VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VYUŽITÍ PAMĚŤOVÉ KARTY SD NA PLATFORMĚ FITKIT

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE Martin Gazda
AUTHOR

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VYUŽÍTÍ PAMĚŤOVÉ KARTY SD NA PLATFORMĚ FITKIT

USING OF SD MEMORY CARD FOR FITKIT PLATFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE Martin Gazda
AUTHOR

VEDOUCÍ PRÁCE Ing. Zdeněk Vašíček
SUPERVISOR

BRNO 2008

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Gazda Martin**
Obor: Informační technologie
Téma: **Využití paměťové karty SD na platformě FITKit**
Kategorie: Počítačová architektura

Pokyny:

1. Seznamte se s paměťmi typu SecureDigital (SD)
2. Seznamte se s SW/HW platformou FITkit
3. Navrhněte řadič paměti SD, který bude možné integrovat do stávajícího systému
4. Implementujte navržený řadič ve VHDL
5. Navrhněte a implementujte sadu knihovních funkcí pro práci s uvedeným typem paměti
6. Demonstrujte funkčnost na vzorové aplikaci

Literatura:

- http://www.sdcard.org (specifikace, normy)
- dále dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- splnění bodů 1-3 zadání

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Vašíček Zdeněk, Ing.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2007
Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Martin Gazda**
Id studenta: 79029
Bytem: Bezručova 32, 741 01 Nový Jičín
Narozen: 10. 05. 1985, Nový Jičín
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   bakalářská práce

Název VŠKP: Využití paměťové karty SD na platformě FITKit
Vedoucí/školitel VŠKP: Vašíček Zdeněk, Ing.
Ústav: Ústav počítačových systémů
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☐ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....	.....
Nabyvatel	Autor

Abstrakt

Hardware s možností rekonfigurace představuje moderní trend ve vývoji nových obvodů. Velké množství vestavných systémů je využíváno jako mobilní zařízení s vysokými nároky na množství zpracovávaných dat i energetickou náročnost připojených periferií. Těmto nárokům vyhovují paměti založené na technologii FLASH,jako např. paměťová karta SD. Tato práce se zabývá možnostmi využití této paměťové karty na vývojové platformě FITkit.

Klíčová slova

FPGA, FITkit, rekonfigurace, paměťová karta, SecureDigital, MMC,SD

Abstract

Reconfigurable hardware architectures represent a modern trend in development of new circuits. A lot of embedded systems is used as mobile devices with high demands on data capacity and power consumption of peripheral devices. This demands are suited with FLASH based memory, like SecureDigital memory card. This bachelors thesis deals with the problem of usage of this memory card on FITkit development platform.

Keywords

FPGA, FITkit, reconfiguration, memory card, SecureDigital, MMC, SD

Citace

Gazda Martin: Využití paměťové karty SD na platformě FITkit. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Využití paměťové karty SD na platformě FITkit

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Zdeňka Vašíčka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jméno Příjmení
Datum

Poděkování

Chtěl bych poděkovat Ing. Vašíčkovi za užitečné rady při řešení projektu a odbornou pomoc při vypracování této bakalářské práce.

© Martin Gazda, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah 1
Úvod 3
1 Paměťová karta Secure Digital 4
1.1 Stručný úvod k FLASH pamětem 4
1.1.1 Historie FLASH pamětí 4
1.1.2 Omezení FLASH pamětí 5
1.2 Stručný úvod k Secure Digital 6
1.2.1 Historie 6
1.2.2 Rychlost a možnosti připojení SD karty 6
1.2.3 Kapacita SD karet 7
1.2.4 Systém souborů 8
1.2.5 Otevřenost standardu 8
1.3 Komunikační protokoly 9
1.3.1 Režim SD Bus 9
1.3.2 SPI protokol 10
1.3.3 Výběr protokolu a inicializace karty 14
1.3.4 Zabezpečení přenášených dat 15
1.4 Popis registrů a připojení SD karty 16
1.4.1 Připojení SD karty 16
1.4.2 Registry 19
1.5 Operace čtení a zápisu dat 21
1.5.1 Čtení dat 21
1.5.2 Zápis dat 22
1.6 Časování 23
2 Navržené řešení 26
2.1 Platforma FITKit 26
2.1.1 Struktura FPGA 26
2.1.2 Připojení SD karty k platformě FITkit 27
2.2 Řadič karet typu SD 28
2.2.1 Implementace řadiče paměťové karty SD 29
2.2.2 Popis řadiče paměťové karty SD 30
2.2.3 Výsledky implementace komponenty 33
3 Demonstrační aplikace 34
3.1 Práce s paměťovou kartou 34

3.1.1 Funkce uživatelského rozhraní 35
Závěr 36
Literatura 37
Seznam příloh 39
Přílohy 40

Úvod

Myšlenka rekonfigurovatelného hardware není zdaleka nová. Sahá až do 60. let minulého století, kdy byly nastíněny principy počítače s programovatelným polem a procesorem řídícím jeho rekonfiguraci [2]. Tehdy byl ovšem tento nápad daleko za technologickými možnostmi doby. Velký rozmach této myšlenky a začátek jejího většího využívání byl zaznamenán až na konci 20. století spolu s radikálním rozvojem křemíkové technologie, který umožňoval umístnit komplexní architekturu na jeden čip.

V současnosti se velká část práce v oblasti rekonfigurovatelných obvodů zaměřuje na vývoj FPGA (Field Programmable Gate Array). Stejně jako většina ostatních zařízení tohoto typu se FPGA skládá z homogenních funkčních jednotek, jejichž rekonfigurace je řízena externí jednotkou. Tou je nejčastěji mikrokontroler.

Vývojová platforma FITkit se tak stává ideálním prostředkem pro vývoj aplikací pro tyto systémy, neboť obsahuje obě dvě výše zmiňované jednotky. Navíc dostatek vstupně výstupních portů a výpočetní kapacity ji předurčuje k vývoji aplikací komplexnějších a rozsáhlejších.

Paměťová náročnost většiny aplikací dnes neúměrně vzrůstá a také objemy dat, které aplikace produkují, či potřebují ke své činnosti se za poslední desetiletí několikrát znásobily. Potřeba úložiště pro tato data je tedy stále aktuální.

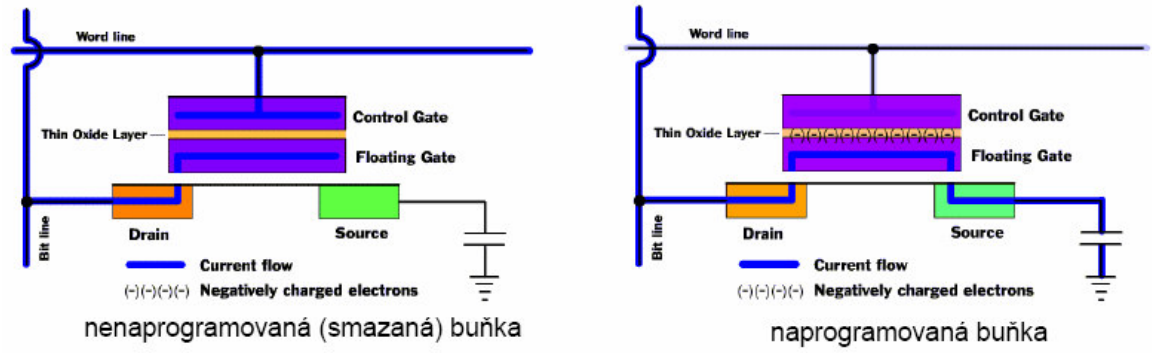
Mnoho aplikací vestavěných systémů jsou přenosná zařízení, náročná na dodržování co možní nejnižší spotřeby a na mechanickou odolnost. V těchto podmínkách přestávají klasická média svými parametry vyhovovat a především média obsahující mechanické části se dostávají na okraj zájmu.

Cílem této práce je tedy navrhnout řadič pro paměťovou kartu SD a její využití na platformě FITkit a rozšířit tak možnosti úložišť pro data aplikací o programovatelnou paměť typu FLASH. Dále vytvořit sadu knihovních funkcí pro mikrokontroler pro práci s tímto řadičem.

1 Paměťová karta Secure Digital

1.1 Stručný úvod k FLASH pamětem

Paměť FLASH je energeticky nezávislá paměť typu EEPROM a je tvořena maticí paměťových buněk, které leží na průsečících jednotlivých řádků a sloupců. Každá z buněk je tvořena unipolárním tranzistorem s dvěma hradly, která oddělena tenkou vrstvičkou oxidu (obrázek Obrázek 1.1). Horní hradlo je připojeno na sběrnici řádků, spodní hradlo je tzv. plovoucí, není připojeno nikam. Na toto spodní hradlo se náboj dostane z horního (řídícího) hradla tunelováním. Náboj je v tomto případě nositelem informace. Nenaprogramovaná paměťová buňka tedy nemá na plovoucím hradle žádný náboj, a proto nemůže dojít k otevření paměťového tranzistoru při přivedení výběrového signálu pro konkrétní naadresovaný řádek paměti. Paměťová buňka má v tomto případě hodnotu log. 1. Hodnotu logické 0 získáme programováním paměťové buňky, tedy vložením náboje do plovoucího hradla, které je pak schopné při výběru řádku paměti tranzistor této buňky otevřít. Mazání FLASH pamětí spočívá v odvedení tohoto náboje. [1]



Obrázek 1.1 Paměťová buňka paměti typu FLASH

Způsob mazání FLASH paměti, spolu s rychlostí přístupu k datům, nezávislosti na napájení a odolnosti proti otřesům, se staly hlavním důvodem vyššího nasazování tohoto typu paměti i v aplikacích, kde se dříve uplatňovaly klasické paměti typu EEPROM a pevné disky. Hlavní nevýhodou pamětí EEPROM bylo mazání paměti po malých blocích, typicky bajtech. Zvýšením velikosti těchto bloků u pamětí FLASH se dosáhlo značného zvýšení rychlosti, zvláště při práci s velkými objemy dat, přestože samotný čas potřebný pro vymazání jednoho bloku se příliš nezměnil. [2]

1.1.1 Historie FLASH pamětí

Paměti typu FLASH (NOR i NAND) objevil v roce 1984 zaměstnanec Toshiby Dr. Fujio Masuoka. Pojmenovat nově objevenou paměť „FLASH“ bylo pak nápadem jeho kolegy Shoji Ariizumi,

kterému rychlost procesu mazání připomínala blesk fotoaparátu (angl. flash). První představení vynálezu proběhlo v San Francisku, Kalifornie na konferenci IEEE 1984 International Electron Devices Meeting (IEDM) v roce 1984.

První komerční prezentace paměti typu FLASH proběhla v roce 1988 firmou Intel. Jednalo se o typ NOR FLASH. Přestože NOR FLASH potřebovaly delší dobu pro zápis i mazání dat, poskytovaly plnou šířku adresové i datové sběrnice, čímž umožňovaly přistoupit na libovolné místo paměti. Tyto vlastnosti je předurčily pro roli nástupce starších ROM čipů, které se využívaly v aplikacích, kde změna uložených dat probíhala zřídka, ne-li vůbec, např. počítačový BIOS, firmware set-top boxu, apod. Zaručovaly přibližně 100000 přepsání. V počátcích paměťových karet se využívaly např. u typu CompactFlash, později byly nahrazeny levnějšími NAND FLASH.

FLASH paměť typu NAND byla představena Toshibou v roce 1989 na ISSCC. Vykazovala kratší dobu zápisu i mazání a pro stejnou kapacitu vyžadovala menší místo na čipu, což umožnilo větší hustotu paměťových prvků a tedy i menší náklady na jeden bit. Tato paměť již neposkytuje plnou šířku adresové sběrnice, data je tedy nutné číst v blocích, přičemž typická velikost bloku jsou stovky až tisíce bitů. V tomto přístupu k datům je velmi podobná sekundárním pamětem, jako jsou pevný disk a optická média. První paměťovou kartou založenou na paměti NAND FLASH je karta SmartMedia. [2]

1.1.2 Omezení FLASH pamětí

1.1.2.1 Blokové mazání

Hlavní konkurenční výhoda FLASH paměti (blokově orientovaný přístup) je bohužel i jedním z hlavních omezení při práci s tímto typem paměti. Na rozdíl od čtení či zápisu dat, které je možné provádět po bajtech nebo slovech, je nutné při mazání vždy vynulovat celý blok, což nejčastěji znamená nastavit všechny bity bloku na hodnotu logická 1. Jinými slovy, paměť typu FLASH (konkrétně NOR FLASH), umožňuje čtení a zápis libovolného místa paměti, neumožňuje už ale jeho vymazání nebo přepsání (s výjimkou případů, kdy změna uložených dat ovlivní pouze bity původně v log. 1, změna log.1 -> log. 0, např. 1111 -> 1001).

1.1.2.2 Omezený počet přepsání

Dalším omezením při nasazení FLASH pamětí je konečný počet přepsání (většina výrobců garantuje hodnotu 100000 úspěšných přepsání pro nultý blok). Toto omezení kompenzují někteří výrobci pomocí firmware nebo souborového systému, který si udržuje informaci o počtu zápisů do jednotlivých bloků a snaží se rozložit zátěž i na další bloky. Tato a další techniky prodlouží životnost FLASH paměti většinou dostatečně v porovnání s životností zařízení ve kterém jsou nasazeny. I když v řadě aplikací FLASH pamětí nejsou minimální ztráty dat závažným problémem, není vhodné

paměti typu FLASH nasazovat v aplikacích, které jsou kriticky závislé na správnosti uložených dat a které předpokládají jejich častou změnu.

1.2 Stručný úvod k Secure Digital

Paměťová karta Secure Digital (SD) je paměťová karta vytvořená pro potřeby přenosných audio a video zařízení. Tomu odpovídají i její parametry z hlediska bezpečnosti, kapacity, výkonu i spotřeby. Byla vyvinuta firmami Matsushita, SanDisk a Toshiba a dnes je využívána především v digitálních fotoaparátech, mobilních telefonech, mobilních zařízeních apod. Kapacita paměti se pohybuje v rozmezí od 8MB až 4GB (specifikace 1.0 standardní karta SD) a 4GB až 32GB v případě SDHC (specifikace 2.0, vysoko kapacitní SD).

1.2.1 Historie

V srpnu roku 1999 firmy Matsushita, SanDisk a Toshiba poprvé ohlašují svůj záměr vytvořit paměťovou kartu nové generace nazvanou Secure Digital. Karta s rozměry 24mm x 32mm x 2,1mm se měla díky vysokému stupni ochrany autorských práv a vysoké hustotě paměťových prvků stát přímým konkurentem paměťové kartě Memory Stick firmy Sony, která byla na trhu již od roku 1998.

Název paměťové karty Secure Digital vznikl při samotné tvorbě, neboť karta vznikla rozšířením již existující karty MMC (Multimedia Card) o šifrovací obvody. Ty byly přidány na popud hudebního průmyslu, který se obával šíření pirátských nahrávek. Ačkoliv mělo šifrování přispět k zabezpečení obsahu proti nelegálnímu šíření, v praxi je využíváno spíše sporadicky.

První velké nasazení karet SD přichází v roce 2000, kdy konsorcium 30 významných firem počítačového a elektronického průmyslu vydává paměťové karty SD s kapacitou 32 a 64 MB.[3]

1.2.2 Rychlost a možnosti připojení SD karty

Paměťová karta SD, stejně jako její předchůdce MMC, podporuje více způsobů připojení a řadu rychlostí. Obě paměťové karty umožňují připojení pomocí rozhraní SPI. Toto rozhraní používá čtyři signály: DATin, DATout, CLK a CS. Výhodou rozhraní SPI je, že se stalo standardní periferií většiny mikrokontrolerů.

Kromě SPI podporuje SD karta komunikace využívající čtyřbitové nebo jednobitové obousměrné sběrnice. V tomto režimu se používají signály: CLK, CMD a DAT v jednobitovém módu a CLK, CMD a DAT[0:3] pro čtyřbitový mód (detaily viz. [4]).

Maximální rychlosti přenosu při blokovém (burst) režimu (rychlost přenosu mezi vyrovnávací pamětí karty a hostem) je závislá na frekvenci připojených hodin a na zvoleném připojení paměťové karty, souhrn podává tabulka Tabulka 1.1.

Paměťová karta	*Maximální frekvence*	*Maximální rychlost přenosu*
MultiMedia Card		
SPI Bus mode	20 MHz	2,5 MB/s
MMC 1-bit mode	20 MHz	2,5 MB/s
SD Card		
SPI Bus mode	25 MHz	3,125 MB/s
SD 1-bit mode	25 Mhz	3,125 MB/s
SD 4-bit mode	25 Mhz	12,5 MB/s

Tabulka 1.1: Maximální přenosové rychlosti a frekvence karet MMC a SD

1.2.2.1 High-Speed Mode

Ačkoliv přenosová rychlost dosahovala až 12,5 MB/s, byla ve verzi 2.0 navýšena na 25 MB/s. Důvodem k tomuto nárůstu rychlosti byla zvyšující se kapacita paměťových karet a také zvyšující se požadavky výrobců na propustnost těchto karet. Zvýšení přenosové rychlosti se dosáhlo zvýšením frekvence na 50MHz. Z důvodu kompatibility jsou paměťové karty SD verze 2.0 jsou po připojení napájecího napětí nastaveny na standardní rychlost a mohou být pomocí příkazu přepnuty do vysokorychlostního módu [4].

1.2.3 Kapacita SD karet

Podle kapacity můžeme paměťové karty SD rozdělit do dvou kategorií a to na karty s nižší a vysokou kapacitou (standard SDHC). Standardní kapacita, která je definována ve všech specifikacích SD karty, uvažuje maximální kapacitu paměti do hodnoty 2GB včetně. Karty s vysokou kapacitou jsou podle verze 2.0 definovány v rozsahu od 2GB do 32GB, přičemž horní hranice kapacity je uměle omezena, teoreticky by bylo možné dosáhnout kapacity až 2048GB.

Ačkoliv SDHC karty mají stejné rozměry i rozložení konektorů, a jsou tedy snadno zaměnitelné se starším typem, nejsou kompatibilní a starší zařízení s nimi nedokáží pracovat. Naopak, pokud je zařízení schopné pracovat s kartami SDHC, je kompatibilita směrem dolů zachována [4].

1.2.3.1 Problém kompatibility u kapacit vyšších než 2GB

Zařízení, která používají SD kartu, identifikují každou paměťovou kartu pomocí 128 bitového identifikačního řetězce, který zasílá karta při inicializaci. U SD karet standardní kapacity je 12 bitů určeno k určení počtů paměťových clusterů v rozsahu od 1 do 4096, 3 bity jsou určeny pro definování počtu bloků na cluster, které se dekódují jako 4 8 16 32 64 128 256 nebo 512 bloků na cluster.

Podle starší specifikace verze 1.x byla velikost bloku přesně 512 bajtů, což určovalo maximální kapacitu 4096 x 512 x 512 = 1 GB paměti. Pozdější verze standardu 1.x umožňovaly pro určení velikosti bloku využít 4 bity, tedy 1024 bajtů, což umožnilo pracovat s paměťovými kartami do

velikosti 2GB. Bohužel zařízení navržená před touto změnou mylně zaměňovala karty s kapacitou 2GB za menší 1GB. Někteří výrobci se snažili kapacitu SD karet zvyšovat zvýšením velikosti bloku na 2048B, tedy na celkovou kapacitu 4GB. Tato modifikace však nebyla součástí specifikace 1.x, a proto tyto karty pracovaly pouze v zařízeních stejného výrobce.

Nová specifikace verze 2.0 pro SD karty s vysokou kapacitou (SDHC) změnila přístup podle kterého se určuje kapacita karty. Pro určení velikosti je použito 22bitů identifikačního řetězce, přičemž v současné době je povoleno využít nižších 16 bitů. Jelikož přírůstek odpovídá 512kB, je v současnosti maximum 32GB. Všechny karty s kapacitou 4GB a vyšší musí nutně splňovat specifikaci 2.0. Dva bity, které byly původně rezervovány jsou nyní využity pro rozlišení typu karty, 00= standard, 01= HC, 10 = rezervováno, 11= rezervováno.

1.2.4 Systém souborů

Jako většina paměťových karet, bývají i SD karty od výrobce naformátovány systémem souborů FAT nebo FAT32. Použití nejrozšířenějšího systému souborů umožňuje práci s kartou v podstatě na jakémkoliv zařízení vybaveném čtečkou paměťových karet typu SD. Výhodou je také možnost použití standardních nástrojů pro práci se systémem FAT (např. ScanDisk) pro opravu nebo obnovu poškozených dat. Ve většině systémů je karta připojena obdobně jako výměnný harddisk, může být tedy naformátována na libovolný systém souborů, který podporuje daný systém, jako např. UFS2/ext2.

Zajímavostí zůstává využití nástrojů pro defragmentaci. Ačkoliv je možné tyto nástroje použít i pro paměťovou kartou SD, nemá proces defragmentace žádný přínos. Defragmentace slouží k optimalizaci přístupu k datům uloženým na zařízeních s otáčejícími se disky a pohybujícími se hlavami, avšak přístup k datům uloženým na pamětí FLASH je čistě náhodný, a tak procesem defragmentace nezískáváme žádné zvýšení výkonu. Naopak, přesouváním dat se snižuje životnost celé karty kvůli omezenému počtu zápisových cyklů.

Paměťové karty SDHC jsou ve většině případů formátovány systémem souborů FAT32.

1.2.5 Otevřenost standardu

K dnešnímu dni existují tři specifikace SD paměťových karet, verze 1.0, 1.1 a 2.0. Získání specifikace SD karty bylo původně podmíněno uzavřením licenčních smluv a placením poplatků. Tyto licenční smlouvy zakazovaly vývoj open-source ovladačů, což vyvolávalo nevoli v komunitě open-source vývojářů, a tak byla uvolněna zjednodušená verze specifikace, kterou je možné získat volně z webových stránek. Přestože většina ovladačů byla napsána před uvolněním specifikace, přispěly tyto informace k vyřešení problémů s kompatibilitou v některých specifických případech.

V roce 2006 vydala asociace SD dokument SD Specification - Simplified Physical Layer Specification (viz. [4]). V tomto dokumentu jsou popsány především vlastnosti a použité protokoly.

Naopak samotná fyzická specifikace, možnosti připojení, časování, popis funkcí připojení a odpojení za běhu, jsou maximálně zjednodušeny nebo vynechány. Pro informace tohoto typu je nezbytné využívat dokumenty vydané firmami Toshiba [5] nebo SanDisk [6].

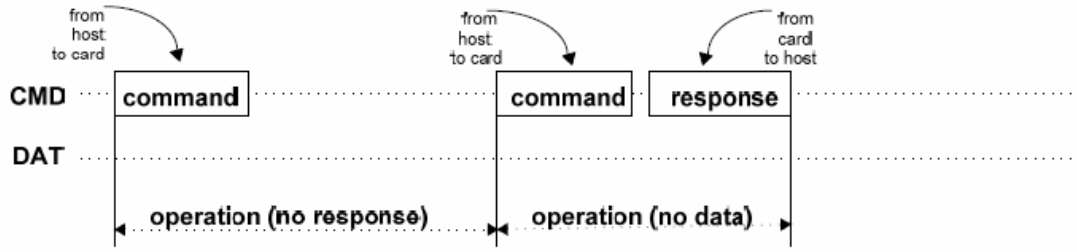
1.3 Komunikační protokoly

1.3.1 Režim SD Bus

Komunikace pomocí protokolu SD Bus je založena na příkazech a sledech bitů, které začínají startbitem a končí stopbitem.

- **Příkaz:** příkaz je sled bitů, který začíná operaci. Je zasílán zařízením jedné konkrétní kartě (adresovaný příkaz) nebo všem připojeným kartám. Příkaz je přenášen sériově po CMD lince.
- **Odpověď:** odpověď je sled bitů, kterou posílá v dané chvíli naadresovaná karta, nebo kterou posílají synchronně všechny připojené karty jako odpověď na dříve přijatý příkaz. Odpověď je přenášena sériově po CMD lince.
- **Data:** data jsou přenášena od z karty do zařízení a nebo opačně. Data jsou přenášena pomocí datových linek.

Adresování jednotlivých paměťových karet se provádí pomocí adres přidělených pro danou relaci během inicializační fáze. Základní operací je transakce příkaz-odpověď, viz obrázek Obr 1.2. Tento typ transakce nese informaci v těle příkazu a odpovědi. Některé operace obsahují, případně očekávají další data.



Obr 1.2 Transakce příkaz a odpověď (převzat z [4])

Datové přenosy do/z paměťové karty jsou prováděny po blocích. Za datovým blokem vždy následují bity kontrolního součtu CRC. Jsou definovány jednoblokové i víceblokové přenosy dat. Přenosy více bloků jsou výhodnější z hlediska rychlosti, musí být ale ukončeny příkazem stop přenášeným přes CMD linku. Přenosy dat je možné realizovat pomocí jedné, nebo více datových linek.

1.3.2 SPI protokol

SPI protokol je dalším komunikačním protokolem, který paměťová karta SD podporuje. Při komunikaci využíváme podmnožinu příkazů protokolu SD Bus. Komunikaci pomocí protokolu SPI je nutné zvolit zasláním speciálního příkazu ihned po připojení napájecího napětí a není možné ji změnit aniž bychom paměťovou kartu znovu inicializovali.

Výhodou SPI protokolu je jeho snadná implementace a možnost využít již vytvořených SPI řadičů. Nevýhodou je ztráta výkonnosti oproti SD Bus módu (viz tabulka Tabulka 1.1).

Komunikace pomocí SPI je orientována bajtově. Každý příkaz nebo datový blok je násobkem osmi bitů a je bajtově zarovnán vzhledem k signálu CS. Hodiny SPI se začínají počítat nastavením signálu CS a každý příkaz nebo datový blok musí začínat na osmi násobku hodinového taktu.

SPI protokol se skládá z příkazů, odpovědí a datových bloků. Veškerá komunikace s paměťovou kartou je řízena hostem. Host začíná každou transakci nastavením signálu CS na hodnotu logická 0.

Nyní si uveďme některé vlastnosti, které odlišují SPI režim od režimu SD Bus. Připojená karta vždy odpovídá na zaslaný příkaz a pokud při čtení dat z karty dojde k chybě, karta odpovídá odpovědí s předem definovaným formátem. Tato odpověď tedy nahrazuje očekávaná data. Během operace zápisu je každý přijatý blok potvrzen odpovědí s předem definovanou strukturou, která obsahuje i hlášení o chybách během zápisu.

Pokud pracujeme s kartou standardní kapacity, pak není velikost datového bloku omezena, může být v rozsahu jediného bajtu i plné velikosti bloku. Operace čtení a zápisu nedokončených bloků je možno zakázat, povolit v řídícím registru CSD. V případě práce s SDHC kartou je pak velikost datového bloku pevně dána a to 512 bajtů, operace čtení a zápisu menších velikostí než je velikost bloku jsou zakázány.

1.3.2.1 Formát příkazu

Každý příkaz začíná start bitem (log. 0) a končí stopbitem (log. 1). Celková délka příkazu je 48 bitů. Každý příkaz obsahuje kontrolní součet, takže chyba, která vznikla během přenosu může být detekována a operaci je možné opakovat. První je vysílán nejvíce významný bit. Význam jednotlivých bitů je uveden v tabulceTabulka 1.2.

Pozice v příkazu	**47**	**46**	**[45:40]**	**[39:8]**	**[7:1]**	**0**
Šířka (bity)	1	1	6	32	7	1
Hodnota	‘0’	‘1’	x	x	x	‘1’
Popis	start bit	směr přenosu	kód příkazu	argument	CRC7	stop bit

Tabulka 1.2: Formát příkazu

Základní příkazy pro práci s paměťovou kartou SD shrnuje tabulka Tabulka 1.3. Speciálním typem příkazů jsou aplikační příkazy (ACMD XX). Těmto příkazům musí vždy předcházet APP_CMD(CMD55), který určuje, že kód následujícího příkazu je nutné interpretovat ne jako standardní příkaz, ale právě jako aplikačně specifický příkaz.

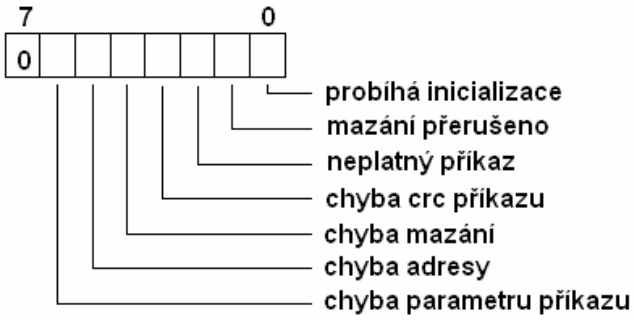
Příkaz	**Název**		**Argument**	**Typ odpovědí**	**Popis**
CMD0	GO_IDLE_ STATE	00h	žádný	R1	Reset karty, výběr kom.protokolu
CMD9	SEND_CSD	09h	žádný	R1	Čtení registru CSD
CMD16	SET_BLOCKLEN	10h	[31:0]velikost bloku	R1	Nastavuje velikost bloku v bajtech pro operace čtení a zápisu.
CMD17	READ_SINGLE_ BLOCK	11h	[31:0] adresa	R1	Čtení bloku o velikosti dané příkazem SET_BLOCKLEN
CMD24	WRITE_BLOCK	18h	[31:0] adresa	R1	Zápis bloku o velikosti dané příkazem SET_BLOCKLEN
CMD55	APP_CMD	37h	žádný	R1	Určuje že příští příkaz je aplikačně specifický
CMD58	READ_OCR	3Ah	žádný	R3	Čtení registru OCR
CMD59	CRC_ON_OFF	3Bh	[31:1] volné [0:0] CRC on/off	R1	Zapnutí, vypnutí CRC
ACMD41	SD_SEND_OP_ COND	29h	žádný	R1	Inicializace karty

Tabulka 1.3: Základní příkazy protokolu SPI

1.3.2.2 Formát odpovědi

Existuje pět formátů odpověďi. Délka odpovědi se pohybuje mezi 8 až 136 bity. Většina odpovědí je zabezpečena kontrolním součtem CRC, přičemž datové bloky jsou vždy zabezpečeny 16bitovým CCITT polynomem.

- odpověď R1



Odpověď R1 zasílá karta hostu po každém příkazu s výjimkou příkazu SEND_STATUS(viz [4]). Nejvíce významný bit je vždy 0, ostatní bity jsou příznaky chyb, aktivní v log. 1. Význam příznakových bitů je následující:

Probíhá inicializace – probíhá inicializace karty, po tuto dobu přijímá karta pouze příkazy CMD0, CMD58, CMD59 a ACMD41

Mazání přerušeno – proces mazání karty byl přerušen, nedošlo k smazání požadované oblasti paměti

Neplatný příkaz – přijat neplatný příkaz

Chyba CRC příkazu – kontrolní součet posledního zaslaného příkazu byl chybný

Chyba mazání – během procesu mazání došlo k neočekávané chybě, např. snaha smazat data chráněná proti přepsání

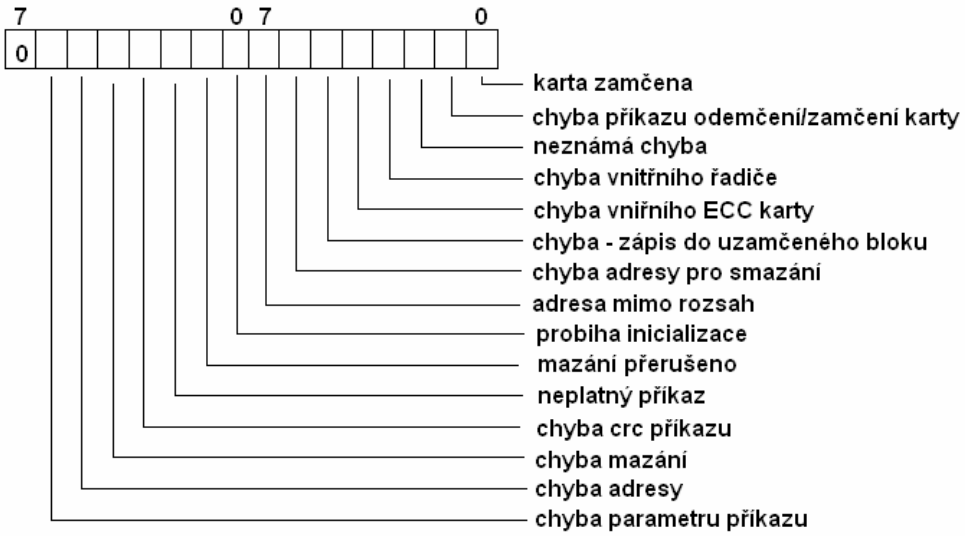
Chyba adresy – adresa zaslaná jako argument posledního příkazu není platná – mimo rozsah adres

Chyba parametru příkazu – argument příkazu (adresa, délka bloku, apod.) je mimo rozsah povolený kartou

- odpověď R1b – rozšíření odpovědi R1, přidává přítomnost busy signálu na DataOut

  Odpověď R1b se využívá v režimu SD Bus, kde hodnotou logická 0 karta indikuje zaneprázdněnost (např. probíhající zápis do paměti FLASH apod.).

- odpověď R2



Odpověď R2 je rozšířením odpovědi R1, horní bajt je odpověď formátu R1. Význam přidaných bitů je následující:

Chyba adresy pro smazání – adresa příkazu mazání neodpovídá adrese žádného ze sektorů

Chyba – zápis do uzamčeného bloku – poslední příkaz zápisu dat do paměti se pokusil zapisovat do bloku chráněného proti přepsání

Chyba vnitřního ECC karty – oprava chyby pomocí ECC selhala

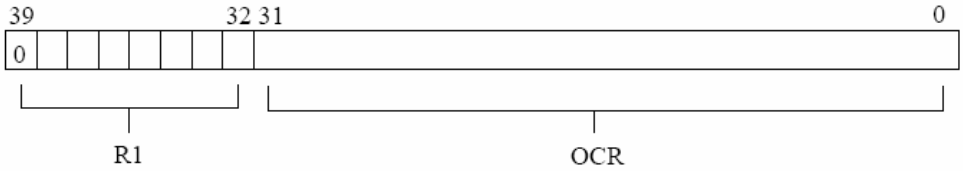
Chyba vnitřního řadiče – při práci vnitřního řadiče karty nastala obecná chyba

Neznámá chyba – během práce s kartou došlo k neznámé chybě

Chyba příkazu odemčení/zamčení karty – chybné heslo pro odemčení/zamčení karty

Karta zamčena – karta je mechanicky zamčena vůči přepsání pomocí přepínače na boku karty

- odpověď R3 – zaslána při čtení registru OCR

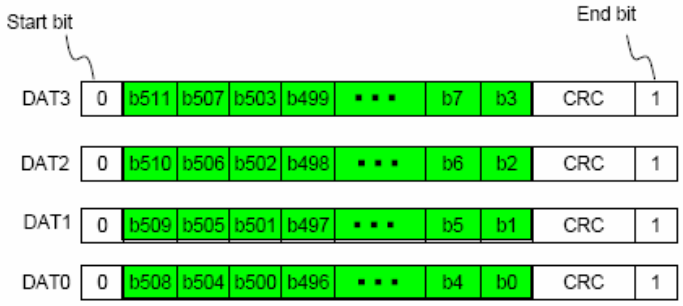


- odpověď R7 – zaslána po příkazu CMD8

Odpověď R7 je zaslána vždy po přijetí příkazu SEND_IF_COND (CMD8). Argument příkazu obsahuje část pro určení aktuálního napájecího napětí hostu. Pokud karta tento rozsah napájecího napětí přijme, zašle hostu, jako součást odpovědi právě tuto část argumentu. Tento příkaz existuje u karet specifikace 2.0 a vyšších.

1.3.2.3 Formát dat

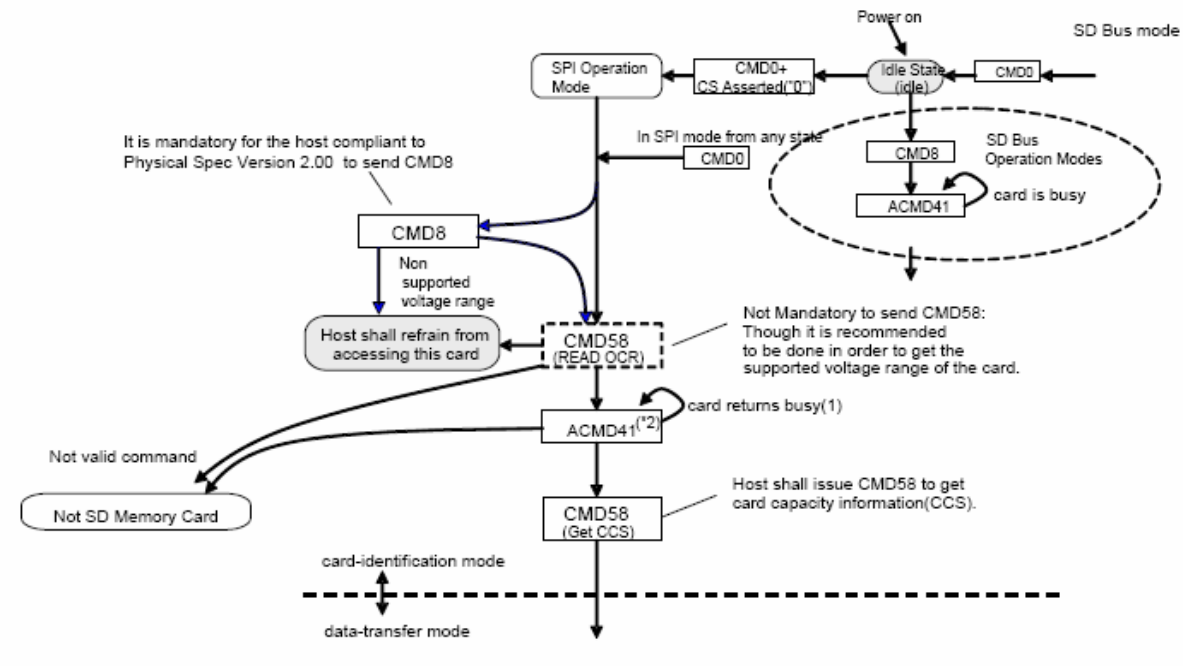
V případě, že jsou pro přenos využity čtyři datové linky, přenášejí se vždy 4 bity současně (viz obrázek Obrázek 1.3). Startbit a stopbit, stejně jako kontrolní součet CRC jsou přenášeny pro každou linku nezávisle. Také vlastní kontrola CRC součtu je prováděna pro jednotlivé linky. Naopak stav CRC součtu nebo příznak zaneprázdnění (busy) jsou posílány pouze po datové lince DAT0 (DAT1 až DAT3 jsou v tomto okamžiku ve stavu vysoké impedance).



Obrázek 1.3 Přenos dat pomocí čtyř datových linek (převzat z [4])

1.3.3 Výběr protokolu a inicializace karty

Po připojení napájecího napětí se paměťová karta nachází v SD Bus módu (Obrázek 1.4). Přepnutí do režimu SPI se provádí zasláním příkazu reset (CMD0) a současným nastavením pinu CS do log. 0. Po úspěšném přepnutí zašle karta odpověď tvaru R1. Jediným způsobem jak se vrátit do SD Bus módu je odpojením a opakovaným připojením napájení.



Obrázek 1.4 Přepnutí paměťové karty SD do SPI módu a inicializace (převzat z [4])

V první fázi se příkazem SEND_IF_COND (CMD8) se ověřuje verze připojené paměťové karty. Pokud je v odpovědi na tento příkaz nastaven bit Illegal Command (neplatný příkaz), jedná se o SD kartu vyrobenou podle specifikace 1.X a nebo se nejedná o kartu SD. Pokud tento příznak nastaven není, je připojená SD karta vyrobena podle specifikace 2.0. Součástí příkazu SEND_IF_COND je také volba rozsahu napájení, host nastavuje aktuální připojené napájení. Odpověď pak potvrdí přijetí tohoto rozsahu napájecích napětí kartou, nebo toto nastavení odmítne. V takovém případě není možné s kartou pracovat. Součástí příkazu CMD8 je řetězec (check pattern), který slouží k ověření, zda příkaz proběhl správně a načtená odpověď je validní.

Poté následuje příkaz READ_OCR (CMD 58). Ten má za úkol znovu ověřit rozsah napájecího napětí, ve kterém karta pracuje.Pokud karta odmítne rozsah napájecího napětí zadaný hostem, není možné dále pokračovat v inicializaci.

Příkaz SD_SEND_OP_COND (ACMD 41) spouští vlastní proces inicializace. V argumentu tohoto příkazu je možné zaslat kartě informaci o podpoře SDHC (příznakem HCS). Ověření, zda je karta vysokokapacitní, se provádí pomocí příkazu READ_OCR, jehož odpověď obsahuje příznakový bit CCS(pokud je 1, jedná se o SDHC). O probíhajícím procesu inicializace karta informuje nastavením příznakového bitu „in idle state“, který je součástí odpovědi příkazu

SD_SEND_OP_COND. Host opakovaně zasílá příkaz, dokud není příznak zrušen. Tím je proces inicializace úspěšně dokončen.

1.3.4 Zabezpečení přenášených dat

V SPI módu umožňuje SD karta zabezpečit přenášená data a příkazy kontrolním součtem CRC. Výchozím režimem je SPI mód bez zabezpečení. V SD Bus mód je však použití zabezpečení vyžadováno a proto první příkaz, přepnutí do SPI módu (CMD0), který je vykonáván ještě v tomto režimu, je nutné zabezpečit CRC kódem. Jelikož je CRC kód pro příkaz CMD0 pevně daný, je vhodné jej uložit jako konstantu (CMD0 = 0x40, 0x0, 0x0, 0x0, 0x0, CRC7 = 0x95). K zapnutí ověřování přenášených dat pomocí CRC slouží příkaz CRC_ON_OFF(CMD59). Ten musí být proveden před prvním příkazem SD_SEND_OP_CODE(ACMD41).

1.3.4.1 CRC7

Cyklický redundantní součet (CRC7) se používá pro zabezpečení příkazů a počítá se na základě následujícího předpisu.

Algoritmus 1.1:

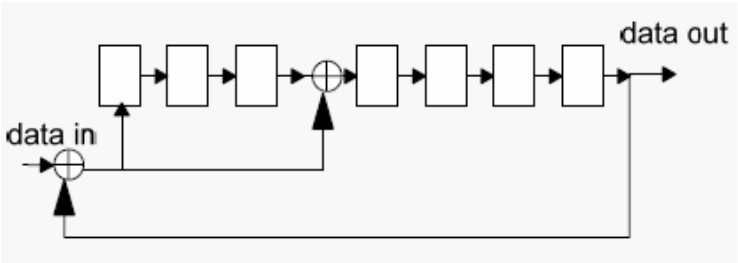
polynom: G(x) = x7 + x3 + 1

M(x) = (první bit) * x^n + (druhý bit) * x^{n-1}+…+ (poslední bit) * x^0

CRC[6..0] = Zbytek [M(x) * x^7 / G(x)]

První bit je nejvíce významný bit dané posloupnosti bitů (příkazu, CSD registru,…). Stupeň polynomu *n* je počet bitů, které budou polynomem zabezpečeny snížený o jedničku, tedy např. pro příkaz s délkou 40 bitů je n = 39.

Samotný výpočet probíhá sériově jak ilustruje obrázek Obrázek 1.5. Tento přístup je jistě vhodnější než paralelní výpočet, neboť odpadá nutnost ukládat data a také doba výpočtu je nesrovnatelně kratší.



Obrázek 1.5 Sériový způsob výpočtu CRC7

1.3.4.2 CRC16

Přenosy dat SD karty jsou zajištěny standardním CCITT polynomem. Výpočet CRC16 se řídí tímto vztahem:

Algoritmus 1.2:

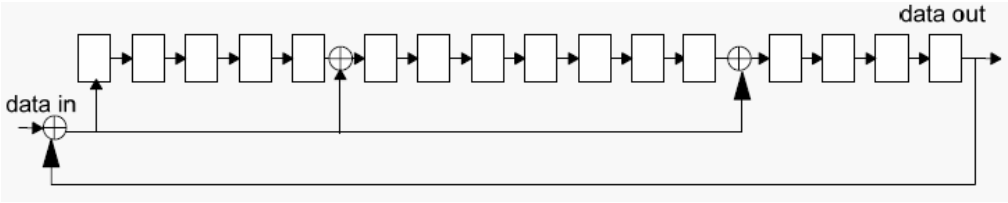
polynom: G(x) = x16 + x12 + x5 + 1

M(x) = (první bit) * x^n + (druhý bit) * x^{n-1}+…+ (poslední bit) * x^0

CRC[6..0] = Zbytek [M(x) * x^{16} / G(x)]

První bit je nejvíce významný bit přenášeného bloku dat. Stupeň polynomu *n* je počet bitů, které budou polynomem zabezpečeny snížený o jedničku, tedy např. pro velikost bloku 512 bajtů je n = 4095. Tento způsob zabezpečení je možné použít pro délku bloku do 2048 bajtů.

Pro výpočet bylo stejně jako u CRC7 využito sériového zapojení (obrázek Obrázek. 1.6).

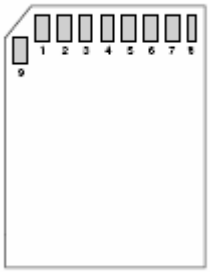


Obrázek. 1.6 Obvod pro výpočet zabezpečení CRC16

1.4 Popis registrů a připojení SD karty

1.4.1 Připojení SD karty

Paměťová karta SD má devět pinů, jsou to CD/DAT3, CMD, V_{SS1}, VDD, CLK, V_{SS2}, DAT0, DAT1 a DAT2 (viz Obrázek. 1.7). Počet připojených pinů a jejich význam závisí na volbě komunikačního protokolu.



Obrázek. 1.7 Layout SD karty

1.4.1.1 SD Bus

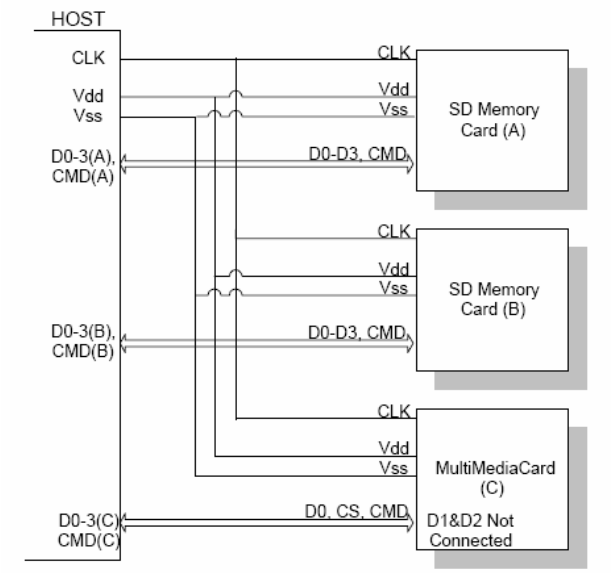
V režimu SD Bus jsou využity všechny piny SD karty. Význam jednotlivých pinů je uveden v tabulce Tabulka 1.4.

Pin	Jméno	Typ	Popis
1	CD/DAT3	I/O	Detekce karty/Datová linka [bit 3]
2	CMD	I/O	Příkaz/Odpověď
3	V_{SS1}	S	Zem napájení
4	V_{DD}	S	Napájení
5	CLK	I	Hodiny
6	V_{SS2}	S	Zem napájení
7	DAT0	I/O	Datová linka [bit 0]
8	DAT1	I/O	Datová linka [bit 1]
9	DAT2	I/O	Datová linka [bit 2]

Tabulka 1.4: SD Bus – funkce jednotlivých pinů

Připojení více zařízení k jednomu HOSTu

Během inicializace jsou jednotlivým kartám přiřazeny logické adresy, příkazy jsou zasílány každé kartě samostatně. Po ukončení inicializační fáze se běžně zasílají příkazy po společné sběrnici, přičemž adresa konkrétní karty je součástí těla příkazu. Data jsou vždy přenášena po samostatných vodičích konkrétní karty (Obrázek. 1.8).



Obrázek. 1.8 Topologie SD Bus módu, inicializace (převzat z [4])

1.4.1.2 SPI

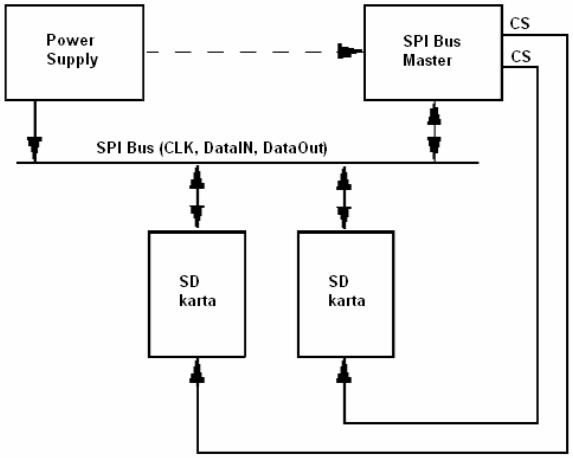
V režimu SPI nejsou využity všechny piny SD karty, ale pouze pět pinů. Kromě pinů pro připojení napájecího napětí (V_{SS1}, VDD, V_{SS2},), jsou to piny CS, DataIn, CLK a DataOut. Význam jednotlivých pinů je uveden v tabulce Tabulka 1.5.

Pin	Jméno	Typ	Popis
1	CS	I	Výběr karty
2	DataIn	I	Vstup pro příkazy/data
3	V_{SS1}	S	Zem napájení
4	V_{DD}	S	Napájení
5	CLK	I	Hodiny
6	V_{SS2}	S	Zem napájení
7	DataOut	O	Výstup pro odpovědi/data
8	RSV	I	rezervován
9	RSV	I	rezervován

Tabulka 1.5: SPI – funkce jednotlivých pinů

Připojení více zařízení k jednomu HOSTu

V režimu SPI je adresování realizováno pomocí signálu CS. Paměťová karta je vybrána nastavením tohoto signálu na log. hodnotu 0 a tato hodnota je udržována po celou dobu práce s kartou. Jednou výjimkou z tohoto pravidla je okamžik zápisu dat do FLASH paměti paměťové karty, kdy je možné se od karty odpojit nastavením signálu CS do log. 1. Každé zařízení proto musí mít vlastní CS signál (Obrázek. 1.9).



Obrázek. 1.9 Topologie SPI módu (převzat z [4])

1.4.2 Registry

Paměťová karta SD obsahuje šest registrů – OCR (Operation Condition Register), CID (Card identification number), CSD(Card specific data:), RCA(Relative card address), DSR(Driver Stage Register) a SCR (SD Configuration Register). S registry je možné manipulovat pouze pomocí příslušných příkazů. Registry OCR, CID, CSD a SCR obsahují obecné konfigurační informace a informace specifické pro konkrétní paměťovou kartu. Registry DSR a RCA obsahují aplikačně specifické informace, např. v režimu SD Bus je v registru RCA uložena relativní adresa karty. Obsahují také bity pomocí kterých je možné rozšířit funkcionalitu karty. V následující části je uveden stručný přehled, podrobnější informace lze nalézt v [4] [4].

1.4.2.1 OCR registr

Tento 32bitový registr uchovává informaci o možnostech napájení paměťové karty. Pokud konkrétní SD karta podporuje práci v daném rozsahu napájecích napětí, je v poli odpovídající tomuto rozsahu přiřazena logická hodnota 1. Dále registr obsahuje příznaky ukončení procesu inicializace a příznak vysokokapacitní SD karty.

1.4.2.2 CID registr

Tento 128bitový registr obsahuje identifikační údaje karty potřebné během identifikační fáze po připojení karty k hostitelskému zařízení.

1.4.2.3 CSD registr

Obsah tohoto 128bitového registru je závislý na verzi paměťové karty, obsahuje např. informace o maximální přístupové době, maximální velikosti bloku s kterými je karta schopná pracovat, maximální odebíraný proud apod. Můžeme zde nalézt bity s následujícím významem:

- **TAAC, NSAC**

  Definuje asynchronní část časování přístupu k datům. Spolu s parametrem NSAC určuje TAAC čas potřebný pro čtení dat (doba mezi přijetím odpovědi typu R1 a prvním bitem čtených dat), definují maximální dobu mezi příkazem pro čtení dat a vystavením prvního bitu odpovědi.

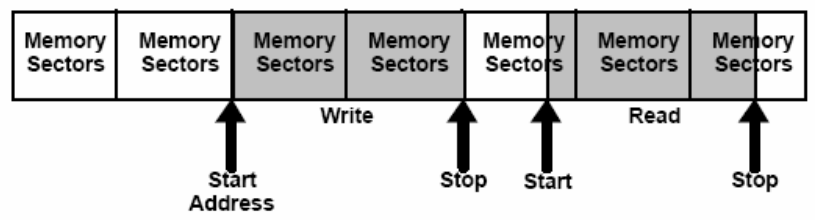
- **READ_BL_LEN, WRITE_BL_LEN, READ_BL_PARTIAL, WRITE_BL_PARTIAL**

  Parametr READ/WRITE_BL_LEN určuje maximální velikost bloků po kterých mohou být data čtena a pohybuje se v rozsahu 512 až 2048 bajtů. Hodnoty READ_BL_LEN a WRITE_BL_LEN jsou u SD karty vždy shodné, přesto že se jedná o dva nezávislé parametry.

Parametr READ/WRITE_BL_PARTIAL určuje, zda je možné pracovat i s bloky menšími než je fyzická velikost bloku paměti FLASH(typicky 512B), přičemž nejmenší možná velikost bloku je jeden bajt.

- **WRITE_BLK_MISALIGN, READ_BLK_MISALIGN**

Tyto dva parametry určují zarovnání zapisovaných, respektive čtených bloků a určují zda může zapisovaný, čtený blok přesahovat fyzický blok paměti (viz Obrázek 1.10). Tyto parametry jsou potenciálně velmi nebezpečné, mohou vést ke špatné interpretaci čtených dat, neboť hostitelské zařízení nemusí vždy znát zarovnání konkrétního bloku a může dojít při zápisu k přepsání cizích dat.



Obrázek 1.10 Zarovnání dat v paměti (převzato z [5])

- **ERASE_BLK_LEN**

Tento parametr určuje po kolika násobcích velikosti bloku probíhá mazání SD karty. Pokud je tato hodnota nastavena na 0, probíhá mazání od začátku sektoru po jeho konec, velikost sektoru určuje parametr SECTOR_SIZE.

1.4.2.4 RCA registr

Jedná se o 16bitový registr, udržující relativní adresu karty a který je možné modifikovat z důvodu rozlišení jednotlivých zařízení připojených ke stejné sběrnici. Nevyužívá se v SPI módu.

1.4.2.5 DSR registr

Jedná se o 16bitový registr, jehož implementaci norma nevyžaduje. Nemá přiřazeno konkrétní použití, často se však využívá k ukládání rozšiřujících informací o možnostech paměťové karty, čímž pomáhá zvýšit výkonnost.

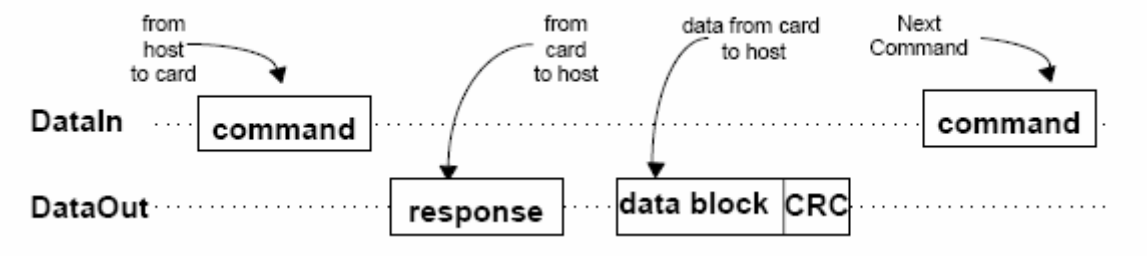
1.4.2.6 SCR registr

Tento 64bitový registr doplňuje registr CSD, jsou v něm uloženy informace o speciálních vlastnostech, které daná karta nabízí, např. podpora zabezpečení. Je nastavován výrobcem.

1.5 Operace čtení a zápisu dat

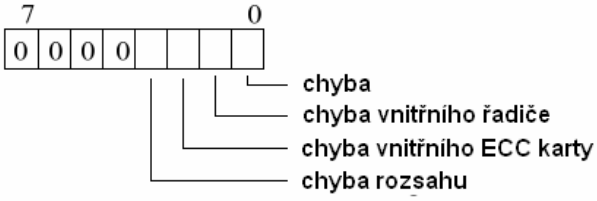
1.5.1 Čtení dat

SPI mód podporuje čtení dat po jednotlivých blocích i vícenásobné čtení bloků (příkazy CMD17 a CMD 18). Po přijetí validního příkazu čtení dat odpovídá karta odpovědí typu R1, za odpovědí následuje blok dat. Pokud je připojená karta standardní velikosti, odpovídá velikost čteného bloku velikosti nastavené pomocí příkazu SET_BLOCKLEN (CMD16). V případě použití karty SDHC, je velikost bloku pevně nastavena na 512 bajtů. Každý blok čtených dat je zabezpečen pomocí CRC16, který následuje bezprostředně za blokem data (Obrázek. 1.11). Pokud provádíme vícenásobné čtení bloků, je kontrolním součtem zabezpečen každý blok.



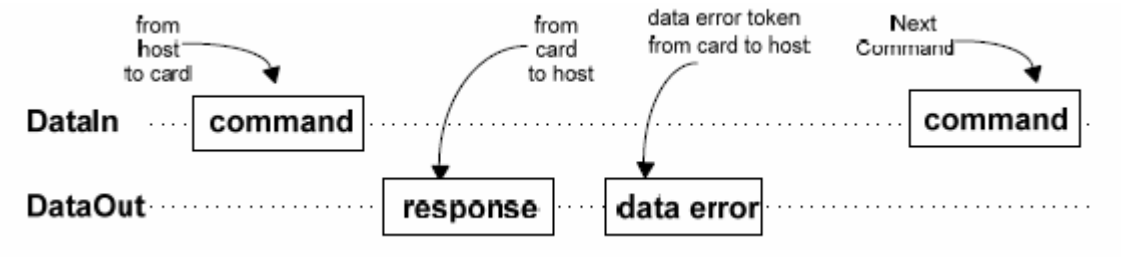
Obrázek. 1.11 Čtení dat (převzat z [4])

Je doporučeno, aby čtený blok byl zarovnán vůči fyzickým blokům uvnitř karty. V případě čtení z karty typu SDHC je toto povinnost.



Obrázek 1.12 Data error token

Jelikož je nutné čekat, než karta vystaví data, je příjem dat je zahájen speciálním řetězcem, který uvozuje datový blok (start block token). V případě chyby karta reaguje chybovým řetězcem (data error token, Obrázek 1.12) namísto očekávaného bloku dat (viz. Obrázek 1.13).



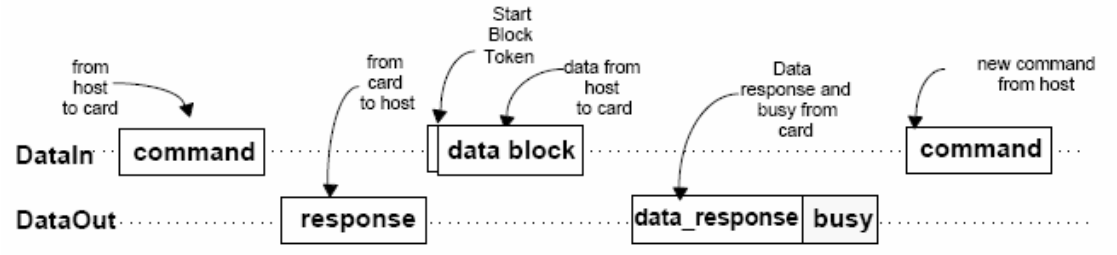
Obrázek 1.13 Chyba čtení dat (převzat z [4] a upraven)

1.5.1.1 Čtení registrů CID,CSD

Obsah registrů CID a CSD je v SPI módu čten pomocí postupu velmi podobnému čtení bloku dat. Při požadavku na obsah těchto registrů zašle karta nejprve standardní odpověď typu R1, která je následována blokem o velikosti 16 bajtů, zakončeným 16bitovým CRC kódem.

1.5.2 Zápis dat

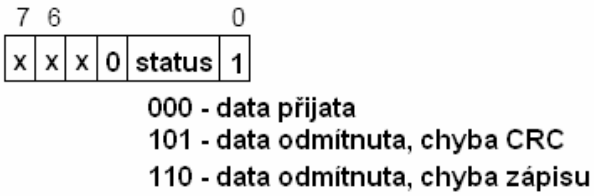
SPI mód podporuje zápis samostatných bloků a zápis více bloků současně. Po přijetí validního příkazu pro čtení (CMD24 nebo CMD25) odpovídá karta odpovědí formátu R1 a čeká na blok dat, který zasílá hostitel. Blok je opět uvozen speciálním řetězcem, zakončení bloku dat kontrolním součtem CRC16, délka bloku a omezení kladené na adresu jsou stejné jako u operace čtení dat (obrázek Obr. 1.14).



Obr. 1.14 Zápis dat (převzat z [4])

Po přijetí bloku dat karta zasílá hostiteli odpověď ve formátu R1. Pokud byl blok dat přijat bez chyb, budou tato data následně zapsána. Po dobu zápisu dat do paměti FLASH indikuje SD karta činnost tak, že udržuje hodnotu pinu DataOut na hodnotě log. 0.

Jakmile je zápis hotov, hostitel by měl zkontrolovat výsledek procesu zápisu pomocí příkazu SEND_STATUS(CMD13). Některé chyby, jako např. adresa mimo rozsah, porušení ochrany proti zápisu, je možné zjistit pouze jakmile proběhne proces zápisu. Bez této kontroly má hostitel pouze informace, zda nedošlo k chybě kontrolního součtu nebo chybě zápisu zaslané pomocí odpovědi Data Response Token (viz. Obrázek 1.15).



Obrázek 1.15 Data Response Token

Během procesu zápisu je možné deaktivovat kartu uvedením signálu CS do log. 1. SD karta odpojí pin DataOut od hostitele. Při opětovné aktivaci karty je na pinu DataOut stále hodnota log. 0 v případě, že proces zápisu ještě nebyl dokončen a všechny příkazy zaslané kartě jsou odmítnuty. Jelikož je možné zápis dat zrušit zasláním příkazu RESET (CMD0), což může vést k poškození dat, je povinností hostitele zabránit zaslání tohoto příkazu v této situaci.

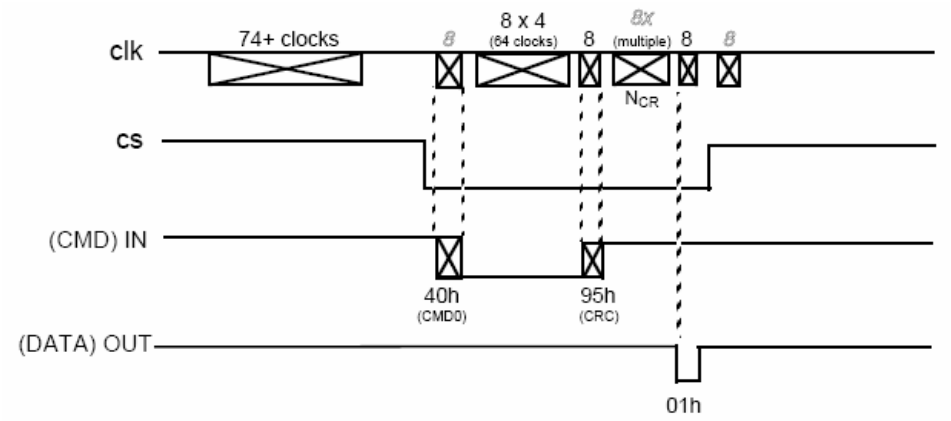
1.6 Časování

Časování SD karty v režimu SPI odpovídá časování karty MMC, ze které paměťová karta SD vychází. Vzhledem k částečné otevřenosti standardu (viz. kapitola 1.2.5) je nutné získat informace o časování SD karty z dokumentace MMC karet a doufat v zachování plné kompatibility směrem dolů, nebo z dokumentace starší SD standardů. Popisky průběhů obsahuje tabulka Tabulka 1.6.

8	počet hodinových taktů
8	počet prázdných hodinových taktů
8x	Osminásobky hodinových taktů
8x	Osminásobky prázdných hodinových taktů
8 x A	pevně daný počet A osminásobků hodinových taktů

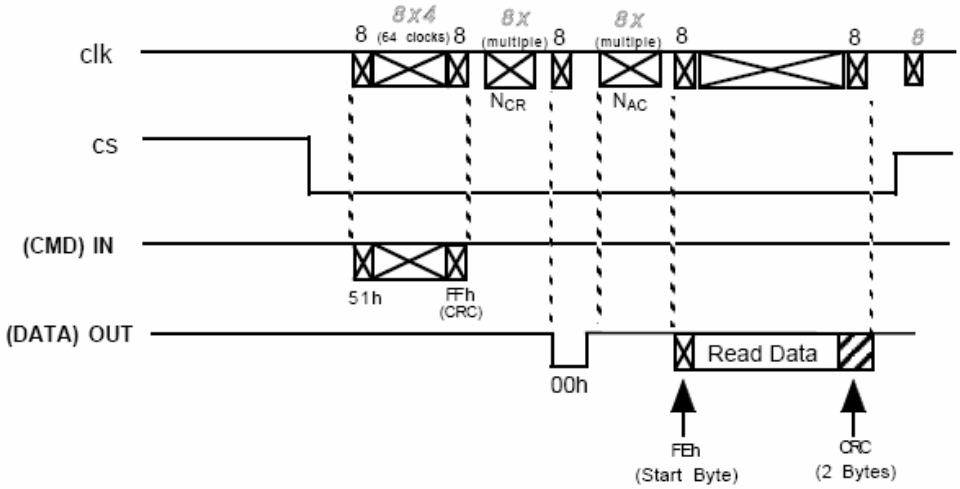
Tabulka 1.6: Vysvětlivky časových průběhů

Reset



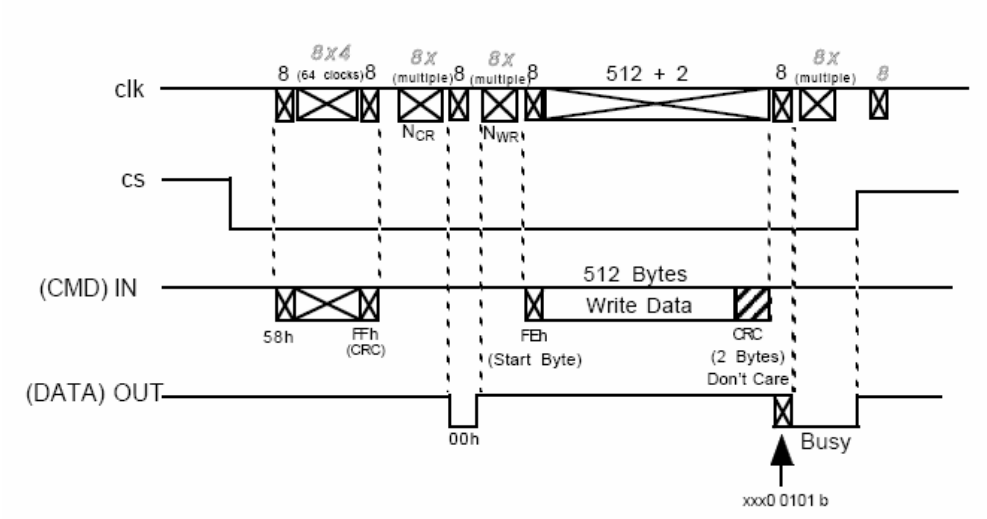
Obrázek 1.16 Reset (převzat z [7[7])

Čtení dat



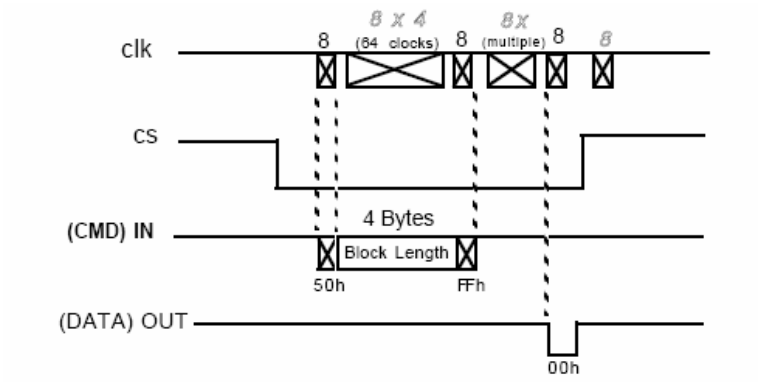
Obrázek 1.17 Čtení dat (převzat z [7[7])

Zápis dat



Obrázek 1.18 Zápis dat (převzat z [7[7])

Nastavení délky bloku



Obrázek 1.19 Nastavení délky bloku (převzat z [7[7])

Veškerá omezení, která jsou kladena na maximální nebo minimální dobu odezvy jsou uvedeny v tabulce Tabulka 1.7, jednotkou je 8 strojových cyklů (tedy 1 = 8 strojových taktů) a f je frekvence hodin

Na všech časových průbězích je možné si všimnout jedné zásadní věci, vkládání prázdných hodinových taktů. Tyto volné takty jsou nezbytné, taktují vnitřní řadič paměťové karty SD. I před ukončením práce s kartou a jejím odpojením je nezbytné vyčkat minimálně osm hodinových taktů na dokončení všech vnitřních operací radiče karty a až poté odpojit napájecí napětí.

	Min	Max
NCS	0	-
NCR	0	8
NRC	1	-
NAC	1	viz. **Chyba! Nenalezen zdroj odkazů.**
NWR	1	-
NEC	0	-
NDS	0	-
NBR	0	1

Tabulka 1.7: Časové konstanty

Vzorec 1.1:

$$\min [\{\{(TAAC * f) + (NSAC * 100)\} * 1/8\}, \{(100ms * f) * 1/8\}]$$

2 Navržené řešení

2.1 Platforma FITKit

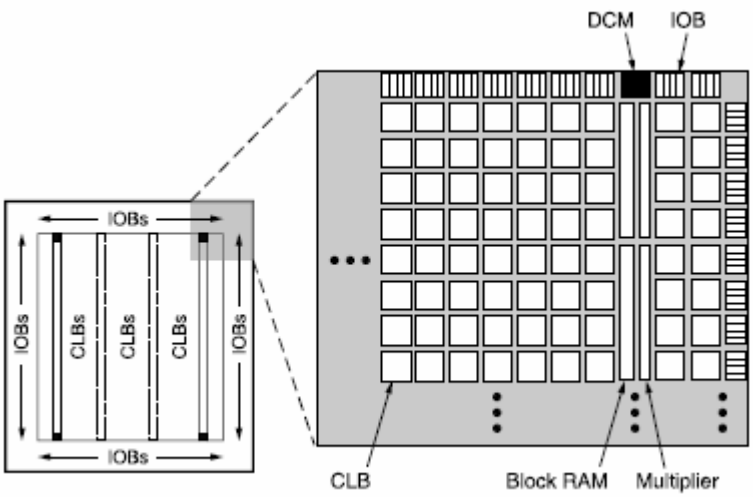
Vývojová platforma FITkit vznikla na Fakultě informačních technologií VUT za účelem zapojení vestavěných systémů do běžné výuky a zlepšení praktických schopnosti studentů při práci s hardware [8].

Obsahuje výkonný mikrokontroler firmy Texas Instruments a řadu periferií. Pro tuto práci je nejvýznamnějším prvkem programovatelné hradlové pole (FPGA) řady Spartan-3 od firmy Xilinx, které bylo využito k implementaci řadiče paměťové karty SD. Mezi další periferie patří např. paměť DRAM, řádkový displej a klávesnice. K dispozici je také rozhraní pro komunikaci s platformou, sériové rozhraní pro naprogramování mikrokontroleru z PC (řízeno USB-UART převodníkem). Dále zde nalezneme další rozhraní pro komunikaci s externími periferiemi např. rozhraní VGA, RS232 atd. Podrobnější informace a schémata lze nalézt v [11].

2.1.1 Struktura FPGA

Rodina FPGA Spartan-3 obsahuje pět programovatelných elementů, jejich umístnění na čipu ilustruje Obrázek 2.1 :

- CLB (Configurable Logic Block). CLB představuje hlavní zdroj pro implementaci synchronních i kombinačních obvodů. Uvnitř CLB jsou dva páry jednotek (Slice). Slices obsahují generátory funkcí (look-up tabulky LUT), prvky k uchování dat (klopné obvody), sadu multiplexorů, které umožňují kombinovat ostatní prvky za účelem sestavení složitější logiky, a další vedlejší logiku.
- IOB (Input/Output Block). IOB je jednotka pro kontrolu toku dat mezi vývody pouzdra a vnitřní logikou. Podporuje obousměrný tok dat i třístavovou logiku včetně dalších vstupně výstupních standardů.
- Bloková RAM (BRAM). Její velikost se liší v závislosti na třídě FPGA.
- Násobičky 18 krát 18 bitů.
- DCM (Digital Clock Manager). Jednotka, která umožňuje distribuci hodinového signálu vnitř FPGA.

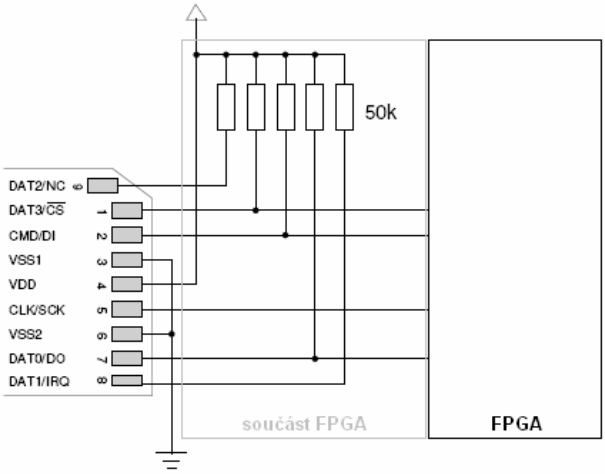


Obrázek 2.1 Struktur FPGA rodiny Spartan-3 (převzato z [10]a upraveno)

FPGA čip na FITkitu je zástupcem nejnižší třídy rodiny Spartan-3. Obsahuje 192 konfiguračních logických bloků, organizovaných do matice 16 krát 12, čtyři násobičky, dvě jednotky DCM a čtyři bloky BRAM o kapacitě 72 kilobitů, organizované do jediného sloupce.

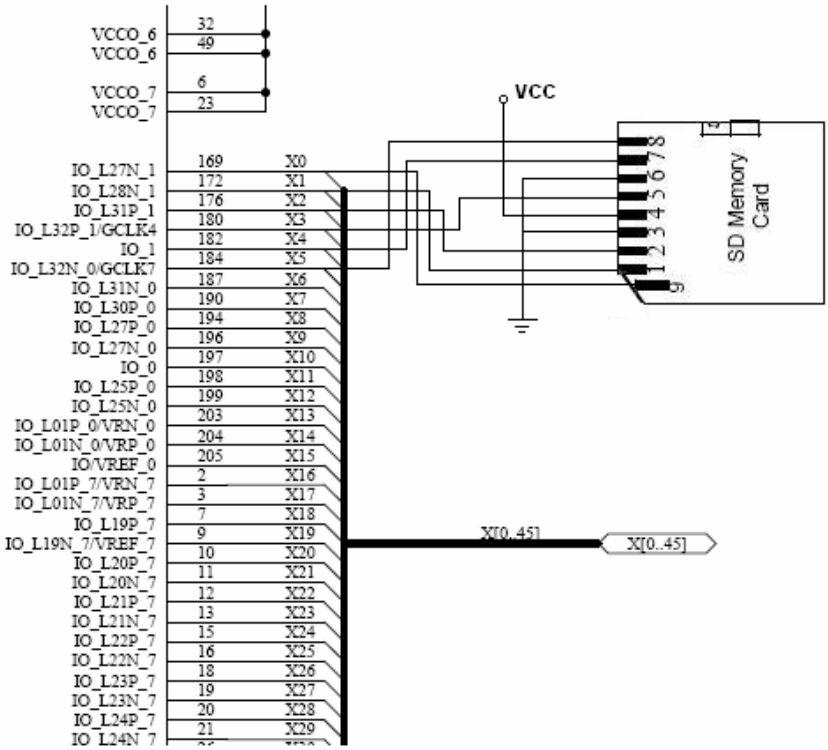
2.1.2 Připojení SD karty k platformě FITkit

Na Obrázek. 2.2 vidíme způsob připojení SD karty k FPGA. Přítomnost rezistorů (tzv. pullup rezistory) i jejich hodnoty jsou dány specifikací a musí být připojeny. Zde je vhodné využít pullup rezistorů obsažených uvnitř FPGA.



Obrázek. 2.2 SPI - Schéma připojení SD karty k FPGA

Na straně FPGA je pak karta připojena na sběrnici X. Detaily této sběrnice můžeme nalézt v[11]. Vlastní přiřazení pinů karty SD a sběrnice X je na Obrázek 2.3.



Obrázek 2.3 Připojení karty SD ke sběrnici X FPGA

2.1.2.1 Top-level entita

V hierarchií entit VHDL projektu je vždy nejvyšší tzv. top-level entita. Její signály odpovídají jednotlivým vývodům FPGA. V rámci vývoje firmware FITkitu byla vytvořena sada základních entit, které je možné v projektech použít. V tomto projektu je nutné využívat sběrnici X (propojení s konektory VGA, RS232, PS2 případně IDE, část sběrnice X vyvedena), je proto nutná entita tlv_pc_ifc ().

V tomto projektu bylo nezbytné editovat i vlastnosti jednotlivých pinů entity, připojení pullup rezistorů. Tyto jsou pro správnou funkčnost paměťové karty SD nezbytné i na pinech, které nejsou v režimu SPI využity.

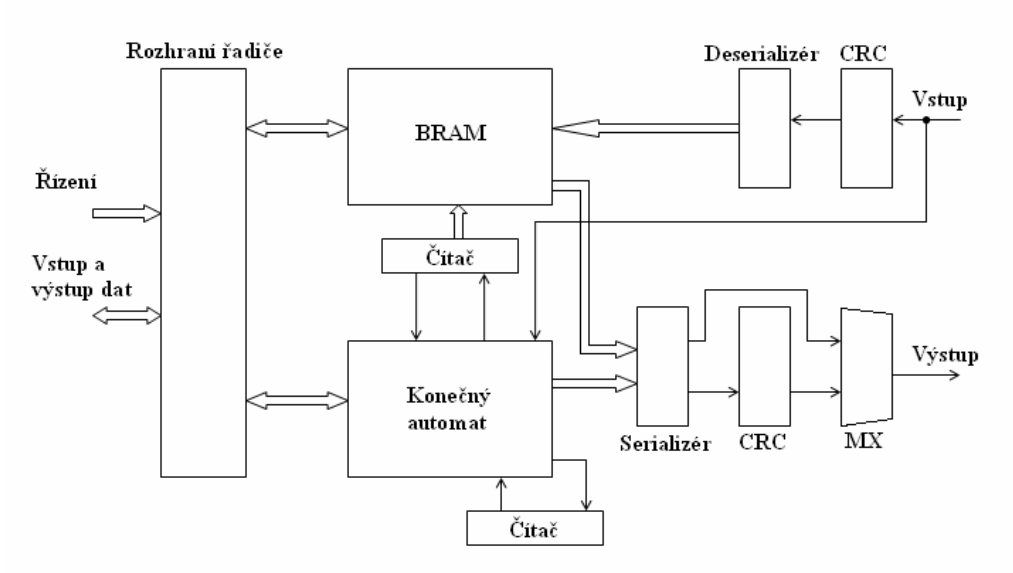
2.2 Řadič karet typu SD

Proces implementace řadiče paměťové karty SD můžeme rozdělit na dvě do jisté míry nezávislé části. Logicky první je návrh firmware, jehož syntézou získáme konfiguraci pro FPGA. V druhé části se zaměříme na návrh vhodného uživatelského rozhraní pro komunikaci s paměťovou kartou SD.

Zdrojové kódy v jazyce VHDL (VHSIC Hardware Description Language) se nacházejí v adresáři SDcard/top, uživatelské rozhraní je napsáno v jazyce C s využitím knihovny *libfitkit*, jeho zdrojové texty se nacházejí v adresáři SDcard/sw.

2.2.1 Implementace řadiče paměťové karty SD

Řadič paměťové karty SD umožňuje čtení, zápis a mazání jednotlivých bloků FLASH paměti. Vlastní komunikace s paměťovou kartou probíhá pomocí protokolu SPI, je tedy možné celý řadič rozdělit na dva logické celky a to obvody výhradně pro příjem a pro odesílání dat.

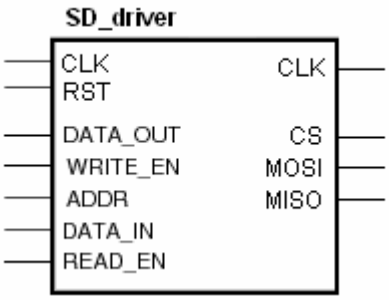


Obrázek 2.4 Blokové schéma řadiče paměťové karty

Další částí řadiče je řídící logika, která se stará o sestavení odesílaných dat a o dekódování dat přijímaných. Součástí řadiče je paměť, která slouží pro uložení přijímaných a odesílaných dat. Navržené blokové schéma uvažuje tyto nejdůležitější součásti řadiče spolu s rozhraním, pomocí kterého probíhá řízení řadiče i vlastní přenos dat mezi mikrokontrolerem a paměťovou kartou, Obrázek 2.4. Toto blokové schéma bylo výchozím konceptem při vlastní implementaci řadiče.

2.2.1.1 Rozhraní řadiče paměťové karty SD

Na obrázku Obrázek 2.5 vidíme rozhraní řadiče paměťové karty SD. Při návrhu tohoto rozhraní bylo nutné zohlednit již existující propojovací systém FITkitu, který slouží pro komunikaci s periferiemi realizovanými uvnitř FPGA.



Obrázek 2.5 Rozhraní řadiče paměťové karty SD

Tento propojovací systém je sada komponent dovolující komunikaci mikrokontroleru a libovolného zařízení připojeného k této sběrnici (detaily viz [12]). Velmi stručně si teď popišme význam jednotlivých signálů na straně paměťové karty:

- **CLK**

  Hodinový signál synchronizující paměťovou kartu a řadič. Signál hodin paměťové karty je nutné řídit a proto není možné jej připojit přímo na hodinový signál rozvedený uvnitř FPGA. Ve fázi inicializace karty nesmí frekvence připojených hodin karty překročit 400kHz. Poté je možné frekvenci zvýšit na maximum v závislosti na typu karty a typu komunikačního protokolu, viz Tabulka 1.1.
- **CS**

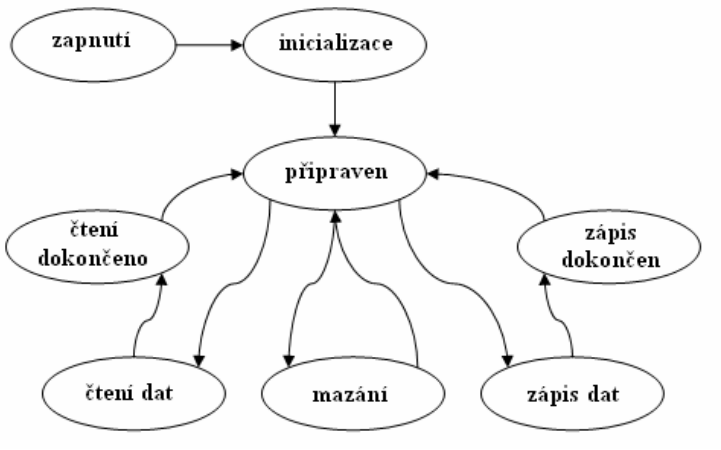
  Tento signál slouží k výběru paměťové karty se kterou bude probíhat následující komunikace, je-li připojeno více paměťových karet, a je udržován na hodnotě log. 0 po celou dobu komunikace.
- **MOSI**

  Data, příkazy – výstup do paměťové karty připojené k řadiči
- **MISO**

  Data, odpovědi – výstup z paměťové karty připojené k řadiči.

2.2.2 Popis řadiče paměťové karty SD

Výsledná podoba řadiče paměťové karty SD vychází z navrhovaného blokového schématu (Obrázek 2.4) a rozšiřuje tento návrh. V následujícím textu se zaměřím na jednotlivé součásti řadiče, jejich funkci, případně implementaci (detailní schéma viz. příloha).



Obrázek 2.6 Stavový diagram hlavního konečného automatu

2.2.2.1 Konečný automat

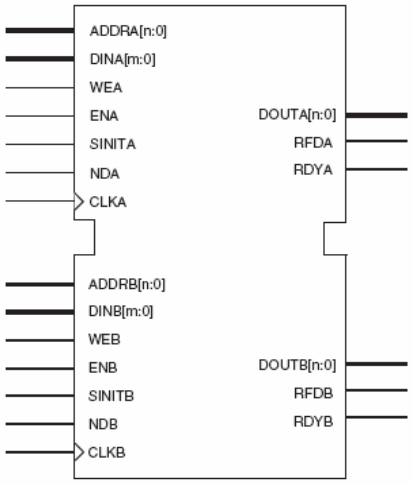
Konečný automat je řídící logikou celého řadiče, implementovaný jako dva Mealyho automaty. Zjednodušený stavový diagram hlavního konečného automatu vidíme na Obrázek 2.6. Výstup tohoto konečného automatu řídí druhý konečný automat, který zajišťuje vlastní vykonání procesu inicializace, čtení, zápisu nebo mazání. Jeho stavy a výstup jsou pak vstupními podmínkami pro uskutečnění přechodu mezi stavy hlavního konečného automatu.

2.2.2.2 Čítače

Řadič obsahuje dva čítače. První čítač slouží pro uchování informace o počtu odesílaných nebo přijímaných bitů, hodnota do které čítá je vždy závislá na aktuálním stavu konečného automatu. Je řízen signály load – načtení nové koncové hodnoty, cntmux – volba koncové hodnoty, done – příznak dokončení čítání. Druhý čítač slouží pro adresaci blokové paměti BRAM, jeho hodnota se zvyšuje vždy při přijetí nebo odeslání jednoho bajtu dat paměťové kartě. Výstupem je tedy adresa, význam ostatních signálů je shodný s prvním čítačem.

2.2.2.3 BRAM

FPGA obvod použitý ve FITkitu obsahuje čtyři blokové paměti BRAM (block RAM) o celkové kapacitě 4 x 16384 bitů. Každá pamět může být instancována uvnitř VHDL kódu jako dvouportová nebo jednoportová. V prvním případě se využívá primitiva RAMB16_Sm_Sn, v druhém případě primitiva RAMB16_Sn, kde konstanty m a n ovlivňují datovou šířku prvního a druhého portu a mohou nabývat následujících hodnot: 1, 2, 4, 9, 18 nebo 36 bitů. V případě datové šířky 9, 18 a 36 bitů má uživatel k dispozici 8, 16 a 32 datových bitů a 1,2 případně 4 paritních bity. Paritní bity mohou být využity také k úschově dat, neboť výpočet parity není prováděn automaticky [10].



Obrázek 2.7 Primitiva RAMB16_Sm_Sn

V řadiči je využito primitiva RAMB16_S9_S36, přičemž paritní bity DINPA a DINPB, ani synchronní reset SINIT nejsou využívány. Řídící signály RE obou portů jsou trvale v log 1, na datových výstupech jsou tedy vždy data odpovídající přivedené adrese.

Datové vstupy a výstupy portu A jsou připojeny na rozhraní řadiče, tedy na stranu komunikující s mikrokontrolerem. Datová sběrnice portu B slouží pro čtení a ukládání dat přijatých nebo zaslaných dat paměťové kartě. Datový vstup portu B je připojen na výstup komponenty realizující převod sériových dat na paralelní. Datový výstup portu B je připojen přes multiplexor na vstup komponenty realizující serializaci dat a jejich odesílání.

2.2.2.4 CRC7, CRC16

Výpočet kontrolního součtu CRC probíhá podle algoritmu Algoritmus 1.1 a Algoritmus 1.2. Nastavením řídícího signál CE na hodnotu log. 1 začne výpočet kontrolního součtu na příchozích sériových datech. Po ukončení výpočtu se nastaví CE na hodnou log. 0 a v dalších 7 nebo 16 taktech dojde k odeslání kontrolního součtu CRC. V případě kontroly přijímaných dat, pouze u CRC16, se po bezchybném přijetí dat a kontrolního součtu nastaví signál CRC_ZERO na 1.

2.2.2.5 SRL_PRL

Tato komponenta provádí převod přijímaných sériových dat na paralelní data. Po nastavení signálu CE na log. 1 se vždy po 8 přijatých bitech nastaví signál DATA_RCVD. Paralelní výstup dat je připojen na vstupní datový port paměti BRAM.

2.2.2.6 PRL_SRL

Tato komponenta slouží k serializaci vstupních paralelních dat. Výběrovým signálem CE, jeho nastavením na hodnotu log.1 a opět log. 0, se provede zápis vstupu do vnitřního registru. Ten je pak s každým taktem hodinového signálu vysouván na výstup komponenty, nejprve nejvíce významné bity. V okamžiku, kdy se odešle i poslední bit uloženého bajtu, nastaví se signál DATA_SND na hodnotu log 1.

2.2.2.7 Rozhraní řadiče

Tato komponenta slouží k vlastnímu připojení řadiče k propojovacímu systému FITkitu. Dále obsahuje signály pro připojení portu A blokové paměti BRAM. Signály STATE a CMD slouží k řízení celého řadiče, signál CMD obsahuje kód příkazu, který se má vykonat, signál STATE obsahuje příznakové bity informující o průběhu těchto příkazů. Signále SD_ADR se přenáší adresa bloku paměťové karty ze které se má číst, nebo na ni zapisovat.

2.2.3 Výsledky implementace komponenty

K vývoji celé implementace byl použit program fy. Xilinx ISE. Tento nástroj slouží ke kompletní analýze zdrojového kódu ve VHDL. Postupně provádí jeho překlad, sestavení hierarchie jednotlivých entit s přihlédnutím ke generickým parametrům návrhu a analýzu konečných automatů, ale také optimalizace a základní analýzu časování. Výsledkem je tzv. netlist, který obsahuje logický popis dizajnu na úrovni meziregistrových přenosů spolu s omezeními.

V druhé fázi je provedena implementace netlistu, která zahrnuje překlad na reprezentaci primitivními logickými hradly. Dále program provede mapování hradlové reprezentace na elementy specifické technologie, v našem případě na prvky uvnitř slices (LUT). V předposlední fázi (place and route) se rozhodne, kde se mapované prvky umístí na čipu, a jak budou propojeny. Nakonec je vytvořen konfigurační soubor, který je nutné nahrát pomocí vhodného terminálového programu do FPGA.

V našem případě bylo zjištěno, že celý obvod pokrývá 28 procent kapacity konfiguračních jednotek (slices) a může být synchronizován signálem o frekvenci přibližně 90 MHz. Podrobnější výsledky syntézy a implementace dizajnu poskytují části logů v příloze.

3 Demonstrační aplikace

Aby bylo možné s paměťovou kartou SD pracovat je třeba navrhnout vhodné uživatelské rozhraní a sadu příkazů pro vykonání několika funkcí. Veškerou komunikaci s paměťovou kartou řídí mikrokontroler. Jako prostředník mezi uživatelem a kartou interpretuje a vykonává příkazy uživatele přijaté přes sériové rozhraní z terminálu a zpět na terminál zasílá načtená data a stavové informace.

3.1 Práce s paměťovou kartou

Základní komunikace s paměťovou kartou probíhá pomocí funkce dostupné v knihovně *libfitkit* pro komunikaci přes SPI s konkrétním řadičem periferního zařízení, funkcí FPGA_SPI_RW_AN_DN.

První parametr této funkce určuje, zda se jedná o operaci čtení nebo zápis (symbolické konstanty SPI_FPGA_ENABLE_READ a SPI_FPGA_ENABLE_WRITE). Následuje parametr bázové adresy SPI dekodéru, který musí být stejný jako adresa zadaná při generickém mapování komponenty SPI_adc ve zdrojovém kódu jazyka VHDL (např. pro adresu o šířce 8 bitů může mít hodnotu 10h). Třetím parametrem je ukazatel na pole o velikosti 32bitů (alokováno jako 4 x 8). Čtvrtým parametrem je šířka adresy v násobcích bajtů a pátým pak šířka přenášených dat, opět jako násobky bajtů.

Výhody velmi jednoduché základní komunikace s řadičem periferního zařízení bylo dosaženo namapováním řídících a stavových registrů do adresního prostoru řadiče periferního zařízení (viz Tabulka 3.1).

Symbolická konstanta	Adresa	Registr/Paměť
není	0h | 1FFh	BRAM
SD_ADR	200h	registr SD_ADR
CMD	201h	registr CMD
STATUS	203h	registr STATUS

Tabulka 3.1: Mapování registrů do adresového prostoru

To umožňuje uživateli zapsat,či číst libovolný bajt paměti BRAM, ze které se pak tento 512 bajtů velký blok přenáší do paměti FLASH nebo z ní čte. Totéž platí o možnosti provést jakoukoliv z operací podporovaných kartou a reagovat na zaslané odpovědi. V tabulce Tabulka 3.2 najdeme přehled základních příkazů. obsahuje význam jednotlivých bitových kombinací registru STATUS. Registr SD_ADR je určený pro uložení adresy bloku paměťové karty.

Je zodpovědností uživatele pokud využívá těchto základních příkazů aby dodržel předepsaný postup práce:

- Operace zápisu

  Nejprve je nutné naplnit 512 bajtů paměti BRAM požadovanými daty. Poté zapsat do registru CMD odpovídající příkaz. Nakonec čekat na potvrzení o zápisu. Do registru CMD vložit příkaz *noop*.
- Operace čtení

  Při operaci čtení je třeba zapsat do registru CMD odpovídající příkaz a čekat na potvrzení platnosti přečtených dat. Poté do registru CMD vložit příkaz *noop*.
- Operace mazání

  Operace mazání začíná zápisem příkazu *erase* do registru CMD. Po potvrzení úspěšnosti operace je nutné registr CMD opět nastavit na příkaz *noop*.

Ve všech předchozích případech platí podmínka, že adresa daného paměťového bloku, nad kterým chceme příkazy provádět, musí být do registru SD_ADR zapsána jako první.

příkaz	**kód příkazu**
noop	00000000h
write	00000001h
read	00000002h
erase	00000003h

Tabulka 3.2: Základní příkazy řadiče paměťové karty SD

3.1.1 Funkce uživatelského rozhraní

Pokud bychom implementovali pouze nízkoúrovňovou komunikaci byla by práce s paměťovou kartou SD zbytečně komplikovaná. Pro snadnou práci zejména s větším objemem dat byly vytvořeny příkazy pro vložení většího počtu bloků, jejich mazání apod.

Funkce, které používá MCU pro implementaci uživatelských příkazů jsou definovány v modulu config.c. Prototypy těchto funkcí jsou deklarovány v hlavičkovém souboru config.h. Pro podrobnější informace o implementaci odkazuji na programovou dokumentaci v adresáři SDDriver/sw/doc/. Syntaxe jednotlivých příkazů s přesným popisem se nachází v souboru uživatelské příručky README.txt v adresáři SDDriver/sw/doc.

Závěr

Využití rekonfigurovatelného hardware má do budoucna jistě velký potenciál a s ním i platformy s čipy založenými na technologii FPGA. Rozšiřování a zkvalitňování aplikací určených pro tento typ obvodu povede jistě k jejich dalšímu masivnímu rozšíření.

Využití paměťové karty SD se ukázalo jako problematické. Tato karta nemá zcela otevřený standard, a proto je nutné shromažďovat informace i z jiných zdrojů, především ze starších specifikací karet MMC, se kterými je paměťová karta SD zpětně kompatibilní. Problematické se ukázalo samotné připojení karty k vývojové platformě FITkit, neboť na tuzemském trhu neexistuje prodejce s paticemi pro karty tohoto druhu.

Vlastní využití paměťové karty SD na platformě FITkit považuji za úspěšné. Ke kartě a k datům na ní obsaženým je možné přistupovat několika způsoby a data bez problémů číst i zapisovat.

Elektronická paměť bez mechanických částí s nízkou spotřebou se zdá být ideální pro využití především v mobilních vestavných systémech. Vývoj v oblasti pevných disků navíc ukazuje, že paměti typu FLASH mohou sloužit i jako náhrada klasických pevných disků.

Literatura

[1] SEKANINA, L.: *Návrh počítačových systémů (INP 14) – Paměti*. [online], Brno, FIT VUT v Brně, [cit. 2008-04-15].
URL < https://www.fit.vutbr.cz/study/courses/INP/private/prednaska/inp14pameti.pdf>

[2] Wikipedia: *Flash memory*. [online], poslední aktualizace 13.05.2008, [cit. 2008-05-14].
URL < http://en.wikipedia.org/wiki/Flash_memory>

[3] Wikipedia: *Secure Digital Card*. [online], poslední aktualizace 13.05.2008, [cit. 2008-05-14].
URL < http://en.wikipedia.org/wiki/Secure_Digital>

[4] SD Group: *SD Specifications - Part 1: Physical Layer Simplified Specification*. [online], Verze 2.00 (2006), ©2001-2006 SD Group (Panasonic, SanDisk, Toshiba) and SD Card Association, poslední aktualizace 25.09.2006, [cit. 2008-05-14].
URL < http://www.sdcard.org/about/memory_card/pls/Simplified_Physical_Layer_Spec.pdf >

[5] TOSHIBA: *TOSHIBA SD Card Specification*. [online], poslední aktualizace 10.02.2003, [cit. 2008-05-14]. URL < http://www.digitalspirit.org/file/index.php/obj-download/docs/sd/TOSHIBA_SD_Card_Specification.pdf >

[6] SANDISK*: SanDisk Secure Digital (SD) Card Product Manual*. [online], Verze 1.9 (2003), ©2003 SanDisk Corporation, poslední aktualizace prosinec 2006, [cit. 2008-05-14].
URL < http://www.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf >

[7] ULRICH, R.: *MMC SD Timming*. [online], [cit. 2008-05-14].
URL < http://www.ulrichradig.de/site/atmel/avr_mmcsd/pdf/MMCSDTimming.pdf >

[8] Stránky projektu FITkit. [online], Brno, FIT VUT v Brně, ©2006.
URL: <http://merlin.fit.vutbr.cz/FITkit>.

[9] FITkit: *Popis kitu*. [online]. Stránky projektu FITkit, Brno, ©2006 [cit. 2008-05-13].
URL: <http://merlin.fit.vutbr.cz/FITkit/hardware.html>.

[10] Xilinx, Inc.: *Xilinx DS099 Spartan-3 Complete data sheet*. [online], ©2003-2006, poslední aktualizace 26.04.2006, [cit. 2008-05-13].
URL: <http://www.xilinx.com/bvdocs/publications/ds099.pdf>.

[11] FUCIK, O.; BARDAS, R.: *FITKIT – BLOCK DIAGRAM*. [online], Verze 1.1, © CAMEA s.r.o. & R.B.E. 2005, [cit. 2008-05-14].
URL < http://merlin.fit.vutbr.cz/FITkit/docs/pdfs/hw_block.pdf >

[12] VAŠÍČEK, Z.: *Firmware / Propojovací systém FITkitu*. [online], Brno, FIT VUT v Brně, ©2006-2008, [cit. 2008-05-14].
URL < http://merlin.fit.vutbr.cz/FITkit/docs/firmware/spifitkit.html >

[13] VAŠÍČEK, Z.: *Aplikace / Využití paměti BRAM v MCU*. [online], Brno, FIT VUT v Brně, ©2006-2008, [cit. 2008-05-14].

URL < http://merlin.fit.vutbr.cz/FITkit/docs/aplikace/app_bramdemo.html >

Seznam příloh

Příloha 1. Detailní schéma řadiče

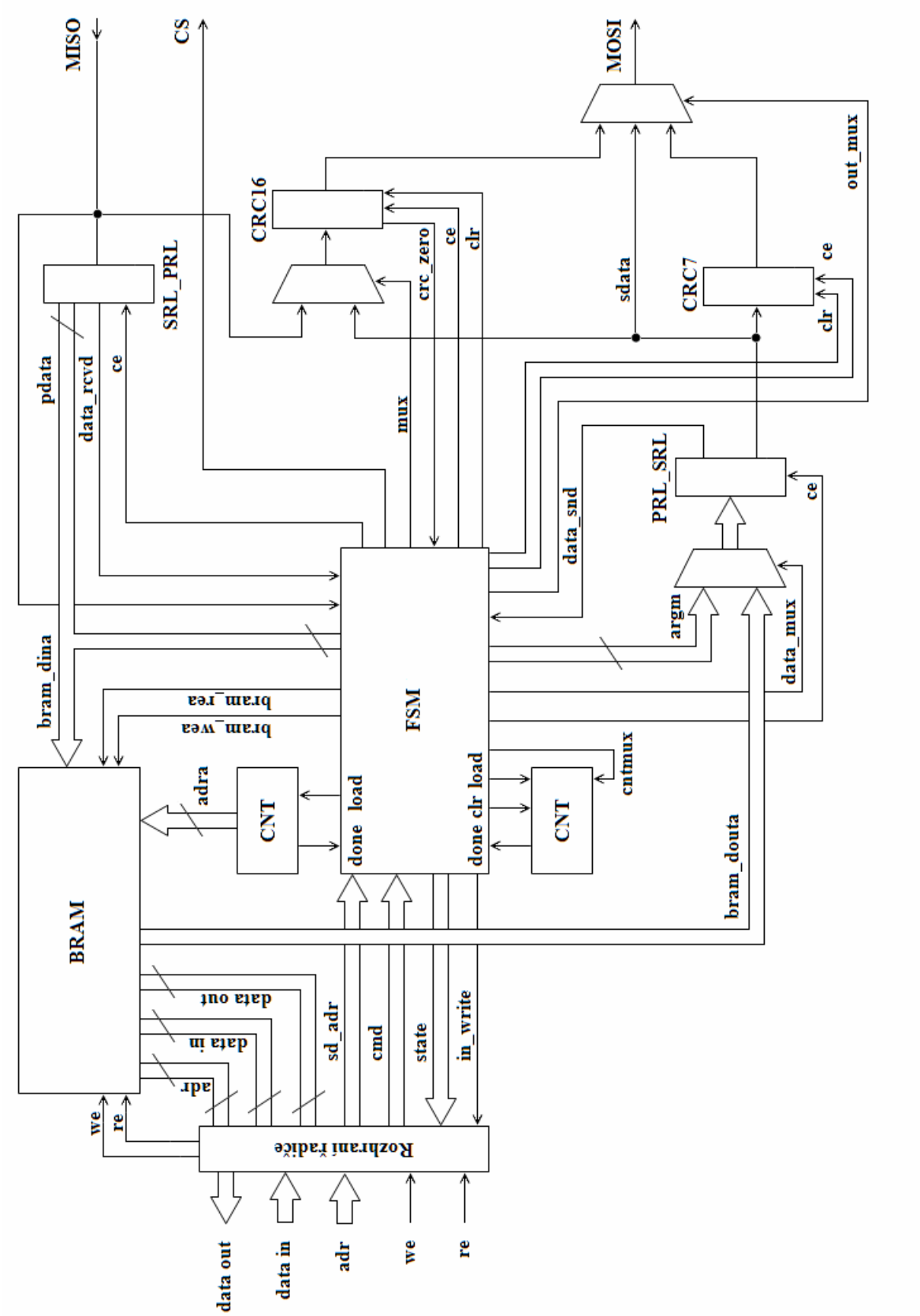
Příloha 2. Úseky logů z překladu VHDL projektu

Příloha 3. Programová dokumentace

Příloha 4. Uživatelská příručka v textovém formátu

Přílohy

Příloha 1. Detailní schéma řadiče



Příloha 2. Úseky logů z překladu VHDL projektu

```
    Synthesizing Unit <fpga>.
Related source file is "C:/Documents and Settings/martin/Plocha/Skola/asd/chips/arch_pc_ifc.vhd".
    Unit <fpga> synthesized.
=========================================================================

HDL Synthesis Report

    Macro Statistics
    # Adders/Subtractors                    : 4
     10-bit adder                           : 1
     3-bit adder                            : 1
     3-bit subtractor                       : 1
     5-bit adder                            : 1
    # Counters                              : 4
     27-bit down counter                    : 1
     3-bit down counter                     : 1
     5-bit up counter                       : 1
     6-bit up counter                       : 1
    # Registers                             : 55
     1-bit register                         : 40
     10-bit register                        : 1
     16-bit register                        : 1
     2-bit register                         : 1
     3-bit register                         : 2
     32-bit register                        : 7
     8-bit register                         : 2
     9-bit register                         : 1
    # Comparators                           : 7
     3-bit comparator greater               : 2
     5-bit comparator greater               : 2
     5-bit comparator less                  : 3
    # Multiplexers                          : 4
     32-bit 4-to-1 multiplexer              : 4
    # Tristates                             : 58
     1-bit tristate buffer                  : 58
    # Xors                                  : 5
```

```
    1-bit xor2                                  : 5
=========================================================================

Timing Summary:
---------------
Speed Grade: -4

    Minimum period: 11.233ns (Maximum Frequency: 89.023MHz)
    Minimum input arrival time before clock: 11.857ns
    Maximum output required time after clock: 16.631ns
    Maximum combinational path delay: 9.078ns
=========================================================================

Device Utilization Summary:

    Number of BUFGMUXs                    2 out of 8      25%
    Number of External IOBs              70 out of 124    56%
      Number of LOCed IOBs               70 out of 70    100%

    Number of RAMB16s                     1 out of 4      25%
    Number of Slices                    448 out of 768    58%
      Number of SLICEMs                   0 out of 384     0%

=========================================================================
```

Příloha 3. Programová dokumentace

Programová dokumentace se nachází na přiloženém CD v adresáři /sw/doc.

Příloha 4. Uživatelská příručka v textovém formátu

Uživatelská příručka (README.txt) obsahující informace o zprovoznění projektu a ovládání uživatelského rozhraní se nachází na přiloženém CD v adresáři /sw/doc.