VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

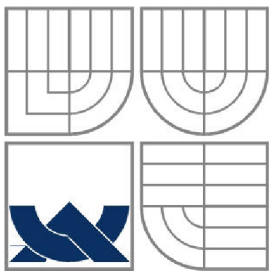
DISK NA BÁZI PAMĚTI FLASH

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

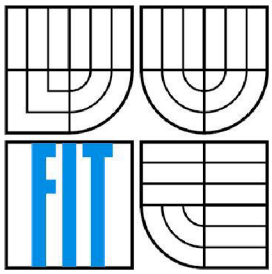
AUTOR PRÁCE
AUTHOR

Bc. MIROSLAV DVOŘÁK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DISK NA BÁZI PAMĚTI FLASH

DRIVE BASED ON FLASH MEMORY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MIROSLAV DVOŘÁK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2011

Abstrakt

Práce se zabývá technologií flash, historií jejího vývoje, současnými aplikacemi této technologie a diskutuje kladné i záporné vlastnosti těchto paměťových médií. Podrobně zkoumá integraci technologie flash do vysokokapacitních úložných zařízení a běžně používané mechanismy, které potlačují nedostatky pamětí flash pro tuto aplikaci. Součástí práce je i rozbor sběrnic, ke kterým se tato velkokapacitní zařízení běžně připojují. Na uvedených teoretických základech je poté postaven návrh vlastního řešení disku na bázi paměti flash. Práce se věnuje zejména výběru dostupné platformy pro připojení disku k osobnímu počítači - USB, výběru vhodných HW komponent pro vývoj vlastního disku, výrobě PCB pro paměťový modul v prostředí Eagle CAD a implementaci potřebného firmware pro MCU a VHDL designu pro FPGA, které zajišťují základní funkcionalitu zařízení. Na závěr práce shrnuje dosažené výsledky a nastiňuje směr dalšího vývoje.

Abstract

The work deals with flash technology, the history of its development, current application of this technology and discusses the advantages and disadvantages of flash memories. It describes the integration of flash technology into mass storage devices and commonly used mechanisms that suppress the flash shortcomings for such application. The next part of the work focuses on analysis of commonly used buses for flash storage devices. Based on these theoretical foundations, text presents way to develop own flash based disk. The work focuses mainly on finding the most accessible platform for connecting the disk to personal computers - USB, on PCB design for storage module in Eagle CAD and implementation of necessary firmware for MCU and VHDL design for FPGA, that provide the disk functionality. At the end the work summarizes the results and outlines the way of further development.

Klíčová slova

Paměť flash, disk bez pohyblivých částí, řadič disku, USB, FPGA, PCB

Keywords

Flash memory, solid state drive, disk controller, USB, FPGA, PCB

Citace

Dvořák Miroslav: *Disk na bázi paměti FLASH*, *diplomová práce*, Brno, FIT VUT v Brně, 2012

Disk na bázi paměti FLASH

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Václava Šimka
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Miroslav Dvořák
30.5.2012

Poděkování

Tímto bych chtěl poděkovat vedoucímu diplomové práce, Ing. Václavu Šimkovi, za jeho odborné vedení, pomoc při výrobě fyzického zařízení, podnětné připomínky a celkově velmi vstřícný přístup.

Obsah

1 Úvod

Číslicové systémy, pod tímto pojmem si můžeme přestavit mnoho, od výkonného superpočítače, až po systém řídící semafory na křižovatce. Všechny tyto systémy, charakterizuje jedna společná vlastnost, a to, že se jedná o systémy, které pracují s informacemi v diskrétní podobě. Reprezentaci takových informací můžeme zjednodušeně označit jako data. Číslicové (digitální) systémy tedy přijímají vstupní data, a ta potom transformují, dle svého specifického programu, na data výstupní, případně nějakou mechanickou/signálovou reakci. Mimo vstupní a výstupní data mohou při oné transformaci vznikat také meziprodukty v podobě dat dočasných. Všechna tato vstupní, výstupní a dočasná data je nutné samozřejmě uchovávat a přenášet pomocí různých typů médií, která se liší především svým:

- určením (sítě – transport dat, RAM – uchování dat, CD uchování i transport dat)
- technologií (CD – optický mechanismus, Floppy disk – magnetický mechanismus)
- kapacitou (CD/DVD)
- rychlostí

Tato média můžeme také hierarchicky rozdělit, podle jejich úlohy v systému na:

- Interní paměť – registy, cache procesoru
- Primární paměť – systémová paměť RAM a paměti rozšiřujících adaptérů
- Sekundární paměť – pevný disk
- Terciární paměť – odpojitelná datová úložiště-sloužící především pro archivaci

Primární paměť byla v posledních letech vyvíjena především s úmyslem zvýšení rychlosti (SDR->DDR->DDR3), zatímco sekundární paměť (mechanický pevný disk) doznal značného navýšení své kapacity, ovšem už nedosáhl takového pokroku v oblasti zrychlování. Je to tedy pevný disk, který dokáže snížit celkový výkon systému v případě intenzivnější práce s daty (ať už se jedná o zápis nebo čtení). Sekundární úložiště v podobě pevných disků se vyskytují nejen ve všech osobních počítačích, výkonných stanicích, ale i v aplikačně specifických řešeních, která pracují s větším objemem dat.

1.1.1 Pevné disky

Pokud zůstaneme na poli osobních počítačů, tak zde platí, že nejběžnějším vysokokapacitním datovým úložištěm je pevný disk na mechanické bázi. Ten je tvořen rotujícími kotouči z magneticky měkkého materiálu, na nějž je informace uložena pomocí zmagnetizování určitého segmentu zapisovací hlavou [6]. Data jsou opět přečtena hlavou, ve které se indukuje elektrický proud při průchodu nad zmagnetizovanými segmenty. Tato technologie má velmi lákavý poměr cena/kapacita, ovšem méně přesvědčivá je odolnost proti poškození, spotřeba a také rychlost. V době, kdy se i nejlevnější přenosné počítače staví na vícejádrových procesorech, začíná být právě tento mechanický pevný disk brzdou pro velkou řadu i běžně používaných aplikací.

Řešením problému s rychlostí/spolehlivostí pro výkonné stanice je sdružování jednotlivých pevných disků do RAID polí. Toto ale určitě není cesta pro běžné nasazení například v přenosných počítačích, nebo dalších specifických systémech, kde je současně důležitým kritériem spotřeba, hmotnost nebo rozměry diskového systému.

Historie kotoučových pevných disků na magnetické bázi sahá do roku 1956, kdy firma IBM představila produkt 350 RAMAC (který měl nahradit magnetické pásky), od té doby prošly vývojem, který spočíval především ve zvyšování hustoty zápisu, tedy potažmo zvyšování kapacity, zmenšování

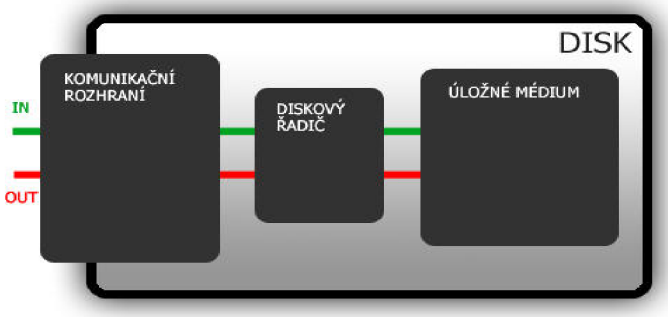
a zrychlování. Zároveň po celou dobu vývoje kotoučových pevných disků existovaly snahy o nalezení alternativy na nemagnetické bázi. Ovšem výsledky těchto snah byly příliš nákladné pro komerční výrobu a často se potýkaly s malou životností, či neschopností uchovávat data bez napájení.

Skutečně první, komerčně úspěšný produkt se objevil až s využitím NAND flash paměti, která je schopná uchovávat data i v době, kdy není napájena.

Způsoby integrace této technologie do pevných disků, jejími přednostmi, postupy jak čelit jejím nedostatkům a konceptuálním nástinem, jak takový disk vytvořit se zabývá tato práce.

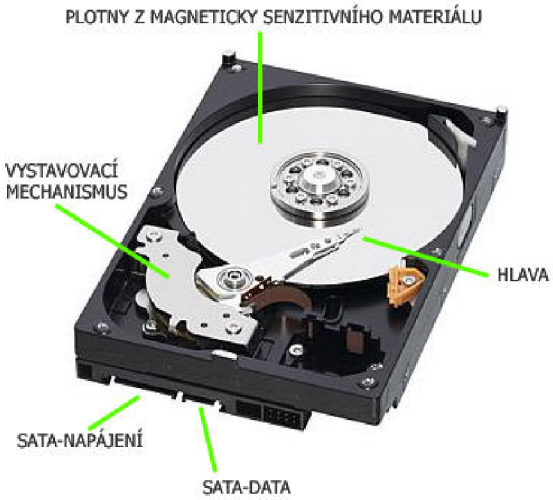
2 Disk na bázi paměti flash

Komerčně nasazené pevné disky na bázi paměti flash se označují zkratkou SSD (solid state drive), mimo tato relativně nová média se více než dekádu těší velké oblibě malé USB disky na bázi paměti flash, zaměřené především na snadný transport dat. Ačkoliv schematický koncept základních stavebních prvků těchto zařízení (viz Obrázek 1 : obecné, zjednodušené schéma disku) je téměř totožný s mechanickými pevnými disky, funkce jednotlivých prvků (s výjimkou komunikačního rozhraní) je zcela odlišná.



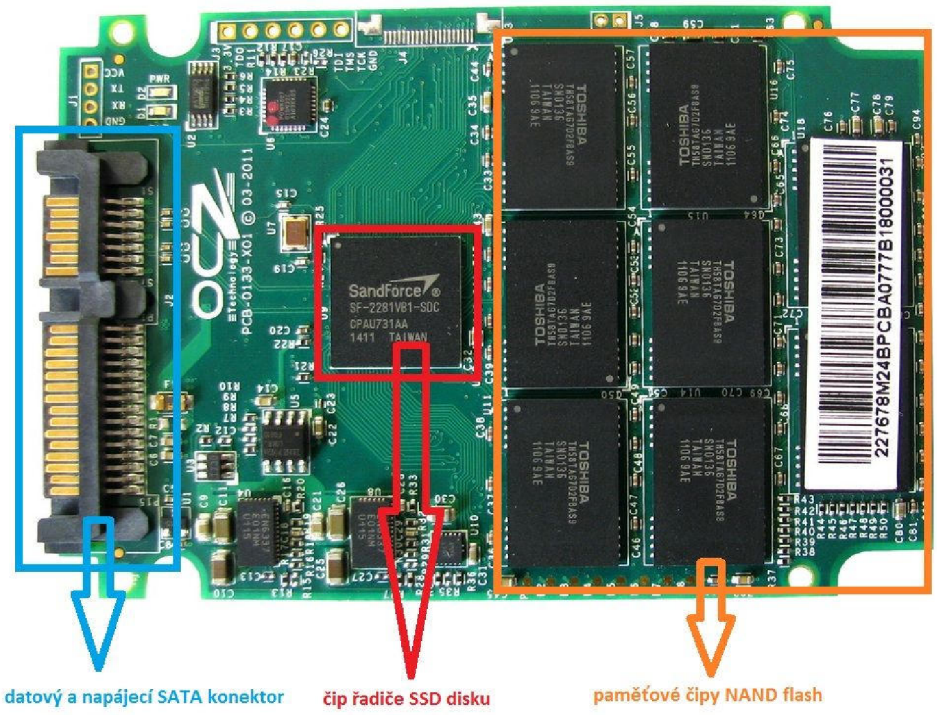
Obrázek 1 : obecné, zjednodušené schéma disku

U mechanických pevných disků si pod pojmem úložné médium představujeme kotouče z magneticky citlivého materiálu a diskový řadič má za úkol, velmi zjednodušeně řečeno, řídit vystavení zapisovacích/čtecích hlav nad rotujícími plotnami a přečtení/zápis hodnoty ze správného sektoru viz Obrázek 2.



Obrázek 2 : Mechanický pevný disk (současnost)

V případě disku na bázi paměti flash (viz Obrázek 3) se bude jednat o úložné médium v podobě pole čipů NAND flash a diskový řadič v základní podobě by mohl být jen řídící člen, který zažádá o blok dat ze správné adresy(čtení), nebo naopak odešle blok dat (zápis). Jak ovšem později uvidíme, je zapotřebí mnohem více, aby tato technologie mohla být komerčně úspěšná.



Obrázek 3 : Uvnitř SSD disku [1]

2.1 Historie vývoje disků na bázi paměti flash

Vývoji disků s paměťmi NAND flash předcházela celá řada pokusů o vývoj disků bez pohyblivých částí, která sahá do historie téměř stejně daleko, jako vývoj samotných kotoučových pevných disků. Důvodem je spotřeba, hmotnost, rozměry i rychlost mechanických disků.

První prototypy disků bez pohyblivých částí byly zkonstruovány už v 50tých letech 20. století, jejich základním stavebním kamenem byla paměť založená na magnetických jádrech a jednalo se o stejnou technologii, která do příchodu polovodičových pamětí plnila v počítačích funkci hlavní paměti. Vývoj tímto směrem byl zastaven s příchodem kotoučových pevných disků, jejichž cena byla nižší.

O dvě dekády později byl vyvinut disk bez pohyblivých částí pro superpočítače firmy IBM, tento typ disku byl již postaven na polovodičové technologii, ale opět kvůli vysokým nákladům na výrobu se tento systém neprosadil.

Na konci 70tých let firma General Instruments vyvinula disk na bázi elektronicky programovatelné ROM, který v mnoha ohledech mohl připomínat dnešní NAND flash paměť, ovšem životnost této technologie byla značně nižší a proto se toto řešení neujalo jako nepraktické [2].

Další slibnou technologií byla tzv. bublinová paměť (viz Obrázek 4), která využívala tenký film magnetického materiálu, který byl schopen udržet magnetický náboj malých oblastí (využit například v počítači Sharp PC-5000). V 80tých letech ovšem tato technologie opět neuspěla díky inovacím provedeným na mechanických pevných discích, které vedly ke snížení jejich ceny a zrychlení [7].

Obrázek 4 : Čip bublinové paměti [7]

Následovaly další systémy založené na technologiích primární paměti, které disponovaly záložním napájením nebo baterií pro uchování dat v případě výpadku primárního napájení (např. BatRam – Santa Clara Systems).

Vůbec první paměť flash (jak NOR, tak NAND), tehdy nazývaná SE-EEPROM (simultaneously erasable EEPROM), pochází také z 80tých let, z laboratoří firmy Toshiba a za jejím vývojem stojí Dr. Fujio Masuoka. Technologie byla představena veřejnosti na International Elektron Device Meetingu v roce 1984 v San Franciscu [2].

Potenciál technologie, konkrétně NOR flash, využila firma Intel v roce 1988, když komerčně nasadila tento druh paměti jako náhradu ROM čipů. Tato paměť umožňuje přístup na konkrétní adresu, na druhou stranu trpí nízkou rychlostí zápisu a mazání.

V roce 1987 firma Toshiba představila oficiálně i technologii NAND flash, která má oproti technologii NOR rychlejší zápis i přepis, ovšem k datům se přistupuje typicky po stránkách, což není vhodné jako náhrada pro programovou ROM, ale tento přístup lze využít u datových úložišť jako paměťové karty, nebo disky [2].

2.2 Komerční řešení s technologií flash

Technologie NAND flash nachází uplatnění všude tam, kde jsou kladeny požadavky na nízkou spotřebu, vyšší odolnost vůči vnějším vlivům (otřesy, nárazy aj.), nižší hmotnost, malé rozměry i nižší přístupovou dobu k datům [5].

2.2.1 Spotřební elektronika

Mezi konkrétní aplikace patří primárně mobilní spotřební elektronika. Do této kategorie můžeme zahrnout například mobilní MP3 přehrávače, mobilní telefony a smartphony, digitální fotoaparáty a kamery aj. Tyto produkty, v podobě, jak je dnes známe, jsou přímým důsledkem vývoje pamětí flash.

2.2.2 USB flash disk

Další využití flash pamětí v podobě tzv. USB flash disku, bylo uvedeno na trh na sklonku roku 2000. Jednalo se o řešení, které mělo přímo nahradit do té doby nejrozšířenější přepisovatelé přenosné médium – floppy disk. První produkty měly kapacitu 8MB (cca 5krát více než klasická disketa). O prvenství při vývoji se hlásí 4 strany (M-Systems, Phison Electronics, Trek Technology a Netac Technology). V současnosti tyto disky dosahují kapacit od 1GB po 256GB.

2.2.3 SSD disky na bázi flash

První SSD disk na bázi pamětí flash byl představen v roce 1995 společností M-Systems. Paměti flash nahradily v koncepci SSD disků do té doby používané DRAM paměti, které na rozdíl od pamětí flash

vyžadovaly nepřerušované napájení, jinak nebyly schopny udržet data. Tyto disky byly určeny pro nasazení ve ztížených podmínkách a armádním průmyslu.

V roce 1999 bylo ohlášeno vydání několika řešení, která měly plnit funkci pevného disku a která byla založena na flash pamětech. Jejich předpokládaná kapacita měla dosáhnout až 18GB. S klesající cenou flash pamětí, stoupala dostupnost těchto SSD disků a zvětšovala se jejich kapacita. V současnosti jsou běžné kapacity od 64GB po 500GB, ale existují i modely s kapacitou 2TB.

Vzhledem k událostem na podzim 2011, kdy došlo vlivem přírodní katastrofy k omezení celosvětové výrobní kapacity klasických mechanických pevných disků a jejich následnému celosvětovému zdražení o více než 100%, je pravděpodobné, že SSD disky se opět přiblížily nahrazení mechanických pevných disků v aplikacích, kde není primárním kritériem uchovávání velkého množství dat. Jejich použití v osobních počítačích jako systémový disk, nebo nasazení v odvětví přenosných počítačů už není pouze ojedinělým jevem.

2.3 Vlastnosti dostupných komerčních disků na bázi paměti flash

Výhody

- Hlavní předností těchto disků, pro většinu zájemců, je jejich rychlost. Běžná řešení dosahují u optimálních dat (data velikostně přizpůsobená velikosti bloku) rychlostí přes 500MB/s čtení i zápis. Důvodem je jednak technologie flash a její vybavovací doba blížící se 0,1ms, ale také koncepce řadiče.
- Nízká poruchovost – průměrná doba mezi poruchami činí 2mil. hodin (cca 228let) [9]
- Vysoká odolnost – tato technologie není tak náchylná na mechanické vlivy jako mechanické pevné disky – snáší lépe pády, otřesy a je schopna operovat v širším teplotním rozmezí. V číslech jsou při pádu v zapnutém stavu cca 23krát odolnější (1500G / 65G) [9]
- SSD jsou malé, lehké a neprodukují žádný hluk
- SSD mají nízkou spotřebu – jejich spotřeba většinou nedosahuje ani 20% spotřeby mechanického pevného disku.
-

Nevýhody

- Náklady na pořízení SSD
- Jiný princip fungování úložného média přináší nové problémy, které je nutné řešit speciálními technikami, většinou implementovanými v rámci řadiče.
- Flash paměť má omezenou životnost – počet čtení/zápisů je vázán na konkrétní výrobní technologii a u prvních generací spotřebních SSD disků, byl toto jeden z největších argumentů v neprospěch SSD disků na bázi paměti flash.

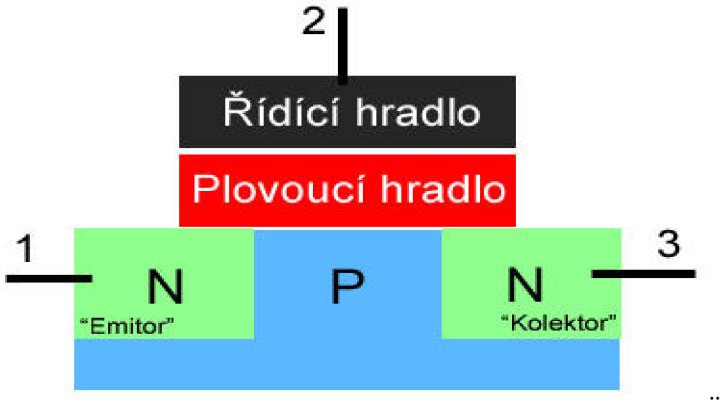
3 Technologie disků s pamětí flash

S postupnou miniaturizací výrobního procesu paměťových čipů stoupá jejich kapacita a zároveň klesají náklady na jejich výrobu, lze tedy očekávat, že bude-li tento trend nadále úspěšně pokračovat, přijde doba, kdy SSD zcela vytlačí současné mechanické pevné disky do ústraní. Jejich výhody značně převyšují jejich zápory. A zápory, které vrhaly stín na tuto technologii se daří úspěšně eliminovat pomocí speciálních postupů, kterými se budeme dále v textu zabývat.

3.1 Flash paměti

Jedná se o nevolatilní, přepisovatelnou paměť s libovolným přístupem. Na rozdíl od předchůdce v podobě EEPROM lze přepisovat přímo jednotlivé bloky, nikoliv jen celý obsah paměti.

Paměť tvoří pole unipolárních tranzistorů viz Obrázek 5. Tranzistor je konstrukčně velmi podobný tranzistoru typu MOSFET, má však navíc plovoucí hradlo, jež je umístěno mezi kontrolní hradlo a kanál tranzistoru. Plovoucí hradlo je od svého okolí kompletně odizolováno slabou vrstvičkou izolantu, to vede k tomu, že pokud se do něj dostanou nějaké elektrony, zůstanou v něm uvězněny až po několik let - to je princip zapamatování si hodnoty. Slabý náboj takto uvězněných elektronů ovlivňuje napětí přivedené na řídící hradlo a tím také ovlivňuje průtok proudu kanálem tranzistoru - což je způsob jakým může být čtena uložená hodnota. Uložení hodnoty potom spočívá v přivedení vyššího než čtecího (kladného) napětí na konektor řídícího hradla, což má za následek, že některé elektrony, které proudí tranzistorovým kanálem projdou skrze izolační obal do plovoucího hradla, a některé z nich zůstanou v onom hradle uvězněny - takto je uchována informace i v případě ztráty napájení. Vymazání obsahu je proces, při němž dojde k uvolnění elektronů zachycených v plovoucím hradle. Toho je docíleno tak, že se na řídící hradlo a zdrojový konektor tranzistoru přivede vyšší napětí opačných pólů, což umožní zachyceným elektronům opět překonat izolační vrstvičku a opustit plovoucí hradlo, tento jev se nazývá Fowler-Nordheimův tunel [5].

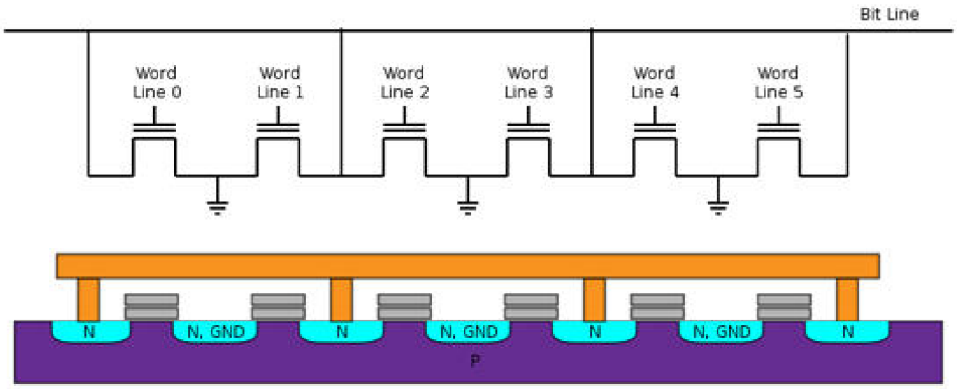


Obrázek 5 : Schéma unipolárního tranzistoru

Podle způsobu jakým jsou tyto unipolární tranzistory integrovány do pole rozlišujeme 2 hlavní druhy paměti flash:

3.1.1 NOR flash

Paměť nazývaná jako NOR flash, je tvořena tranzistory zapojenými tak, že každý jednotlivý tranzistor je připojen jedním konektorem přímo k zemi a druhým na bitový výstup (Obrázek 6). Takové zapojení vede k tomu, že se pole chová jako NOR hradlo [2].



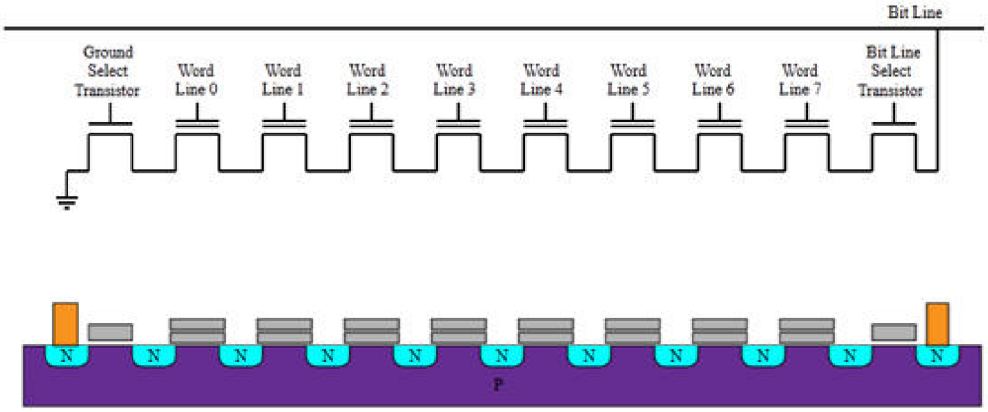
Obrázek 6 : Schéma paměti NOR flash [2]

V základu uchovávají jednotlivé tranzistory hodnotu ekvivalentní logické hodnotě 1, jelikož při přivedení patřičného napětí na řídící hradlo (2), bude protékat proud tranzistorem. Uložení hodnoty ekvivalentní logické 0 poté probíhá tak, že je přivedeno zvýšené napětí na konektor řídícího hradla, to má za následek, že proud elektronů je dostatečně silný, aby umožnil některým z nich projít skrze slabou izolační vrstvu do plovoucího hradla.

Opětovné vymazání paměťové buňky (její nastavení a hodnotu ekvivalentní logické 1) poté proběhne tak, že je přivedeno vyšší rozdílné napětí na konektor řídícího hradla a emitoru (nazývaného taktéž jako zdroj). To má za následek, že elektrony zachycené v plovoucím hradle jsou vypuzeny. Celé pole unipolárních tranzistorů je rozděleno do menších segmentů, které mohou být vymazány samostatně.

3.1.2 NAND flash

NAND flash byla vyvinuta se snahou maximalizovat kapacitu vztaženou na plochu čipu, zároveň se snaží dosáhnout co možná nejnižší ceny vzhledem ke kapacitě. Využívá shodnou technologii jako NOR flash - unipolární tranzistory, ovšem liší se v jejich zapojení. U tohoto druhu paměti jsou unipolární tranzistory zapojeny v sérii (Obrázek 7). To má za následek, že hodnoty uchovávané v takové sérii musí být čteny naráz, stejně tak zápis je prováděn na celé sérii (narozdíl, od technologie NOR, kde lze adresovat jednotlivé tranzistory = bity). Tuto sérii nazýváme stránkou a stránky jsou sdružovány v bloky. Operace mazání je potom prováděna nad celým blokem (nikoliv slektivně na úrovni stránek) [2].



Obrázek 7 : Schéma paměti NAND flash [2]

Čtení v takovémto zapojení podle značení v obrázku Obrázek 7 : Schéma paměti NAND probíhá tak, že na všechny "word line" konektory je přivedeno napětí těsně nad hodnotu napětí, která je nutná pro průtok proudu naprogramovaným tranzistorem a na jeden vybraný "word line" je přivedeno napětí těsně nad úroveň napětí, která je nutná pro průtok proudu vymazaným tranzistorem. Potom bude sérií protékat elektrický proud (bit line bude nastavena na hodnotu ekvivalentní logické 0), právě pokud vybraný tranzistor(bit) není naprogramován.

Bitové linky jednotlivých stránek jsou poté propojeny dalšími tranzistory, i přesto je ale technologie NAND o cca 45% méně prostorově náročná, než technologie NOR.

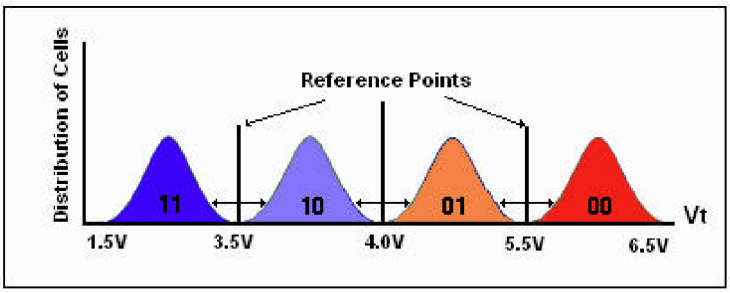
3.1.3 Nejdůležitější rozdíly mezi NOR a NAND flash

- obě technologie se liší v zapojení jednotlivých paměťových buněk
- rozhraní, které aplikuje čtení a zápis do paměti je kompletně odlišné, zatímco NOR technologie umožňuje náhodný přístup po bitech, technologie NAND umožňuje přístup pouze po stránkách [5].

3.1.4 SLC, MLC, TLC

Funkčnost popsaná v kapitole 3.1.1 a 3.1.2 se vztahuje přímo na technologii SLC (single level cell), což znamená, že každá buňka uchovává právě jeden bit informace. SLC je technologie NAND flash, která má nejdelší životnost, nejnižší výskyt chyb a je také nejrychlejší. Zároveň má ale nejnižší hustotu dat na plochu čipu, tudíž nejméně příznivý poměr cena/kapacita. Tyto paměti jsou v současnosti používány v dražších SSD discích [4].

MLC (multi level cell) je technologie, která umožňuje uchovat v každé buňce 2 bity informace. Princip spočívá ve schopnosti udržovat na plovoucím hradle více úrovní náboje, což vede k tomu, že jsme schopni reprodukovat z napětí na výstupu tranzistoru i více logických úrovní (v tomto případě celkem 4: 00-plně naprogramovaná buňka, 01 - částečně naprogramovaná buňka, 10 - částečně vymazaná buňka, 11 plně vymazaná buňka). Reprezentaci jednotlivých logických úrovní vidíme na obrázku Obrázek 8 : rozložení logických úrovní MLC NAND flash [4].



Obrázek 8 : rozložení logických úrovní MLC NAND flash [4]

TLC (tripple level cell) je už pouze složitější analogií, k technologii MLC, kde jsou ještě více prohloubeny rozdílné vlastnosti mezi popsanými SLC a MLC. TLC by měla na trh přinést velkokapacitní užitkové SSD disky za rozumnou cenu. Prvním komerčním diskem, který měl být vybaven tímto typem úložného média měl být disk OCZ Octane a měl mít kapacitu 1TB. Disk byl nakonec uveden na trh pouze s použitím technologie MLC a kapacitou 128GB. Technologie TLC tedy ještě nebyla komerčně nasazena.

3.1.5 Standardizace výroby flash paměťových čipů

Jelikož výrobní technologie, kterou jsou flash paměti vyráběny se neustále rychle vyvíjí a pro výrobce zařízení využívajících takovéto čipy je velmi nepraktické, aby byli nuceni měnit návrh svého produktu ať už kvůli změnám v technologii nebo při změně dodavatele NAND flash čipů, byly vytvořeny standardy, které zaručují, že jakýkoliv čip splňující tuto normu bude použitelný v návrhu, který byl s ohledem na tuto normu vyvinut.

ONFI (Open NAND Flash Interface) [3] je standard, který se vyvíjí od roku 2006 a v současnosti se dočkal revize 3.0. Revize 1.0 definovala rozhraní pro rychlejší integraci zařízení do hostitelských platforem bez nutnosti zásahu do firmware. Je definována také propustnost, která je limitována hodnotou 50 MB/s. Následné revize zvyšují definovanou propustnost a přidávají také například podporu EZ NAND flash, což jsou čipy, které disponují integrovanou ECC korekcí chyb. Za vznikem tohoto standardu stojí firmy jako Intel, Micron, Sony, Sandisk, Marvell, Seagate, Sandforce aj.

Tuto normu ovšem nepřijali 2 giganti: Samsung a Toshiba, kteří mají svůj vlastní standard zvaný Toggle, jehož vývoj započal později (revize 1.0 v roce 2009, revize 2.0 v roce 2011). Výhody plynoucí z tohoto standardu jsou srovnatelné s ONFI. Jelikož oba standardy se v současné době ubírají směrem ke standardizaci v rámci JEDEC (vývoj otevřených standardů v oblasti mikroelektroniky), lze očekávat sloučení obou standardů.

Rozhraní pamětí flash je stejné jako u paměti SRAM s rozdílem v úrovních napájecího napětí potřebných pro programování a mazání.

3.1.6 Výrobní proces

Výrobní proces je často označován číselnou hodnotou, která udává, velikost nejmenší součástky, kterou lze tímto procesem zhotovit. V posledních letech, pro polovodičové součástky platí, že jejich výrobní proces se zmenšuje o cca 70% každé 2-3roky. Flash paměti jsou v současnosti v tomto ohledu jednou z nejagresivněji se vyvíjejících oblastí mezi polovodičovými součástkami. Zmenšování výrobního procesu s sebou přináší především možnost zvětšování kapacity na jednom čipu, snižování výrobních nákladů (zmenšuje se množství použitého materiálu), snižování spotřeby a zároveň vyzařování odpadního tepla. V současnosti se největší množství komerčně nasazovaných NAND flash čipů vyrábí 25nm technologií, jsou už také nasazovány čipy vyráběné 20nm výrobním procesem.

3.1.7 Životnost flash pamětí

Životnost pamětí flash je možné určovat jako počet zápisových cyklů. Budeme-li brát v úvahu technologii SLC, potom NOR flash dosahuje v současnosti nejhůře shodné životnosti jako NAND flash nebo lepší. U technologií MLC a TLC se výhoda NOR flash zmenšuje [4].

- SLC NAND cca 100 000 zápisů
- MLC NAND cca 5000- 10 000zápisů
- TLC NAND cca 100-500 zápisů
- SLC NOR 100 000-1 000 000 zápisů
- MLC NOR 100 000 zápisů.

3.1.8 Synchronní / asychronní flash paměť

Synchronní flash paměť je flash paměť, která dokáže vykonat teoreticky 2násobek čtecích/zapisovacích operací za shodnou jednotku času, než asynchronní flash paměť. Jedná se samozřejmě o nákladnější technologii a její zmíněné vlastnosti je dosaženo tím, že tato paměť dokáže vykonávat dané operace nejen s vzestupnou hranou řídícího hodinového signálu, ale také se sestupnou hranou (obdoba DDR RAM). Rozvod takového centrálního hodinového signálu je definován v normě ONFI rev.2.0. Použitelnost této technologie je tedy vázána na řadič, který respektuje specifikaci minimálně ONFI rev. 2.0.

3.1.9 Budoucnost technologie flash

Mimo zdokonalování výrobního procesu se očekává nástup technologií, které udrží více informací v jedné buňce MLC, TLC, ... Sníženou spolehlivost těchto postupů bude nutné kompenzovat integrováním dokonalejších opravných mechanismů. Dalším možným směrem vývoje je vytvoření 3D-tranzistorových matic, namísto pouhého současného plošného uspořádání.

Zároveň existuje několik dalších slibných technologií, které by se mohly v budoucnu objevit v některých aplikacích, kde dnes vídáme paměti flash:

- **FeRAM** - nevolatilní pamět, založena na feroelektrické vrstvě (materiál, který je spontánně elektricky polarizovaný a jehož polaritu je možné změnit elektrickým polem). Dosahuje lepších zápisových rychlostí než flash, menší spotřeby a zároveň životnost této technologie se pohybuje až někde kolem 10^{16} zápisových cyklů. Na druhou stranu nedosahuje tak velké hustoty dat na čipu, jako paměti flash a její cena je vyšší. [11]
- **MRAM** (Magnetoresistive Random-Access Memory) - je nevolatilní paměť založená na uchování magnetického náboje (namísto elektrického), dosahuje obdobné rychlosti jako SRAM, obdobné hustoty na čipu jako DRAM (při zachování shodného výrobního procesu), jelikož buňky nepotřebují obnovovat má menší spotřebu než DRAM a netrpí degradací funkčních členů jako je tomu u flash pamětí. Technologie je vyvíjena již od devadesátých let, v současnosti představuje hlavní problém výrobní proces. Jelikož poptávka po všech polovodičových součástkách vyrobených výrobním procesem poslední generace (flash, DRAM, ...) převyšuje nabídku, nedovolí si žádný velký výrobce polovodičů určit celou továrnu s nejmodernějším výrobním procesem této nové technologii. V současnosti se komerčně nasazené MRAM vyrábějí několik generací starým 180nm výrobním procesem [12].
- **PMC** (Programmable metalization cell) - je technologie vyvíjená pod záštitou Univerzity v Arizoně, je založena na přemisťování iontů v pevném elektrolytu. Je určena k přímému nahrazení flash pamětí, dosahuje vyšší hustoty, nižší spotřeby a delší životnosti [13].

- **PCM** - je nejmladší z technologií (2008), je založena na prvcích 16 skupiny periodické tabulky prvků (síry, selenia, telluru), kdy z nich tvořené chalkogenidové sklo je schopno měnit svoji strukturu na základě zahřívání při průchodu el. proudu. (krystalické, amorfní). Technologie je schopna dosáhnout vyšší hustoty než současné NAND flash, ale trpí jinými neduhy (tepelná izolace mezi buňkami, přeskočení proudu do sousedních buňek atd.) komerčně je tedy technologie stále nenasaditelná [14].

3.2 Diskový řadič

Je další komponentou flash disků, která má přímý a velmi podstatný vliv na vlastnosti celého zařízení. Je to právě řadič, který umožňuje adekvátní použití pamětí technologie flash v SSD pevných discích v počítačích a dalších zařízeních. Samotná technologie pamětí flash není úplně ideální pro danou aplikaci (životnost, složitost atomických diskových operací), ale při integraci speciálních postupů, které zohledňují vlastnosti flash pamětí a jejich omezení je zřejmé, že tato technologie dosahuje zajímavých výsledků. Diskový řadič integruje a realizuje takové postupy.

Obecně se řadič skládá z procesoru, který vykonává program firmwaru, řadiče hostitelského rozhraní, obvodů rozhraní flash pamětí a obvodu, který má na starosti mapování logických adres na adresy fyzické. Další obvyklou částí je paměť procesoru, která v sobě může implementovat cache podobnou diskové cache, ale také uchovává informace o mapování adres nebo metadata jednotlivých stránek paměťového subsystému - s tím souvisí dále popsané techniky jako bad block mapping, wear leveling apod. Časté je i použití obvodu, který realizuje kontrolu správnosti dat, který je opět součástí technik řešících spolehlivost disků na bázi flash paměti (Obrázek 9).

Přesný popis daných technologií a algoritmů je často dobře stráženým výrobním tajemstvím každého producenta těchto řadičů a jsou chráněny patenty.

3.2.1 Paměť cache

V provozním režimu se mimo klasickou diskovou cache používá také pro uchování mapovacích informací (logicko-fyzické mapování adres) nebo třeba i metadat jednotlivých stránek. Paměť cache je díky způsobu, jakým je k ní nutné přistupovat (nejlépe bitově pro všechny operace) většinou realizována technologií RAM (DRAM) a tedy nezůstane zachována při výpadku napájení. Problém je v tom, že mapovací data (i metadata) jsou životně důležitá pro správné fungování disku, proto existuje buď další nevolatilní paměť, do které se tato data při vypínací sekvenci přesunou, nebo je pro tato data vyhrazena část v datové paměti. V každém případě, žádný z těchto způsobů neřeší záchranu dat v případě náhlého výpadku napájení. Proto současné SSD disky uchovávají metadata přímo v konkrétních blocích v datové paměti (to má za následek složitější algoritmy, pro práci s datovým úložištěm jako je zápis, přepis atd.) A v případě mapovacích dat se u disků určených pro kritické aplikace využívá záložních napájecích mechanismů v podobě baterie, nebo systému kondenzátorů, které v případě výpadku napájení poskytnou záložní energii pro proces uložení kritických dat z volatelní cache, do nevolatilní paměti. Tyto tabulky s kritickými daty nejsou jediným případem, kdy může výpadek napájení ohrozit integritu dat na disku. Jelikož diskové operace u flash disků jsou mnohem komplexnější než operace u běžných mechanických disků, může dojít k porušení integrity i při výpadku napájení v době, kdy probíhá například přepis, garbage-collection aj.

3.2.2 Základní diskové operace v podání SSD disku

Mezi základní diskové operace patří čtení, zápis, vymazání a přepis dat, tyto operace mají samozřejmě zcela odlišný průběh, než u mechanických pevných disků. Z principů, jak jsou tyto operace prováděny plyne řada omezení.

Čtení dat
Čtení dat je poměrně přímočarou operací, jak by se dalo čekat z konstrukce NAND flash, přídavná logika obsahuje registr, jehož velikost odpovídá velikosti stránky (tedy sérii unipolárních tranzistorů). Při zpracování požadavku na přečtení určité stránky je fyzicky provedena operace "čtení hodnot" série unipolárních tranzistorů, které představují danou stránku a výsledek je uložen do daného registru. Přečtení 4kb stránky u technologie SLC NAND trvá přibližně 25µs [2].

Zápis dat
Z povahy pamětí flash plyne, že data mohou být zapisována pouze do vymazaných stránek, při prvotním použití jsou všechny buňky vymazány a řadič se tedy snaží nová data zapsat vždy do zcela prázdné stránky, v případě, že prázdné stránky nejsou k dispozici, bude se vždy jednat o operaci přepsání stránky. Zápis 4kb stránky do volného místa v SLC NAND flash paměti trvá přibližně 250µs [2].

Vymazání dat
Mazání dat v pamětech flash neprobíhá na úrovni jednotlivých stránek, ale na úrovni celých bloků. Proto pokud bychom chtěli mazat selektivně některá vybraná data, není to možné. Selektivní vymazání daných částí paměti umožňuje technika zvaná garbage collection. Spočívá v tom, že data bloku, která jsou stále aktuální, jsou přečtena a uložena na předem uvolněné místo v paměti, je upraveno mapování těchto dat a starý, neaktuální blok je celý vymazán. Vymazání bloku SLC NAND flash paměti o velikosti 256kiB trvá přibližně 2ms [2].

Přepis dat
Jelikož SSD disk používá mapování z logického prostoru na fyzický prostor, je přepis proveden tak, že se příchozí data zapíšou na zcela nové umístění, s tím je upraven i logický odkaz na data na konkrétní fyzickou adresu. Data určená k přepisu nejsou ale stále vymazána a blokují tedy prostor pro nový zápis [2].

Z výše popsaného jednoznačně plyne, že disk je nejrychlejší, pokud není zaplněn. Dále, že pro rychlý přepis za pomoci zápisu dat na volné místo v paměti a přemapování adresy je nutné, aby existovala rezerva se stále uvolněným místem (není vhodně disk zcela zaplnit).

3.2.3 Základní techniky řadiče pro zvýšení životnosti / spolehlivosti

3.2.3.1 Mapování vadných bloků

Životností rozumíme počet zápisů/vymazání buňky, který předchází stavu, kdy buňka není schopna udržet požadovanou hodnotu. Proto každý řadič disponuje tabulkou, ve které uchovává adresy vadných bloků. Detekce a označení bloku za vadný a jeho následné přemapování na jiné umístění se provádí vždy při operaci zápisu. Důvodem je fakt, že i když jsou některé logické hodnoty chybné, je možnost, že specializované zotavovací algoritmy dokáží data opět opravit. Proto je důležité zachovat blok adresovaný pro případ čtení. Pokud by ovšem přišel požadavek na přepsání takového vadného bloku, pomíjí důvod pro jeho zachování a nová data jsou jednoduše přemapována jinam a daný blok označen jako chybný.

Existence takovéto techniky práce s flash pamětí umožňuje použít i čipy, které jsou již od své výroby znehodnoceny vadnými bloky. Ve skutečnosti prakticky žádný vyrobený flash čip nemá všechny bloky v pořádku, ale díky této technice je možné je nasadit a minimální množství odpadu při výrobě má příznivý vliv na cenu.

3.2.3.2 Wear leveling

Mimo označování vadných bloků je dobré také vzniku takových bloků předcházet, vznikají především opotřebováním. Základní technikou je zabránit nerovnoměrnému opotřebení bloků. Tedy rozkládat rovnoměrně data mezi všechny bloky paměťového zařízení. Nejjednodušší algoritmus implementující tuto funkcionalitu je založen na uchovávání počtů zápisů na konkrétní fyzické adresy a upřednostnění při dalším zápisu těch s nižší hodnotou.

3.2.4 Základní techniky řadiče pro zvýšení/udržení rychlosti

3.2.4.1 Zapojení flash paměťových čipů

Ačkoliv přístupová doba flash pamětí je oproti mechanickému pevnému disku výrazně nižší (~0,1ms/~5ms), rychlost sekvenčního čtení dat už tak jasně ve prospěch flash technologie nehovoří. Přesto komerční SSD disky současnosti dokáží dosáhnout přenosových rychlostí okolo 500MB/s. Prvním krokem k takovým rychlostem je paralelní zapojení více flash paměťových čipů do pole, které by mohlo být přirovnáno k RAID 0 zapojení klasických disků. Toto zapojení může mít i několik úrovní.

3.2.4.2 TRIM

Současné operační systémy, které jsou optimalizované pro práci s mechanickými pevnými disky u nichž se přepis z hlediska náročnosti operace ničím neliší od zápisu, způsobují výkonnostní problémy u SSD disků u nichž je operace přepisu značně náročnější. Pokud operační systém, nebo uživatel vymaže nějaká data, jsou informace o těchto datech odstraněny ze systému souborů, ovšem fyzicky zůstávají na disku až do chvíle, než dojde k jejich přepisu. To vede k tomu, že při typické operaci disku SSD jako je garbage collection, se zpracovávají stále i data, která už mohou být operačním systémem dávno označena za neplatná / permanentně vymazaná. S postupující dobou užívání disku má potom tento problém stále výraznější dopad na jeho rychlost (potažmo i životnost - dochází k

zápisu už nevalidních dat). Toto řeší příkaz TRIM, ten umožňuje označit vymazaná data na disku a ta nejsou dále zpracovávána při garbage collection. SATA příkaz TRIM musí podporovat operační systém (podpora byla zavedena u Windows 7, Linux kernel >= 2.6.33, Os X >= 10.6.8), dále jej musí samozřejmě podporovat i firmware disku. První generace SSD disku tento příkaz nepodporovaly a i některé následné komerční produkty nepodporují TRIM například v RAID 0 zapojení.

3.2.4.3 Over-provisioning

Je zřejmé, že stejně tak jako absence příkazu TRIM, může neblaze ovlivnit výkon SSD disku i přílišné zaplnění jeho kapacity. Proto disky nezpřístupňují svoji kompletní kapacitu operačnímu systému, ale nechávají si skryté místo jako jakousi rezervu. Jelikož mapování z logických adres na fyzické probíhá dynamicky, jsme schopni adresovat ve skutečnosti všechny bloky disku, ale nikoliv naplnit celou jeho kapacitu. Tento přístup má několik dalších výhod. Například v rámci wear leveling techniky je díky tomuto přístupu zátěž rozprostřena mezi větší množství buňek. Dále při mapování vadných bloků nedochází nutně ke snižování kapacity disku, vadné bloky se nahradí z rezervní kapacity. Při ceně současných flash pamětí je tato technika poměrně přístupným řešením.

3.2.5 Vybrané další techniky implementované do řadičů

3.2.5.1 Oprava chyb (ECC)

Mimo EZ-NAND čipy, které jsou schopny provádět ECC korekci chyb interně, existuje alternativa, která zpřístupňuje techniky opravy chyb i za použití klasických NAND flash pamětí. Při ukládání dat je v rámci bloku připravena redundantní kapacita pro uchování opravných kódů. Tyto kódy jsou pak čteny společně s daty a algoritmy provedení kontroly / opravy dat jsou již v režii řadiče. Tato funkčnost umožňuje provádění techniky bad-block mapping.

3.2.5.2 Cache

Většina řadičů SSD (až na SandForce řadiče) disponuje cache. Tu využívá jednak jako klasickou diskovou cache například při přednačítání dat atd., ale zároveň je v ní možné uchovávat po dobu běhu tabulku vadných bloků, adresovací tabulku aj. Disky s rozhraním PCI-E (OCZ revodrive) využívají místo vlastní cache systémovou paměť počítače.

3.2.5.3 Šifrování

Disk může být vybaven také interními šifrovacími algoritmy.

3.2.5.4 Komprese dat

Kompresní algoritmy, mají vliv na reálnou kapacitu paměťového média. V případě SSD disků dobře navržené kompresní algoritmy snižují zatížení rozhraní paměťového subsystému, snižují množství dat, která je nutné znovu zapisovat na jiné umístění při mazání bloků / přepisování stránek (viz 3.2.6.1 Write amplification).

3.2.6 Další specifika disků na bázi pamětí flash

3.2.6.1 Write amplification

Takzvané násobení zápisového zatížení vychází z faktu, že při přepisech, a mazání stránek se data dané buňky, která nejsou operací dočtena znovu zapisují na další, nové umístění na disku, aniž by o to operační systém žádal, jedná se o vlastnost této technologie a zároveň v kombinaci s jednou její stinnou stránkou - relativně krátkou životností flash pamětí - se jedná o nežádoucí efekt. I s ohledem na to, že při těchto operacích dochází k dalšímu (teoreticky nadbytečnému) zatěžování přenosového pásma paměťového subsystému. Každý z algoritmů a technik navržený pro SSD disky by měl brát tento fenomén v potaz a snažit se minimalizovat jeho výskyt.

Write amplification je měřitelná veličina, její hodnota je podíl počtu příkazů zápisu uvnitř paměťového subsystému a počet skutečných žádostí o zápis, vyvolaných hostitelským zařízením. Bez použití kompresních algoritmů je hodnota vždy větší nebo rovna jedné. Ovšem například řadiče SandForce používají pokročilé komprimační algoritmy a jejich typická hodnota write amplification se pohybuje okolo 0,5 [10].

Největší vliv na hodnotu této veličiny mají [10]:

- Efektivita algoritmů, provádějících garbage collection
- Over-provisioning - čím více prostoru je k dipozici, tím menší WA
- TRIM
- Zaplněnost kapacity jednotky

3.2.6.2 Zarovnání diskových oddílů

V současnosti pracuje logika disku nejčastěji s 4kB bloky, ovšem většina operačních systémů zarovnává své diskové oddíly po 512 bytech, to umožňuje vznik nezarovnaných diskových oddílů. Blok dat, který by se měl vlézt do jednoho fyzického bloku disku může potom fyzicky zasahovat i do dvou bloků. Tento fakt způsobuje nejen degradaci výkonu (místo jedné operace zápisu je nutné vykonat 2), ale také snížení životnosti. Mimo samotný operační systém, který zpravidla vytváří špatně zarovnané diskové oddíly může být na vině i RAID řadič apod. Nejjednodušší řešení spočívá v použití speciálního SW při vytváření oddílů, který oddíly zarovná korektně.

3.2.7 Komerční řadiče na trhu

Většina komerčních řešení se velmi rychle vyvíjela a většinou příchod nějaké nové technologie v rámci řadiče znamenal příchod nové generace SSD disků. Mezi nejvýznamnější výrobce patří SnadForce, Intel, Indilinx, Samsung, JMicron, Marvell. Nejúspěšnější jsou řadiče SandForce, které patří k nejrychlejším, jejich zvláštností je, že nepoužívají žádnou cache, jen malou integrovanou přímo v rámci procesoru řadiče. Každý řadič má několik kanálů, kterými jsou k němu flash čipy připojeny (RAID 0), většinou u menších kapacit (60GB) nejsou tyto kanály využity všechny a v porovnání se svými kapacitně většími variantami dosahují nižšího výkonu (nevyužívají plný potenciál řadiče) [1]. Obdobné problémy může způsobit i přechod na modernější výrobní procesy u NAND flash čipů, kdy vlivem vyšší míry miniaturizace může stačit k dosažení dané kapacity menší počet NAND flash čipů, což vede opět k osazování disků nižším počtem čipů a nevyužití všech kanálů a degradaci výkonu - viz OCZ Vertex 2 a přechod z 34nm paměťových čipů na 25nm [1].

3.3 Nejpoužívanější komunikační rozhraní

Pomocí tohoto rozhraní se disky připojují k hostitelskému systému. V případě SSD disků jejich nástup s sebou nepřinesl žádné nové rozhraní, ale pouze adoptovaly již dostupná řešení. Ovšem v některých případech využívají taková rozhraní, která nejsou pro klasické PC disky obvyklá (např. PCIe). Řadič komunikačního rozhranní je u běžných komerčních řešení implementován přímo v řadiči SSD disku.

Mezi nejrozšířenější rozhraní SSD disků patří tedy SATA (rev. 2, rev. 3), PCI Express, USB a v serverových řešeních SAS a Fibre Channel. Většina dodávaných SSD řadičů má integrován řadič SATA. Proto i řešení do PCIe (např. OCZ RevoDrive) se skládá vlastně ze 2 SSD SATA disků na kartě, zapojených do SATA RAID0, který je realizován samostatným SATA řadičem, ten je také integrován na dané kartě a následně připojen k řadiči sběrnice PCIe.

3.3.1 SATA

Jedná se o sériovou sběrnici, která od roku 2000 dospěla už do třetí revize. Jednotlivé revize se liší především svojí propustností (1,5Gb/s, 3Gb/s, 6Gb/s). Sběrnice je point-to-point, na každou linku je možné k hostitelskému zařízení připojit pouze jediné zařízení. Většina mechanických pevných disků si co do propustnosti stále vystačí pouze s první revizí SATA I. Zatímco nejnovější SSD disky již využívají propustnost, kterou poskytuje rozhraní SATA III. Fyzická podoba této sběrnice je zpravidla 7mi žilový kabel, kde 2 páry přenášejí diferenciálními signály data (in/out), zbylé tři linky jsou připojeny k zemi.

3.3.1.1 SATA Protokol

Sata protokol se skládá ze tří vrstev: fyzická, linková a transportní. Fyzická vrstva definuje standardy na úrovni vedení elektrických signálů, jejich úrovní, specifikace vodiče (kabelu). Fyzická vrstva má dále na starosti detekci připojených zařízení, inicializaci spojení, nastavení správných komunikačních parametrů (rychlost linky 1,5Gb/s, 3,0 Gb/s, 6,0Gb/s), synchronizaci zařízení a realizaci kódování 8b/10b (synchronizace).

Vyjednávání o použití určité rychlosti se nazývá OOB (Out Of Band Signaling), skládá se ze speciální sekvence signálů a primitiv zaslaných mezi hostem a připojeným zařízením viz **Chyba! Nenalezen zdroj odkazů.** [8].

Po inicializaci linky předává řízení fyzická vrstva vrstvě linkové (fyzická vrstva nadále realizuje pouze kódování). Linková vrstva je dále zodpovědná za příjem a odesílání paketů (FIS) po sběrnici. Tyto pakety obsahují buď řídící primitiva nebo data. Každý paket je opatřen hlavičkou, která určuje jeho typ a tělem, které nese data/parametry primitiv. SATA protokol umožňuje přenos maximálně 8192B v rámci datové části jednoho FIS (u SAS je to 1024B) [8].

3.3.1.2 SATA příkazy

SATA příkazy jsou shodné s rozhraním ATA/ATAPI (zpětná kompatibilita) a navíc SATA rozhraní definuje některé nové, které jsou specifické jen pro technologii SATA (celkem 4 a 3 příkazy ATA byly pro rozhraní SATA modifikovány).

Všechny ATA příkazy jsou inicializovány zápisem patřičných hodnot do sady 1-bajtových registrů zařízení (sada těchto registrů se nazývá Command block). Registr č. 7 je použit pro kód příkazu, který má být proveden a registry 1-6 slouží pro uchování parametrů příkazu - zápis parametrů musí předcházet zápisu příkazu, protože zápisem do registru 7 se zahajuje vykonávání příkazu.

3.3.2 USB

Další nejrozšířenější sběrnicí je sběrnice USB. Narozdíl od SATA, ktera by mohla být považována za výhradně interní počítačovou sběrnici (její externí modifikací je sběrnice eSATA), je USB určeno pro připojování externích/periferních zařízení. Protokol sběrnice je navržen s ohledem na univerzalitu a možnost použití pro velkou řadu rozmanitých zařízení od HID, přes tiskárny, až po mobilní úložná zařízení, kde mezi zcela nejrozšířenější dnes patří právě disk na bázi paměti flash.

Velmi důležitou a v době představení převratnou funkcionalitu představuje technologie Plug&Play, která umožňuje zařízení připojovat a odpojovat za běhu systému, bez nutnosti restartu

USB vzniklo v 90. letech minulého století jako náhrada sériového portu RS-232. USB za 16let svojí existence dospělo již do třetí revize:

- revize 1.0 (15.1.1996) - low-speed (1,5Mbit/s), full-speed (12Mbit/s)
- revize 1.1 (23.9.1998) - zpřesněna specifikace - první široce rozšířená revize
- revize 2.0 (27.4.2000) - přidává high-speed mód (480Mbit/s)
- revize 3.0 (17.11.2008) - nový standard - super-speed režim (5Gbit/s)

Hnacím motorem tohoto rozvoje je především vzrůstající potřeba vyšší propustnosti. USB 2.0 je revize, která měla umožnit především připojení multimediálních zařízení, vyžadujících vyšší než full-speed přenosové rychlosti (např. digitální kamery), jednalo se o reakci na sběrnici IEEE1394 od společnosti Apple, která byla přímo mířena pro připojení digitálních zařízení pro zachytávání obrazu. USB 3.0 zase reaguje na stále častější využití USB sběrnice pro připojování externích úložných disků, kde přenosová rychlost high-speed režimu nebyla dostačující.

Revize USB 3.0 navíc přináší změnu na fyzické úrovni. Místo 4 typických vodičů pro nižší revize USB (DATA+, DATA-, VCC+5V, GND), rozšiřuje tyto vodiče o dalších 5 vodičů, které realizují super-speed přenosový režim. V rámci USB 3.0 je tedy implementována duální sběrnice (2vodiče high/full-speed, 5 vodiče super-speed, 2 vodiče pro napájení). Zároveň ve specifikacích poklesl s USB 3.0 požadavek na nejnižší operační napětí (z 4,4V na 4V). USB 3.0 dále upravuje komunikaci a ustavuje asynchronní model namísto komunikace na bázi opakované výzvy (poll)

V roce 2010 bylo přidáno do specifikace využití USB portu pro nabíjení nekonfigurovaných zařízení - baterií. Které má umožnit odběr proud až 1,5A .

Sběrnice USB zachovává napříč svými revizemi zpětnou kompatibilitu na obou stranách, jak sběrnice tak i zařízení by měla být podle specifikací zpětně kompatibilní. Zařízení nativně operující v high-speed režimu by tedy měla být schopna v případě nutnosti fungovat i ve full-speed režimu. Analogicky funguje zpětná kompatibilita i pro super-speed zařízení.

Skutečná propustnost sběrnice záleží na spoustě faktorů. Některé plynou přímo ze specifikace a protokolu, kde žádnému zařízení není povoleno rezervovat plnou šířku pásma. U standardu USB 2.0, potažmo high-speed režimu je povoleno jedinému zařízení rezervovat maximálně 80% šířky pásma.

Dalším aspektem je režie samotného komunikačního protokolu mezi hostem a zařízením, kde podle představitele USB-IF Compliance programu (měření reálných veličin při použití USB zařízení, specifikace standardů a vydávání certifikací o splnění konkrétního USB standardu) z reálných měření plyne, že v závislosti na konkrétním užití se režie pohybuje v průměru mezi 10-15% [14]. Pro data je dále použito kódování 8/10(USB 3.0), které také snižuje hodnotu skutečně přenesených dat v rámci přiděleného pásma. Pro hrubý orientační výpočet doby trvání přenosu lze tedy sestavit vzorec:

$$T = \frac{X \cdot 8}{B \cdot U \cdot R}$$

Rovnice 1 : Teoretický, zjednodušený výpočet doby přenosu dat přes sběrnici USB

kde T je celkový čas datového transféru , X je velikost přenášených dat v MB, B je maximální přenosová rychlost celého pásma definovaná standardem, U je desetinné číslo vyjadřující maximální část šířky pásma, která je přidělitelná jedinému zařízení, R je desetinné číslo reprezentující skutečně přenášená data - tedy po odečtení režie. Pro high-speed režim za použití výše zmíněných hodnot dojdeme k závěru, že maximální přenosová rychlost skutečných dat se pohybuje někde pod hranicí 35-38MB/s. V reálném systému jsou dosažené hodnoty ještě nižší, jelikož zařízení často nedostane přiděleno ani 80% šířky pásma. V prostředí Windows Vista a novějších je možnost sledovat šířku přiděleného pásma konkrétnímu zařízení ve "správci zařízení" -> ve vlastnostech univerzálního hostitelského řadiče ->na karřtě "upřesnit".

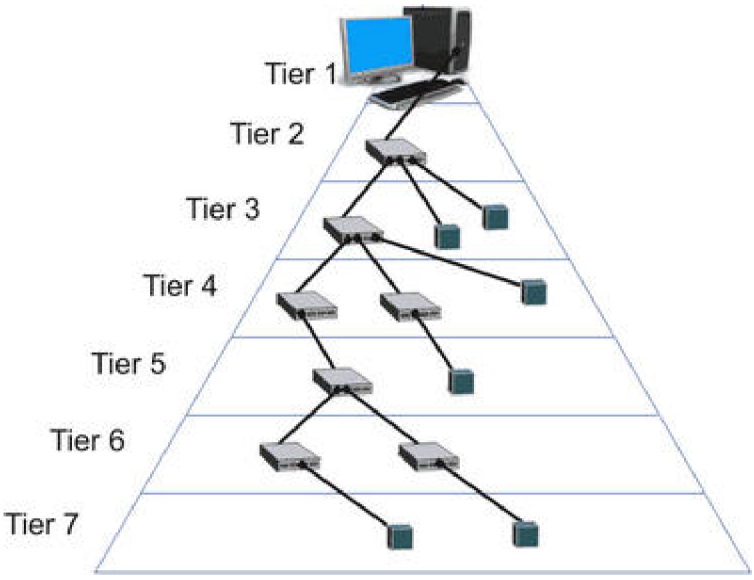
USB revize (mód)	USB 1.0 (full-speed)	USB 2.0 (high-speed)	USB 3.0 (super-speed)
Média (Data Size)	Teoretická doba přenosu		
MP3 / obrázek (4MB)	5.3sec	0,1s	0,1s
256MB Flash disk	5,7min	8,5s	0,8s
Obsah CD (700MB)	15,4min	23,1s	2,3s
Obsah DVD (4,7GB)	1,71hod	2,54min	15,6s
Blu ray (25GB)	9,3hod	13,9min	70s

Tabulka 1 - orientační přehled trvání přenosů klasických médií

Dalším důležitým omezením plynoucím ze specifikací je omezení délky kabelu, pro revizi 1.1 na 3m a pro revizi 2.0 na 5m. Revize 3.0 udává pouze maximální doporučenou délku, a to 3m. Pomocí rozbočovačů je možné vytvořit kabelové spojení až do 6-ti násobku uvedené maximální délky pro jeden kabel.

3.3.2.1 Topologie USB sběrnice

Celá sběrnice je paketová Master-Slave orientovaná. Veškerá aktivita tedy vychází z jednoho Master root USB portu, který má nad provozem sběrnice výhradní kontrolu a žádné jiné připojené zařízení není oprávněno po sběrnici vysílat bez předchozí výzvy. Celá topologie má potom stromový charakter, kde takzvané huby (rozbočovače) fungují jako uzlové body. Tento systém větvení je možné aplikovat i na více úrovních - lze řadit rozbočovače za sebe do série a tímto způsobem je možné připojit až 127 zařízení. Jelikož spousta zařízení se z USB také napájí, je součástí standardu také bod, který zakazuje, zapojování hubů, napájených pouze ze sběrnice do série.



Obrázek 9 : Topologie USB

Obrázek 10 znázorňuje možnosti zapojení zařízení a rozbočovačů na sběrnici USB. Master root HUB leží na úrovni 1 a většinou to bývá rozbočovač z nějž vedou všechny USB porty počítače. Celkem lze do série zapojit maximálně 6 rozbočovačů, což vytváří strom o 7mi patrech. Do revize 2.0 představoval USB hub více méně opakovač, který data ze svého upstream portu šířil do všech downstream portů. Revize 3.0 přináší unikátní adresování každého paketu a tedy každý paket je směrován pouze do příslušného downstream portu.

3.3.2.2 Napájení

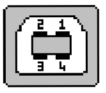
Každá ze zmíněných revizí, mimo rozdíl v rychlosti, garantuje i jiný minimální dodávaný proud. USB 2.0 garantuje 500mA, USB 3.0 garantuje až 900mA. Tyto hodnoty jsou garantované, prakticky poté záleží na konkrétním systému, kolik proudu je schopen dodat. Garantovaný proud je poté dělen na jednotky o velikosti 100mA - ať už pro potřeby dělení mezi jednotlivá zařízení, tak i pro potřeby ze sběrnice napájených rozbočovačů, kde k takovému hubu není umožněno připojit více než 4 další zařízení, jelikož každé zařízení dostane jednu proudovou jednotku (100mA USB2.0/ 150mA USB3.0) a jedna je ponechána pro samotný hub. Napájení zařízení přímo ze sběrnice je proces podřizující se striktně specifikacím a každé připojené zařízení je nuceno již při samotné inicializaci připojení deklarovat, zda má vlastní zdroj napájení a jaké jsou jeho nároky na napájení ze sběrnice. Systém je poté oprávněn zařízení, jehož napájecí nároky není schopen uspokojit, odmítnout.

3.3.2.3 USB konektory

USB 1.x a 2.0 dle standardu USB-IF poskytuje 5 různých konektorů (existují další proprietální mutace jako: Mini-4P od Sony, nebo Mini-A a Mini-AB, které se již nepoužívají) :

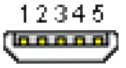
- Standard-A : nejběžnější typ u osobních počítačů

- Standard-B : běžný typ u periferních zařízení

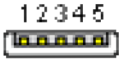


- Mini-B: nejběžnější typ u mobilních zařízení (MP3, fotoaparáty)

- Micro-B: nejčastěji používaný u mobilních telefonů, GPS, PDA



- Micro-A: viz Micro-B



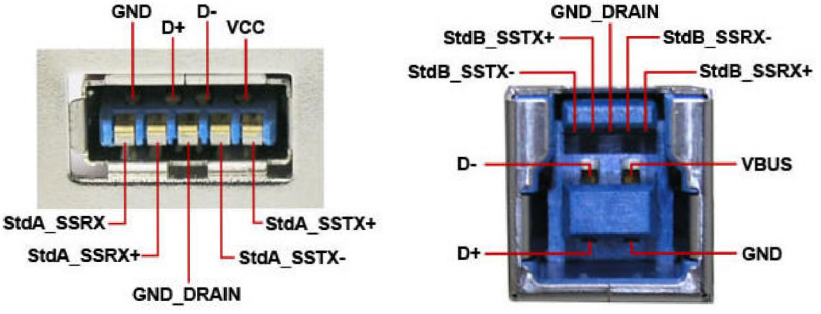
Pin	Název	Barva vodiče	Popis
1	VBUS	Červená	+5 V
2	D–	Bílá	Data –
3	D+	Zelená	Data +
4	GND	Černá	Zem

Tabulka 2 : Vodiče USB 1.x, 2.0 Standard-A a Standard-B [15]

Pin	Název	Barva vodiče	Popis
1	VBUS	Červená	+5 V
2	D–	Bílá	Data –
3	D+	Zelená	Data +
4	-	-	Nepoužitý
5	GND	Černá	Signal ground

Tabulka 3 : Vodiče USB 1.x, 2.0, mini a micro[15]

Konektory pro USB 3.0 jsou celkem 4: Standard-A, Standard-B, Powered-B (ma navíc 2 vodiče pro DPWR a DGND) a varianta mini.

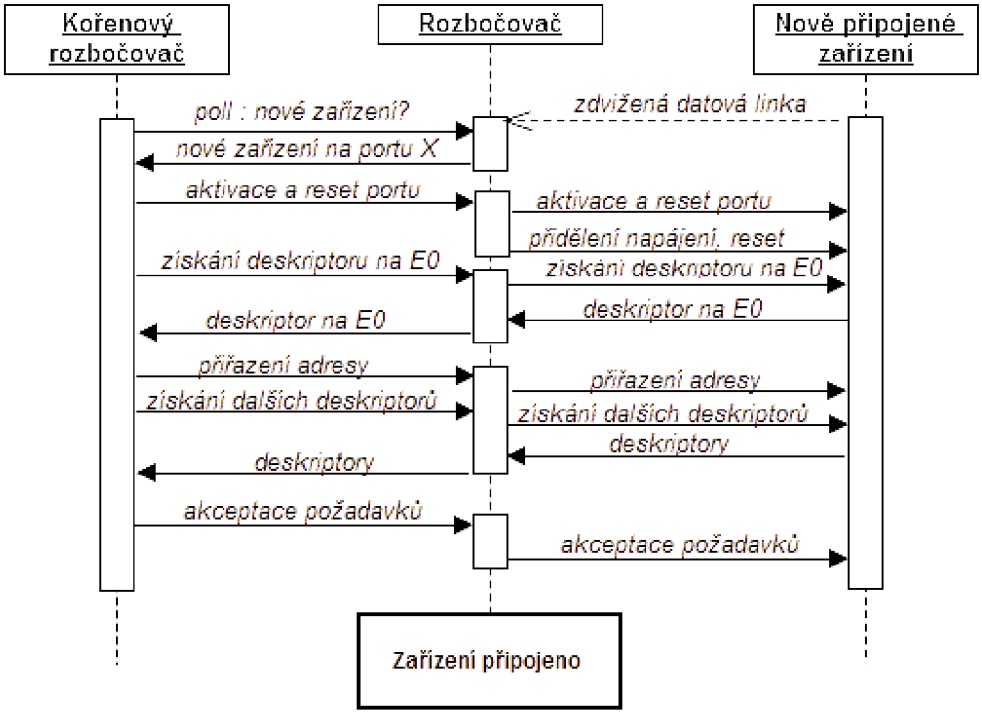


Obrázek 10 : vlevo: USB 3.0 standard-A, vpravo: USB 3.0 standard-B

3.3.2.4 Plug&Play

Je nesporná výhoda sběrnice USB, která v prvních letech její existence nebyla rozhodně standardem. Jedná se o mechanismus autokonfigurace zařízení systémem bez nutnosti zásahu uživatele (manuální konfigurace) a zároveň bez nutnosti restartování systému - sběrnice je schopná přijmout nové zařízení a změnit konfiguraci "za běhu". Nově připojené zařízení na sběrnici detekuje HUB podle zdvižené datové linky, na což navazuje sled těchto kroků:

1. Kořenový rozbočovač polluje rozbočovače na sběrnici s žádostí o hlášení nových zařízení. Rozbočovač, ke kterému bylo zařízení připojeno odpoví patřičně na takový požadavek kořenovému rozbočovači.
2. Po získání čísla portu nového zařízení kořenový rozbočovač tento port aktivuje a resetuje.
3. Hub, ke kterému je zařízení připojeno uvolní pro zařízení jeden díl napájení - 100mA a zašle zařízení signál reset o délce 10ms.
4. Než zařízení získá svoji adresu reaguje na první dotazy na adrese 0 (pouze Endpoint 0). Hostitel od zařízení v tomto provizorně ustaveném režimu získá základní konfigurační informace v podobě takzvaných deskriptorů - jedná se o deskriptory popisující komunikační schopnosti zařízení - řádově pár bytů.
5. Hostitelský hub přiřadí na základě získaných informací zařízení jeho adresu na sběrnici.
6. Na nové adrese získá hostitel od zařízení další popisovače, které specifikují mimo jiné i nároky zařízení na napájení ze sběrnice.
7. Hostitel ověří, zda má pro zařízení dostatečné prostředky (šířku pásma, napájení) a pokud ano, zařízení připojí. V opačném případě zůstane zařízení nepřipojeno.



Obrázek 11 : Zjednodušený šekvenční diagram procesu připojení nového zařízení na sběrnici

3.3.2.5 Protokol sběrnice USB

USB je sběrnice paketová, kde veškeré transakce iniciuje hlavní rozbočovač. Typickým sledem transakcí je:

1. Token paket - hlavička identifikující následující transakci, adresu koncového zařízení a endpoint
2. Datový paket - samotný obsah transakce
3. Status paket - potvrzení přijetí obsahu, případně zpráva o chybě

USB pakety se nejčastěji skládají z těchto segmentů [14]:

- **Sync** - tento segment obsahuje každý paket, jeho úkolem je synchronizovat hodiny vysílače a přijímače, 8bit pro low a full-speed, 32bit pro high-speed.
- **PID** - paket ID - identifikuje typ packetu, 8bit, kteér jsou tvořeny 4bitovým kódem + 4bitovým komplementem kódu.

TOKEN paket					**DATA paket**				
PID	0001	1001	0101	1101	PID	0011	1011	0111	1111
Popis	OUT Token	IN Token	SOF Token	SETUP Token	Popis	DATA0	DATA1	DATA2	MDATA
HANDSHAKE paket					**SPECIAL paket**				
PID	0010	1010	1110	0110	PID	1100	1100	1000	0100
Popis	ACK Handshake	NAK Handsake	STALL Handsake	NYET	Popis	PREamble	ERR	Split	Ping

Tabulka 4 : hodnoty PID

- **ADDR** - délka 7bit (127 adres zařízení), adresa 0 je rezervována pro nově připojené zařízení
- **ENDP** - 4bitový end-point field - specifikuje pro který endpoint je přenos určen (16 možných endpointů).
- **CRC** - kontrolní součet, token paket 5bit, data paket 16bit.
- **EOP -** konec paketu

Každé zařízení na USB sběrnici disponuje určitým počtem endpointů. Endpoint je zakončení komunikačního kanálu mezi rozbočovačem a zařízením. Zařízení při rozpoznání své adresy, provede požadovanou operaci (např. zápis, čtení) s bufferem endpointu, který je identifikován pomocí ENDP adresy. Každé zařízení musí disponovat endpointem 0, což je konfigurační endpoint a přes něj je zařízení inicializováno. Použití dalších endpointů je již záležitostí konkrétní aplikace.

Logické spojení mezi hostitelem a koncovým zařízením se nazývá roura (pipe), která má přiděleny parametry jako přidělená šířka pásma, typ přenosu (Control, Bulk, Iso, Interrupt), velikost paketů a směr datového toku. Rozeznáváme 2 typy rour: **Stream pipe** - nemá definovaný formát, může být řízena jak hostitelem, tak samotným zařízením, podporuje iso, bulk a přerušovací přenosy. **Message pipe** - má přesně stanovený formát, je řízena výhradně hostitelem a je schopna pouze kontrolního přenosu [14].

3.3.2.6 Druhy přenosů na sběrnici USB

Celkem rozeznáváme 4 druhy přenosů na sběrnici USB:

1. **Kontrolní přenos** - jedná se o přenos, který slouží nejčastěji pro přenos inicializačních dat připojeného zařízení - například přenos popisovačů zařízení. Druhým využitím tohoto

přenosu je přenos informací o stavu zařízení. Samotný přenos se skládá ze zaslání setup paketu a datového přenosu následovaného potvrzením o úspěšném vykonání transakce.

2. **Přenos přerušení** - jedná se žádost zařízení potřebujícího urgentní obsluhu ze strany hostitele. Vzhledem ke koncepci USB sběrnice není možné, aby zařízení svévolně vyslalo takovou žádost a musí čekat na poll hostitele a až na tuto výzvu zařízení odpoví svou žádostí. Tento druh přenosu má garantovanou latenci obsluhy a je jednosměrný.

3. **Izochronní přenos** - proudový přenos, garantuje dostupné přenosové pásmo, ale negarantuje doručení dat. Navrženo pro přenášení proudových dat u nichž převládá potřeba kontinuity datového přenosu nad její přesností - ideální pro přenášení převážně živého multimediálního obsahu.

4. **Bulk přenos** - jedná se o přenos, který garantuje správnost doručených dat, nikoliv ovšem latenci jejich doručení. Disponuje mechanizmy pro odhalení chyb a jejich opravy (opakování přenosu). Na sběrnici je tomuto druhu přenosu vyhrazeno přenosové pásmo až po alokaci všech pásem ostatních.

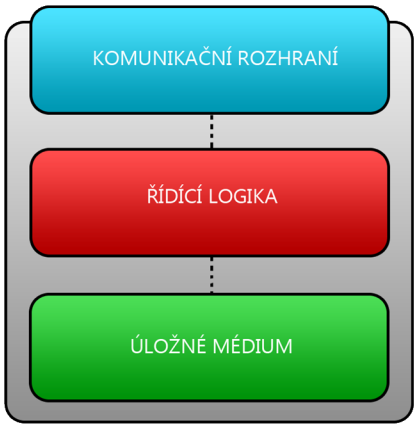
3.3.2.7 Další vývoj sběrnice USB

Vývoj sběrnice USB se v současnosti odklonil od zvyšování propustnosti a místo toho usiluje o zlepšení mobility v podobě takzvaného Wireless (bezdrátového) USB. V roce 2010 již byla standardizována revize 1.1 tohoto rozhraní. Hlavním rozdílem oproti "drátové" sběrnici USB je existence pouze jediného - kořenového - rozbočovače. Jelikož se zařízení připojují bezdrátově, odpadá nutnost použití sekundárních propojovacích mezičlánků. Výjimkou je pouze koncept zpětné kompatibility, který má umožnit připojení stávajících USB zařízení k bezdrátové sběrnici. Pro tento účel existují 2 mezičlánky označované jako HWA (Host wireless adapter) a DWA (Device wireless adapter). HWA se na straně hostitele připojuje do standardního USB portu a slouží jako bezdrátový rozbočovač pro Wireless USB zařízení. DWA se naopak připojuje k USB zařízením, která nejsou nativně Wireless USB kompatibilní a slouží jako bezdrátový adaptér.

4 Koncepce vlastního disku s technologií flash

Cílem celé práce je na dříve zmíněných teoretických základech postavit návrh vlastního disku na bázi paměti flash, který by byl připojitelný k osobnímu počítači a zvládal základní diskové operace. V této praktické části je zejména důležité zvolit vhodnou platformu pro realizaci, která nejen, že umožňuje vytvoření daného zařízení, ale je také dostupná v rozumném množství (myšleno "kusy") a čase, a jejíž potenciál by umožnil v návaznosti na tuto práci postavit plnohodnotné funkční zařízení, odpovídající běžným potřebám.

Základní koncept zařízení prakticky kopíruje koncept běžných komerčních řešení jako jsou SSD disky nebo fash disky. Celý disk by se dal rozdělit na 3 hlavní funkční celky, realizující určitou specifickou činnost. Základem každého datového úložiště, tedy i tohoto disku, je úložné médium - v tomto případě se jedná výhradně o paměti NAND flash. Toto médium je nutné řídit specificky navrženou logikou, která má absolutní kontrolu nad úložným médiem. A posledním článkem je komunikační rozhraní, které připojuje disk k externí sběrnici a zajišťuje komunikaci s hostitelským systémem.



Obrázek 12 : Základní koncept vyvíjeného disku

Volba vhodné technologie pro každý funkční celek musí předcházet fázi tvorby detailnějšího návrhu zařízení a dala by se rozdělit na dva zásadní podproblémy:

1. Prvním z nich je volba vhodného komunikačního rozhraní, které bude zajišťovat propojení mezi osobním počítačem a samotným zařízením.

2. Druhým je výběr technických prostředků pro vývoj řídící logiky, která zpracovává příkazy přijaté pomocí komunikačního rozhraní a realizuje všechny potřebné funkce pro práci s paměťovým subsystémem.

Samostatným problémem potom zůstává volba pamětí flash, kde se můžeme opřít o poměrně podrobný teoretický základ části 3.1.

4.1 Volba vhodného externího rozhraní

S odkazem na kapitolu 3.3 lze pro připojení flash disku k osobnímu počítači uvažovat sběrnice PCIe, SATA a USB. Každá ze zmíněných má nesporné klady, ale často také mnoho nevýhod, které jsou především pro kusový vývoj prototypu s omezenými možnostmi v některých případech téměř nepřekonatelné.

4.1.1 Možnosti sběrnice PCIe

Sběrnice PCIe je určitě zajímavou a nevšední možností, poskytuje výhodu především ve své rychlosti. Komerčně vyráběná zařízení pro tuto platformu jsou zaměřena tedy právě především na rychlost a pokud by hlavním cílem práce mělo být vytvoření disku s absolutním důrazem na rychlost, byla by tato sběrnice vhodný kandidát.

Avšak už při bližší analýze vyráběných komerčních zařízení je zřejmé, že se jedná o poměrně inovativní směr. Fakticky neexistuje žádný běžně dostupný produkt, který by byl nativním PCIe diskem. Většina řešení na bázi paměti flash integruje převodník z PCIe na SATA (např. u OCZ Revo drive) a tedy se nejedná o nativní PCIe disk.

Důvodů pro takovéto řešení najdeme hned několik. První z nich bude rozhodně levnější vývoj i výroba - hardware pro sběrnici SATA je již vyvinut, SSD řadiče pro SATA rozhraní vyrábí hned několik výrobců a SATA rozhraní je tedy již dobře otestované a v nejbližších letech po něm bude určitě větší poptávka, než po nativním PCIe řešení. Nikomu se tdy nevyplatí tedy kvůli okrajovému segmentu s poptávkou po vysoké přenosové rychlosti do této technologie investovat. Tento trend je potvrzen již z historie, kdy můžeme nalézt mnoho snah vyvíjet rychlé disky pro interní sběrnici (např. Gigabyte RAMdisk), které si ovšem nikdy nenašly cestu k masovějšímu nasazení.

Pro tuto práci se tedy tato sběrnice nehodí především pro nedostatečnou teoretickou oporu v podobě běžně vyráběných zařízení, neexistuje ani žádný vyráběný HW, který mohl tuto implementaci usnadnit. Veškerou práci by musel vykonat generický programovatelný hardware, pravděpodobně FPGA, které má jistá omezení v počtu využitých PCIe linek, jelikož uvažované, dostupné modely nejsou schopny výkonem obsloužit více než 4-8linek - tím se částečně degraduje jediná výhoda PCIe - rychlost. Dalším faktem je, že v běžných operačních systémech pochopitelně není přítomen žádný generický ovladač, který by podporoval disk připojený přes toto rozhraní - není pro tento účel navrženo.

4.1.2 Možnosti sběrnice SATA

SATA se dlouhou dobu jevil jako ideální kandidát. Jedná se o sběrnici primárně navrženou pro připojení úložných zařízení k osobnímu počítači, je bezesporu nejrozšířenější platformou pro danou aplikaci současnosti z čehož plyne i rozsáhlá ovladačová podpora napříč všemi běžně užívanými operačními systémy. Její rychlost je zcela dostačující pro běžné aplikace a s novou revizí SATA3 představuje vhodnou sběrnici i pro nejrychlejší SSD disky současnosti.

I přes její rozšíření nejsou teoretické podklady potřebné pro vývoj zařízení běžně sehnatelné v kompletní a komplexní podobě. Samotná specifikace standardu SATA, která je spravovaná sata-io není volně dostupná, ale pouze za poplatek, který má úsměvnou hodnotu pro velké výrobce a znepříjemňuje tak pouze přístup k informacím případným zájemcům z řad širší veřejnosti.

Zcela nepřekonatelným problémem se potom ovšem stává dostupnost hardwaru, který by zjednodušil vývoj koncových zařízení pro tuto sběrnici. Nejschůdnější by byla možnost použití samostatného integrovaného obvodu, který by realizoval funkci SATA device controlleru (řadiče

zařízení). Ovšem takových obvodů na trhu není mnoho, většina integrovaných SATA řadičů jsou hostitelské řadiče, určené pro připojení běžných SATA disků k vestavěným systémům. Integrované SATA řadiče pro koncová zařízení se buď dodávají pouze ve velkých kvantitách pro výrobní linky, nebo je silně nevyhovující jejich dostupnost, dokumentace jejich užití atp.

Alternativou jak využít rozhraní SATA je implementovat vlastní řadič v programovatelném obvodu FPGA, případně využít některý z komerčně nabízených řešení v podobě soft IP core řešení, které by realizovalo funkci SATA řadiče za použití FPGA. Takové řešení s sebou přináší i spoustu nevýhod. Nutnost provozovat na jednom FPGA jak třetí stranou dodaný řadič SATA zařízení, tak vlastní řešení implementující řadič pamětí flash s sebou přináší riziko těžko laditelných chyb způsobených časováním apod., dále by do jisté míry prodražilo vývoj, jelikož v rámci implementace SATA řadiče v FPGA jsou kladeny velké nároky na I/O rozhraní FPGA, které by realizovalo nejnižší, fyzickou vrstvu protokolu SATA.

Vzhledem k nedostupnosti kusového množství potřebného hardware v rozumném čase a úskalím plynoucím z použití alternativy v podobě soft IP core je bohužel použití SATA sběrnice nevhodnou a neefektivní volbou pro realizaci tohoto projektu.

4.1.3 Možnosti sběrnice USB

USB sběrnice je taktéž jednou z nejrozšířenějších sběrnic vůbec. Pro tuto sběrnici existuje nepřeberné množství zařízení, je výborně a hlavně dostupně zdokumentovaná. Její protokol sice nepatří k nejtriviálnějším, což je daň za míru univerzality, kterou tato sběrnice nabízí. Umožňuje připojit k osobnímu počítači prakticky jakoukoliv periferii, včetně úložného zařízení. Hardware pro vývoj zařízení na této platformě je nejdostupnější ze všech zmiňovaných možností. A pro vývoj zařízení nabízí v kontrastu s rozhraním SATA hned několik integrovaných řadičů. Mírnou slabinou této sběrnice při použití pro připojení úložných zařízení, je její rychlost. Revize 2.0 dosahuje dle kapitoly 3.3.2 přibližně 35MB/s v ideálních podmínkách, což může být pro externí disk limitující. Revize 3.0 nabízí až 10x vyšší propustnost v porovnání s USB 2.0, ovšem dostupnost integrovaných řadičů pro tuto revizi je zatím omezena - prakticky na produkt firmy Cypress (CYUSB3011), který je na trhu teprve od dubna 2011. Vzhledem k zamýšleným vlastnostem výsledného zařízení, kde rychlost nepředstavuje rozhodující kritérium lze tedy uvažovat i o revizi 2.0.

Vzhledem k popsaným vlastnostem zástupců jednotlivých sběrnic byla sběrnice USB shledána jako nejvhodnější kandidát z hlediska efektivity práce, dostupnosti, ale také množství přístupných teoretických podkladů.

4.2 Volba technologie řídící logiky

Jednoznačným kandidátem pro realizaci řídící logiky je programovatelný FPGA obvod. Takové řešení poskytuje dostatek volnosti při vývoji řadiče paměťového subsystému, je flexibilní i s ohledem na případná budoucí rozšíření zařízení a poskytuje dostatek výkonu pro danou aplikaci a disponuje rozumnými možnostmi pro připojení obvodu, který realizuje řadič externí sběrnice.

4.2.1 FPGA

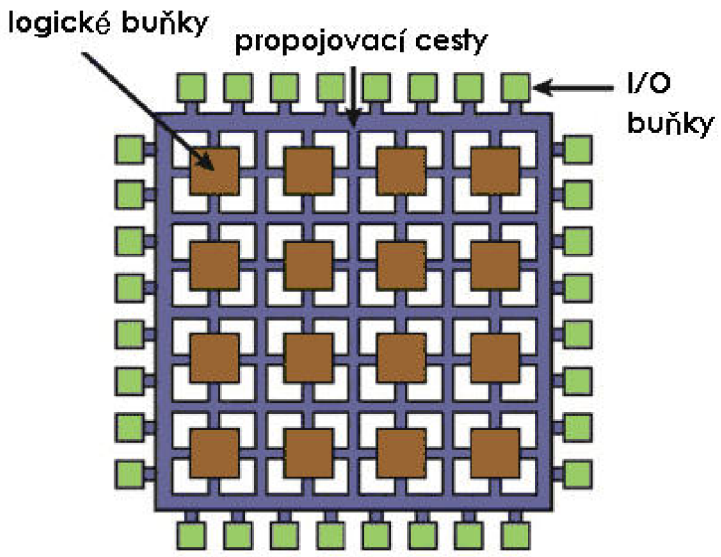
Následující podkapitola vychází z [16].

FPGA je zkratka pro field-programmable-gate-array, přeloženo do češtiny hradlové pole. Jedná se o polovodičové zařízení, které je možné konfigurovat k provádění logických funkcí. Klasické FPGA je tvořeno na okrajích I/O bloky a uvnitř logickými bloky (viz Obrázek 13 : Architektura FPGA), které mohou provádět jak komplexní logické funkce, tak prosté funkce jako AND nebo OR, bloky jsou propojeny propojovacími cestami. Dalšími přítomnými prvky jsou moduly RAM, násobičky atd. Mezi I/O bloky mohou být také zpožďovací linky na násobení frekvence atd. Velikost FPGA se udává počtem těchto logických bloků, například (10x10).

Architektura od firmy Xilinx má většinou takovou koncepci, že každá buňka (SLICE) obsahuje dvě LUT (Look Up Table), do níž můžeme vložit jakoukoliv čtyřvstupovou logickou funkci, a dva klopné obvody, pro každou LUT jeden.

Nejčastěji se technologie FPGA používají pro synchronní design, jen v případě velmi jednoduchých aplikací se používá pro design asynchronní. Tato technologie je nasazována téměř na jakoukoliv aplikaci, od realizace DSP počínaje, přes obranné a zdravotnické systémy, až po simulaci vyvíjených HW návrhů.

Návrh designu pro FPGA bývá vytvořen buď v nějakém HDL (VHDL / Verilog) jazyce nebo jako schématický diagram. Nástroje pro zpracování těchto návrhů vytvářejí takzvaný netlist (technologickou mapu pro specifické FPGA), na jehož základě je možné ověřit správnou funkčnost návrhu, z něj je poté vytvořen binární soubor, který je možné nahrát většinou pomocí software dodávaného výrobcem do FPGA zařízení.



Obrázek 13 : Architektura FPGA

4.2.1.1 Výběr výrobce a modelu

Mezi přední současné výrobce FPGA obvodů patří firma Xilinx společně s firmou Altera. Nabídka firmy Xilins obsahuje 2 rodiny FPGA čipů, Virtex a Spartan (od 7mé generace byla rodina čipů Spartan rozdělena na Atrix a Kintex), tyto FPGA jsou poměrně běžně dostupné, byly tedy jasnými kandidáty.

Virtex
je rodina navržená především pro výkon a náročné aplikace. Mimo programovatelné logiky disponuje širokou škálou aplikačně specifických obvodů jako jsou například FIFO a ECC obvody, DSP bloky, řadič PCIe, Ethernetu a další. Modelová roadmapa této rodiny obsahuje i obvody ve speciálních pouzdrech odolnějších například proti radiaci, které umožňují použití například ve vesmírných programech. Poslední řada Virtex-7 je vyrobena 28nm výrobním procesem a podle čísel výrobce má poskytnout až o 50% vyšší výkon než předchozí 40nm řada Virtex-6. Tyto komplexní obvody s důrazem na rychlost a spolehlivost jsou pro účely tohoto projektu zbytečně naddimenzované a drahé.

Kintex
je jakousi střední cestou, založena na architektuře rodiny Virtex, ovšem vzhledem k integraci menšího počtu aplikačně specifických obvodů a také menší velikosti samotného pole logických buněk se jedná o energeticky i nákladově úspornější řešení.

Artix
je rodinou určenou pro nenáročné aplikace. Jeho nižší výkon v porovnání s ostatními FPGA Xilinx 7mé generace dělá z tohoto čipu energeticky šetrné zařízení. S jeho nasazením se i díky poměrně příznivé ceně počítá ve spotřební elektronice, především potom v mobilních zařízeních napájených za baterie. Bohužel jak Kintex, tak Artix nejsou zatím běžně dostupná řešení (Q1 2012).

Spartan
je rodina určena pro méně náročné aplikace a poskytuje zajímavé řešení s ohledem na cenu těchto čipů. Jednotlivé modely této rodiny se liší především svojí velikostí a tím, zda disponují GTP trasceivery (LXT) nebo nikoliv (LX). Tato rodina je ze zmíněných nejvhodnějším kandidátem pro danou aplikaci.

4.2.2 Xilinx Spartan6

Je FPGA vyrobené 45nm výrobním procesem určené pro mainstreamové použití. Velikost v počtu logických buněk se pohybuje od 3480 až po 147443. LUT mají 6 vstupů a 2 registry. RAM bloky disponuují velikostí 18Kb (2x9Kb). Více informací viz specifikace Xilinx Spartan 6 [17].

4.3 Výběr flash pamětí

Na základě kapitoly 3.1 je pro tento druh aplikace samozřejmostí užití pamětí typu NAND flash, které svojí kapacitou dosahují rozumné velikosti řádově několik jednotek GB. Dalším požadavkem je standardizované rozhranní, alespoň asynchronní ONFI 1.0.

Při bližším prozkoumání katalogů běžných prodejců jako arrow.com, avnet.cz nebo digi-key.com se opět setkáváme s problémem dostupnosti HW v kusovém množství a zároveň v rozumném čase. Při požadavku na kapacitu čipů alespoň 8GB (64Gb) se výběr zůžil prakticky na jediný model dostupný v digi-key.com a to *Micron MT29F64G08CAAA*. I v základním návrhu je vhodné použít minimálně 2 paměťové čipy, jelikož se tím snižuje možnost kritických komplikací v případě, že jedna paměť nebude funkční. Dalším důvodem je snadná možnost rozšíření funkčnosti disku o jednoduchý RAID na úrovni řadiče paměťového média.

4.4 Návrh disku dle zvolených technologií

Zvolené technologie nabízí několik možností, jak dále pokračovat ve vývoji zařízení.

První z nich je použití jednotlivých funkčních článků (USB řadič zařízení, FPGA Xilinx Spartan-6 a Micron MT29G64G08CAAA) a navržení kompletního schématu, které by bylo na míru postaveno dané aplikaci a následný návrh desky plošných spojů, její výroba a osazení. Tento postup nechává velký prostor pro vznik chyb, jejichž náprava je časově velmi náročná. Oprava chyby při vývoji PCB může ve spojení s výrobou PCB znamenat až několikatýdenní zpoždění a samostatné odhalení některých nedostatků nemusí být vždy přímočaré.

Druhou variantou je použití některého z vývojových kitů, který je postaven na zvolených technologiích a umožňuje vývoj požadovaného zařízení. Jelikož jak sběrnice USB, tak FPGA Xilinx Spartan-6 jsou na trhu již nějakou dobu, existuje několik dostupných řešení založených na těchto technologiích. Výhodou takového přístupu je otestovaný funkční hardwarový základ, který urychlí do jisté míry počátky vývoje a nabídne technickou podporu ze strany výrobce. Jako další plus lze vnímat to, že většina takovýchto kitů je osazena i dalšími technologiemi, jako je například paměť RAM, možnost připojení non-volatilní paměti z které je možné při inicializaci zařízení FPGA automaticky naprogramovat a další, které lze využít pro případné budoucí rozšíření zařízení.

Na základě těchto faktů byla zvolena možnost postavení disku na některém z vývojových USB-FPGA kitů.

4.4.1 USB-FPGA kit

USB-FPGA osazenými různými FPGA jak značky Xilinx tak Altera existuje na trhu poměrně mnoho. Při zúžení výběru na ty, které disponují FPGA Spartan-6 při zohlednění všech dalších požadavků řešení německé firmy ztex (http://ztex.de). Mimo FPGA Spartan-6, jsou kity vybavovány i mikrokontrolerem *Cypress CY7C68013A EZ-USB FX2*, který realizuje funkci USB řadiče USB2.0. Výrobce zatím nenabízí kity s USB řadičem například *Cypress CYUSB3014*, které jsou kompatibilní se sběrnicí USB 3.0. Dle vyjádření výrobce se tato zažízení chystají na čtvrtý kvartál roku 2012. Tento fakt ovšem nepředstavuje problém, jelikož zařízení je navrhováno s důrazem na návrh funkční architektury, nikoliv na rychlost, lze tedy předpokládat, že užití sběrnice USB 2.0 se nestane úzkým hrdlem celého zařízení.

Sortiment kitů ztex je poměrně bohatý včetně několika rozšiřujících modulů. Firma ke svým kitům poskytuje i velice základní SDK balíček pro vývoj firmwaru pro USB řadič.

Pro aplikaci plně vyhovují 2 verze kitů:

1. ZTEX 1.11c FPGA Spartan-6 25LX, USB řadič EZ-USB FX2 (*Cypress CY7C68013A*), dále také 64MB DDR SDRAM, microSD socket pro načtení FPGA bitstreamu pomocí firmware.

2. ZTEX 1.15a - FPGA Spartan-6 45LX, USB řadič EZ-USB FX2 (*Cypress CY7C68013A*), dále také 128MB DDR2 SDRAM, microSD socket pro načtení FPGA bitstreamu pomocí firmware.

ZTEX 1.11c by ve všech aspektech byl bezpochyby dostačující platformou, ovšem s ohledem na budoucí rozšiřitelnost je nutné brát v potaz i počet FPGA GPIO vývodů. Běžné rozhraní flash pamětí má dle normy ONFI 17 vývodů z toho jsou 4 napájecí, tedy připojení každá flash paměť potřebuje na propojení s FPGA 13 vodičů. ZTEX 1.15a disponuje 91 FPGA GPIO vývody, tedy k tomuto kitu lze

připojit až 7 pamětí. S ohledem právě na počet vyvedených IO pinů byl zvolen kit ZTEX USB-FPGA kit 1.15a.

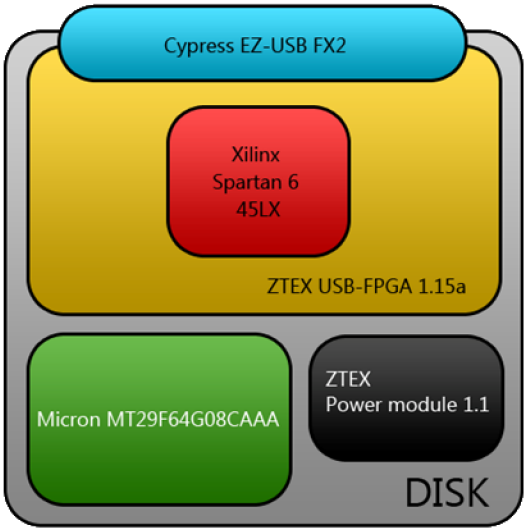
4.4.2 Základní návrh architektury

Na základě zvolených komponent je možné navrhnout základní architekturu celého zařízení. S návrhem souvisí i problematika napájení, jelikož vybraný kit je nutné napájet externě, případně vytvořit obvod (dle referenčního designu), který by napájel zařízení ze sběrnice USB. Sběrnice USB 2.0 garantuje proud 500mA. Teoretický odhad na horní mez spotřeby vyvíjeného zařízení počítá s těmito hodnotami, které jsou udávány výrobci zařízení:

- Flash paměť: max **50mA** každý čip
- USB řadič *CY7C68013A*: max **85mA**
- Xilinx Spartan-6: dle odhadu estimačních nástrojů výrobce by se měla spotřeba pohybovat mezi **200-500mA**

Součet těchto hodnot vede k jednoznačnému závěru, že bude vhodné použít externí zdroj napájení, minimálně s ohledem na následnou rozšiřitelnost zařízení. Firma ZTEX vyrábí ke svým kitům i napájecí modul, který je možné připojit na jakýkoliv stabilní zdroj o napětí 4-16V.

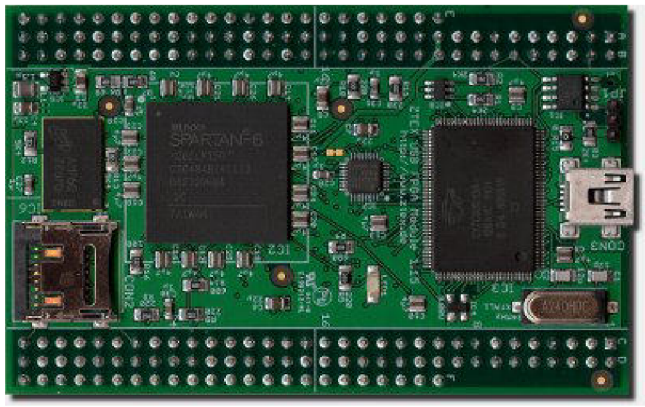
Disk tedy sestává ze 3 samostatných bloků (jak znázorňuje Obrázek 14 : Blokový diagram architektury disku) a to řídící modul v podobě USB-FPGA kitu, paměťový modul a napájecí modul.



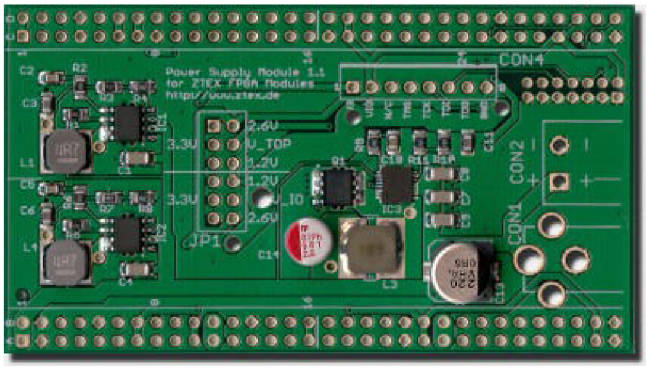
Obrázek 14 : Blokový diagram architektury disku

4.4.3 Fyzická podoba navrhovaného zařízení

Dva z modulů jsou komerčně dostupné výrobky firmy ZTEX, paměťový modul je navržen jako samostatný modul představovaný samostatně navrženou deskou tištěných spojů, kdy způsob jeho propojení s dalšími moduly se zaměřuje především na kompaktnost celého výsledného zařízení.

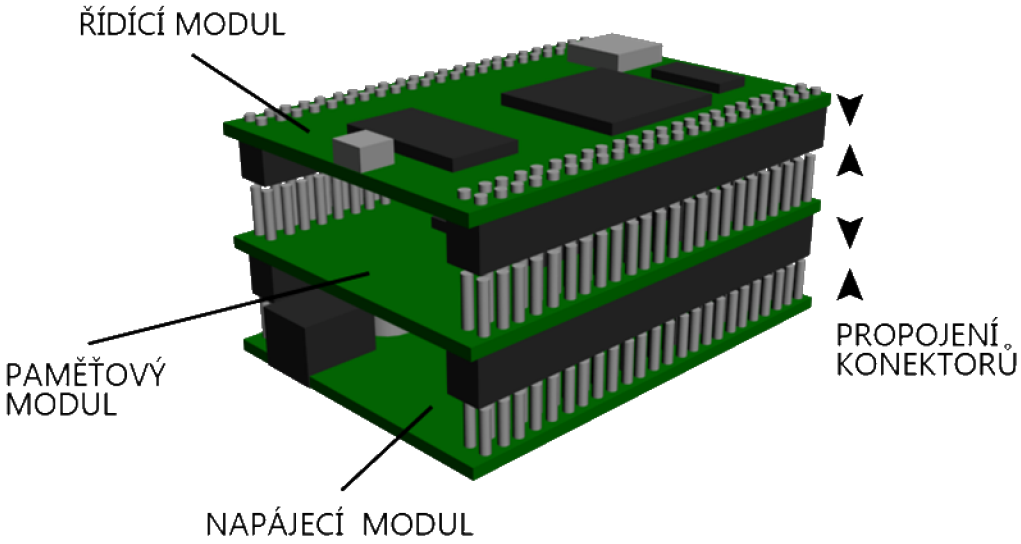


Obrázek 15 : ZTEX USB-FPGA 1.15a - Řídící modul



Obrázek 16 : ZTEX Power module - napájecí modul

Koncepce fyzického propojení všech modulů vychází z designu samotného kitu a aplikovaného vertikálního propojení mezi napájecím modulem a USB-FPGA modulem. Paměťový modul je vložen mezi oba moduly ZTEX. K propojení slouží speciální pin-header konektory se samičím i samčím zakončením.



Obrázek 17 : Model fyzického zařízení a způsob propojení jednotlivých modulů

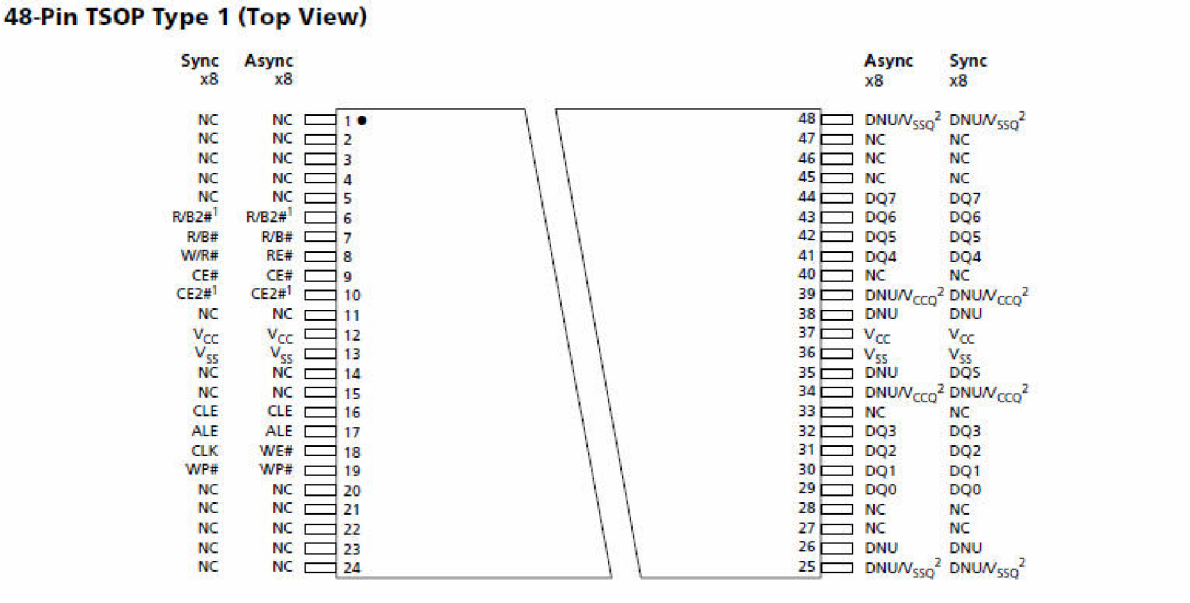
5 Paměťový modul

Paměťový modul nese v základním návrhu 2 čipy flash pamětí Micron MT29F64G08CAAA v pouzdře TSOP-48 (piny umístěny po stranách čipu). Jedná se o desku plošných spojů, která realizuje patřičné propojení mezi samotnými flash čipy a konektorem vedoucím k řídícímu modulu.

5.1 Schéma zapojení

Pro návrh schématu zapojení a vytvoření podkladů pro vývoj PCB byla použita zkušební verze CAD návrhového prostředí EAGLE [http://cadsoftusa.com] ve verzi 6. Zdrojové soubory jsou k dispozici na přiloženém CD. Data vytvořená ve verzi 6 nejsou zpětně kompatibilní s nižšími verzemi softwaru. Před samotným návrhem schématu je nutné věnovat pozornost rozmístění jednotlivých pinů na pouzdře paměťových čipů a vývodům FPGA na jednotlivé IO piny řídícího modulu.

5.1.1 Mapa výstupních pinů MT29F64G08CAAA (TSOP-48)



Obrázek 18 : Rozmístění výstupů flash paměti

- NC - no connection
- DQ 0 - DQ 7 - poloduplexní datová, příkazová a adresová sběrnice
- VCC - vstupní napájení 3.3V
- VSS - zem
- ALE - pokud je nastaven, načte adresu z DQx sběrnice do adresového registru
- CLE - pokud je nastaven, načte z DQx sběrnice kód příkazu
- R/B# - read/busy signál
- DNU - do not use
- CE# - chip enable - zapínání/vypínání logických jednotek (použitá flash paměť má pouze jedinou funkční jednotku)
- WE# - povolení zápisu při použití asynchronního rozhraní

Poznámka: uvedené signály mají popsanou funkci pouze při použití asynchronní komunikace, při užití synchronní komunikace dle standardu ONFI 2.0, který použitý čip splňuje je nutné význam a užití signálů zkontrolovat v manuálu k paměti flash (je součástí přiloženého CD).

5.1.2 Mapa pinů ZTEX USB-FPGA 1.15a

	E	A	B	
1		4..35V	4..35V	1
2		GND	GND	2
3		INT4	T0	3
4		T1	T2	4
5		BKPT	GND	5
6		SCL	SDA	6
7		3.3V	3.3V	7
8		TxD0	RxD0	8
9		A20~IO_L16N_1	A19~IO_L16P_1	9
10	1.2V	2.5V	2.5V	10
11	1.2V	1.2V	1.2V	11
12	VREF	A18~IO_L66N_SCP0_0	B18~IO_L66P_SCP1_0	12
13	C16~IO_L65N_SCP2_0	D17~IO_L65P_SCP3_0	GND	13
14	B16~IO_L63P_SCP7_0	A17~IO_L64N_SCP4_0	C17~IO_L64P_SCP5_0	14
15	A16~IO_L63N_SCP6_0	C14~IO_L46N_0	D15~IO_L46P_0	15
16	B12~IO_L36P_GCLK15_0	3.3V	3.3V	16
17	A12~IO_L36N_GCLK14_0	A11~IO_L35N_GCLK16_0	C11~IO_L35P_GCLK17_0	17
18	B14~IO_L50P_0	C13~IO_L48P_0	A13~IO_L48N_0	18
19	A14~IO_L50N_0	C12~IO_L37N_GCLK12_0	D11~IO_L37P_GCLK13_0	19
20	D12~IO_L47N_0_NC45	C15~IO_L62P_0	F10~IO_L38P_0	20
21	D13~IO_L47P_0_NC45	GND	GND	21
22	A10~IO_L34N_GCLK18_0	VCCO_IO	VCCO_IO	22
23	B10~IO_L34P_GCLK19_0	1.2V	1.2V	23
24	C9~IO_L8P_0	C10~IO_L33N_0	D10~IO_L33P_0	24
25	C7~IO_L5P_0	D8~IO_L32N_0	D9~IO_L32P_0	25
26	A7~IO_L5N_0	A8~IO_L6N_0	B8~IO_L6P_0	26
27	B6~IO_L4P_0	C8~IO_L7N_0	D7~IO_L7P_0	27
28	VREF	C6~IO_L3N_0	D6~IO_L3P_0	28
29	3.3V	A5~IO_L2N_0	C5~IO_L2P_0	29
30	3.3V	B3~IO_L1P_HSWAPEN_0	A6~IO_L4N_0	30
31	GND	VCCO_IO	VCCO_IO	31
32	GND	GND	GND	32

Tabulka 5 : ZTEX USB-FPGA 1.15a pinlayout (1/2)

	C	D	F	
1	USB_5V	USB_5V		1
2	5V (unused)	5V (unused)		2
3	L20~IO_L43P_GCLK5_M1DQ4_1	5V (unused)		3
4	GND	GND		4
5	INT5#	WAKEUP*		5
6	3.3V	3.3V		6
7				7
8	GND	V20~IO_L71N_1		8
9	PE7/GPIFADR8	Y22~IO_L59N_1		9
10	PE6/T2EX	AA22~IO_L63N_1	AA21~IO_L63P_1	10
11	PE5/INT6	Y21~IO_L59P_1	AB21~IO_L61N_1	11
12	PE4/RXD1OUT	W20~IO_L53P_1	Y20~IO_L67N_1	12
13	PE3/RXD0OUT	AA20~IO_L61P_1	AB20~IO_L65N_1	13
14	PE2/T2OUT	V19~IO_L71P_1	AB19~IO_L65P_1	14
15	PE1/T1OUT	Y19~IO_L67P_1	AB18~IO_L2N_CMPMOSI_2	15
16	PE0/T0OUT	V18~IO_L73N_1	AA18~IO_L2P_CMPCLK_2	16
17	GND	Y15~IO_L5P_2	AA16~IO_L4P_2	17
18	2.5V	2.5V	AB15~IO_L5N_2	18
19	1.2V	V15~IO_L13N_D10_2	W14~IO_L16P_2	19
20	Y11~IO_L31P_GCLK31_D14_2	W15~IO_L14P_D11_2	Y16~IO_L14N_D12_2	20
21	AA12~IO_L30P_GCLK1_D13_2	AA14~IO_L15P_2	AB14~IO_L15N_2	21
22	Y10~IO_L29N_GCLK2_2	AB12~IO_L30N_GCLK0_USERCCLK_2	AB11~IO_L31N_GCLK30_D15_2	22
23	AB10~IO_L32N_GCLK28_2	AA10~IO_L32P_GCLK29_2	W11~IO_L29P_GCLK3_2	23
24	Y13~IO_L41P_2	T14~IO_L20P_2	U14~IO_L20N_2	24
25	W9~IO_L47P_2	W12~IO_L42P_2	Y12~IO_L42N_2	25
26	GND	GND	VREF	26
27			Y8~IO_L47N_2	27
28	1.2V	1.2V	AB7~IO_L63N_2	28
29	2.5V	TDI	Y7~IO_L63P_2	29
30		TMS	AB6~IO_L64N_D9_2	30
31		TCK	AA6~IO_L64P_D8_2	31
32	GND	TDO	VREF	32

Tabulka 6 : ZTEX USB-FPGA 1.15a pinlayout (2/2)

5.1.3 Propojení paměťového modulu s řídícím modulem

Pro připojení každé paměti je zapotřebí na straně řídícího modulu celkem 15 IO pinů a 4 piny jsou vyhrazeny pro napájení. Pro připojení dvou flash pamětí využijeme tedy celkem 30 IO pinů řídícího modulu. Mapování jednotlivých pinů paměťového modulu na vstupně výstupní piny řídícího modulu bylo během vývoje PCB několikrát upravováno s ohledem na délky vodičů v PCB paměťového modulu a jejich uspořádání.

FLASH 1			
pin function	FLASH pin	USB-FPGA module pin	FPGA pin
R/B	7	A26	A8~IO_L6N_0
RE	8	B26	B8~IO_L6P_0
CE	9	A25	D8~IO_L32N_0
VCC	12	A16/B16/E29/E30	-
VSS	13	C17/C26/D26/C32	-
CLE	16	B25	D9~IO_L32P_0
ALE	17	A24	C10~IO_L33N_0
WE	18	E24	C9~IO_L8P_0
WP	19	B24	D10~IO_L33P_0
DQ0	29	A30	B3~IO_L1P_HSWAPEN_0
DQ1	30	B30	A6~IO_L4N_0
DQ2	31	A29	A5~IO_L2N_0
DQ3	32	B29	C5~IO_L2P_0
DQ4	41	A28	C6~IO_L3N_0
DQ5	42	B28	D6~IO_L3P_0
DQ6	43	A27	C8~IO_L7N_0
DQ7	44	B27	D7~IO_L7P_0
VSS	36	C17/C26/D26/C32	-
VCC	37	A16/B16/E29/E30	-

Tabulka 7 : Propojení flash paměti s FPGA (1/2)

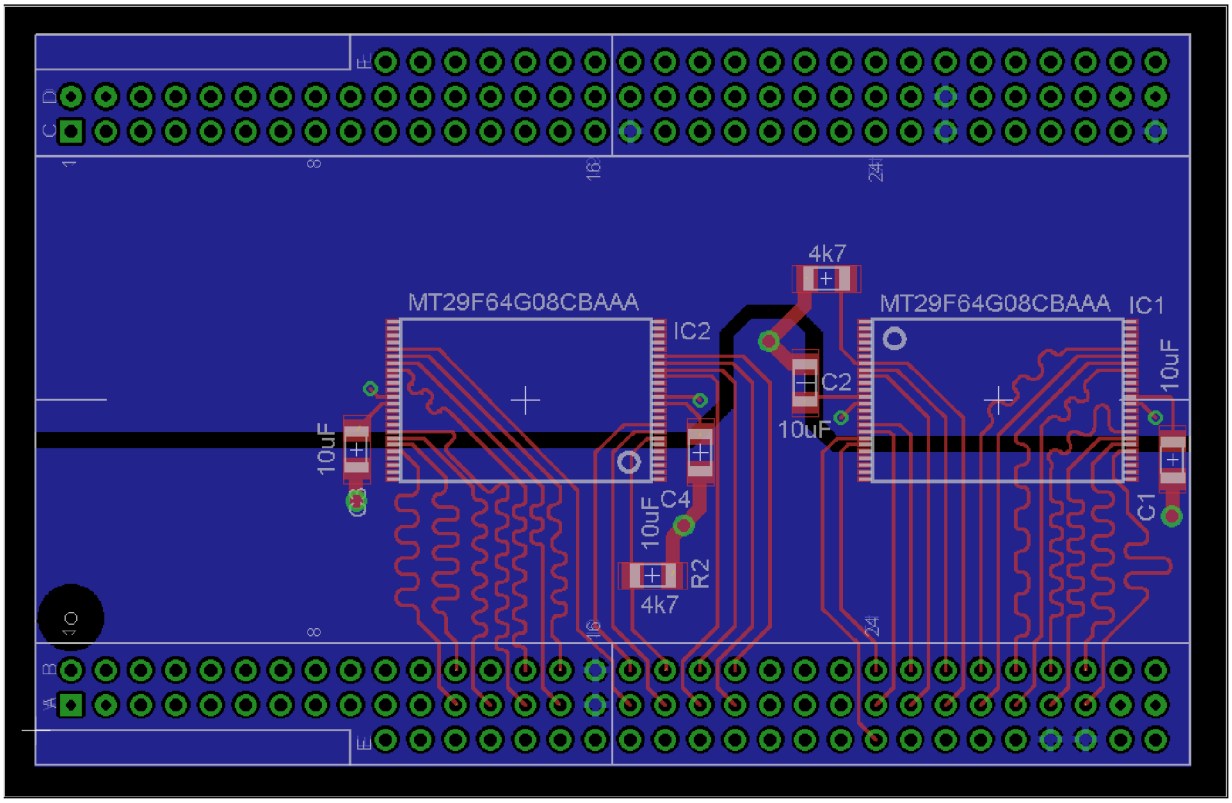
FLASH 2			
pin function	FLASH pin	USB-FPGA module pin	FPGA pin
R/B	7	B18	A13~IO_L48N_0
RE	8	A18	C13~IO_L48P_0
CE	9	B17	C11~IO_L35P_GCLK17_0
VCC	12	A16/B16/E29/E30	-
VSS	13	C17/C26/D26/C32	-
CLE	16	A19	C12~IO_L37N_GCLK12_0
ALE	17	B19	D11~IO_L37P_GCLK13_0
WE	18	A20	C15~IO_L62P_0
WP	19	B20	F10~IO_L38P_0
DQ0	29	A17	A11~IO_L35N_GCLK16_0
DQ1	30	B15	D15~IO_L46P_0
DQ2	31	A15	C14~IO_L46N_0
DQ3	32	B14	C17~IO_L64P_SCP5_0
DQ4	41	A14	A17~IO_L64N_SCP4_0
DQ5	42	A13	D17~IO_L65P_SCP3_0
DQ6	43	B12	B18~IO_L66P_SCP1_0
DQ7	44	A12	A18~IO_L66N_SCP0_0
VSS	36	C17/C26/D26/C32	-
VCC	37	A16/B16/E29/E30	-

Tabulka 8 : Propojení flash paměti s FPGA (2/2)

Schéma zapojení z Eagle cadu je k nalezení v příloze A a zdrojový soubor na přiloženém CD.

5.1.4 Návrh PCB paměťového modulu

PCB paměťového modulu bylo navrženo na základě schématu (příloha A) v prostředí Eagle cad. Vodiče jsou navrženy ručně, nebyl použit žádný algoritmus pro automatizaci. Jako výborný průvodce při vývoji desky plošných spojů posloužila série tutoriálů na adrese: http://www.sparkfun.com/tutorials/category/1 . Navržená deska plošných spojů je jednovrstvá při použití obou stran. Ve čtvrté revizi návrhu se již podařilo umístit všechny datové vodiče na jednu stranu a rozvod napájení na stranu opačnou. K rozvodu napájení v podobě 3,3V a země jsou použity 2 polygony (v obrázku Obrázek 19 : Návrh PCB paměťového modulu fialovou barvou). Datové vodiče jsou široké 0,25mm, což je dáno hlavně požadavky vývodů paměťových pouzder, které požadují, aby šířka přivedeného vodiče byla v rozmezí 0,17-0,27mm. Napájecí vodiče na přední straně modulu jsou přiváděny vodiči o šířce 1mm a až v blízkosti pinu paměti se jejich šířka mění taktéž na 0,25mm. Ke každému vstupu napájení je předřazen stabilizační kondenzátor. Tyto kondenzátory jsou celkem 4 a jedná se o keramické 6,3V (očekávané napětí na napájecím vodiči je 3,3V) SMD kondenzátory v pouzdře 1206C s kapacitou 10µF. Každá z pamětí potřebuje mimo VCC napájení napájet ještě pull-up rezistor, který je připojen k R/B pinu (otevřená propusť). Tyto reziostory jsou SMD 4K7 Ohm v pouzdře taktéž 1206C. Tyto komponenty jsou běžně k dostání například v GM electronics.



Obrázek 19 : Návrh PCB paměťového modulu

Paralelní datové cesty - sběrnice DQx - byly navíc za pomoci meandrování upraveny tak, aby byly všechny délky těchto vodičů shodné, případně aby rozdíl délek byl v tolerančním rozmezí 10%. Vodiče nejsou shodně dlouhé v rámci samotného paměťového modulu, jelikož bylo nezbytné zohlednit i délku dalšího vodiče v rámci řídícího modulu, který propojuje I/O konektor se samotným FPGA. V prostředí Eagle CAD slouží pro výpočet délky jednotlivých vodičů makro *length.ulp*.

Tabulka 9 : Délky DQx vodičů prvního paměťového modulu a Tabulka 10 : Délky DQx vodičů druhého paměťového modulu znázorňují přehledně jednotlivé délky vodičů, všechny hodnoty vzdáleností uvedené v těchto tabulkách jsou v milimetrech.

- Sloupec Kit trace -označení příslušného vodiče v rámci řídícího modulu,
- TRACE - označení vodiče ve schématu paměťového modulu,
- Kit ln - délka příslušného vodiče v rámci řídícího modulu,
- PCB ln - délka vodiče paměťového modulu před meandrováním,
- sum - součet délek daného vodiče na řídícím modulu a paměťovém modulu,
- diff max - rozdíl hodnot sum nejdelšího vodiče dané sběrnice a hodnoty sum daného vodiče, určuje o kolik milimetrů je nutné vodič meandrováním nastavit,
- meander - předpokládaná délka vodiče paměťového modulu po meandrování
- real meander - skutečná délka vodiče po meandrování
- % - procentuální vyjádření délky po meandrování vztažené k délce ve sloupci meander

FLASH1	Kit trace	TRACE	Kit ln	PCB ln	sum	diff max	meander	real meander	%
0	A30	N1	18,21	20,35	38,56	10,26	30,61	30,61	100
1	B30	N2	19,15	18,76	37,91	10,91	29,67	29,67	100
2	A29	N3	17,82	23,31	41,13	7,69	31	31	100
3	B29	N4	16,86	21,72	38,58	10,24	31,96	32,11	100,4693
4	A28	N5	17,27	29,41	46,68	2,14	31,55	31,56	100,0317
5	B28	N6	15,59	27,82	43,41	5,41	33,23	33,23	100
6	A27	N7	16,46	32,36	48,82	0	32,36	32,36	100
7	B27	N8	14,41	30,77	45,18	3,64	34,41	34,41	100

Tabulka 9 : Délky DQx vodičů prvního paměťového modulu

FLASH2	Kit trace	TRACE	Kit ln	PCB ln	sum	diff max	meander	real meander	%
0	A17	N20	18,65	35,2	53,85	0	35,2	35,2	100
1	B15	N23	18,96	29,6	48,56	5,29	34,89	34,89	100
2	A15	N24	20,05	31,19	51,24	2,61	33,8	33,15	98,07692
3	B14	N25	18,75	26,65	45,4	8,45	35,1	35,1	100
4	A14	N26	19,13	26,58	45,71	8,14	34,72	34,6	99,65438
5	A13	N27	23,23	23,83	47,06	6,79	30,62	30,62	100
6	B12	N28	22,04	19,29	41,33	12,52	31,81	31,81	100
7	A12	N29	23,37	20,88	44,25	9,6	30,48	30,48	100

Tabulka 10 : Délky DQx vodičů druhého paměťového modulu

Obecně zpoždění signálu PCB vodiče závisí na několika faktorech: jako parametry vodiče (šířka, tloušťka), délka vodivé cesty i její tvar (není dobrou praxí dělat například pravoúhlé zatáčky), i dielektrická konstanta izolačního materiálu.

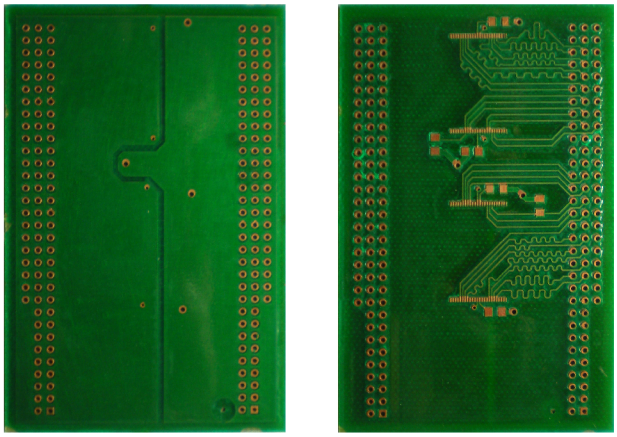
Výroba PCB

Z návrhu desky plošných spojů je po ukončení návrhové fáze vygenerovaná sada souborů, které popisují navržené PCB tak, aby jej bylo možné strojově vyrobit. Za standard, který přijímá drtivá většina výrobců PCB jsou považovány tzv. gerber soubory. Jedná se v podstatě o textové soubory popisující design PCB. Každý soubor nese odlišné informace:

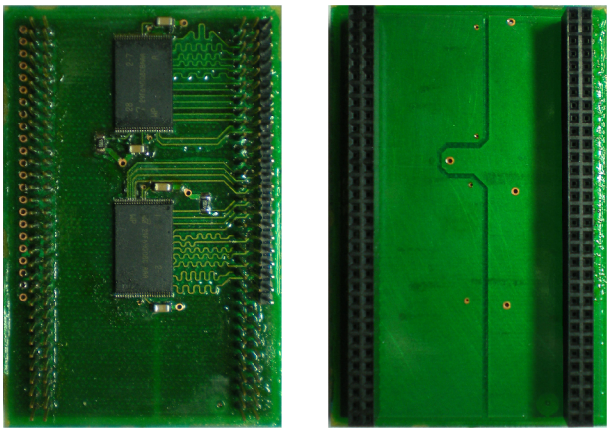
- soubor nesoucí souřadnice a popis děr k vyvrtání (*.dri),
- horní vrstva vodivých cest (*.GTO)
- horní "potisk" (*.GTO)
- horní mapa pájivých vývodů (*.GTS)
- spodní vrstva vodivých cest (*.GBL)
- spodní potisk (*.GBO) - není standardem
- spodní vrstva pájivých vývodů (*.GBS)

Zdrojové gerber soubory pro výrobu PCB jsou přiloženy na CD, stejně jako soubory obsahující nastavení CAM projektoru pro jejich generování z návrhových souborů Eagle cad.
Výroba desky plošných spojů se velmi zjednodušeně skládá z 5 kroků:

1. vyvrtání otvorů, podle mapy otvorů a očištění povrchu měděné folie
2. zakrytí motivu plošného spoje leptuvzdornou vrstvou
3. odleptání nepotřebné mědi a následné odstranění leptuvzdorné vrstvy
4. nanesení krycích a ochranných vrstev
5. mechanické opracování desky a nanesení potisku



Tabulka 11 : Neosazená deska tištěných spojů

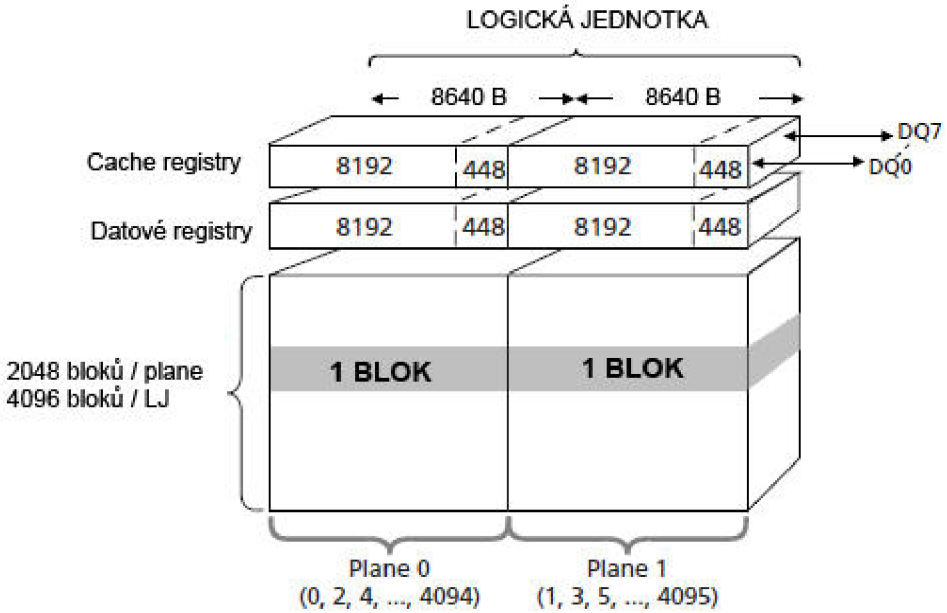


Tabulka 12 : Osazená deska tištěných spojů

5.2 Vnitřní uspořádání paměti MT29F64G08CAAA

Základní úložnou jednotkou je jedna stránka, její velikost činí 8KB + 448B, stránky jsou sdružovány do bloků, kde každý blok má 256 stránek, tedy dohromady 2048KB + 112KB. Zde je vhodné vzpomenout fakt, že typickou vlastností flash pamětí je schopnost programovat stránku, ale mazat pouze celé bloky. Bloky potom tvoří větší celky tzv. plane, které obsahují 2048 bloků, s celkovou kapacitou 2048 x (2048KB + 112KB), což činí 4320MB. Každá logická jednotka má potom takové plane dva a tedy celková kapacita logické jednotky je 8640MB.

Modely pamětí 128G, 256G, 512G, jsou potom architekturou identické s jediným rozdílem, obsahují patřičný počet logických jednotek. Použitý model 64G obsahuje logickou jednotku pouze jednu.



Obrázek 20 : Znázornění uspořádání datového úložiště

Každá stránka disponuje kapacitou pro metadata (448B), ukládají se sem například kontrolní data pro ECC nebo příznaky vadných buněk.

Adresování probíhá v trojrozměrné mřížce:

1. sloupce: CA0-CA13, adresy 0-8639 (21C0h)
2. stránka: PA0 - PA7, adresy 0-255 (FFh)
3. blok: BA8 - BA19

Adresování probíhá zpravidla v 5ti cyklech:

1. CA0 => DQ0 ... CA7 => DQ7
2. CA8 => DQ0 ... CA13 => DQ5, DQ6 => LOW, DQ7 =>LOW
3. PA0 => DQ0 ... PA7 => DQ7
4. BA8 => DQ0 ... BA15 => DQ7
5. BA16 => DQ0 ... BA19 => DQ3, DQ4-DQ7 => LOW

Kroky 3,4,5, tedy adresy stránky a bloku, jsou nazývány souhrnně adresa "řádku" (row).
Pozn.: V pátém kroku je DQ4 určeno pro výběr logické jednotky, pokud má zařízení jedinou, měla by být hodnota nastavena na LOW.

5.3 Rozhraní paměti MT29F64G08CAAA

Jedná se o NAND flash MLC paměťové moduly, které odpovídají specifikacím ONFI 2.2. To znamená, že disponují jak asynchronním, tak synchronním komunikačním rozhraním. V rámci práce je využit asynchronní komunikační protokol, ačkoliv synchronní protokol je rychlejší (dle specifikace až 4x), část řadiče asynchronního protokolu implementovaná v FPGA je názorněji a snáze implementovatelná pomocí konečného stavového automatu.

5.3.1 Asynchronní rozhraní

Asynchronní rozhraní představuje třístavová polo-duplexní multiplexovaná sběrnice, která slouží pro přenos příkazů, adres i dat a řídících signálů, které vyjadřují aktivní stav na hodnotě logické 0 (LOW).

5.3.1.1 Asynchronní Enable / Standby

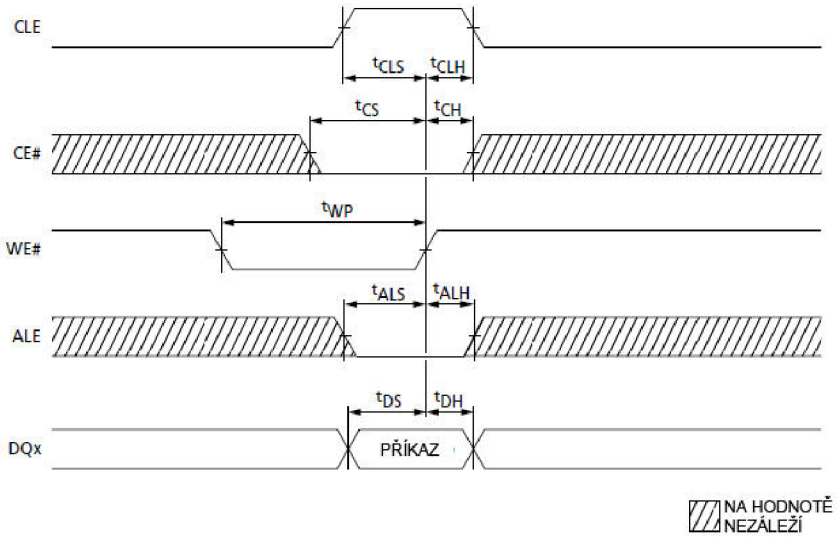
Signál CE slouží k aktivování / deaktivování jednotky (u paměťových čipů, které mají více logických jednotek slouží taktéž k výběru konkrétní logické jednotky, s kterou se aktuálně komunikuje). Pokud je hodnota CE signálu LOW, jednotka je aktivní a přijímá všechny další řídící signály. Jednotka je deaktivována, pokud má signál CE hodnotu HIGH. Jednotka tuto změnu reflektuje i v případě, že je momentálně zaneprázdněna a to ihned po dokončení dané operace.

5.3.1.2 Asynchronní nečinnost sběrnice (bus Idle)

Tento stav nastává při nastavení hodnot signálům CE = LOW, WE = HIGH a RE = HIGH - jednotka zůstává aktivní, ale nepřijímá, ani nevysílá žádné signály na sběrnici DQ.

5.3.1.3 Asynchronní zadání příkazu

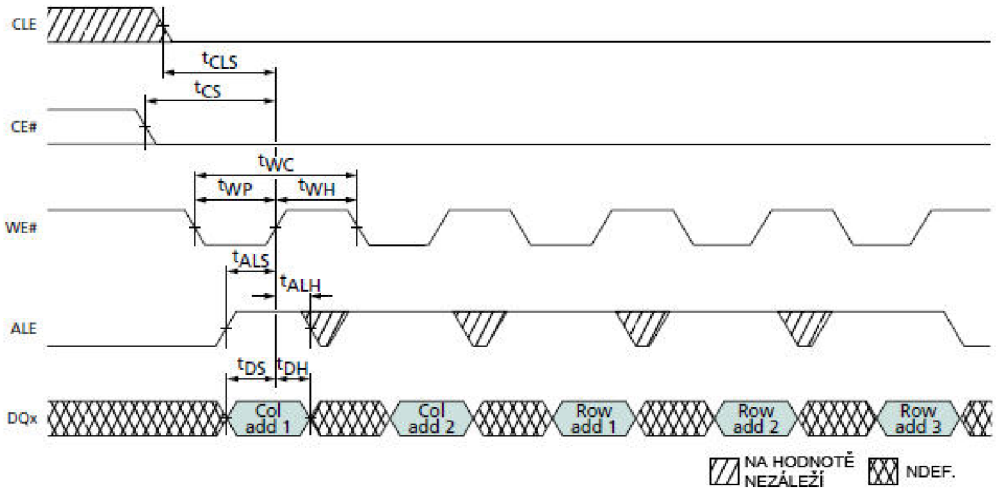
Příkaz je po sběrnici DQ[7:0] přiveden do příkazového registru při vzestupné hraně signálu WE, přičemž CE = LOW, ALE = LOW, CLE = HIGH, RE = HIGH. V případě, že je jednotka právě zaneprázdněna, je příkaz ignorován. Výjimku tvoří příkazy RESET (FFh) nebo READ STATUS (70h).



Obrázek 21 : Diagram asynchronního zadání příkazu

5.3.1.4 Asynchronní zadání adresy

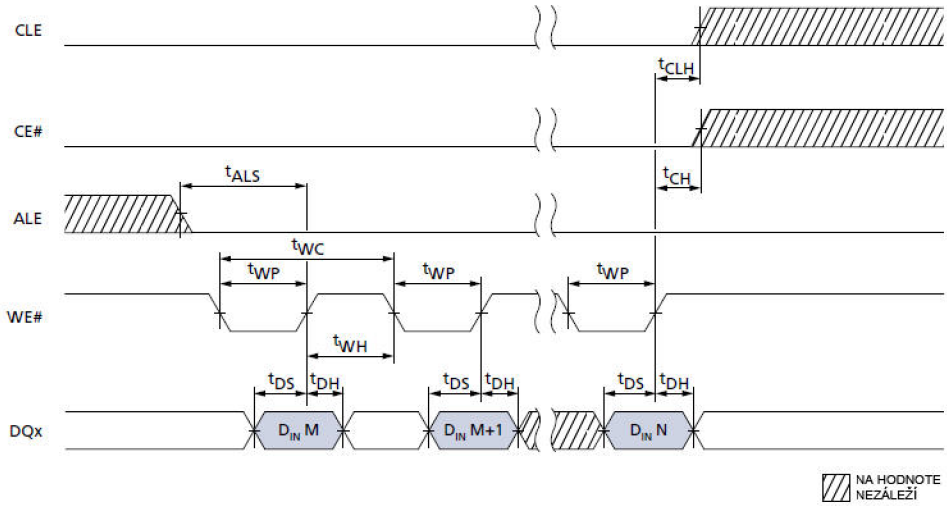
Adresa je zapsána do adresového registru prostřednictvím sběrnice DQ[7:0], přičemž zápis je proveden vždy s vzestupnou hranou WE, když CE = LOW, ALE = HIGH, CLE = LOW, RE = HIGH. Bity, které nejsou součástí adresy, musí mít logickou hodnotu 0. Počet cyklů, potřebných pro konkrétní příkaz se můžu lišit. V případě, že je jednotka zaneprázdněna, není adresa přijata.



Obrázek 22 : Diagram asynchronního zadání adresy

5.3.1.5 Asynchronní vstup dat

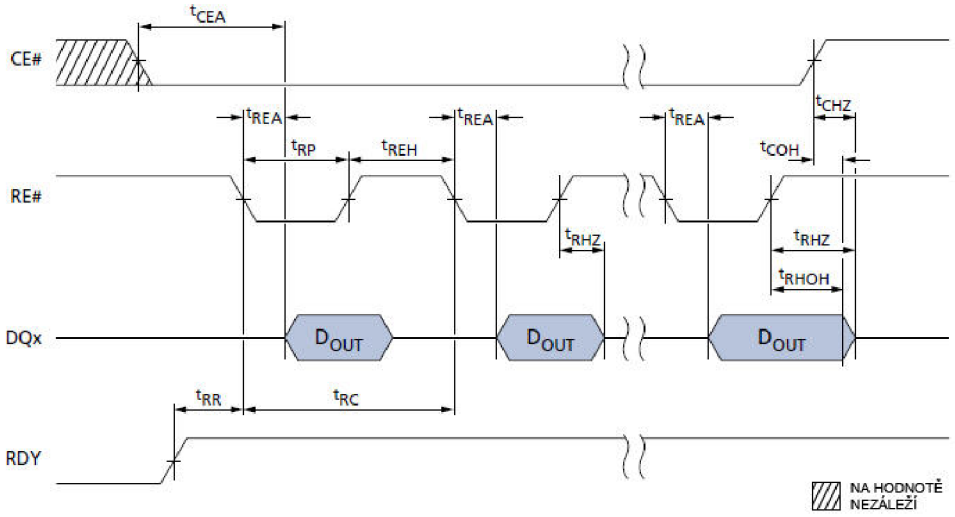
Data jsou zapsána do cache registru skrze sběrnici DQ[7:0] s vzestupnou hranou signálu WE, když CE = LOW, ALE = LOW, CLE = LOW, RE = HIGH. Pokud je jednotka zaneprázdněna, data nepřijme.



Obrázek 23 : Diagram asynchronního vstupu dat

5.3.1.6 Asynchronní výstup dat

Datový výstup následuje zpravidla po operaci čtení, případně po operaci pro získání stavu jednotky. Data jsou z cache vložena na sběrnici DQ[7:0] se sestupnou hranou signálu RE, při CE = LOW, ALE = LOW, CLE = LOW, WE = HIGH.



Obrázek 24 : Diagram asynchronního výstupu dat

Při implementaci je nunté zohlednit také časovaní, hodnoty pro použitý rychlostní mód 0 jsou zobrazeny v následujících tabulkách 13 a 14.

tCS	tALS	tCLS	tWP	tDS	tCLH	tCH	tALH	tDH	tWH	tWC
> 70ns	> 50ns	> 50ns	> 50ns	> 40ns	> 20ns	> 20ns	> 20ns	> 20ns	> 30ns	> 100ns

Tabulka 13 : Časování pro příkazové a adresové operace

tCEA	tREA	tRP	tREH	tRHZ	tCOH	tCHZ	tRR	tRC	tRHOH	tRLOH
< 100ns	< 40ns	> 50ns	> 30ns	< 200ns	> 0ns	> 20ns	> 40ns	> 100ns	> 0ns	> 0ns

Tabulka 14 : Časování pro datové operace

5.3.2 Inicializace pamětí

Paměť je automaticky inicializována při přivedení a ustálení napájecího napětí Vcc. Dokončení prvotní inicializace paměti je signalizována pomocí ready/busy (R/B) signálu, který má po dobu inicializace hodnotu LOW a po dokončení inicializace nabývá hodnoty HIGH. V případě, že řadič není schopen sledovat R/B signál měl by na dokončení inicializace čekat 100µs od dosažení Vcc(MIN) a Vccq(MIN).

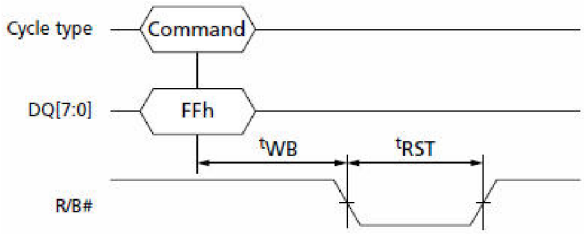
Následně je nutné provést prvotní RESET (FFh) celé paměti, který ji uvede do "známého" stavu a po jehož provedení se paměť dostane do normálního provozního stavu. Provedení resetu může řadič paměti sledovat opět pomocí R/B signálu, případně může použít příkaz READ STATUS (70h). Po provedení prvotního resetu se paměť nachází v asynchronním režimu, v rychlostním režimu 0. Tento model podporuje až rychlostní režim 5, ovšem model v pouzdře TSOP48 pouze rychlostní režim 4.

5.4 Nejdůležitější operace

Základními diskovými operacemi jsou čtení, zápis a mazání dat. Pro základní funkcionalitu je nutné uvažovat operaci reset pro uvedení zařízení do známého / výchozího stavu a také operaci read status, která slouží jako zpětná vazba pro řadič paměti a umožňuje tak zjistit stav právě probíhající operace.

5.4.1 RESET (FFh)

Reset přepne zařízení do známého, počátečního stavu. Teto příkaz je také první příkaz, který musí být spuštěn po inicializaci paměti. Jeho zadáním dojde k přerušení všech probíhajících operací (pokud je volán při probíhajícím programování stránky, nebo mazání bloku, hrozí, že v paměťovém poli zůstanou částečně zapsaná / vymazaná data). Po jeho úspěšném dokončení se paměť nachází v asynchronním režimu, v rychlostním módu 0 a příkazový registr je připraven k přijetí dalšího příkazu.

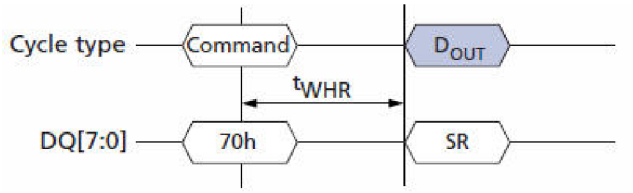


Obrázek 25 : Diagram průběhu operace RESET

Informace o časování pro mód 0: tWB = max 200ns, tRST max 5µs pokud je zařízení ve stavu ready (až 500µs, pokud zařízení není ve stavu ready).

5.4.2 READ STATUS (70h)

Zadáním tohoto příkazu do příkazového registru je vrácen v data-out cyklu stav jednotky. Pokud paměťový čip obsahuje více logických jednotek, je nutné ve spojení s většinou příkazů používat READ SATATUS ENHANCED (78h), který umožňuje volbu logické jednotky, která má vrátit svůj stav. V tomto případě, kdy je použit paměťový modul pouze s jedinou logickou jednotkou, je možné používat za všech okolností příkaz READ STATUS (70h)

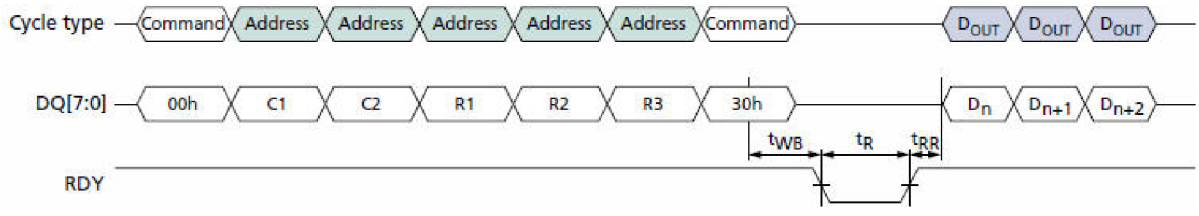


Obrázek 26 : Diagram průběhu operace READ STATUS

Informace o časování pro mód 0: tWHR: min 120ns.

5.4.3 READ PAGE (00h-30h)

Přečte stránku flash paměti. Příkaz 00h přepne jednotku do režimu čtení, poté je přeneseno 5 adresových cyklů, které identifikují požadovanou stránku. Následuje příkaz 30h, který zkopíruje data z flash paměti do příslušného cache registru a provede data-output transfer.

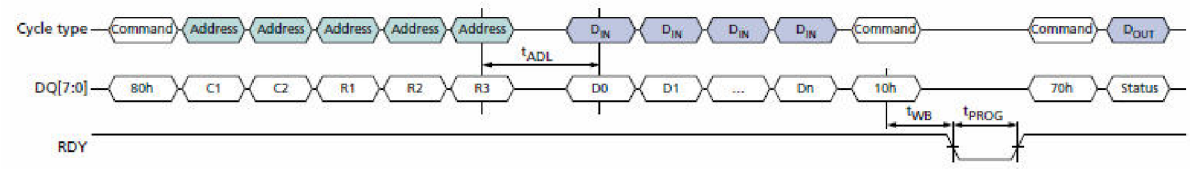


Obrázek 27 : Diagram průběhu operace READ PAGE

Informace o časování pro mód 0: tWB = max 200ns, tR = max 75µs, tRR = min 40ns.

5.4.4 PROGRAM PAGE (80h-10h)

Umožňuje řídícímu obvodu uložit data do cache a následně data přenést na zvolenou adresu v paměti. Příkaz je přijmut pouze za předpokladu, že se jednotka nachází ve stavu "ready" (RDY=1, ARDY=1). Zadáním příkazu 80h do příkazového registru celá operace začíná, tento příkaz také smaže veškerý obsah cache registru. Následně je nunté v 5ti cyklech přenést adresu programované stránky a nakonec realizovat data-input přenos (v jeho průběhu je možné použít i příkazy CHANGE WRITE COLUMN, nebo CHANGE ROW ADDRESS). Po dokončení datového přenosu je zapsán příkaz 10h do příkazového registru, čímž je označen konec zápisu.

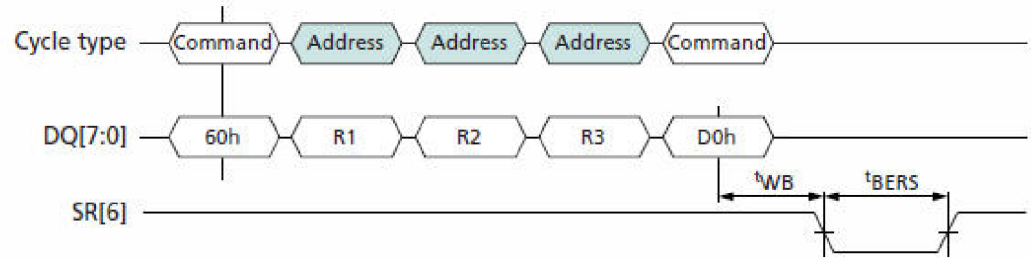


Obrázek 28 : Diagram průběhu operace PROGRAM PAGE

Informace o časování pro mód 0: tWB = max 200ns, tPROG = max 2600µs (typ 1300µs).

5.4.5 ERASE BLOCK (60h-D0h)

Vymaže data bloku adresovaného v adresovém registru. Příkaz je přijmut pouze za předpokladu, že se jednotka nachází ve stavu "ready" (RDY=1, ARDY=1). Úkon začíná zápisem příkazu 60h do příkazového registru, následuje přenos adresy řádku ve 3 krocích (adresa stránky je zanedbána). Poté proběhne odstranění dat zadáním příkazu D0h. Pro sledování dokončení operace je možné sledovat signál R/B, případně se jednotky opakovaně dotázat na její status pomocí příkazu 70h.

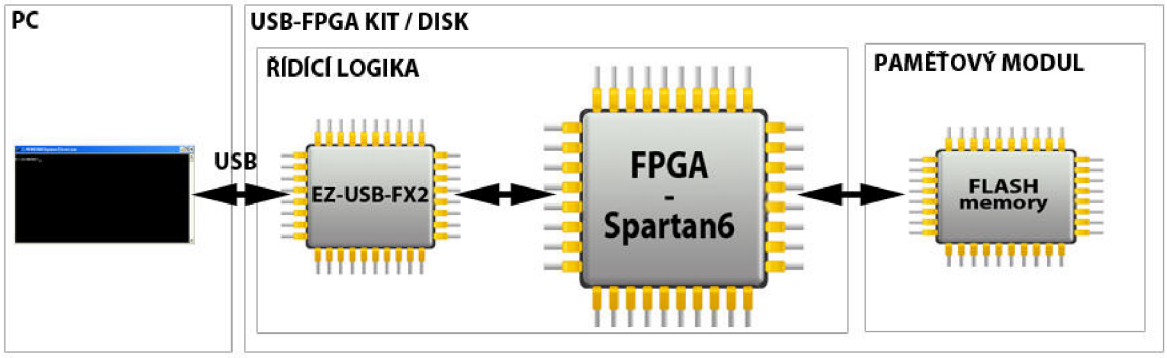


Obrázek 29 : Diagram průběhu operace ERASE BLOCK

Informace o časování pro mód 0: tWB = max 200ns, tBERS = max 10ms (typ 3,8ms).

6 Implementace

Pro názornost a jednoduchost ladění při vývoji je disk navržen jako USB zařízení s konzolovým přístupem, pro které je možné jednotlivé příkazy zadávat ručně přímo v konzoli. Tato implementace je dále snadno rozšiřitelná na klasický disk[19].



Obrázek 30 : Vizualizace návrhu

V následujícím popisu implementace se počítá, že je zařízení ve stavu, kdy bylo právě spuštěno bez načteného uživatelského firmware jak v USB řadiči, tak v FPGA. V USB řadiči je načten z EEPROM původní firmware výrobce, tak aby se zařízení inicializovalo na USB sběrnici a bylo možné s ním komunikovat.

6.1 Aplikace

Po spuštění aplikace hledá na USB sběrnici zařízení s patřičným ID výrobce (převzato s SDK k vývojovému kitu) , poté, co je nalezeno, je načten do zařízení vytvořený a zkompilovaný firmware pro EZ-USB-FX2. Po načtení firmwaru je skrze něj naprogramováno i FPGA patřičným bitsreamem. Pomocí přepínačů příkazové řádky je možné vynutit nahrání firmwaru nebo naprogramování FPGA i v již inicializovaném / naprogramovaném stavu. Poté je vytvořeno spojení mezi firmwarem a aplikací pomocí 2 endpointů. Konkrétně endpoint EP2 (adresa 0x82) pro komunikaci směrem zařízení -> host a EP4 (adresa 0x04) pro komunikaci směrem host -> zařízení. Na straně zařízení jsou oba endpointy opatřeny dvěma buffery o velikosti 512B (bufferované endpointy slouží ke snížení čekacích dob při přenosu).

Po ustavení spojení přechází aplikace do dvoustavové smyčky, v prvním stavu čeká na zadání řetězce ze standardního vstupu - tento řetězec představuje příkaz a jeho parametry. A následně čeká na odpověď zařízení.

6.2 Firmware pro řadič sběrnice USB

Tato část řešení je určena ke zpracování zpráv, které přicházejí po sběrnici USB. Zpracování Bulk komunikačního protokolu do značné míry usnadňuje SDK dodávané k vývojovému kitu. Pro všechny

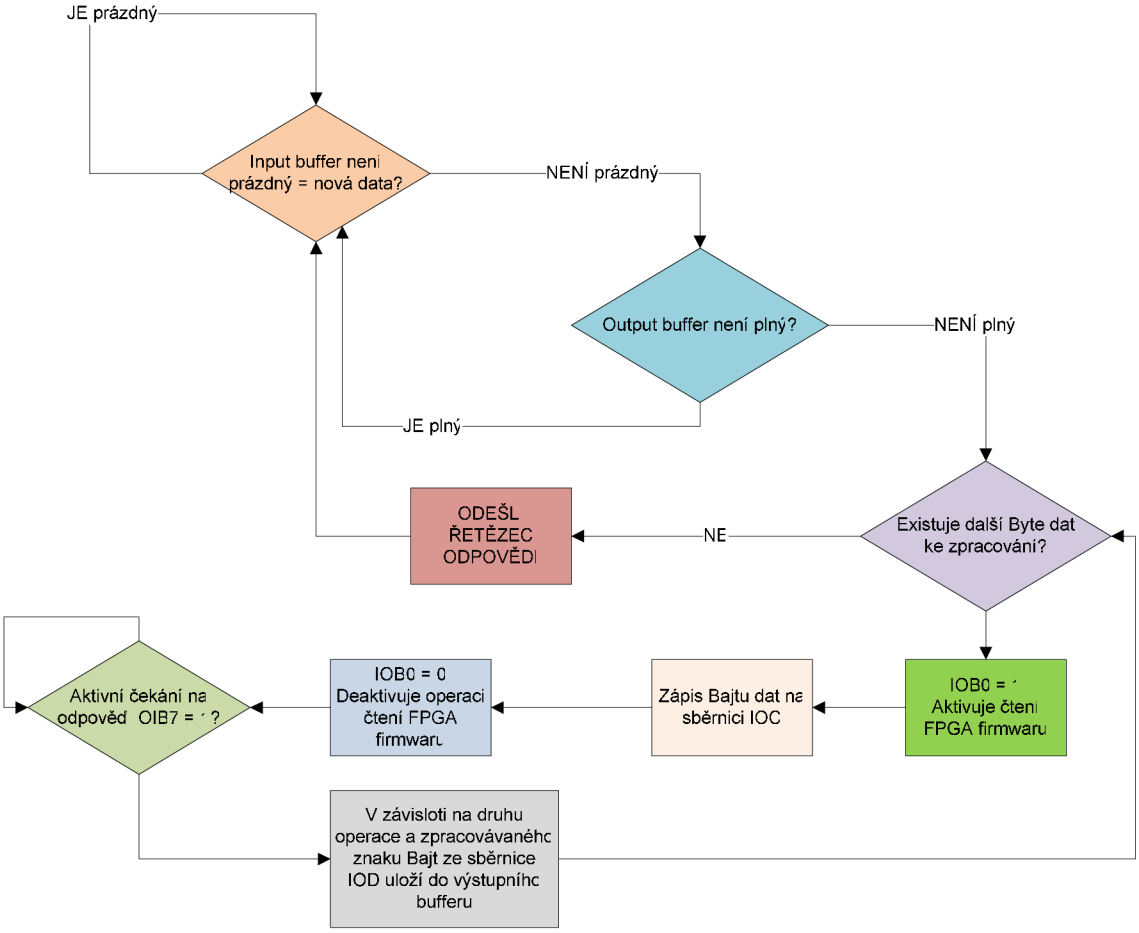
standardní úkony existují příslušná makra, jejichž definici je možno nalézt ve vložených hlavičkových souborech. V rámci firmwaru jsou definované komunikační kanály takto:

```
EP_CONFIG(2,0,BULK,IN,512,2);
EP_CONFIG(4,0,BULK,OUT,512,2);
```

Kde jednotlivé hodnoty mají takovýto význam:

1. identifikátor endpointu,
2. identifikátor rozhraní,
3. USB komunikační protokol,
4. směr přenosu (z pohledu hostitelského zařízení),
5. velikost bufferu v Bajtech,
6. počet stínových bufferů

Definice makra EP_CONFIG se nachází v souboru ztex-conf.h, řádek 186. Pomocí systému maker je nastaven také identifikátor zařízení a kanál pro programování FPGA (v tomto případě je využit EP4 v high-speed režimu). Následuje definice makra, které se spouští po ukončení programování FPGA - jeho nejpodstatnějšími funkcemi jsou: nastavení frekvence hodinového signálu, reset USB komunikačních kanálů po naprogramování FPGA a také aktivace RESETovacího signálu pro firmware v FPGA.



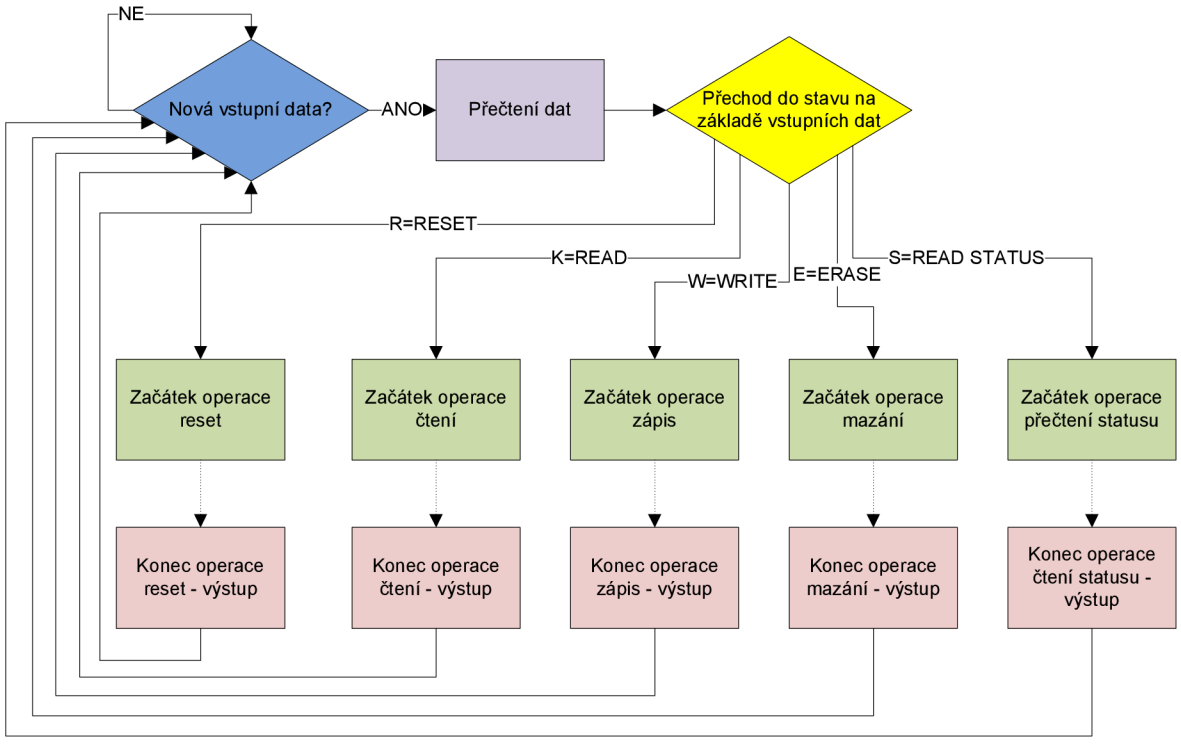
Obrázek 31 : Diagram znázorňující chování hlavní smyčky firmwaru

6.3 Firmware pro FPGA

Je navržen jako konečný stavový automat s dvěma rozhraními, který řídí paměťové operace v závislosti na vstupech z řadiče USB rozhraní. Počáteční stav čeká v cyklické smyčce na příjem identifikátoru operace, po jeho přijetí tento identifikátor vyhodnotí a přejde do patřičného stavu, který řídí danou operaci. Operace můžeme rozdělit celkem do 4 skupin, podle toho, jaká data jsou k jejich provedení zapotřebí a jaká data vystupují:

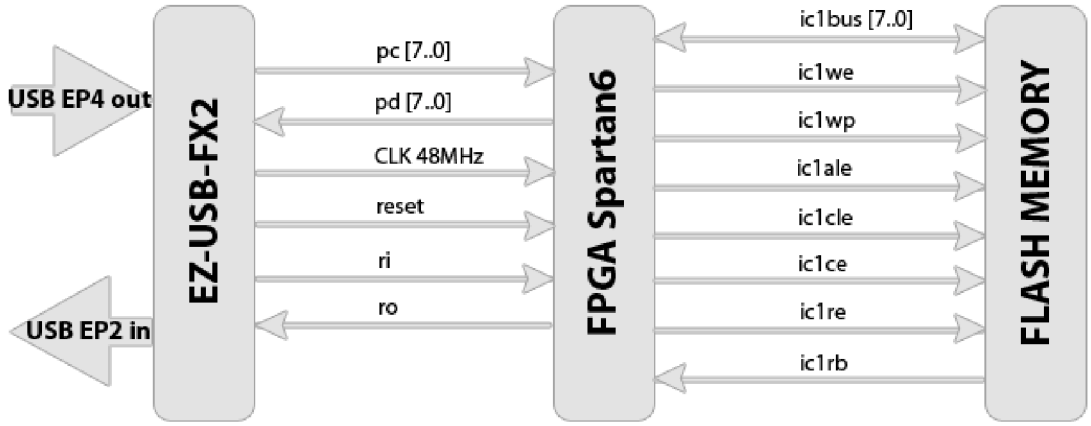
1. Operace bez dalších vstupů s 1B výstupem - tyto operace jsou vykonány pouze na základě svého operátoru (například RESET) a výstupem je 1B informace (buď příznak o úspěšném dokončení operace, nebo výstupní hodnota - například u operace READ STATUS).
2. Operace vyžadující delší vstup a 1B výstup - takovou operací může být například operace PROGRAM PAGE (vstupují také adresa a data), případně ERASE BLOCK (vstupuje adresa). A výstupem je pouze příznak o dokončení provádění operace.
3. Operace vyžadující delší vstup i výstup - například operace READ PAGE. Ta vyžaduje na vstup zadání adresy a výstupem jsou data.

Podle typu operace přizpůsobuje i firmware EZ-USB-FX svůj výstup, který odesílá spuštěné aplikaci na hostitelském počítači.



Obrázek 32 : Zjednodušený a zobecněný diagram stavového automatu realizujícího flash paměťový řadič

Příkazy jsou v souladu se specifikací uvedenou v kapitole 5. Časování bylo ověřeno pomocí modelu v prostředí Modelsim. Podrobnější náhled je součástí přílohy C.



Obrázek 33 : Mapa řídících a propojovacích signálů

Na obrázku 34 jsou uvedeny všechny řídící signály vystupující z řadiče implementovaného v FPGA, stejně jako datové sběrnice a signály, které řídí samotný řadič. Firmware řadiče USB sběrnice nastaví při své vlastní inicializaci signál reset, kterým uvede do pohotovostního stavu (stav 0 konečného stavového automatu) program firmwaru v FPGA. Hlavní hodinový signál má kmitočet 48MHz, což odpovídá periodě ~20ns. Sběrnice *pc* je vstupní sběrnicí a nese data, která přicházející přes USB. Sběrnice *pd* je naopak sběrnicí výstupní a odesílá data do hostitelské aplikace. Obě tyto sběrnice jsou široké 1B. Signály *ri* a *ro* indikují připravenost dat ke čtení. Signál *ri* je vysílán řadičem USB sběrnice vždy, když je připraven k přečtení nový bajt dat. Naopak signál *ro* nastavuje firmware v FPGA vždy, když je na sběrnici *pd* připravena odpověď. Signály propojující rozhraní flash paměti a jejího řadiče v FPGA odpovídají specifikaci ONFI a jejich řízení je založeno na faktech kapitoly 5.

6.3.1 Formát příkazů a výstupů

Příkazy jsou zadávány jako řetězce do konzolové aplikace a následně potvrzeny k odeslání. Jejich formát je následující:

1B	Až 8*5B	xB
Identifikátor příkazu	Adresa zadaná v binárním kódu	Data v binárním kódu

Data a adresy jsou zadávány v char-binárním kódu kvůli snazšímu zpracování a transparentnosti. Výstup je potom pouze znak "K" jako znak úspěšného provedení operace případně data, kde každý Bajt je vyjádřen svojí dekadickou hodnotou, tedy 0-255.

7 USB mass storage class

Dalším krokem k vytvoření běžně použitelného úložného média je použití USB mass storage třídy. Tuto třídu používají běžné komerční USB flash disky. Její výhodou je, že k úložnému médiu přistupuje čistě jako k blokovému zařízení, disk tedy nemusí vůbec řešit souborový systém a pracuje pouze s daty. Toto rozšíření nepředstavuje žádné velké změny pro řadič pamětí (firmware pro FPGA), jelikož bude stále řešit nízkoúrovňové operace. Větších změn musí dostát firmware USB řadiče, nejen, že se zcela změní komunikační rozhraní, ale řadič musí být navíc schopen na základě přijatých dat vyvolat adekvátní operaci v řadiči paměti.

7.1 Vysokokapacitní paměťové zařízení

Je identifikováno hodnotou třídy 8. Tato třída poskytuje různé přístupové protokoly, je možné volit také z více druhů příkazových sad (ty jsou definovány podtřídou). Standardem je podtřída číslo 6, která používá *SCSI Transparent command set*. Třída obsahuje i podtřídu číslo 1, která pro komunikaci se zařízením využívá *Reduced block commands*. několik zdrojů uvádí (viz například http://www.xat.nl/en/riscos/sw/usb/class.htm, nebo wiki apod.) že je to typická podtřída pro USB flash disky.

Za použití volně dostupné zkušební verze softwaru pro sledování komunikace na sběrnici USB - *USBlyzer 2.0* (http://www.usblyzer.com/), byla provedena analýza popisovačů 3 soudobých USB flash disků a 2 dalších úložných zařízení za účelem zjištění, kterou sadu příkazů tato zařízení skutečně používají.

- Corsair Flash Voyager 4GB - **podtřída 06h SCSI - Transparent command set**
- Patriot Xporter XT 4GB - **podtřída 06h SCSI - Transparent command set**
- Kingston DT R400 16GB - **podtřída 06h SCSI - Transparent command set**
- Externí pevný disk Verbatim (model: 53008 500GB) - **podtřída 06h SCSI - Transparent command set**
- Nokia E72 (v režimu vysokokapacitní paměťové zařízení) - **podtřída 06h SCSI - Transparent command set**

Z tohoto jednoznačně plyne, že pro USB úložná zařízení je využívána podtřída se sadou příkazů SCSI - Transparen command set.

Velkým kladem třídy "*vysokokapacitní paměťové zařízení*" je bezesporu existence generického ovladače, jehož prostřednictvím je možné disky připojovat k osobnímu počítači bez nutnosti instalovat jakýkoliv další podpůrný software. Tento ovladač je v operačních systémech windows oficiálně přítomen od verze Windows 2000 (Windows NT5) již v základní instalaci. Tohoto ovladače pro přístup k úložnému médiu by mělo využít i vyvíjené zařízení, které je tedy koncipováno jako USB CLASS 01h, USB SUBCLASS 06h.

7.1.1 USB Mass storage class

Pod touto třídou lze využít metodu přenosů, buď Control/Bulk/Interrupt, která je vhodná výhradně pro externí FDD mechaniku a není využitelná pro přenosy rychlejší než full-speed [18], nebo metodu využívající pouze Bulk přenosy. Metodu založenou výhradně na Bulk přenosech využívají i všechna výše analyzovaná zařízení. Vzhledem k povaze vyvíjeného zařízení je Bulk-only transfer metoda jedinou použitelnou možností.

7.1.1.1 Popisovače zařízení

Při vývoji firmware zařízení je nutné nastavit správně popisovače zařízení, které zařízení odesílá hostiteli v průběhu procesu inicializace. Na jejich základě hostitelský systém volí ovladač, který zařízení přiřadí, zjistí také na kterých Endpointech lze se zařízením komunikovat, jaký komunikační protokol zvolit, maximální velikost odesílaných paketů a další informace. Základními popisovači jsou:

- Popisovač zařízení
- Popisovač konfigurace
- Popisovač rozhraní
- Popisovače jednotlivých endpointů

Pro zjištění chování zařízení MSC na sběrnici USB bylo využito mimo velmi nedetailních specifikací také postupu "reverzního inženýrství". Tento postup spočíval v analýze komunikace výše zmíněných USB vysokokapacitních paměťových zařízení s hostitelským systémem pomocí nástroje USBlyzer. Následně byly dohledány informace k analýzou získaným hodnotám a na těchto základech bylo navržen základní firmware pro vyvíjené zařízení. Zvláštností je fakt, že veškerá zařízení používají některá nastavení, která jsou ve specifikacích z již let 1999-2003 považována za zastaralá/již opuštěná.

Popisovače zařízení jsou konfigurovány pomocí frameworku, který je dodáván s vývojovým kitem ZTEX (kokrétně se jedná o hlavičkový soubor ztex-descriptors.h). Jejich nastavení pro účely vyvíjeného zařízení jsou součástí přílohy D.

7.1.1.2 USB Mass storage class - Bulk only transportní protokol

Podkapitola je založena na [20].

Použití tohoto protokolu je deklarováno v popisovači rozhraní (bit 7 - hodnota 50h). Jedná se o protokol, který je používán pro všechna úložná zařízení připojená přes sběrnici USB (výjimkou může být externí disketová mechanika). Poskytuje obalovou vrstvu v prostředí sběrnice USB pro protokol, který realizuje datové operace. Protokol datových operací je určen taktéž v popisovači rozhraní (bit 6). Jak bylo zmíněno výše, nejpoužívanějším je SCSI Transparent command set.

Jak plyne ze specifikace sběrnice USB, i v rámci tohoto protokolu vychází všechny žádosti ze strany hostitele. Komunikace v rámci BOTP má celkem 3 fáze:

1. Příkaz - *zaslání příkazu zařízení*
2. Data - *pokud to povaha příkazu vyžaduje, následuje datová fáze*
3. Status - *hlášení po dokončení operace (vše v pořádku / oznámení chyby...)*

Příkaz

Příkaz je zařízení zaslán v podobě tzv. "command block wrapper (CBW)". Ten se skládá z:

- *dCBWSignature* - identifikátor, že se jedná o CBW - jeho hodnota je vždy 43425355h (little endian)
- *dCBWTag* - jednoznačný identifikátor daného příkazu v rámci komunikace (tento tag musí obsahovat také status)
- *dCBWDataTransferLength* - délka datového přenosu v Bytech
- *bmCBWFlags* - příznaky : směr přenosu - 0 = out / 1 = in
- *bCBWLUN* - identifikátor logické jednotky zařízení
- *bCBWCBLength* - délka následujícího příkazu v Bytech
- *CBWCB* - samotný příkaz dle zvolené příkazové sady (SCSI transparent command set)

Byte/bit	7	6	5	4	3	2	1	0
0-3	*dCBWSignature*							
4-7	*dCBWTag*							
8-11	*dCBWDataTransferLength*							
12	*bmCBWFlags*							
13	-				*bCBWLUN*			
14	-			*bCBWCBLength*				
15-30	*CBWCB*							

Tabulka 15 : Command Block Wrapper

Data

Datový přenos korespondující se zaslaným příkazem v CBWCB, data mohou být přenášena obousměrně, avšak pouze jediným směrem v rámci daného příkazu. Může se jednat například o operace jako READ (in), WRITE (out), nebo INQUIRY (in). Očekávaná délka datového přenosu je dána v CBW a případné neshody s touto hodnotou jsou navráceny ve status odezvě.

Status

Je odpověď zařízení hostovi na právě provedenou operaci, která se nazývá "command status wrapper (CSW) a její význam je především kontrolní. Skládá se z:

- *dCSWSignature* - identifikátor status paketu - jeho hodnota je vždy 53425355h (little endian)
- *dCSWTag* - odpovídá tagu, kterým byl označen příkaz
- *dCSWDataResidue* - tato hodnota udává jakýkoliv rozdíl mezi očekávanou délkou datového přenosu a skutečnou délkou v Bytech
- *dCSWStatus* - status: 00h = příkaz vykonán, 01h = příkaz selhal, 02h chyba fáze

Byte/bit	7	6	5	4	3	2	1	0
0-3	*dCSWSignature*							
4-7	*dCSWTag*							
8-11	*dCSWDataResidue*							
12	*dCSWStatus*							

Tabulka 16 : Command Status Wrapper

7.1.1.3 povinné SCSI příkazy

Povinnými příkazy by měly být logicky všechny, které jsou takto označeny v příslušném standardu, ovšem v praxi je běžné, že zařízení neimplementují všechny. Skutečné minimum pro funkční MSC zařízení je:

- INQUIRY - dotaz na konfiguraci zařízení, vrací různá data, viz specifikace na CD
- READ CAPACITY (10) - žádost o obdržení kapacity zařízení - odpovědí jsou 4B : adresa posledního bloku, 4B : velikost bloku v Bajtech.
- READ (10) - operace čtení - udává adresu začátku čtení a následně délku čteného úseku, odpovědí je patřičný počet Bajtů dat.
- REQUEST SENSE - vrací hlášení o chybách.
- TEST UNIT READY - testuje připravenost jednotky přijmout / zaslat data.
- WRITE (10) - informuje o adrese, kam zapsat a kolik dat.

8 Použité nástroje

Pro vývoj firmwaru řadiče sběrnice USB a obslužné aplikace bylo využito základů SDK dodávaného k Ztex vývojovému kitu. Toto SDK je založeno na JAVA knihovnách, které slouží pro vývoj aplikace. Podrobná dokumentace JAVA API je dostupná na stránkách výrobce: http://www.ztex.de/firmware-kit/docs/java/index.html . Samotný firmware je napsán v jazyce C za použití hlavičkových souborů ze zmíněného SDK, které poskytují spoustu užitečných maker, usnadňujících použití sběrnice USB. Pro úspěšnou kompilaci je nutné mít v systému Java JDK 6 nebo novější, ANSI C kompiler SDCC, zde je nutné dávat velký pozor na verzi - nejnovější verze 3.1 není stabilní a výsledkem bylo často zcela nepředvídatelné chování. Pro účely této práce byla nakonec použita verze SDCC 3.0, která se chová stabilně. Při překladu pod platformou windows je nutné taktéž nainstalovat i nástroj MSYS, který zpřístupňuje kolekci nativně unixových nástrojů jako bash, make, gawk, grep aj. Mimo Java Development Kit, je samozřejmě nutné pro samotné spuštění nainstalovat taktéž Java Runtime Environment.

Pro vývoj FPGA firmware byl použit nástroj Xilinx ISE ve verzi 14.1, pomocí tohoto nástroje byly VHDL zdrojové kódy taktéž syntetizovány a byl vytvořen bitstream soubor pro naprogramování FPGA. Simulace VHDL designů byly provedeny v prostředí Modelsim PE Studen Edition v 10.1. Samotné naprogramování FPGA je zajišťeno přímo při spouštění hostitelské aplikace.

Pro přístup k zařízení pod systémem windows je možné použít ovladač libusb-win32, při vývoji byl použit ve verzi 1.2.5.0, tento ovladač může být nutné přiřadit několikrát, jelikož zařízení běžně pracuje s více USB ID (ZTEX, Cypress).

9 Navrhovaná rozšíření

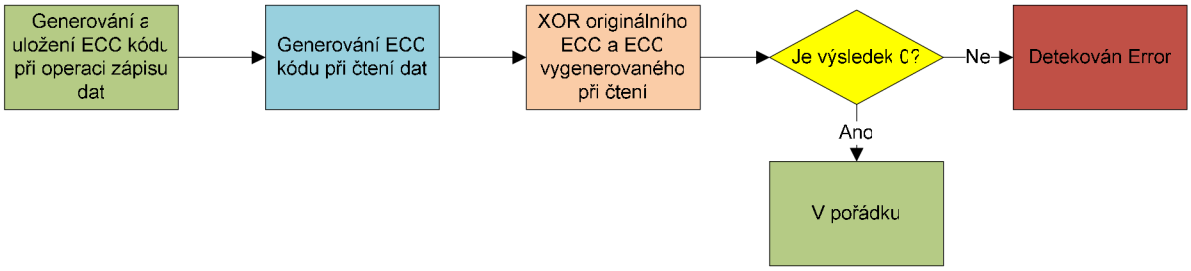
Mimo připojení disku jako nativního USB disku za použití generického ovladače existují další rozšíření, která by z demonstračního zařízení vytvořila plnohodnotný použitelný disk pro běžné aplikace. Tato rozšíření se týkají převážně spolehlivosti a rychlosti zařízení.

9.1 Zvýšení spolehlivosti

S ohledem na dříve popsané nedostatky pamětí flash pro použití jako sekundárního a off-line úložného média by bylo vhodné implementovat alespoň základní techniky, které se snaží vliv těchto nedostatků na spolehlivost paměti potlačit.

9.1.1 EEC a mapování vadných bloků

Výrobce použitých pamětí *Micron* garantuje, že paměti mají minimálně 3996 funkčních bloků z 4096, dále že každý vadný blok má již při opuštění výrobního procesu označeny vadné bloky a to značkou 00h na počátku metadatového prostoru každého bloku - bajt 8192. Každá paměť má také zkontrolovaný a funkční blok na fyzické adrese 00h.



Obrázek 34 : Diagram detekce vadných bloků

Pro výpočet ECC se používají například Hammingův algoritmus (snadný na implementaci, omezené možnosti opravy kódu), Reed-Solomonův algoritmus (robusní, složitý na implementaci, vyžaduje hodně systémových prostředků), BCH algoritmus. ECC je možné zapsat do metadatového prostoru a v případě detekce chyby označit vadný blok tak, jako to dělá výrobce.

V případě, že by bylo zařízení používáno jako klasicky připojený disk řízený generickým ovladačem systému, bude nutné explicitně zajistit, že po detekci a označení vadného bloku nebude blok již nikdy použit. Mimo příznaků v podobě metadat každého bloku by bylo vhodné užít i tabulky, která by uchovávala adresy vadných bloků. Vzhledem k tomu, že chybné bloky jsou automaticky vyřazeny z funkčnosti, je nutné implementovat asociativní tabulku mapování logických adres na fyzické, která bude zohledňovat tento fakt. Tabulku udržující seznam nefunkčních bloku a asociativní tabulku pro mapování adres by bylo vhodné sloučit v jednu.

9.2 Zvýšení rychlosti

Během vývoje zařízení byl brán především zřetel na transparentnost řešení (řadič komunikačního rozhraní skutečně zpracovává data a řadič disku používá základní operace). Míst, kde by se dalo navrženo řešení vylepšit s ohledem na rychlost je hned několik:

9.2.1 Řadič komunikačního rozhraní

Řadič komunikačního rozhraní *Cypress EZ-USB-FX2* je schopen USB high speed přenosu, ovšem pouze za předpokladu, že není využit procesor řadiče, ten totiž svým výkonem neodpovídá požadované propustnosti. V případě, že by fronty endpointů byly řízeny vnějším řadičem z FPGA (nikoliv interním procesorem), je možné dosáhnout vyšší rychlosti při komunikaci po sběrnici USB.

9.2.2 Použitý mód časování pamětí

S ohledem na řídící hodinový signál FPGA byl u pamětí použit základní režim 0, který znamená vyšší latence při změnách signálů. Použité paměti jsou schopny operovat až v režimu 4 (dle ONFI standardu), který přináší v průměru 4 násobné zrychlení.

9.2.3 Paralelizace

Vyvíjený prototyp disponuje dvěma paměťovými čipy, jelikož využití zařízení v podobě klasického disku s sebou přináší nutnost vytvoření dynamické tabulky pro mapování adres, je možné inspirovat se RAID režimy u diskových polí a implementovat analogické řešení na úrovni řadiče flash pamětí v FPGA. Mimo zvýšení rychlosti a kapacity je možné samozřejmě uvažovat i zvýšení spolehlivosti třeba jednoduchým zrcadlením.

9.2.4 Použití komplexnějších operací paměti

Paměť disponuje operacemi, které umožňují snadnější a rychlejší operaci s většími objemy dat. Ať už se jedná o operace umožňující měnit adresu v průběhu čtení, zápisu (není nutné zapisovat data na médium souvisle), nebo umožňující data řadit do fronty apod.

9.2.5 Použití cache

USB-FPGA kit disponuje 128MB paměti DDR2. Cache by mimo data měla sloužit pro uchování tabulky pro mapování adres v aktivním režimu. Avšak tuto funkcionalitu by bylo možné implementovat snadno přímo v FPGA, jeho volná kapacita by k tomuto účelu bohatě postačovala.

9.3 Programování FPGA z flash

Pro běžné použití by samozřejmě bylo vhodné, aby bylo zařízení schopné uchovat bitstream pro FPGA i ve vypnutém stavu a při spuštění s ním opět naprogramovalo FPGA. Kit je za tímto účelem vybaven slotem na microSD kartu s SPI rozhraním, kterou lze pro tyto účely použít.

9.4 USB flash disk spolupracující s OS

Implementací výše uvedených vylepšení, především týkajících se spolehlivosti, a za použití předpřipraveného firmwaru pro USB mass storage class je možné vyvinou běžně použitelný USB disk, který bude pracovat při použití v OS běžně přítomných ovladačů .

10 Závěr a zhodnocení

Práce se podrobně zaměřuje na technologii pamětí flash, jejich možnosti pro uchování dat a integraci tohoto paměťového média do diskových zařízení. Poukazuje na slabiny technologie flash pro danou aplikaci a nastiňuje postupy, které potlačují vliv těchto záporných vlastností. Zmiňuje také konkurenční technologie a porovnává je s ohledem na použití v diskových jednotkách. Disky na bázi pamětí flash je možné připojit ke spoustě běžných rozhraní, v rámci diplomové práce jsou tedy okrajově zmíněny ty nejběžnější varianty - mnohdy se zajímavým závěrem, jako například absence nativního SSD disku pro PCIe rozhraní na běžném trhu.

Praktická část práce se potom zaměřuje na možnosti návrhu a konstrukce vlastního disku na bázi paměti flash. Prvním a překvapivě poměrně nelehkým krokem je výběr vhodných komponent, kdy s ohledem na dostupnost elektronických součástek musel být návrh několikrát měněn. Problém většinou vyvstává v situaci, kdy je kladen požadavek na nízké - kusové - množství a rychlou dostupnost. To se netýká jen samotných modulů NAND flash pamětí, ale také řadičů komunikačního rozhraní - najít dostupný integrovaný obvod, který by realizoval SATA řadič pro zařízení je takřka nemožné. Nakonec bylo zvoleno nejrozšířenější rozhraní pro periferní zařízení - USB. Jelikož byly komponenty objednávány z vlastních zdrojů, rozhodl jsem se, s ohledem na budoucí využití, investovat do vývojového kitu s USB rozraním. Jediné po kusech dostupné paměťové čipy s kapacitou 8GB byly nakonec dodány z USA.

Pro zvolený hardware bylo zapotřebí navrhnout koncept celého zařízení a následně vyvinout paměťový modul. S tím souvisel návrh desky plošných spojů v prostředí Eagle Cad a osvojení si prvních zkušeností s designem PCB.

Vyvinutý firmware implementovaný jako konečný stavový automat demonstruje základní principy fungování řadiče paměti flash standardu ONFI. Implementace tohoto firmwaru se ukázala být v běžných podmínkách velmi složitá, především díky nemožnosti efektivně ladit a testovat prototypové zařízení. Dalším implementačně nastíněným rozšířením je firmware pro řadič USB sběrnice, pomocí kterého je zařízení demonstrativně připojeno a inicializováno jako USB velkokapacitní paměťové zařízení, pouze s použitím generického ovladače OS Windows. Pro tento úkol bylo nutné vzhledem k neúplným nebo těžko dostupným informacím detailně analyzovat provoz na sběrnici USB pomocí SW sondy. Při této příležitosti bylo také zjištěno, že velkokapacitní USB flash zařízení nevyužívají příkazovou sadu RBC jak se ve většině zdrojů uvádí, ale SCSI transparent command set.

Práci hodnotím celkově jako velmi přínosnou, mimo rozšíření teoretických základů, bylo nutné pracovat se spoustou specializovaného software, zkusil jsem navrhnout a použít první vlastní PCB a také počet různorodých programovacích jazyků v rámci jednoho projektu je poměrně vysoký. Vyvíjený prototyp má v případě aplikace dalších popsaných rozšíření potenciál stát se plnohodnotným USB flash diskem.

11 Literatura

[1] Hort Tomáš: *Technologie a zajímavosti z oblasti SSD disků* [online] 22.11.2011 [cit 29.12.2011]. Dostupné z URL: http://pctuning.tyden.cz/hardware/disky-cd-dvd-br/22588-technologie-a-zajimavosti-z-oblasti-ssd-disku

[2] Kolektiv autorů projektu wikipedia.org: *Flash memory* [online] 2.1.2012 [cit 3.1.2012] Dostupné z URL: http://en.wikipedia.org/wiki/Flash_memory

[3] Amber Huffman: *Open NAND Flash Interface Specification revision 1.0* [online] 28.12.2006 [cit 3.1.2012]. Dostupné z URL: http://onfi.org/wp-content/uploads/2009/02/onfi_1_0_gold.pdf

[4] Super Talent Technology Inc.: *SLC vs. MLC: An Analysis of Flash Memory* [online] 20.3.2008 [cit 26.12.2011] Dostupné z URL: http://www.supertalent.com/datasheets/SLC_vs_MLC%20whitepaper.pdf

[5] Tišnovský Pavel: *Technologie Flash Pamětí a Způsoby Jejich Využití* [online] 25.9.2008 [cit 27.12.2011] Dostupné z URL: http://www.root.cz/clanky/technologie-flash-pameti-a-zpusoby-jejich-vyuziti

[6] Kolektiv autorů projektu wikipedia.org: *Hard disk drive* [online] 27.12.2011 [cit 27.12.2011] Dostupné z URL: http://en.wikipedia.org/wiki/Hard_disk_drive

[7] Kolektiv autorů projektu wikipedia.org: *Bubble memory* [online] 7.12.2011 [cit 27.12.2011] Dostupné z URL: http://en.wikipedia.org/wiki/Bubble_memory

[8] Anderson Don: SATA Storage Technology, Mindshare Press, San Francisco, 2007, 463 s, ISBN 978-0-9770878-1-5

[9] Kolektiv autorů projektu wikipedia.org: *Solid-state drive* [online] 21.12.2011 [cit 27.12.2011] Dostupné z URL: http://en.wikipedia.org/wiki/Solid-state_drive

[10] Kolektiv autorů projektu wikipedia.org: *Write amplification* [online] 20.12.2011 [cit 28.12.2011] Dostupné z URL: http://en.wikipedia.org/wiki/Write_amplification

[11] Kolektiv autorů projektu wikipedia.org: *Ferroelectric RAM* [online] 4.4.2012 [cit 10.4.2012] Dostupné z URL: http://en.wikipedia.org/wiki/Ferroelectric_RAM

[12] Kolektiv autorů projektu wikipedia.org: *Magnetoresistive random access meory* [online] 26.12.2011 [cit 28.12.2011] Dostupné z URL: http://en.wikipedia.org/wiki/Magnetoresistive_random_access_memory

[13] Kolektiv autorů projektu wikipedia.org: *Programmable metallization cell* [online] 14.5.2011 [cit 28.12.2011] Dostupné z URL: http://en.wikipedia.org/wiki/Programmable_metallization_cell

[14] Kolektiv autorů projektu wikipedia.org: *Phase-change memory* [online] 25.12.2011 [cit 28.12.2011] Dostupné z URL: http://en.wikipedia.org/wiki/Phase-change_memory

[15] Kolektiv autorů projektu wikipedia.org: *Universal Serial Bus* [online] 10.5.2011 [cit 11.5.2011] Dostupné z http://en.wikipedia.org/wiki/Universal_Serial_Bus

[16] Kolektiv autorů projektu wikipedia.org: *Field-programmable gate array* [online]. 12.5.2012 [cit. 13.5.2012] Dostupné z URL: http://en.wikipedia.org/wiki/Field-programmable_gate_array

[17] Xilinx: *Spartan-6 Family Overview* [online] 25.10.2011 [cit. 13.5.2012] Dostupné z URL: http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf

[18] Usb.org: Universal Serial Bus Mass Storage Class Overview [online] 23.06.2003 [cit. 14.5.2012] Dostupné z URL: http://www.usb.org/developers/devclass_docs/usb_msc_overview_1.2.pdf

[19] Axelson Jan: USB Mass storage, Lakevies Research, Madison, 2006, 287s, ISBN13 978-1-931448-05-5

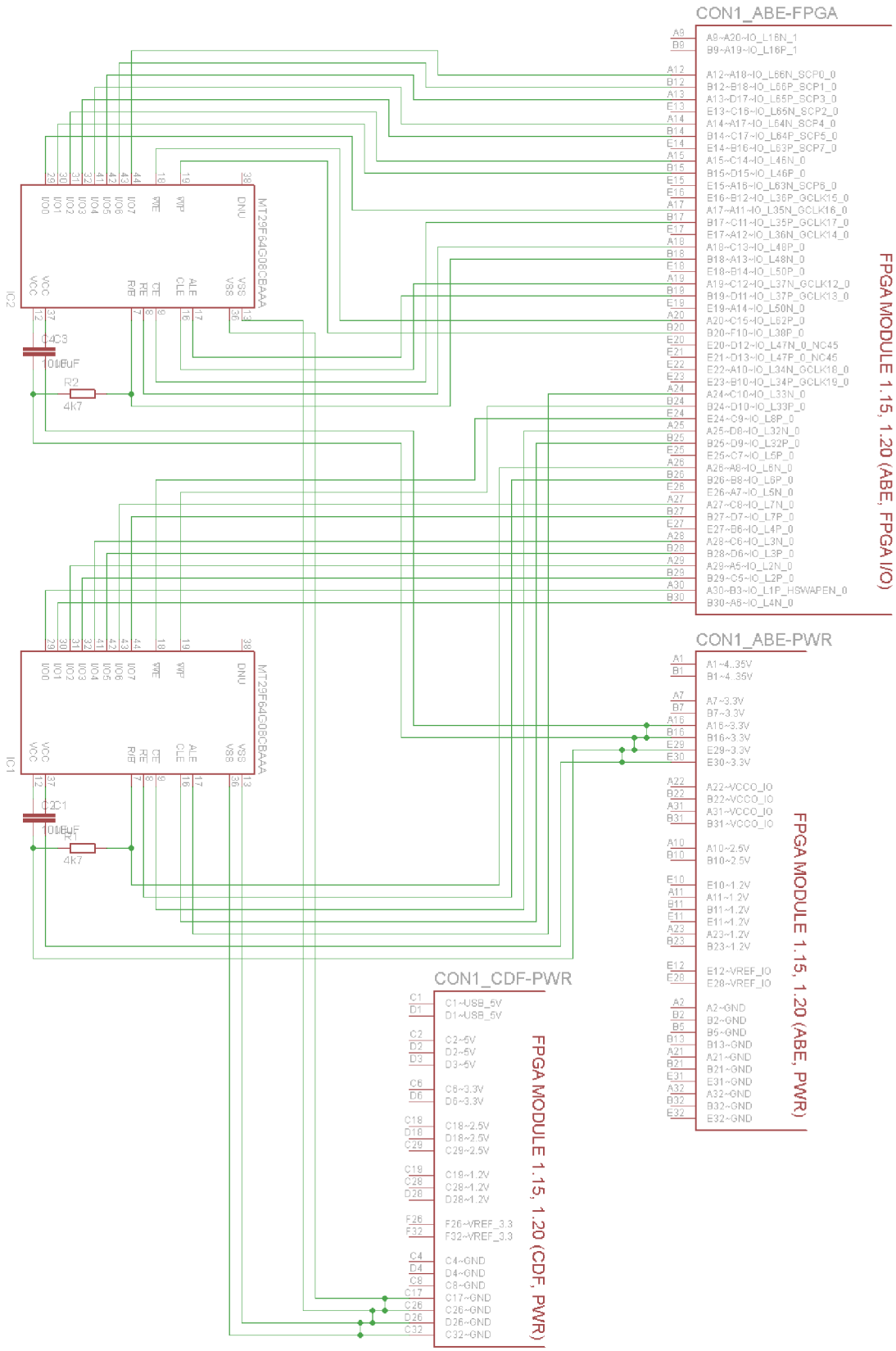
[20] Usb.org: Universal Serial Bus Mass Storage Class Bulk-Only Transport [online] 31.09.1999 [cit. 14.5.2012], Dostupné z URL: http://www.usb.org/developers/devclass_docs/usbmassbulk_10.pdf

Seznam příloh:

Příloha A. Schéma zapojení EAGLE cad paměťového modulu.
Příloha B Popisovače USB zařízení.
Příloha C Rozbor implementace vybraných příkazů.
Příloha D Nastavení popisovačů pro MSC.

Obsah přiloženého CD:
Použité podklady (specifikace)
Zdrojové kódy
Manuál
Scháma zapojení a desky plošných spojů
Gerber soubory PCB
Tato zpráva
Vybrané nástroje

Příloha A - schéma zapojení paměťového modulu



Příloha B (1/2) - popisovače USB zařízení

Offset	Field	Size	Value	Description
0	*bLength*	Byte	12h	Size of this descriptor in bytes.
1	*bDescriptorType*	Byte	01h	DEVICE descriptor type.
2	*bcdUSB*	Word	????h	*USB Specification* Release Number in Binary-Coded Decimal (i.e. 2.10 = 210h). This field identifies the release of the *USB Specification* with which the device and is descriptors are compliant.
4	*bDeviceClass*	Byte	00h	**Class is specified in the interface descriptor.**
5	*bDeviceSubClass*	Byte	00h	**Subclass is specified in the interface descriptor.**
6	*bDeviceProtocol*	Byte	00h	**Protocol is specified in the interface descriptor.**
7	*bMaxPacketSize0*	Byte	??h	Maximum packet size for endpoint zero. (only 8, 16, 32, or 64 are valid **(08h, 10h, 20h, 40h)**).
8	*idVendor*	Word	????h	Vendor ID (assigned by the USB-**IF**).
10	*idProduct*	Word	????h	Product ID (assigned by the manufacturer).
12	*bcdDevice*	Word	????h	Device release number in binary-coded decimal.
14	*iManufacturer*	Byte	??h	Index of string descriptor describing the manufacturer.
15	*iProduct*	Byte	??h	Index of string descriptor describing this product.
16	*iSerialNumber*	Byte	??h	Index of string descriptor describing the device's serial number. **(Details in 4.1.1 below)**
17	*bNumConfigurations*	Byte	??h	Number of possible configurations.

Tabulka B-1: Popisovač zařízení

Offset	Field	Size	Value	Description
0	*bLength*	Byte	09h	Size of this descriptor in bytes.
1	*bDescriptorType*	Byte	02h	CONFIGURATION Descriptor Type.
2	*wTotalLength*	Word	????h	Total length of data returned for this configuration. Includes the combined length of all descriptors (configuration, interface, endpoint, and class- or vendor-specific) returned for this configuration.
4	*bNumInterfaces*	Byte	??h	Number of interfaces supported by this configuration. **The device shall support at least the Bulk-Only Data Interface.**
5	*bConfigurationValue*	Byte	??h	Value to use as an argument to the *SetConfiguration()* request to select this configuration.
6	*iConfiguration*	Byte	??h	Index of string descriptor describing this configuration.
7	*bmAttributes*	Byte	?0h	Configuration characteristics: **Bit** — **Description** 7 — Reserved (set to one) 6 — Self-powered 5 — Remote Wakeup 4..0 — Reserved (reset to zero) **Bit** 7 is reserved and must be set to one for historical reasons. **For a full description of this *bmAttributes* bitmap, see the *USB 1.1 Specification*.**
8	*MaxPower*	Byte	??h	Maximum power consumption of the USB device from the bus in this specific configuration when the device is fully operational. Expressed in 2mA units (i.e. 50 = 100mA)

Tabulka B-2 : Popisovač konfigurace

Příloha B (2/2) - popisovače USB zařízení

Offset	Field	Size	Value	Description
0	*bLength*	Byte	09h	Size of this descriptor in bytes.
1	*bDescriptorType*	Byte	04h	INTERFACE Descriptor Type.
2	*bInterfaceNumber*	Byte	0?h	Number of interface. Zero-based value identifying the index in the array of concurrent interfaces supported by this configuration.
3	*bAlternateSetting*	Byte	??h	Value used to select alternate setting for the interface identified in the prior field.
4	*bNumEndpoints*	Byte	??h	Number of endpoints used by this interface (excluding endpoint zero). **This value shall be at least 2.**
5	*bInterfaceClass*	Byte	08h	**MASS STORAGE Class.**
6	*bInterfaceSubClass*	Byte	0?h	Subclass code (assigned by the USB-IF). **Indicates which industry standard command block definition to use. Does not specify a type of storage device such as a floppy disk or CD-ROM drive.** (See *USB Mass Storage Overview Specification*)
7	*bInterfaceProtocol*	Byte	50h	**BULK-ONLY TRANSPORT.** (See *USB Mass Storage Overview Specification*)
8	*iInterface*	Byte	??h	Index to string descriptor describing this interface.

Tabulka B-3 : Popisovač rozharní

Offset	Field	Size	Value	Description
0	*bLength*	Byte	07h	Size of this descriptor in bytes.
1	*bDescriptorType*	Byte	05h	ENDPOINT Descriptor Type.
2	*bEndpointAddress*	Byte	8?h	The address of this endpoint on the USB device. The address is encoded as follows. Bit — Description 3..0 — The endpoint number 6..4 — Reserved, set to 0 7 — 1 = In
3	*bmAttributes*	Byte	02h	This is a Bulk endpoint.
4	*wMaxPacketSize*	Word	00??h	Maximum packet size. Shall be 8, 16, 32 or 64 bytes (08h, 10h, 20h, 40h).
6	*bInterval*	Byte	00h	Does not apply to Bulk endpoints.

Tabulka B-4 : Popisovač výstupního endpointu

Offset	Field	Size	Value	Description
0	*bLength*	Byte	07h	Size of this descriptor in bytes.
1	*bDescriptorType*	Byte	05h	ENDPOINT descriptor type.
2	*bEndpointAddress*	Byte	0?h	The address of this endpoint on the USB device. This address is encoded as follows: Bit — Description 3..0 — Endpoint number 6..4 — Reserved, set to 0 7 — 0 = Out
3	*bmAttributes*	Byte	02h	This is a Bulk endpoint.
4	*wMaxPacketSize*	Word	00??h	Maximum packet size. Shall be 8, 16, 32 or 64 bytes (08h, 10h, 20h, or 40h).
6	*bInterval*	Byte	00h	Does not apply to Bulk endpoints.

Tabulka B-5 : Popisovač vstupního endpointu

Příloha C - detail provedení příkazu

Simulace úvodní RESET operace

Příloha D - nastavení popisovačů pro MSC

Connection Status Device connected
Current Configuration 1
Speed High (480 Mbit/s)
Device Address 1
Number Of Open Pipes 2

Device Descriptor DIP-USB DISK

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	221Ah	
10	idProduct	2	0100h	
12	bcdDevice	2	0000h	0.00
14	iManufacturer	1	01h	"ZTEX."
15	iProduct	1	02h	"DIP-USB DISK."
16	iSerialNumber	1	00h	
17	bNumConfigurations	1	01h	

Device Qualifier Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	0Ah	
1	bDescriptorType	1	06h	Device Qualifier
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	bNumConfigurations	1	01h	
9	bReserved	1	00h	

Configuration Descriptor 1

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	C0h	Self Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.1......	Yes
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	32h	100 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	FFh	

Endpoint Descriptor 02 2 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	FFh	

Configuration Descriptor 1

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	04h	
7	bmAttributes	1	C0h	Self Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.1......	Yes
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	32h	100 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	FFh	

Endpoint Descriptor 02 2 Out, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	FFh	