

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2009

Petr Vamberský

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Simulátor EPROM

Petr Vamberský

Bakalářská práce

2009

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra elektrotechniky
Akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr VAMBERSKÝ**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**

Název tématu: **Simulátor EPROM**

Z á s a d y p r o v y p r a c o v á n í :

Navrhněte zařízení, které bude simulovat činnost paměti EPROM. Zařízení se jednou ze svých částí bude připojovat do cílového zařízení (DIL patice), kde svými elektrickými a mechanickými parametry bude plně emulovat činnost paměti EPROM typu M27V201 (pouze čtení). Cílovým zařízením bude vývojový kit ADSP-21065L EZ-LITE od firmy Analog Devices. Nahrávání obsahu bude probíhat z nadřazeného počítače třídy PC, a sice přes rozhraní USB. Pro tyto účely vytvořte program pro operační systém Microsoft Windows. Program musí do simulátoru umožnit nahrát binární soubory a soubory ve formátu Intel-HEX. Funkčnost zařízení ověřte stavbou vzorku. Osnova práce: Návrh konstrukčního uspořádání Stanovení technických parametrů a návrh koncepce zařízení Návrh zařízení Vývoj řídicího programu Ověření funkčnosti zařízení

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Nagel Ch., Evjen B., Glynn J., Skinner M.,W.: C 2005 - Programujeme profesionálně, Computer Press, 2005 Petzold, Ch.: Programování Microsoft Windows Forms v jazyce C, Computer Press, 2006 Váňa, V.: Mikrokontroléry ATMEL AVR - popis procesoru a instrukční soubor, BEN - technická literatura, Praha 2003 Katalogové listy, aplikační poznámky a další informace dosažitelné u výrobců použitých součástí na stránkách www

Vedoucí bakalářské práce:

Ing. Martin Hájek
Katedra elektrotechniky

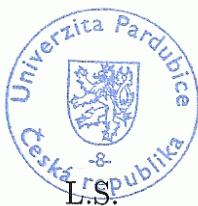
Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**



doc. Ing. Simeon Karamazov, Dr.

děkan



Ing. Zdeněk Němec, Ph.D.

vedoucí katedry

V Pardubicích dne 31. března 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 13. 8. 2009

Petr Vamberský

Souhrn

Cílem práce je návrh zařízení simulujícího funkci paměti EPROM a vytvoření ovládacího programu pro počítač třídy PC s operačním systémem Windows. Zařízení bude s PC propojeno rozhraním USB. Skrze toto rozhraní do simulátoru pomocí ovládacího programu možno nahrát uživatelská data.

Klíčová slova

Simulátor, Emulátor, EPROM, M27V201, AVR, SRAM, USB, C#, Intel-HEX

Abstract

This work is focussed on design of device imitating EPROM memory function and creation of controll program for computer PC class containing operating system Windows. The device will be connected with PC by USB interface. It will be possibility to record user data into simulator through this interface.

Keywords

Simulator, Emulator, EPROM, M27V201, AVR, SRAM, USB, C#, Intel-HEX

Poděkování

Touto cestou bych rád poděkoval svému vedoucímu práce ing. Martinu Hájkovi za přínosné rady a tipy v průběhu celé práce stejně jako za prokázanou trpělivost, které byla nutná nemalá dávka. Děkuji.

Petr Vamberský

Obsah

1	Úvod.....	10
2	Požadavky na zařízení.....	11
2.1	Fyzické propojení.....	11
2.2	Elektrické parametry	11
2.2.1	Stejnoseměrné (DC) parametry	11
2.2.2	Střídavé (AC) parametry	12
3	Navrhované řešení.....	13
3.1	Rozhraní simulátor – cílové zařízení	13
3.2	Rozhraní simulátor – PC.....	14
3.3	Propojení obou rozhraní.....	14
4	Hardware	16
4.1	Návrh zapojení – blokové schéma	16
4.2	Výběr součástek	18
4.2.1	SRAM.....	18
4.2.2	Budiče.....	18
4.2.3	Řídící procesor.....	19
4.3	Kontrola elektrických parametrů simulátoru	20
4.3.1	Stejnoseměrné (DC) parametry	20
4.3.2	Střídavé (AC) parametry	21
4.4	Příjem/odesílání dat z/do PC.....	23
4.5	Zápis/čtení dat do/z SRAM.....	25
4.6	Napájení	26
4.7	Schéma zapojení a DPS	26
5	Software	27
5.1	Ovládací program pro PC a jeho obsluha	27
5.1.1	Struktura programu.....	28
5.1.2	Formát Intel-HEX.....	30
5.2	Řídící program MCU (firmware simulátoru).....	32
5.3	Komunikační protokol	33
6	Testy a měření	34
7	Závěr	35

Seznam obrázků

Obr.1 Časový diagram pro čtení z EPROM.....	12
Obr.2 Vnitřní blokové schéma simulátoru	17
Obr.3 Časový diagram SRAM	22
Obr.4 Časový diagram čtení z FT245R	24
Obr.6 GUI ovládacího programu	27
Obr.7 Struktura tříd v programu	28
Obr.8 Obecný formát záznamu	30
Obr.9 Obecný formát zprávy komunikačního protokolu	33
Obr.10 Výsledek měření parametru T_{GLQW}	34

Seznam tabulek

Tab.1 Stejnoseměrné parametry EPROM.....	11
Tab.2 Střídavé parametry EPROM	11
Tab.3 Stejnoseměrné parametry budičů.....	20
Tab.4 Střídavé parametry budičů	21
Tab.5 Střídavé parametry SRAM.....	22
Tab.6 Časování čtení z FTDI	24
Tab.7 Časování zápisu do FT245R	25
Tab.8 Typy zpráv v komunikačním protokolu.....	33

Seznam zkratek

PC	Personal Computer, osobní počítač
USB	Universal Serial Bus, rozhraní PC pro připojení dalších komponent
DIL	Dual In Line, označení pouzdra elektronické součástky
EPROM	Elektricky programovatelná paměť určená jen pro čtení
SRAM	Statická paměť s náhodným přístupem, určená pro čtení i zápis
I/O	Vstup/výstup
MCU	Microcontroller unit, jednočipový mikropočítač
FIFO	First In First Out, označení paměti (posuvného registru)
ISP	In System Programming, programování v aplikaci
LDO	Low DropOut, označení nízkoubytkového stabilizátoru napětí
GUI	Graphical User Interface, grafické uživatelské rozhraní aplikace
IDE	Integrated Development Environment, vývojové prostředí umožňující návrh, ladění a kompilaci aplikace
DPS	Deska plošných spojů

1 Úvod

Cílem této práce je navrhnout emulátor paměti typu EPROM, konkrétně typu M27V201.

Emulátor EPROM je zařízení, které je v laboratořích k vidění stále méně. Důvodem je masivní nástup MCU s vnitřní pamětí, přechod na SMD, případně přítomnost nástrojů umožňujících ladění aplikace bez jejího nahrání do programové paměti.

Toto zařízení bude sloužit ve školních laboratořích při cvičeních, ve kterých je využíván vývojový kit Analog Devices ADSP-21065L EZ-LITE.

Vlastní text práce je členěn do několika kapitol. První se zabývá zjištěním požadovaných parametrů výsledného zařízení. Ve druhé je proveden rozbor možných řešení zadání. Třetí se zabývá již konkrétním vybraným návrhem a způsobem realizace jeho hardwarové části. Čtvrtá kapitola popisuje vytvořené softwarové vybavení. Poslední z kapitol jsou popsány měření a testy zhotoveného prototypu.

2 Požadavky na zařízení

2.1 Fyzické propojení

Osazená paměť v cílovém zařízení má pouzdro PDIP32 (podtyp DIL). Pro připojení simulátoru bude tedy využit konektor rozměrově odpovídající právě tomuto pouzdru. Jedná se o „samořeznou patici“ na plochý kabel. Pomocí tohoto kabelu bude simulátor komunikovat s cílovým zařízením.

S PC má být zařízení propojeno přes USB. Za tímto účelem vybavíme simulátor konektorem female USB-B. K propojení poté poslouží kabel, který je běžně k dostání v prodejnách s příslušenstvím k PC. Jde o tentýž USB kabel kterým se připojuje tiskárna.

2.2 Elektrické parametry

Simulovaná EPROM má několik verzí lišících se svojí rychlostí. Jelikož nebyl typ upřesněn, budeme tedy dále pracovat s nejméně příznivou situací a tedy nejrychlejším typem. Elektrické parametry lze rozdělit na stejnosměrné a střídavé, oboje je nutné dodržet.

2.2.1 Stejnosměrné (DC) parametry

Tyto parametry jsou uvedeny v následující tabulce. Z ní nás ovšem zajímají pouze V_{IL} , V_{IH} , V_{OL} a V_{OH} . Tabulka je převzata z datasheetu a tedy v angličtině.

Symbol	Parameter	Test Condition	Min	Max	Unit
I_{LI}	Input Leakage Current	$0V \leq V_{IN} \leq V_{CC}$		± 10	μA
I_{LO}	Output Leakage Current	$0V \leq V_{OUT} \leq V_{CC}$		± 10	μA
I_{CC}	Supply Current	$\bar{E} = V_{IL}, \bar{O} = V_{IL}, I_{OUT} = 0mA, f = 5MHz, V_{CC} \leq 3.6V$		15	mA
I_{CC1}	Supply Current (Standby) TTL	$\bar{E} = V_{IH}$		1	mA
I_{CC2}	Supply Current (Standby) CMOS	$E > V_{CC} - 0.2V, V_{CC} \leq 3.6V$		20	μA
I_{PP}	Program Current	$V_{PP} = V_{CC}$		10	μA
V_{IL}	Input Low Voltage		-0.3	0.8	V
$V_{IH}^{(2)}$	Input High Voltage		2	$V_{CC} + 1$	V
V_{OL}	Output Low Voltage	$I_{OL} = 2.1mA$		0.4	V
V_{OH}	Output High Voltage TTL	$I_{OH} = -400\mu A$	2.4		V
	Output High Voltage CMOS	$I_{OH} = -100\mu A$	$V_{CC} - 0.7V$		V

Tab.1 Stejnosměrné parametry EPROM

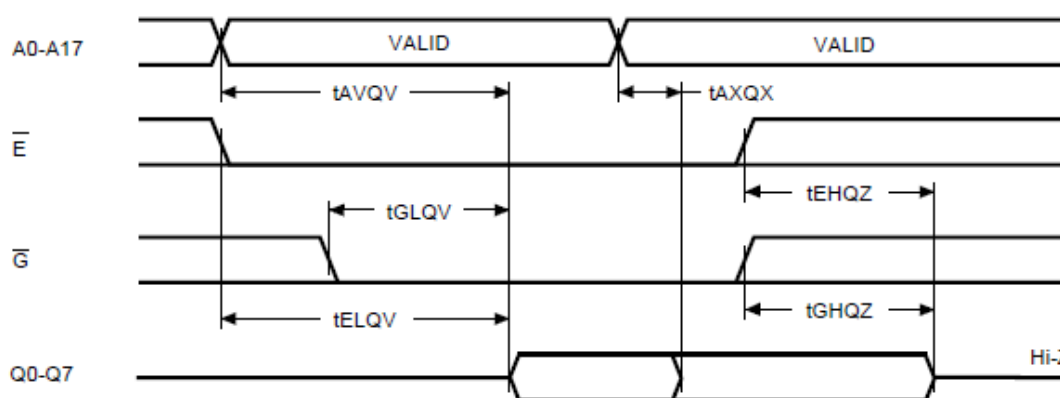
Dalším důležitým parametrem, který zde ovšem není uveden, je napájecí napětí této EPROM, které je v rozmezí 3,0 ~ 3,6V. Toto je nutné považovat za horní mez výstupního napětí, neboť neznáme možnosti cílového zařízení v tomto směru a jejich překročení by mohlo znamenat také jeho zničení.

2.2.2 Střídavé (AC) parametry

Výčet těchto parametrů je opět v tabulce níže, dále pak časový diagram pro jejich přiblížení. Zde je již nutné brát v úvahu všechny. Jde o zachycení stavů na vstupech paměti a její odpovídající reakce na výstupech. Má-li simulátor z hlediska těchto parametrů vyhovět, bude nutné, aby jeho reakční doby byly kratší nebo alespoň stejné jako zde uvedené.

Symbol	Alt	Parameter	Test Condition	M27V201				Unit
				-120		-150		
				Min	Max	Min	Max	
t_{AVQV}	t_{ACC}	Address Valid to Output Valid	$\bar{E} = V_{IL}, \bar{G} = V_{IL}$		120		150	ns
t_{ELQV}	t_{CE}	Chip Enable Low to Output Valid	$\bar{G} = V_{IL}$		120		150	ns
t_{GLQV}	t_{OE}	Output Enable Low to Output Valid	$\bar{E} = V_{IL}$		50		60	ns
$t_{EHQZ}^{(2)}$	t_{DF}	Chip Enable High to Output Hi-Z	$\bar{G} = V_{IL}$	0	40	0	50	ns
$t_{GHQZ}^{(2)}$	t_{DF}	Output Enable High to Output Hi-Z	$\bar{E} = V_{IL}$	0	40	0	50	ns
t_{AXQX}	t_{OH}	Address Transition to Output Transition	$\bar{E} = V_{IL}, \bar{G} = V_{IL}$	0		0		ns

Tab.2 Střídavé parametry EPROM



Obr.1 Časový diagram pro čtení z EPROM

3 Navrhované řešení

3.1 Rozhraní simulátor – cílové zařízení

Vytvořit toto rozhraní je možné několika způsoby. Jeden z návrhů bylo simulovat chování EPROM pomocí MCU. Toto řešení se ovšem ukázalo jako nevhodné. Požítý procesor by totiž musel vzhledem k požadavkům na odezvu být schopen číst vstupy a následně měnit výstupy rychlostí 10^7 změn za vteřinu. Takováto rychlost by vyžadovala procesor s rychlostí alespoň 100MHz a jednocyklovým jádrem. Dalším neskromným požadavkem by byla interní paměť velikosti 128kB.

Další možností je využití součástky pinově kompatibilní se stejnou funkcí týkající se čtení dat. Mezi použitelné obvody patří paralelní EEPROM, FLASH EPROM a SRAM.

Většina EEPROM dostupných v ČR nespĺňuje naše požadavky co do velikosti, rychlosti tak i napájecího napájení. Dalším nezanedbatelným omezením je jejich konečný počet přeprogramování do doby, než začnou selhávat paměťové buňky. Na druhou stranu by zde byla výhoda v zachování nahraných dat i po vypnutí zařízení. Tento požadavek ale není na vyvíjené zařízení kladen.

FLASH EPROM mají stejně jako předchozí EEPROM nedostatek v konečném počtu přeprogramování a nižší rychlosti zápisu oproti vítěznému návrhu. I zde by byla data zachována po vynutí zařízení.

Po zvážení výhod a nevýhod předchozích obvodů bylo rozhodnuto pro využití SRAM. Tento typ paměti je konstrukčně určen pro libovolné neomezené přepisování a čtení dat. Data jsou v ní ovšem uchována jen po dobu, kdy je na ni přivedeno ustálené napájecí napětí. Pro potřeby čtení není obtížné nalézt vhodný typ z hlediska řídicích signálů i napěťových úrovní, který bude odpovídat právě simulované M27V201.

3.2 Rozhraní simulátor – PC

K připojení zařízení k PC má sloužit rozhraní USB. Jeho implementace je možná více způsoby. Jeden z nich by byl vhodný z důvodu snížení počtu součástek, ceny a zvýšení spolehlivosti. Jde o naprogramování logiky USB do MCU který je nutno (viz. další podkapitola) stejně použít. Toto řešení by ovšem bylo časově velice náročné. Dalším negativem by byla nutnost napsat pro takto vytvořené USB zařízení ovladače pro Windows. Nicméně to je vysoko nad rámec této práce.

Dalším možným řešením je využití integrovaných obvodů sloužících jako převodníky z USB na jiné rozhraní. Vzhledem k předchozím zkušenostem bylo rozhodnuto pro využití obvodu společnosti Future Technology Devices International Ltd. (<http://www.ftdichip.com>) a to konkrétně FT245RL.

Převodník zajistí elektrické i softwarové požadavky na USB zařízení. Výrobce ke svým obvodům dodává ovladače pro Windows. Dodávané ovladače jsou dvojího druhu. Nativní, pomocí nichž lze s obvodem komunikovat přímo z klientské aplikace. druhou možností je využití VCP (Virtual Com Port) driverů. Tyto ovladače zajistí vytvoření virtuálního sériového portu, přes který se s obvodem komunikuje.

Obslužná aplikace pro zařízení bude naprogramována v jazyce C#. Vzhledem k jednoduchosti obsluhy sériového portu v tomto jazyce bylo využito právě VCP ovladačů.

Z pohledu obslužného MCU se obvod chová jako paralelní brána, skrz kterou lze přijímat data odeslaná z PC nebo je naopak do počítače posílat. Obou možností bude využito.

3.3 Propojení obou rozhraní

V kapitole 3.1 jsme dospěli k závěru, že na straně připojení k cílovému zařízení bude nejvhodnější z hlediska kompatibility využít SRAM. Ovšem než bude možno tuto paměť připojit na výstup, je nezbytné do ní data napřed nahrát.

K řešení znovu vede více cest. Jedna z možností je použití tzv. BitBang módu konvertoru rozhraní FT245RL. Díky tomuto režimu a za pomoci nativních ovladačů je možné manipulovat jednotlivými piny obvodu. Přidáním několika

dalších součástek jako jsou registry a různá hradla by šlo data do SRAM nahrát a případně i přečíst.

Takové řešení má ovšem několik nevýhod:

- 1) zvýšení počtu součástek doprovázené zvýšenou spotřebou a snížením spolehlivosti
- 2) složitější ovládací software díky závislosti na nativních ovladačích a BitBang módu
- 3) nižší rychlost zápisu díky zpožděním vznikajícím při komunikaci přes USB.

Druhou mnohem výhodnější volbou je použití MCU, který bude zprostředkovávat komunikaci mezi FTDI a SRAM. Díky mikrokontroléru se vyhneme přidávání dalších zbytečných integrovaných obvodů nutných v předchozím případě.

Takovéto zapojení řeší všechny nevýhody toho předchozího:

- 1) počet přidaných součástek se omezí pouze na MCU a budiče sběrnic potřebné v každém případě (viz. kapitola 4.1). Vzhledem k faktu, že procesor bude pracovat pouze v okamžiku zápisu dat, může být po celou zbývající dobu přepnut do režimu spánku čímž se spotřeba ještě sníží
- 2) přibude zde nutnost vytvořit řídicí program pro MCU. Tato je však bohatě vykompenzována jeho jednoduchostí v porovnání s použitím nativních ovladačů a BitBang módu
- 3) vyšší rychlost zápisu automatickému využití FIFO integrované v obvodech FT245. Tato paměť při vhodném návrhu komunikačního protokolu významně snižuje vliv zpoždění vznikajících při komunikaci mezi FT245 a PC.

4 Hardware

4.1 Návrh zapojení – blokové schéma

Z předchozího textu vyplývá, že v simulátoru využijeme statickou RAM, která se bude starat o dodávání dat cílovému zařízení dle jeho požadavků. Pro připojení k PC dobře poslouží obvod FT245 a o zápis dat do SRAM se bude starat mikroprocesor. Ovšem v tuto chvíli vzniká problém. SRAM disponuje pouze 1 adresovou sběrnicí, 1 datovou sběrnicí, několika řídicími signály. Obě sběrnice i většina řídicích signálů je využívána jak při zápisu, tak i při čtení. Nyní je důležité zajistit, aby na těchto sběrnicích a řídicích signálech nevznikaly kolize. Datová sběrnice bude z důvodu úspory pinů procesoru společná pro všechny aktivní integrované obvody.

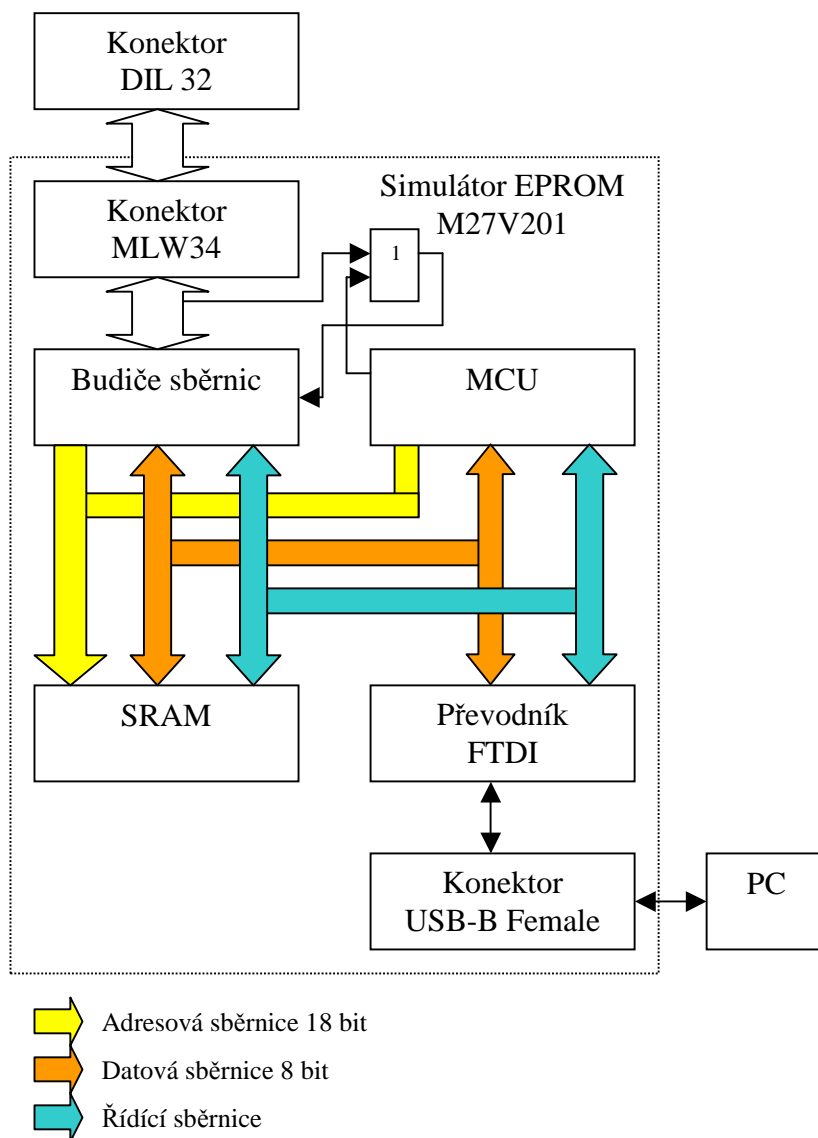
Příkladem budiž situace, v níž se cílové zařízení pokouší přečíst data z SRAM a zároveň se do ní pokusí procesor data zapsat. Cílové zařízení nastaví na SRAM adresu požadovaných dat, totéž provede i procesor. Je vysoce nepravděpodobné, že nastavované adresy oběma aktivními prvky budou za všech okolností stejné. Na bitových vodičích, kde se budou adresy lišit, vznikne zkrat. V lepším případě to oba aktivní prvky vydrží a na sběrnicí bude pouze nedefinovaná úroveň, v horším případě se některý nebo více aktivních obvodů zničí.

Jako řídicí procesor byl vzhledem k předchozím zkušenostem a vhodným vlastnostem vybrán jeden z 8-bitových mikrokontrolérů rodiny AVR firmy Atmel (<http://www.atmel.com>). Konkrétní typ bude upřesněn později. Tyto procesory mohou svoje vstupně/výstupní brány nastavit do stavu vysoké impedance, což znamená, že vhodným naprogramováním můžeme kolizím od tohoto prvku zabránit. Dalším obvodem je převodník FT245. Na společných sběrnicích se bude vyskytovat jeho datový port. Tento port se za určitých okolností chová jako výstupní nebo je ve stavu vysoké impedance. Jeho chování ovlivňují stavy řídicích signálů obvodu. Připojením těchto signálů k MCU, lze opět vhodným programem kolizím zabránit. O SRAM platí z hlediska vzniku kolizí totéž, co o FT245.

Jediným aktivním prvkem, jehož chování nelze ovlivnit, zůstává cílové zařízení. Způsob jak docílit potřebného zamezení možnosti ovlivňovat sběrnice je jeho dočasné odpojení. Bylo by ovšem nevhodné to provádět fyzickým vyndáváním

konektoru. Do cesty signálům vložíme obvod, který to zařídí. Jedná se o tzv. neinvertující budič sběrnice. Tento IO přenáší stavy všech svých vstupů na výstupy do doby, kdy je to povoleno jeho negovaným vstupem „gate“. Po připojení tohoto vstupu k MCU a jeho vhodným řízením nastane situace shodná s FT245 i SRAM a nic již nebrání v úplné kontrole přístupu ke sběrnicím.

Po vyřešení přístupu ke sběrnicím, lze konečně vytvořit blokové schéma, od něhož se bude další funkce jednotlivých obvodů včetně požadovaných elektrických parametrů simulátoru odvíjet.



Obr.2 Vnitřní blokové schéma simulátoru

4.2 Výběr součástek

Nyní, po sestavení blokového schématu, lze přistoupit k výběru součástek, jež zajistí funkčnost jednotlivých bloků. V tento okamžik je vybrán pouze jeden konkrétní obvod a to převodník FfT245RL. Zbývá tedy nalézt vhodný typ SRAM, budiče a procesoru.

4.2.1 SRAM

Tato paměť hraje hlavní roli pro splnění AC parametrů kladených na simulátor. Musí být tedy ve všech parametrech rychlejší než nahrazovaná EPROM a to alespoň o 15ns, které budou třeba jako rezerva pro budiče.

Po několika marných pokusech získat v ČR součástku, která by splňovala potřebné parametry, byly vzneseny požadavky u jednoho z prodejců a výběr byl ponechán na něm. Nakonec byla dodána SRAM **K6X4008T1F-B** v rychlostní verzi **70ns** výrobce **SAMSUNG Electronics CO., LTD.** Tato paměť má sice dvojnásobnou kapacitu, ale to není na překážku, neboť můžeme přebytečný adresový vstup připojit na stálo na jednu logickou úroveň a získáme potřebnou poloviční kapacitu. Co se týká rychlostních požadavků, ty byly splněny téměř přesně. Bohužel 1 z nich byl překročen o 5ns, což se později projevilo.

4.2.2 Budiče

Musejí být schopny odpojit všechny vodiče vedoucí z/do cílového zařízení využívané při čtení (zbylé nebudou použity). Jedná se o 18ti bitovou adresovou sběrnici, 8mi bitovou datovou sběrnici a 2 řídicí vodiče (/Enable a /Gate). Celkem je to tedy 28 linek.

Bohužel možnosti získání této součástky v Čechách jsou ještě horší než u předchozí paměti. Požadované napájecí napětí je 3,3V a doba průchodu signálu menší než 8,5ns. Ve výsledku to tedy znamenalo, že je nezbytné objednat budič v zahraničí. Volba padla na SN74AHC244Q v pouzdru SOIC společnosti Texas Instruments Incorporated (<http://www.ti.com/>) Pro řízení 28 datových linek budou potřeba 4 tyto obvody (každý je 8-bitový).

4.2.3 Řídící procesor

Na procesor nejsou kladeny žádné zvláštní požadavky ohledně periférií nebo rychlosti. Jediným omezením je dostatečný počet I/O linek, možnost programování v aplikaci pomocí rozhraní JTAG nebo ISP a napájecí napětí 3,3V. Procesor bude totiž ve verzi SMD a jeho opakovaná demontáž a montáž by byla přinejmenším problematická. Další výhodnou ovšem nikoli nezbytnou vlastností je možnost využití 1 externího přerušení.

Z blokového schématu je zřejmé, že procesor musí být schopen obsloužit zápis do SRAM, čtení dat z FTDI a ovládat budiče. Společná datová sběrnice má šířku 8 bitů, jelikož je společná, pro potřeby součtu potřebných pinů ji nebudu u jednotlivých obvodů uvažovat. Pro obsluhu SRAM je třeba minimálně 19 linek, ovšem doporučena možnost úplného řízení, což vyžaduje 21. Pro čtení dat z FTDI jsou nezbytné 2 linky, opět doporučena možnost úplného řízení, tedy 4 linky. Budiče vyžadují 1 linku. Dále by bylo vhodné mít nějak možnost informovat uživatele o stavu simulátoru alespoň pomocí 1 LED, tedy další 1 linka.

Minimální počet I/O linek procesoru je tedy 29, doporučeno je 34. Vzhledem k minimálnímu cenovému rozdílu mezi procesory pro minimální a doporučenou konfiguraci použijeme doporučenou. Konkrétní typ MCU, který nám vyhovuje je **ATmega8515L** firmy ATMEL (<http://www.atmel.com>).

Napájení procesoru může být v rozmezí 2,7~5,5V, programování v aplikaci je umožněno pomocí ISP, obsahuje externí přerušení a disponuje 35 vstupně/výstupními linkami. Díky použití MCU s doporučeným počtem I/O je možno do ovládacího programu pro Windows zabudovat i funkce ověření správnosti nahraných dat.

Za účelem naprogramování procesoru pomocí ISP bude navrhované zařízení obsahovat konektor MLW10. Vzhledem k přítomnosti ISP rozhraní procesoru na stejném portu jako bude umístěna datová sběrnice budou toto rozhraní a sběrnice propojeny ochrannými rezistory. Konektor MLW10 bude zapojen dle požadavků programátoru UniProg-USB firmy PK-Design (<http://www.pk-design.net>) pro připojení pomocí kabelu pro ISP.

4.3 Kontrola elektrických parametrů simulátoru

4.3.1 Stejnoseměrné (DC) parametry

Jelikož jsou budiče klasické logické obvody a nikoli spínače, jsou stejnosměrné parametry celého simulátoru dány pouze jejich vlastnostmi. Následuje výřez z katalogového listu výrobce, důležité jsou opět zvýrazněné hodnoty, kde nebylo možno vybrat shodné testovací podmínky, byly vybrány přísnější.

recommended operating conditions (see Note 3)

		MIN	MAX	UNIT
V _{CC}	Supply voltage	2	5.5	V
V _{IH}	High-level input voltage	V _{CC} = 2 V	1.5	V
		V _{CC} = 3 V	2.1	
		V _{CC} = 5.5 V	3.85	
V _{IL}	Low-level input voltage	V _{CC} = 2 V	0.5	V
		V _{CC} = 3 V	0.9	
		V _{CC} = 5.5 V	1.65	

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	V _{CC}	T _A = 25°C			MIN	MAX	UNIT
			MIN	TYP	MAX			
V _{OH}	I _{OH} = -50 μA	2 V	1.9	2	1.9	V		
		3 V	2.9	3	2.9			
		4.5 V	4.4	4.5	4.4			
	I _{OH} = -4 mA	3 V	2.58	2.48				
	I _{OH} = -8 mA	4.5 V	3.94		3.8			
V _{OL}	I _{OL} = 50 μA	2 V			0.1	0.1	V	
		3 V			0.1	0.1		
		4.5 V			0.1	0.1		
	I _{OL} = 4 mA	3 V		0.36	0.5			
	I _{OL} = 8 mA	4.5 V		0.36	0.5			

Tab.3 Stejnoseměrné parametry budičů

Hned první parametr V_{IH} – tedy minimální vstupní napětí pro vysokou úroveň signálu mají použité budiče o 0,1V přísnější než nahrazovaná EPROM. Je tedy nutné zjistit, zda to nemůže ohrozit správnou funkčnost simulátoru po připojení do cílového zařízení. Tím je vývojový kit ADSP-21065L EZ-LITE. Kit je osazen procesorem SHARC ADSP-21065L. Jeho minimální výstupní napětí pro logickou „1“ je 2,4V. To spolehlivě překročí napětí požadované budiči a neovlivní funkčnost kitu po připojení simulátoru. Z tohoto hlediska je tedy možné tento parametr považovat za **VYHOVUJÍCÍ**.

V_{IL} – maximální napětí pro log. „0“ na vstupu. Budiče tolerují ještě o 0,1V více než EPROM, parametr **VYHOVUJE**.

V_{OH} – minimální napětí na výstupu při log. „1“. Budiče garantují napětí o 0,08V vyšší než EPROM, parametr **VYHOVUJE**.

V_{OL} – maximální napětí na výstupu pro log. „0“. Při prvním pohledu by tato vlastnost nevyhověla, nicméně při bližším pohledu zjistíme následující: udávaná hodnota SRAM je 0,4V ovšem za podmínek, že zatěžovací proud nepřekročí 2,1mA. Hodnota u budičů je oproti tomu stanovena při téměř dvojnásobném zatěžovacím proudu. Obvod SN74AHC244 je vyroben technologií CMOS. Výstupní napětí pod zátěží je tedy určeno výstupním FET tranzistorem. V logických obvodech pracují tyto prvky ve spínacím režimu a pracovní bod se nachází v jejich odporové oblasti, kde se chovají jako rezistor. Dle datasheetu se na výstupu může vyskytnout napětí 0,5V při zátěži 4mA. Použitím Ohmova zákona lze zjistit ekvivalentní odpor výstupního tranzistoru.

$$R = \frac{U}{I} = \frac{0,5V}{0,004A} = 125\Omega$$

Opětnou aplikací Ohmova zákona a toho výsledku na testovací podmínky nahrazované EPROM vyjde následující:

$$U = R * I = 125\Omega * 0,0021A = 0,26V$$

Tedy i tento parametr **VYHOVUJE**

4.3.2 Střídavé (AC) parametry

Oproti stejnosměrným parametrům je zde situace o poznání složitější. V různých funkčních situacích je totiž nezbytné zjistit způsob reakce simulátoru, tedy kolik obvodů a kolikrát se účastní dané operace. Následují časové parametry budičů, SRAM a časový diagram SRAM pro objasnění významu jednotlivých zkratk.

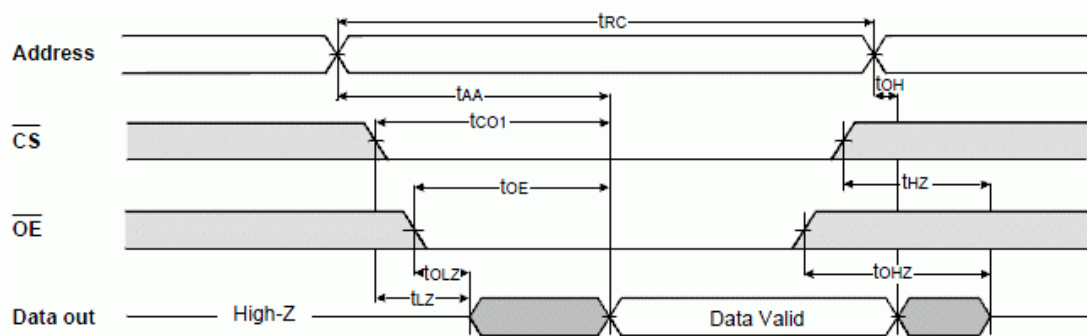
switching characteristics over recommended operating free-air temperature range, $V_{CC} = 3.3V \pm 0.3V$ (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	LOAD CAPACITANCE	$T_A = 25^\circ C$			MIN	MAX	UNIT
				MIN	TYP	MAX			
t_{PLH}	A	Y	$C_L = 15\text{ pF}$	5.8	8.4	1	10	ns	
t_{PHL}				5.8	8.4	1	10		
t_{PZH}	\overline{OE}	Y	$C_L = 15\text{ pF}$	6.6	10.6	1	12.5	ns	
t_{PZL}				6.6	10.6	1	12.5		
t_{PHZ}	\overline{OE}	Y	$C_L = 15\text{ pF}$	5	9.7	1	11	ns	
t_{PLZ}				5	9.7	1	11		

Tab.4 Střídavé parametry budičů

Parameter List		Symbol	Speed Bins						Units
			55ns ¹⁾		70ns		85ns		
			Min	Max	Min	Max	Min	Max	
Read	Read cycle time	t _{RC}	55	-	70	-	85	-	ns
	Address access time	t _{AA}	-	55	-	70	-	85	ns
	Chip select to output	t _{CO}	-	55	-	70	-	85	ns
	Output enable to valid output	t _{OE}	-	25	-	35	-	40	ns
	Chip select to low-Z output	t _{LZ}	10	-	10	-	10	-	ns
	Output enable to low-Z output	t _{OLZ}	5	-	5	-	5	-	ns
	Chip disable to high-Z output	t _{HZ}	0	20	0	25	0	25	ns
	Output disable to high-Z output	t _{OHZ}	0	20	0	25	0	25	ns
	Output hold from address change	t _{OH}	10	-	10	-	10	-	ns

Tab.5 Střídavé parametry SRAM



Obr.3 Časový diagram SRAM

Porovnání je nutné provést u odpovídajících dob zpoždění. To lze pouze při znalosti následujícího: vstup /E u EPROM je funkčně shodný se vstupem /CS u SRAM, totéž platí pro /G u EPROM a /OE u SRAM.

$T_{AVQV} = \max 120\text{ns}$. Je to doba, která uplyne od změny adresy do ustálení platných dat na výstupu. V simulátoru to znamená: průchod adresy budičem, ustálení výstupu SRAM a průchod dat zpět budičem. Celková doba je tedy $2 \cdot T_{PLH} + t_{AA} = 2 \cdot 10\text{ns} + 70\text{ns} = 90\text{ns}$. Parametr **VYHOVUJE**.

$T_{ELQV} = \max 120\text{ns}$. Doba, která uplyne od aktivace vstupu /E do ustálení platných dat na výstupu. V simulátoru to znamená: průchod signálu /E budičem, ustálení výstupu SRAM a průchod dat zpět budičem. Celková doba je tedy $2 \cdot T_{PLH} + t_{CO} = 2 \cdot 10\text{ns} + 70\text{ns} = 90\text{ns}$. Parametr **VYHOVUJE**.

$T_{GLQV} = \max 50\text{ns}$. Doba, která uplyne od aktivace vstupu /G do ustálení platných dat na výstupu. V simulátoru to znamená: průchod signálu /G budičem, ustálení výstupu SRAM a průchod dat zpět budičem. Celková doba je tedy $2 \cdot T_{PLH} + t_{OE} = 2 \cdot 10\text{ns} + 35\text{ns} = 55\text{ns}$. Zde se projevil onen nesplněný požadavek na SRAM. Parametr nevyhovuje, ale pouze na první pohled.

Rozebereme-li si situaci podrobněji, zjistíme, že t_{OE} platí u SRAM za podmínky, že je zatěžovací kapacita výstupu 30pF. Ovšem kapacita vstupů budiče je maximálně 10pF, tj. pouze 33%. Stejně tak vstupní kapacita SRAM je jen 8pF, přičemž zpoždění budiče je měřeno pro zátěž 15pF, takže i zde je zátěž jen 53%. Oba obvody tedy budou ve skutečnosti rychlejší. **Ověření splnění tohoto parametru bude provedeno měřením na logickém analyzátoru.**

T_{EHQZ} a $T_{GHQZ} = \max 40ns$. Doba, která uplyne od deaktivace vstupu /E do přepnutí výstupu do stavu vysoké impedance. V simulátoru je budič, který má na starosti připojení datové sběrnice k cílovému zařízení, řízen kromě MCU i vstupy /E a /G. Doba je tedy závislá pouze na budiči a diodách, které tvoří logický součet. Jsou typu Schottky, konkrétně BAT85 mající dobu zotavení $<5ns$. Celková doba činí $t_{PHZ} + t_{tr} = 11ns + 5ns = 16ns$. Parametr **VYHOVUJE**.

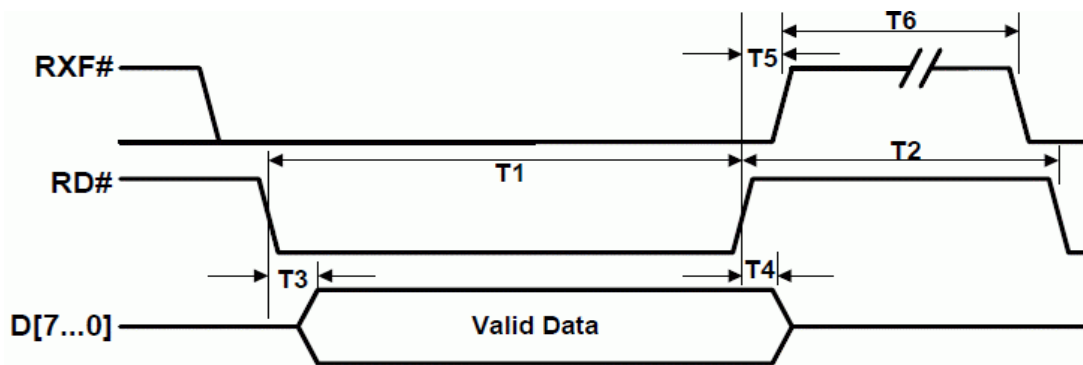
Parametr t_{AXQX} nemá v katalogovém listu přesné vyobrazení, nemá definovanou svoji horní mez, která by mohla být důležitá, nelze ho proto posoudit.

4.4 Příjem/odesílání dat z/do PC

Z kapitoly 3.2 plyne, že pro komunikaci přes USB je využit obvod FT245R. V tomto převodníku je integrována veškerá logika nezbytná pro obsluhu tohoto datového spojení. Pro navazující zařízení je tedy toto rozhraní neviditelné, místo něj ke komunikaci slouží paralelní datová 8-bitová brána a čtveřice řídicích vodičů (/RD, WR, /RXF, /TXE).

Logická „0“ na výstupu /RXF znamená, že byla po USB přijata data a jsou k dispozici ve vnitřní přijímací FIFO. Když nejsou žádná přijatá data, pin je v „1“.

Čtení dat zajišťuje vstupní pin /RD. V době, kdy neprobíhá čtení dat, je nutné udržovat tento vstup v „1“. Po výskytu dat v přijímací FIFO lze tato přečíst přepnutím do „0“. V ten okamžik se změní stav paralelní datové brány z vysoké impedance na výstup, na kterém se vyskytne nejstarší dosud nepřečtený bajt. Po přepnutí /RD zpět do „1“ je datový port vypnut (přepnut do stavu vysoké impedance) a pro čtení je připraven další byte, pokud nějaký ve FIFO je.

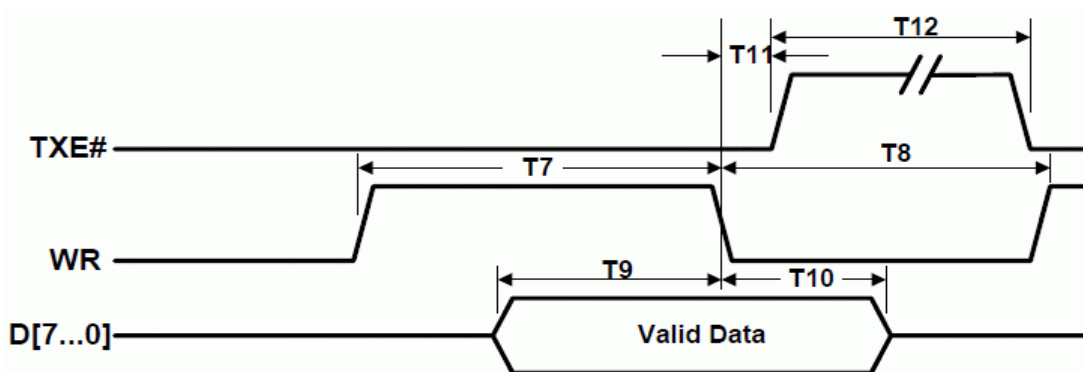


Obr.4 Časový diagram čtení z FT245R

Time	Description	Minimum	Maximum	Unit
T1	RD# Active Pulse Width	50	-	ns
T2	RD# to RD# Pre-Charge Time	50 + T6	-	ns
T3	RD# Active to Valid Data*	20	50	ns
T4	Valid Data Hold Time from RD# Inactive*	0	-	ns
T5	RD# Inactive to RXF#	0	25	ns
T6	RXF# Inactive After RD Cycle	80	-	ns

Tab.6 Časování čtení z FTDI

Zápis dat vyžívá dvojice pinů /TXE a WR. Je-li FT245 schopen přijmout data do své vysílací FIFO, /TXE je v log. „0“. Samotný zápis je pak proveden přivedením vstupu WR do „1“ a jeho opětovným vrácením do „0“.



Obr.5 Časový diagram zápisu dat FT245R 1

Time	Description	Minimum	Maximum	Unit
T7	WR Active Pulse Width	50	-	ns
T8	WR to WR Pre-Charge Time	50	-	ns
T9	Valid data setup to WR falling edge*	20	-	ns
T10	Valid Data Hold Time from WR Inactive*	0	-	ns
T11	WR Inactive to TXE#	5	25	ns
T12	TXE# Inactive After WR Cycle	80	-	ns

Tab.7 Časování zápisu do FT245R

Ovladače pro Windows jsou na přiloženém CD v souboru „CDM 2.04.06 WHQL Certified.zip“

4.5 Zápis/čtení dat do/z SRAM

Rozhraní paměti je tvořeno 18ti bitovou adresovou sběrnicí (1 bit ma trvale přiřazenu log. hodnotu viz. kapitola 4.2.1), 8mi bitovou datovou sběrnicí a řídicími signály /WE, /CS a /OE. Pro možnost připojení na sdílenou datovou sběrnicí se nelze obejít bez znalosti podmínek, za kterých je datový výstup ve stavu vysoké impedance. Toto nastává vždy když /CS = „1“ nebo /OE = „1“.

Vstupem /CS (Chip Select) je určen provozní režim čipu. Je-li neaktivní (log. „1“), paměť je v úsporném (Standby) módu se sníženou spotřebou. V tomto stavu jsou všechny funkce neaktivní. Pro operace jako je čtení/zápis dat je nezbytné, aby byl /CS aktivován (log. „0“).

Čtení dat lze provádět dvěma způsoby. Prvním je řízení vstupů /CS a /OE (Output Enable). Za okamžik poté, co jsou oba nastaveny do „0“ se datová sběrnice stane výstupní a objeví se na ní data ležící v paměťové buňce dané adresou na adresové sběrnicí. Onen okamžik je dán parametry paměti (viz. obr.3 a tab.5). U druhého způsobu jsou vstupy /CS a /OE neustále v log. „0“ a mění se pouze adresa. Nutno podotknout, že použitím tohoto postupu zůstává datový port neustále výstupním a na sdílenou datovou sběrnicí nesmí ostatní obvody zapisovat.

Zápis probíhá aktivací vstupu /CS a /WE (Write Enable). Informace na datové sběrnicí je uložena do paměťové buňky dané adresou.

4.6 Napájení

Simulátor obsahuje aktivní prvky, vyžaduje tedy pro svoji práci zdroj elektrické energie. Napájecí napětí simulované EPROM je v rozmezí 3~3.6V, simulátor bude tedy napájen 3.3V.

Toto napětí lze obstarat z externího zdroje, nebo využít možnosti zásobování energií z USB portu PC. USB v PC je schopno dodat 0,5A (0,1A u notebooku), jmenovité napětí je 5V.

O tom, zda bude reálné využití napájení portem USB rozhodují proudové požadavky zařízení vyhovuje-li napájecí napětí. Celkový proud zařízení je dán součtem napájecích proudů použitých obvodů. MCU pracuje na frekvenci interního RC oscilátoru 8MHz při napětí 3,3V Dle datasheetu to znamená proudový odběr do 8mA. FT245R: <15mA, SRAM: <25mA. Odběr budičů (SN74AHC244Q) závisí především na jeho zatěžovacím proudu. Samotný obvod má zanedbatelnou spotřebu (<1mA), ta je tedy dána zátěží způsobenou připojeným zařízením. Tu bohužel nelze předem určit. Pro potřeby určení proudového odběru tedy využijeme maximální stálý přípustný proud napájecími piny budičů, ten činí 50mA.

Celkem je to tedy:

$$I_{ccmax} = 8mA + 15mA + 25mA + 4 \times 50mA = 248mA$$

Tento proud jsou schopny dodat porty stolních PC, ovšem nikoli všech notebooků. Simulátor tedy bude napájen z USB, bude ovšem obsahovat konektor pro připojení externího zdroje. Pro stabilizaci napájecího napětí byl vybrán LDO stabilizátor „**REG102-3.3**“. Jeho maximální vstupní napětí je 10V a odebíraný proud 250mA. Ve stabilizátoru je integrována proudová a tepelná ochrana.

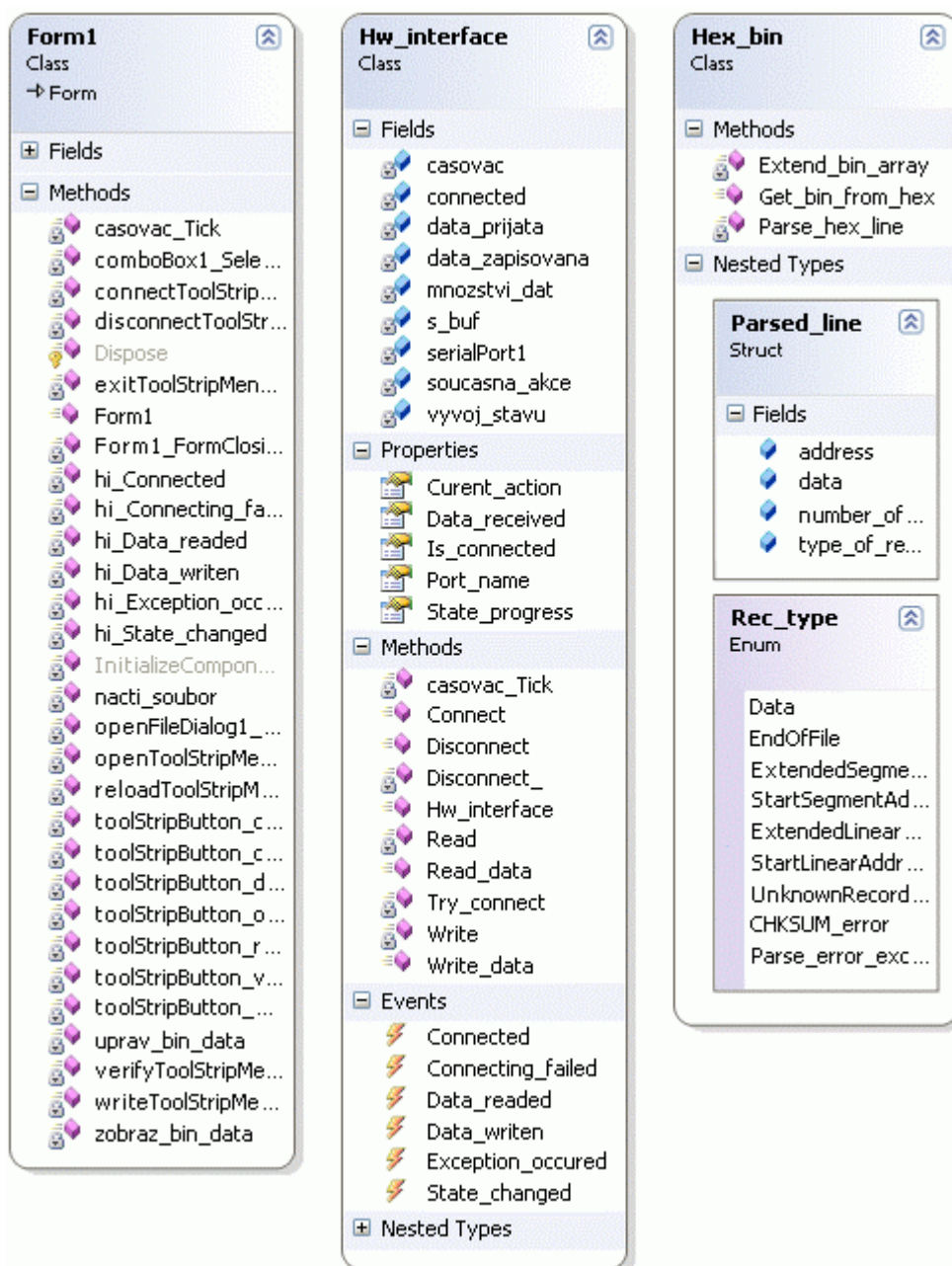
4.7 Schéma zapojení a DPS

Návrh schématu zapojení simulátoru včetně návrhu DPS bylo provedeno v návrhovém systému EAGLE 5.3 společnosti CadSoft (<http://www.cadsoft.de>).

Soubory projektu pro EAGLE a data potřebná pro výrobu DPS se nalézají na příloženém CD v souboru „*Simulator_EAGLE_prototyp.zip*“.

5.1.1 Struktura programu

Na následujícím obrázku je znázorněna struktura vytvořených tříd a doplněné odvozené třídy.



Obr.7 Struktura tříd v programu

Dle zadání musí být program schopen nahrát do simulátoru data ve vstupním formátu Intel-HEX nebo binární data. Za tímto účelem byly vytvořeny třídy „Hw_interface“ a „Hex_bin“. První jmenovaná slouží pro komunikaci se simulátorem a samotné nahrání dat do něho. Data jsou přenášena vždy binárně. Pokud jsou binární i data vstupní, není nutná žádná konverze. Oproti tomu při použití vstupního formátu Intel-HEX je data nutné před odesláním do simulátoru převést

na binární. Tato činnost je dána na starost právě třídě `Hex_bin`. Spolupráci těchto částí a zároveň komunikaci s uživatelem má na starosti odvozená třída „`Form1`“. Generována je automaticky vývojovým prostředím, vždy je ji ovšem nutno doplnit o požadovanou funkčnost. Celkem program obsahuje 6 tříd. Tři z nich jsou jmenované výše, dalšími pak jsou „`Program`“, „`Resources`“ a „`Settings`“. Tyto další třídy jsou opět generovány automaticky a často není nutné (stejně jako v tomto případě) do nich zasahovat. Zde se zmíním pouze o dvou z nich. Třída „`Program`“ je vstupním bodem do aplikace a spouští se odsud její grafické rozhraní. V „`Resources`“ jsou uloženy dodatečné zdroje využívané při tvorbě aplikace jako jsou například obrázky.

Hex_bin je statickou třídou určenou, jak je uvedeno výše, k převodu dat z formátu Intel-HEX na binární data. Obsahuje jedinou veřejnou metodu `Get_bin_from_hex(string hex_str, out byte[] bin)`. Prvním parametrem je řetězec s tady v HEX formátu načtený z otevřeného souboru. Druhým parametrem je pole s převedenými daty do binární podoby. V případě, že se při převodu vyskytne chyba, vrácené pole má hodnotu `null` a metoda skončí výjimkou popisující vzniklou chybu. Neveřejná metoda `Parsed_line Parse_hex_line(string line)` je využívána metodou `Get_bin_from_hex` pro dekódování jednoho záznamu (viz. kapitola 5.1.2) ze vstupních dat. Při vytváření pole s výstupními daty metodou `Get_bin_from_hex` ovšem není předem známo výsledné množství dat. To je určeno až po dokončení konverze. Na jejím začátku je tedy vytvořeno pole určité velikosti a teprve ukáže-li se jeho velikost v průběhu konverze jako nedostatečná, využije se poslední metody třídy `Hex_bin`, tedy `Extend_bin_array(ref byte[] bin)`. Jejím jediným parametrem je pole s daty, které má být zvětšeno. Každým zavoláním metody je velikost pole zdvojnásobena.

Ve třídě **Hw_interface** je zapouzdřena veškerá funkčnost související s komunikací se simulátorem, obsahuje logiku komunikačního protokolu. Jde o navázání/ukončení komunikace se simulátorem o které se starají „`Connect()`“ a „`Disconnect()`“, dále pak samotné přenosy dat do simulátoru (`Write_data`) a z něj zpět (`Read_data`). Žádná z metod neinformuje nadřazený objekt o průběhu své činnosti, tyto informace jsou předávány pomocí událostí a vlastností, které třída obsahuje. Instance této třídy je vytvořena ve třídě **Form1** (obsluhující GUI) a právě ta se stará o předání informací o průběhu akcí uživateli prostřednictvím stavového řádku programu. V této třídě je na první pohled poměrně mnoho metod, většina

z nich jsou ovšem pouze jednoduché obsluhy událostí komponent na GUI. Za zmínku zde stojí jen několik z nich. V první řadě jde o metody začínající řetězcem „hi_“. Jsou to události třídy Hw_interface, díky nimž lze předávat informace o průběhu jednotlivých akcí uživateli. Bez nich by jinak bylo možné předat uživateli zprávu o výsledku akce až po jejím dokončení. Během průběhu akce by aplikace vypadala zablokovaná, což bylo nežádoucí obzvláště před optimalizací komunikačního protokolu, kdy nahrání i malého množství dat trvalo relativně dlouhou dobu (několik vteřin). Další důležitou metodou je nacti_soubor(). Jejím úkolem je v závislosti na vstupních datech zařídit případnou konverzi, přizpůsobit data pro odeslání a navrhnout minimální množství dat potřebné pro přenos do simulátoru. Uživatel si sice může zvolit hodnotu větší, ovšem kromě prodloužení doby zápisu a verifikace to jiné změny v chování aplikace a simulátoru nepřinese. Zobrazení data určených pro přenos do simulátoru je provedeno zavoláním metody zobraz_bin_data().

Zdrojové kódy projektu pro Microsoft Visual Studio 2008 jsou na přiloženém CD v souboru „Simulator_PC_C#.zip“.

5.1.2 Formát Intel-HEX

Jde o textový (ASCII) formát pro reprezentaci binárních dat. Je akceptován jako vstupní formát mnoha hardwarových emulátorů a programátorů. Díky textovému zobrazení je pak možné ukládat binární data i na nebinární média a zobrazit na monitoru či tisknout.

Struktura HEX objektového souboru je tvořena záznamy. Každý záznam má strukturu zobrazenou na následujícím obrázku.

RECORD MARK ' : '	RECLLEN	LOAD OFFSET	RECTYP	INFO or DATA	CHKSUM
1-byte	1-byte	2-bytes	1-byte	n-bytes	1-byte

Obr.8 Obecný formát záznamu

- Každý záznam začíná částí „RECORD MARK“ obsahující znak s ASCII kódem 0x3AH tedy středník „;“.
- Následující část „RECLLEN“ udávající počet bytů informace nebo dat následujících za polem „RECTYP“. Maximální hodnota je 255 (0xFFH), číslo je zapsáno v 16kové soustavě.

- Každý záznam obsahuje „LOAD OFFSET“, je jím určena 16ti bitová adresa od které mají být uložena následující data. Toto pole má význam jen v záznamu typu „Data Record“, v ostatních záznamech, ve kterých není pole využito, by mělo obsahovat adresu „0000“.
- „RECTYP“ je polem určujícím typ záznamu. Pole je použito pro interpretaci zbývajících dat v záznamu. Kódy pro jednotlivé typy záznamů jsou následující:

„00“	Data Record
„01“	End of File Record
„02“	Extended Segment Address Record
„03“	Start Segment Address Record
„04“	Extended Linear Address Record
„05“	Start Linear Address Record

- V každém záznamu se nalézá pole INFO/DATA proměnné velikosti. Je tvořeno jedním nebo více byty kódovanými jako páry šestnáctkových číslic. Význam je závislý na poli RECTYP
- Záznam končí polem CHKSUM, které obsahuje kontrolní součet záznamu. Ten je dvojkovým doplňkem čísla, které vznikne součtem jednotlivých ASCII šestnáctkových párů převedených do binární formy. Součet je počítán od pole RECLLEN včetně do posledního bytu pole INFO/DATA včetně. Součet všech binárních hodnot těchto bytů s binární hodnotou pole CHKSUM je roven nule.

Bližší informace o jednotlivých typech záznamů lze najít v [3].

Program podporuje typy záznamů běžně používané v programátorech. Konkrétně jsou to „Data Record“, „End of File Record“, „Extended Segment Address Record“ a „Extended Linear Address Record“.

5.2 Řídící program MCU (firmware simulátoru)

Pro vytvoření programu řídicího procesoru simulátoru bylo využito IDE AVR Studio 4 společnosti ATMEL ve spolupráci s překladačem WinAVR jazyka C. Vývojové prostředí pro procesory AVR je dostupné zdarma z webu tvůrce http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725, WinAVR pak na adrese <http://sourceforge.net/projects/winavr/files/>.

Program je rozdělen do tří modulů. Prvním je soubor „main.c“, z něhož je program po resetu spuštěn. Jedním z jeho úkolů je inicializace všech použitých periférií. Během ní se provádí například nastavení pinů řídicí sběrnice pro zamezení kolizím na dalších sběrnících, aktivace Brown-out detektoru, povolení přerušení, nastavení časovačů či vymazání SRAM. Jeho hlavní náplní však je implementace komunikačního protokolu a koordinace datových přenosů při komunikaci s PC a SRAM.

Druhým modulem je soubor „sram.c“. Obsahuje funkce `sr_write()`, `sr_read()` a `sr_clear()` určené pro práci s pamětí SRAM v níž jsou uložena data simulované EPROM. Společným rysem všech tří funkcí je nutnost vykonat požadovanou akci bez výskytu kolize na kterékoli sběrnici. Všechny funkce tedy musí na začátku vypnout budiče sběrnic a zajistit si tak k nim výhradní přístup. Po dokončení akce je nutno budiče opět zapnout aby bylo umožněno cílovému zařízení využívat požadovaných funkcí simulátoru. Funkce `sr_write()` slouží pro zápis dat do SRAM, `sr_read()` k jejich čtení a pomocí `sr_clear()` je paměť mazána (přepsána hodnotou 0xFFH).

Úkolem modulu „usb.c“ je zajištění příjmu a odesílání dat z/do PC. Odesílání dat je prováděno pomocí `usb_odesli_data()`. Jejich příjem je zajištěn v obsluze externího přerušení `ISR(INT0_vect)`, které je vyvoláno vždy, když jsou obvodem FTDI přijata data a to nezávisle na případně aktuálně prováděné operaci. Z tohoto důvodu při práci s SRAM je nutné v main zajistit zamezení vzniku přerušení od FT245R a tím přepsání právě využívaných dat daty novými ještě před jejich zpracováním. Jediným způsobem, jak lze z obsluhy přerušení spolupracovat se zbytkem kódu jsou totiž globální proměnné a právě globální pole s daty je sdílené všemi funkcemi s těmito daty pracujícími.

Zdrojové kódy projektu pro AVR Studio 4 jsou na přiloženém CD v souboru „Simulator_AVR_C.zip“.

5.3 Komunikační protokol

Komunikační protokol mezi simulátorem a PC je tvořen zprávami v následujícím formátu:

Hlavička: 1B	Data: 0 – 16B	Kontrolní byte: 0 – 1B
--------------	---------------	------------------------

Obr.9 Obecný formát zprávy komunikačního protokolu

Hlavička: určuje přítomnost a význam následujících polí. Číslem dekódovaným z horních 4 bitů je dána požadovaná akce dle následující tabulky:

1	ECHO: dotaz z PC, zda je simulátor připraven, využíváno při autodetekci. Dolní 4 bity hlavičky jsou 0. Zpráva kromě hlavičky neobsahuje další byty.
2	OK: odpověď na ECHO, simulátor je připraven. Dolní 4 bity hlavičky jsou 0. Zpráva kromě hlavičky neobsahuje další byty.
3	DATA AMOUNT: informace o množství dat v nadcházejícím přenosu. To je dáno součinem mocniny čísla 2^X a 2^{13} , kde X vznikne dekódováním dolních 4 bitů hlavičky. Příklad: hlavička je 0x32. Množství dat je tedy $2^2 * 2^{13} = 32768B = 32kB$. Zpráva kromě hlavičky neobsahuje další byty.
4	SRAM WRITE: zpráva obsahuje data určená k zápisu do SRAM. Dolní 4 bity hlavičky jsou 0. Pole „Data“ obsahuje 16B dat, pole „Kontrolní součet“ je XOR předešlých bytů zprávy včetně hlavičky.
5	SRAM READ: Směr PC -> simulátor: jsou požadována data v množství daném „DATA AMOUNT“. Dolní 4 bity hlavičky jsou 0. Zpráva kromě hlavičky neobsahuje další byty. Směr simulátor -> zpráva má shodný formát jako „SRAM WRITE“ ovšem s hlavičkou „SRAM READ“
6	SRAM RESET: simulátor provede reset SRAM (přepsání všech buněk hodnotou 0xFFH). Dolní 4 bity hlavičky jsou 0. Zpráva kromě hlavičky neobsahuje další byty.
7	ERROR: v případě reakce na zprávu „SRAM READ“ nebo „SRAM WRITE“ je přenos zastaven, v případě reakce na ostatní zprávy je poslední zpráva opakována.

Tab.8 Typy zpráv v komunikačním protokolu

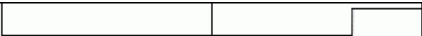
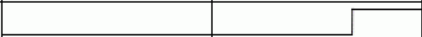
Před vysláním 1. zprávy „SRAM WRITE“ nebo „SRAM READ“ během jednoho přenosu je nutné vyslat „DATA AMOUNT“. V prvním případě má význam resetu adresy v mcu, ve druhém pak určuje množství očekávaných dat.

Původní návrh komunikačního protokolu vyžadoval okamžité potvrzování zpráv. Díky zpožděním při přenosu přes USB však byla komunikace pomalá a tak byla přizpůsobena pro využití bufferu ftdi a omezení vlivu latencí usb.

6 Testy a měření

Ještě před samotným měření byla zjištěn rozdíl v předpokládaném a skutečném zapojení konektoru DIP32. Pin č.1 na konektoru neodpovídal vodiči č.1 na kabelu. Oprava u prototypu spočívala v záměně párů vodičů kabelu před zajištěním v konektoru.

Předmětem měření se stal pouze parametr T_{GLQV} jehož maximální hodnota nesmí překročit 50ns (viz. kapitola 4.3.2).

Net[0]	X		
Net[1]	X		
Net[2]	X		
Net[3]	X		
Net[4]	X		
Net[5]	X		
Net[6]	X		
Net[7]	1		
OF	X		Some signals are NOT connected through probe
Clock	X		Some signals are NOT connected through probe
Tx	X		Some signals are NOT connected through probe
			30ns

Obr.10 Výsledek měření parametru T_{GLQV}

Z výstupu logického analyzátoru plyne, že měřený parametr se svou hodnotou nepřekračující 30ns **vyhovuje**. Všechny diskutované parametry simulátoru by tedy měly být vzhledem k nahrazované paměti M27V201 vyhovující a nic nebrání odzkoušení spolupráce simulátoru a kitu.

Ještě před testem spolupráce kitu a simulátoru je třeba podotknout, že po nahrání nových dat do simulátoru je nezbytné kit resetovat.

Způsob propojení simulátoru s vývojovým kitem ADSP-21065L EZ-LITE je vyobrazen v příloze B. Za účelem otestování zařízení byl vytvořen krátký program, uložen byl ve formátu Intel-HEX. Následovalo jeho načtení ovládacího programu v PC a dále nahrání do simulátoru. Po resetu vývojového kitu byl vykonán vytvořený program. **Test byl úspěšný.** Později při opětovné kontrole parametrů byl zjištěna chyba ve způsobu řízení jednoho z budičů, která zapříčiňuje nemožnost využívat další obvody nacházející se v kitu na stejné datové sběrnici. Oprava spočívala v přidání 3 diod typu schottky, jednoho rezistoru a drobné úpravě DPS.

7 Závěr

Cílem práce byl návrh zařízení simulujícího činnost paměti EPROM M27V201 z hlediska mechanických i elektrických parametrů.

Zpočátku byl proveden ideový návrh simulátoru, ze kterého vzešly 4 možná řešení. Na konec bylo zvolena možnost obsahující řídicí procesor, paměť SRAM, budiče sběrnic a převodník USB na paralelní sběrnici FT245R.

Pro simulátor byla navržena 2stranná DPS. Po jejím osazení se ukázalo, že došlo k chybnému propojení konektoru DIP32. Toto bylo vyřešeno úpravou propojovacího kabelu prototypu. Následoval test spolupráce simulátoru s cílovým zařízením, který proběhl úspěšně.

Později při opětovné kontrole funkčnosti simulátoru byla odhalena ještě jedna chyba spočívající v nesprávném řízení jednoho z budičů. Tato chyba má za následek omezenou funkčnost simulátoru ve smyslu nemožnosti využití jiných obvodů nacházejících se ve vývojovém kitu na stejné sběrnici. K opravě je nutné dodat 4 součástky a modifikovat DPS. Tuto úpravu je možné provést i na prototypu.

Pro zlepšení dynamických parametrů simulátoru navrhuji použití budičů SN74LVC2244A na místo současných SN74AHC244 nebo paměti SRAM s rychlostním označením 55ns

Pro obsluhu zařízení z PC jsem napsal SW v jazyce C#. Kromě všech požadovaných funkcí obsahuje i některé navíc, jako je například verifikace zapsaných dat. Reakce GUI by bylo možné vylepšit přepsáním programu do více vláken.

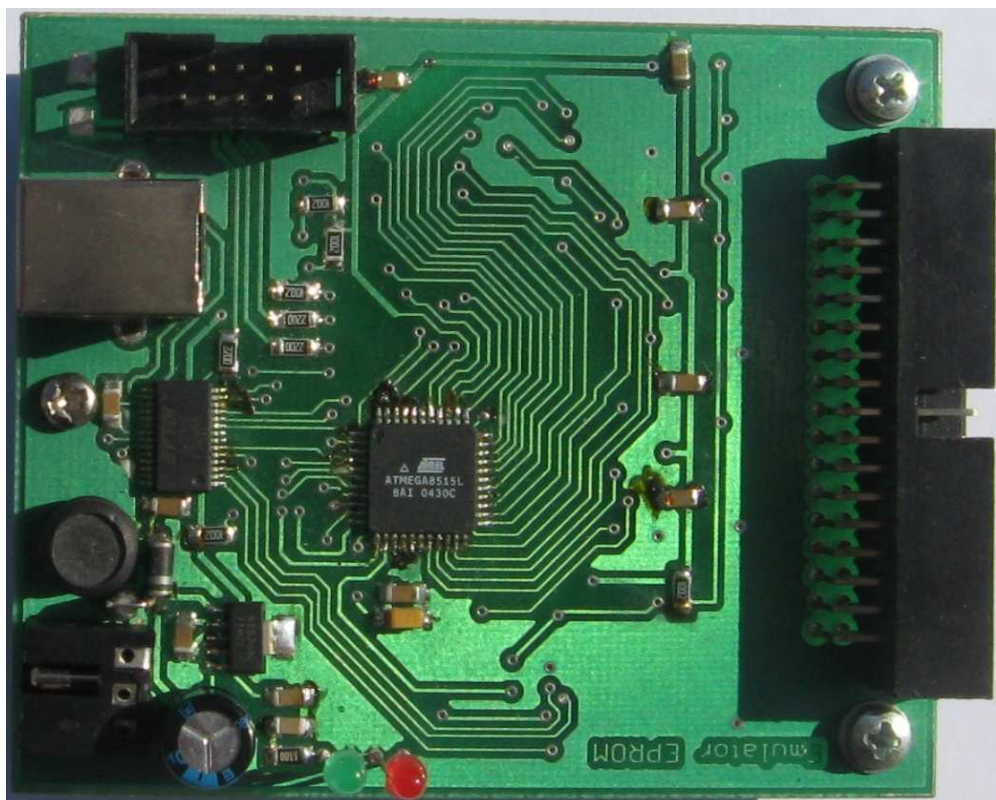
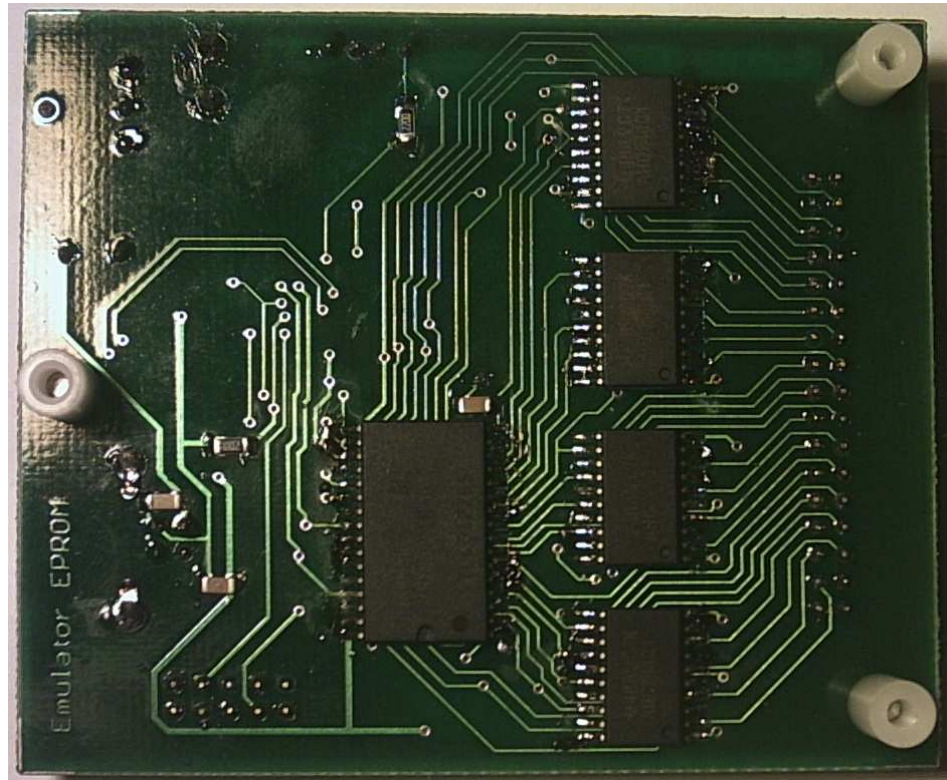
Testování zařízení v praxi proběhlo bez větších problémů, spolupráce s vývojovým kitem fungovala taktéž. Simulátor je možné po odstranění výše uvedených chyb nasadit do provozu v laboratořích.

Seznam literatury

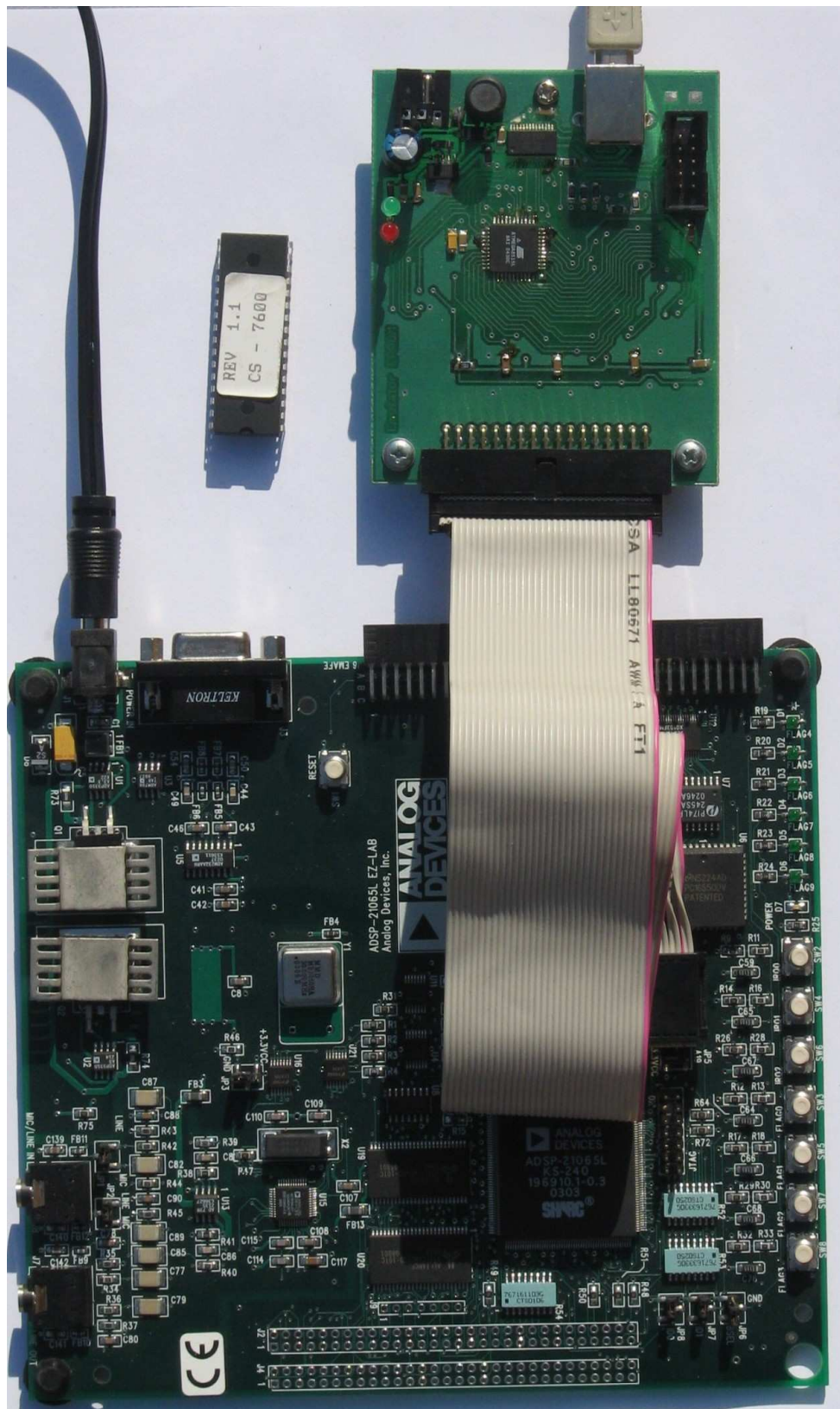
- [1] KERNIGHAN, Brian W., RITCHIE, Dennis M. Programovací jazyk C : ANSI C99. 1. vyd. Brno : Computer Press, 2006. 286 s. ISBN 80-251-0897-X.
- [2] BURKHARD, Mann. C pro mikrokontroléry : uC & Praxe. Překlad Václav Losík. 1. vyd. Praha : BEN - technická literatura, 2003. 280 s., 1 CD-ROM. ISBN 80-7300-077-6.
- [3] Intel Corporation. Hexadecimal Object File Format Specification [online]. c2000 [cit. 2009-08-10]. Formát pdf. Text v angličtině. Dostupný z WWW: <<http://microsym.com/editor/assets/intelhex.pdf>>.
- [4] SELLS, Chris. C# a WinForms : programování formulářů Windows. Překlad: RNDr. Jan Pokorný. 1. české vyd. Brno : ZONER Press, 2005. 648 s. ISBN 80-86815-25-0.
- [5] SHARP, John. Microsoft Visual C# 2008 : Krok za krokem. 1. české vyd. BRNO : Computer Press, 2008. 592 s., 1 DVD-ROM. ISBN 978-80-251-2027-9.
- [6] PK Design. UniProg-USB v1.0 [online]. 2007 [cit. 2009-08-10]. Dostupný z WWW: <http://www.pk-design.net/Datasheets/UniProgUSB_v10_doc_ver_v11_20070820.pdf>.
- [7] STMicroelectronics. M27V201 [online]. 2000 [cit. 2009-08-10]. Dostupný z WWW: <<http://www.datasheetcatalog.org/datasheet/stmicroelectronics/2391.pdf>>.
- [8] SAMSUNG Electronics. K6X4008T1F Family [online]. 2003 [cit. 2009-08-10]. Dostupný z WWW: <http://www.datasheetcatalog.org/datasheets/185/264557_DS.pdf>.
- [9] Texas Instruments. SN74AHC244Q : OCTAL BUFFER/DRIVER WITH 3-STATE OUTPUTS [online]. c2002 [cit. 2009-08-10]. Dostupný z WWW: <<http://focus.ti.com/lit/ds/symlink/sn74ahc244q.pdf>>.
- [10] Analog Devices. ADSP-21065L EZ-KIT Lite : Evaluation System Manual [online]. 2003 [cit. 2009-08-10]. Dostupný z WWW: <http://www.analog.com/static/imported-files/eval_kit_manuals_legacy/62854542ADSP_21065L_EZ_KIT_Lite_Manual_Rev2.pdf>.

- [11] Analog Devices. DSP Microcomputer : ADSP-21065L [online]. 2003 [cit. 2009-08-10]. Dostupný z WWW: <http://www.analog.com/static/imported-files/data_sheets/ADSP-21065L.pdf>.
- [12] ATMEL. AVR042 : AVR Hardware Design Considerations [online]. 2008 [cit. 2009-08-10]. Dostupný z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc2521.pdf>.

Příloha A – fotografie prototypu simulátoru



Příloha B – fotografie propojení simulátoru s vývojovým kitem



Příloha C – fotografie vývojového kitu s osazenou EPROM M27V201

