím a znalostmi, které mám. I přes veškerou snahu, kterou jsem do tvorby animací vložil, věřím, že by stále šli jednotlivé animace vylepšovat a upravovat. Jako dobré vylepšení mě napadl hlasový doprovod, ale to je časově velmi náročné a na to mi bohužel nezbyl čas. U každé animace jsem se snažil o ucelenou strukturu, aby byla jednotlivá cvičení co nejpodobnější a nemátli tak studenta přílišnými změnami.

Ačkoliv jsem si to ze začátku myslel, tak tato práce není jen o animacích. U každého cvičení je v první řadě zadání. Zadání, které je vloženo u každého cvičení bylo po domluvě s vedoucím práce převzato ze seminářů předmětu. Další nezbytně nutnou komponentou jsou obrázky, které se vyskytují v každém cvičení. Některé obrázky se zaměřují na vysvětlení fungování některých součástí, jiné zase součástky pouze zobrazují. Na konci každého cvičení v e-kurzu je tip pro řešení a těžko bych hledal lepší formát než právě obrázek. Proto u každého cvičení je v tipech pro řešení obrázek jak by přibližně mělo finální zapojení vypadat. U těchto obrázků jsem si dal záležet na rozlišení, aby byl dobře vidět každý spoj a každá součástka. Vzhledem k malé podpoře tabulek v programu ProAuthor jsem nakonec převedl i tabulky a rovnice do obrázkového formátu, abych je mohl použít v některých cvičeních, nebo dokonce i autotestech. Autotesty, respektive sady otázek pro autotesty jsou další nedílnou součástí nutných komponent pro vytvoření e-kurzu. Veškeré otázky jsem se snažil vytvořit tak, aby korespondovali s konkrétním cvičením a zároveň prověřili naučené znalosti studentů. U některých otázek je vysvětlení proč tomu tak je, u ostatních to nebylo zapotřebí.

Animace, obrázky, autotesty a zadání jsem importoval do e-kurzu v ProAuthoru a vytvořil tak základ pro nastávající kurz. Výhodou programu ProAuthor je hojně používání na katedře a tak případný student, který bude e-kurz dokončovat, by neměl mít žádný problém s orientací v programu.

Z osobní zkušenosti vím, že sestavit některá zapojení není jednoduché a pevně věřím, že tyto mnou vytvořené komponenty pomohou novým studentům lépe a rychleji pochopit probíranou látku.

Všechny komponenty, které jsem vytvářel, jsou na přiloženém CD, včetně těch, které jsem nepoužil při tvorbě e-kurzu. Víím, že se na dokončování bude ještě hodně pracovat, tak mi přišlo vhodné tyto nepoužité komponenty poskytnout v případě budoucích úprav. Musím přiznat, že je docela pěkný pocit vědět, že se za pár let budou studenti učit z něčeho, co jsem pomáhal vytvářet.

Bakalářskou práci mám rozdělenou na dvě části, v první části se věnuji animacím a programům, ve kterých jsem tvořil. V druhé části se věnuji už konkrétním cvičením, která jsou součástí harmonogramu předmětu ČPS. Vzhledem k tomu, že prioritou mé práce bylo vytvořit animace, obrázky a autotesty, se u každého cvičení podrobněji zabývám pouze tím, jak jsem postupoval při tvorbě simulačního modelu. Všechny nezbytné informace k teorii jsou již vysvětleny v e-kurzu Číslicové prvky a systémy, na který v několika případech odkazuji.

Vzhledem k dnešní době, kdy elektronické materiály a skripta jsou téměř samozřejmostí, věřím, že po dokončení bude tento kurz prospěšný jak vyučujícím tak studentům.

RESUMÉ

V této práci jsem vytvořil několik komponent, které věřím, pomohou studentům při studiu předmětu Číslicové prvky a systémy. Zaměřil jsem se především na tvorbu animací, obrázků a autotestů. Zde v práci jsou uvedeny jen vybrané ukázky některých komponent, všechny ostatní komponenty jsou na přiloženém CD. Všechny mnou vytvořené komponenty jsem importoval do šablony e-kurzu.

Myslím si, že internetové e-kurzy jsou u studentů velmi oblíbené a tak věřím, že po dokončení tohoto kurzu bude studenty hojně využíván.

In this work I have created several components that will help students in the study of the subject Číslicové prvky a systémy. I focused mainly on creating animations, images, and auto-tests. Here in work are given only showcases some of the components, all other components are on the enclosed CD. All the components were created by me. I imported components into a template of e-course.

I think that internet e-courses are very popular among students and so I believe that after completion this course, will be widely used by students.

SEZNAM OBRÁZKŮ A TABULEK

Obrázek 1 - vzhled přehrávače - Captivate default	5
Obrázek 2 - prostředí programu Adobe Captivate 8	5
Obrázek 3 - prostředí programu Multisim.....	6
Obrázek 4 - prostředí programu ProAuthor	7
Obrázek 5 - Generátor datových slov	8
Obrázek 6 - Logický konvertor	9
Obrázek 7 - Cvičení 1 - návrh řešení	11
Obrázek 8 - Ukázka animace 1. cvičení.....	12
Obrázek 9 - Cvičení 2 - návrh řešení	14
Obrázek 10 - Ukázka animace 2. cvičení.....	15
Obrázek 11 - Ukázka animace 3. cvičení.....	18
Obrázek 12 - Cvičení 3 - návrh řešení	19
Obrázek 13 - Cvičení 4 - návrh řešení	21
Obrázek 14 - Ukázka animace 4. cvičení.....	22
Obrázek 15 - Cvičení 5 - návrh řešení	25
Obrázek 16 - Ukázka animace 5. cvičení.....	26
Obrázek 17 - Vnitřní struktura integrovaných obvodů	28
Obrázek 18 - Ukázka animace 6. cvičení.....	29
Obrázek 19 - Cvičení 6 - návrh řešení	30
Obrázek 20 - Cvičení 7 - návrh řešení	32
Obrázek 21 - Ukázka animace 7. cvičení.....	33
Obrázek 22 - Dekodér 1/16 Obrázek 23 - Dekodér 0/16	34
Obrázek 24 - Cvičení 2 - návrh řešení	35
Obrázek 25 - Ukázka animace 8. cvičení.....	36
Obrázek 26 - Multiplexer 1/16 Obrázek 27 - Multiplexer 1/8.....	38
Obrázek 28 - Ukázka animace 9. cvičení.....	40
Obrázek 29 - Cvičení 9 - návrh řešení	40
Obrázek 30 - Multiplexery s tří-stavovými výstupy	41
Obrázek 31 - Cvičení 10 - návrh řešení	41
Obrázek 32 - Ukázka animace 10. cvičení	42
Obrázek 33 - Schéma paměťové buňky	44
Obrázek 34 - Subobvod paměťové buňky	44
Obrázek 35 - Schéma osmibitové polovodičové paměti	45
Obrázek 36 - Subobvod osmibitové polovodičové paměti	45
Obrázek 37 - Ukázka animace 11. cvičení	47
Obrázek 38- Cvičení 11 - návrh řešení	48
Pravdivostní tabulky - pro dvě, tři a čtyři vstupní proměnné.....	10
Pravdivostní tabulka 1 - pro $f = AB + ABC + ABCD$	13
Pravdivostní tabulka 2 - Tvorba výstupní proměnné Y	17
Pravdivostní tabulka 3 - Tvorba výstupní proměnné Y_n s neurčitými stavy	31
Pravdivostní tabulka 4 - Úsporná varianta multiplexeru.....	39

SEZNAM POUŽITÉ LITERATURY

1. ADOBE® CAPTIVATE® HELP. *Adobe*. [online]. 10.13.2014 [cit. 2015-03-17].
Dostupné z: https://helpx.adobe.com/pdf/captivate_reference.pdf
2. ANTOŠOVÁ, Marcela a Vratislav DAVÍDEK. *Číslicová technika: [učebnice]*. 2. vyd. České Budějovice: Kopp, 2007, 286 s. ISBN 978-80-7232-314-2.
3. *Grafika.cz – vše o počítačové grafice*. [online]. [cit. 2015-04-10]. Dostupné z: <http://www.grafika.cz/>
4. JURÁNEK, Antonín. *MultiSIM - elektronická laboratoř na PC: schémata a zapojení*. 1. vyd. Praha: BEN - technická literatura, 2008, 287 s. ISBN 978-80-7300-194-0.
5. MATOUŠEK, David. *Číslicová technika: základy konstruktérské praxe*. 1. vyd. Praha: BEN - technická literatura, 2001, 207 s. ISBN 80-7300-025-3.
6. MICHALÍK, Petr a Pavel BENAJTR. eBook. *Multisim NI 10*. [online]. 7.7.2011 [cit. 2015-02-25]. Dostupné z: <http://www.kvd.zcu.cz/cz/materialy/msm/msm/msm/>
7. MICHALÍK, Petr a Petr SEMRÁD. eBook. *Číslicové prvky a systémy*. [online]. 8.3.2007 [cit. 2014-11-11]. Dostupné z: <http://www.kvd.zcu.cz/cz/materialy/cps/cps/cps/>
8. Multisim™- User Guide. *National Instruments*. [online]. 2007 [cit. 2015-03-11]. Dostupné z: <http://www.ni.com/pdf/manuals/374483a.pdf>
9. SWF FILE FORMAT SPECIFICATION VERSION 19. *Adobe*. [online]. 2006-2012 [cit. 2015-03-11]. Dostupné z: <http://www.images.adobe.com/content/dam/Adobe/en/devnet/swf/pdf/swf-file-format-spec.pdf>

PŘÍLOHY

Na přiloženém CD se nacházejí tyto přílohy:

1. Bakalářská práce ve formátu docx a pdf
2. Animace ve formátu cptx a swf
3. Simulace zapojení ve formátu ms10
4. Obrázky ve formátu png
5. Kurz vytvořený v ProAuthoru