2008/13 – 9.4.2008

elektrorevue
ISSN 1213 - 1539

Komprese JPEG 2000 a akcelerace pomocí DSP

Ing. Marek Kváš
Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,
Ústav automatizace a měřící techniky, Kolejní 4, 612 00 Brno, Česká republika
e-mail: marek.kvas@phd.feec.vutbr.cz

Jako následník velmi známé a velmi rozšířené obrazové komprese JPEG byl vydán v roce 2000 nový kompresní standard JPEG 2000. Tato nová komprese přináší výrazné zlepšení kvality obrazu, které je ovšem zaplaceno vyššími nároky na paměť a výpočetní výkon. Tento článek nejprve ukazuje rozdíly mezi JPEG a JPEG 2000. V další části uvádí základní principy, na kterých je komprese JPEG 2000 postavena. A ve své poslední kapitole popisuje pokus o hardwarovou akceleraci knihovny pro kompresi JPEG 2000 pomocí PCI karty, založené na signálovém procesoru TMS320C6415.

1 Úvod

Oblast systémů pro zpracování obrazu se v posledních letech vyvíjí obrovskou rychlostí. Rychle stoupá rozlišení čipů pro snímání obrazu, barevná hloubka zpracovávaných obrazů a také snímkovací frekvence, se kterou je možné obrazová data získávat. Zvyšování všech těchto parametrů sebou nese prudký nárůst objemu dat, která snímače vyprodukují a která je nutné následně zpracovat. Stále častěji se objevují požadavky na zpracování obrazu v reálném čase. Pro takovéto výpočetně náročné aplikace se využívá skupiny procesorů označovaných jako digitální signálové procesory (dále již jen DSP).

Od procesorů pro všeobecné použití se DSP výrazně liší svojí architekturou, která je optimalizovaná pro aplikace zpracovávající velmi rychle velké toky dat. Důraz je kladen na paměťový subsystém s rozšířenými možnostmi přímého přístupu do paměti (Direct Memory Access -dále jen DMA), širokými sběrnicemi a různými módy adresování. DSP často obsahuje více výpočetních jednotek, které umožňují paralelní zpracování dat, a instrukční sada bývá rozšířena o speciální instrukce používané při zpracování signálů. Tyto vlastnosti dodávají DSP obrovský výpočetní výkon. Oproti procesorům pro všeobecné použití užívaným v počítačích PC zůstávají DSP však nesrovnatelně jednodušší (co do počtu tranzistorů na čipu). Jednoduchost má za následek nižší spotřebu energie i při vysokých taktovacích kmitočtech a velkém výpočetním výkonu.

Velmi efektivní je kombinace počítače typu PC a k němu připojených karet, založených na signálových procesorech. Pro PC máme k dispozici velmi dobře dostupné komfortní nástroje a obrovské množství již připravených, snadno nebo i volně dostupných, aplikací. DSP pak využíváme pouze pro zpracování výpočetně náročných algoritmů, čímž se aplikace pro DSP může výrazně zjednodušit. DSP pak vykonává jen velmi dobře definované opakující se úlohy, pro které je jejich architektura optimalizována.

Právě takový systém, složený z PC a DSP, je popsán v následujícím článku. Je řešen problém akcelerace komprese JPEG 2000. Projekt je založen na knihovně OpenJPEG, která je jako opensource, včetně dokumentace, volně k dispozici na stránkách jejich autorů [1].

elektrorevue
ISSN 1213 - 1539

2 Komprese JPEG 2000

Ať již vědomě nebo nevědomě, každý z nás, uživatelů internetu, mnohokrát denně využívá služeb obrazové komprese, již léta známé pod označením JPEG. Podle tohoto standardu, požívaného již od roku 1992, jsou komprimovány prakticky všechny fotografie, které jsou běžně dostupné na internetu. Do formátu JPEG ukládají fotografie v podstatě všechny digitální fotoaparáty na trhu. Modely, které nejsou určeny pro profesionální nebo poloprofesionální použití, ani jinou možnost nemají. Málokterý formát se dočká tak masového nasazení.

Po osmi letech od svého vzniku se JPEG dočkal svého přímého následovníka - komprese JPEG 2000. Vzhledem k tomu, jak dlouhá doba osm let ve světě digitálních technologií je, to jistě nikoho nepřekvapí. Opravdu překvapující však je, že od chvíle uvedení JPEG 2000 uplynulo dalších sedm let a i přes své znatelně lepší vlastnosti je stále málo známý a používaný spíše výjimečně nebo pro specifičtější aplikace. Důvodem může být i to, že za lepší kvalitu a další výhody platíme daň v podobě vyšších nároků na paměť a výpočetní náročnost.

Podívejme se tedy nejprve na krátké srovnání nového standardu JPEG 2000 se starším formátem JPEG a poté na základní principy, na kterých je JPEG 2000 založen.

2.1 Vlastnosti JPEG 2000 a srovnání s JPEG

Vznik standardu JPEG 2000 byl výsledkem snahy najít kompresní metodu vhodnou pro přirozené obrazy (fotografie), která by nejen efektivněji redukovala požadavky na prostor pro uložení obrazu, ale umožňovala také pohodlnou práci s tímto obrazem – jeho editaci, extrakci částí obrazu, získání náhledů v různých rozlišeních a podobně. Kromě efektivní ztrátové komprese měl nový formát nabízet rovněž bezeztrátovou kompresi. Nejdůležitější vlastnosti JPEG 2000 se dají shrnout do tohoto seznamu [2], [3], [5]:

Obr. 1 Výchozí obrázek pro kompresi

- Bezeztrátová i ztrátová komprese – kompresní poměr ztrátové komprese je o 20 % až 30 % lepší než u JPEG; bezeztrátová varianta dosahuje kompresního poměru 1:2, tedy data po kompresi mají poloviční velikost než před kompresí
- Vysoká kvalita pro velké kompresní poměry – schopnost dobře pracovat s obrázky s poměrem nižším než 0.25 bit/pixel (poměr počtu bitů, které reprezentují obraz po kompresi, ku počtu pixel v obraze)
- Široké možnosti pro progresivní přenos s ohledem na různá kritéria – z dat, která postupně přicházejí po přenosové lince, je možné rekonstruovat obraz v nízké kvalitě a s přibývajícími daty kvalitu zvyšovat. Jiným příkladem může být požadavek, aby byla přenášena nejprve malá část obrazu a až po dosažení její plné kvality zbytek obrazu.
- Přístup k částem obrazu bez nutnosti dekomprese celého obrazu
- Podpora operací s obrazem v komprimovaném stavu bez nutnosti dekomprese – otočení (90°, 180°, 270°), zrcadlení

elektrorevue
ISSN 1213 - 1539

- Podpora pro „Oblasti zájmu“ (ROI – Region of Interest). Část obrazu může mít prioritu nebo být komprimována odlišným způsobem – může být například uložena ve vyšší kvalitě, nebo přenášena přednostně.
- Podpora pro různé barevné modely s různým počtem složek
- Vysoká odolnost proti chybám v bitstreamu
- Využití prakticky libovolných metadat – neobrazových informací (např. komentář co je na obrázku k vidění) uložených spolu s obrazem v jednom souboru
- Možnost zpracovat velké obrázky – JPEG má omezení 64000 x 64000 pixel
- Možnost zpracování složených dokumentů (zejména text + grafika) a možnost pracovat s počítačovou grafikou s ostrými přechody (pro JPEG nevhodné)

Komprese JPEG 2000 (tedy jeho výkonné jádro) byl přijat mezinárodní standardizační organizací jako ISO/IEC 15444-1 v prosinci 2000 [2], [4]. Dále následovaly normy pro různé rozšíření komprese a formátu, např.:

Part 1 – již zmíněné kompresní jádro

Part 2 – rozšíření o nové možnosti dekompozice vlnkovou transformací, kódování ROI, specifikace velké množiny metadat, nový rozšířený formát .JPX

Part 3 – Motion JPEG 2000 – možnost komprimovat video

Poslední částí, která vstoupila v platnost v roce 2003, je Part 12, která má stejný text jako část 12 normy pro MPEG4 a snaží se vytvořit jednotný formát pro ukládání sekvencí multimediálních dat.

Vidíme, že množina vlastností je obrovská a standard je poměrně obsáhlý a složitý. Zajímavější ovšem je podívat se na skutečné výsledky komprese pomocí JPEG 2000 a srovnat je s výsledky komprese JPEG. Toto srovnání nám také ukáže rozdíl ve způsobu práce obou kompresí.

Na Obr. 2 máme sadu známých fotografií Lena, které jsou komprimovány v různých kompresních poměrech jak pomocí JPEG, tak pomocí JPEG 2000 (výchozí vidíme na Obr. 1). Pro kompresi do formátu JPEG byl použit běžný prohlížeč obrázků IrFan View. Pro kompresi do formátu JPEG 2000 byl použit volně dostupný kodek OpenJPEG [1], jehož část byla použita pro implementaci popsanou v druhé polovině tohoto článku. V levém sloupci vidíte snímky komprimované metodou JPEG, v pravém sloupci metodou JPEG 2000.

Při prvním pohledu na horní dvojici obrázků, které jsou komprimovány s kompresním poměrem 20, nejsou patrné žádné rozdíly. Při bližším pohledu na obrázek komprimovaný JPEG je možné na ostřejších hranách (hrana klobouku) vidět nepatrné stopy artefaktů vzniklých kompresí. Na obrázku komprimovaném JPEG 2000 je rozdíl oproti originálu ve vyhlazení některých ploch.

Druhá sada obrázků nám prozrazuje o způsobu práce obou kompresí trochu více. Tyto obrázky jsou komprimovány v poměru 1:70. Na levém obrázku, zpracovaném klasickým algoritmem JPEG, již vidíme, pro JPEG typické, obdélníkové oblasti, které mají jednotnou barvu a ostré, dobře viditelné hrany. Tyto ostré přechody působí velmi rušivě. JPEG dělí obraz na oblasti 8x8 pixel, které následně podrobuje diskrétní kosinové transformaci (DCT) a kvantizaci. Při vyšších kompresních poměrech dochází k tomu, že vyšší koeficienty DCT vymizí, celá tato oblast má jednotnou barvu a neexistuje vazba na sousední oblasti. U JPEG 2000 toto pozorovat nemůžeme. JPEG 2000 provádí diskrétní vlnkovou transformaci

elektrorevue
ISSN 1213 - 1539

(DWT) nad celým obrazem, takže se zvyšující se kvantizací se vytrácí detaily, obraz se jakoby rozmazává, ale prostorové návaznosti zůstávají zachovány a nevznikají žádné ostré přechody. Obraz tak vypadá méně narušený a přirozenější. Přesněji řečeno, obraz je před DWT rozdělen do stejně velkých (oblasti u okrajů obrazu se mohou velikostí lišit) obdélníkových oblastí (tzv. tile – dlaždice). Velikost těchto oblastí může uživatel nastavit až na velikost celého obrazu. Obraz je pak zpracován DWT najednou. Rozdělení na části snižuje paměťové nároky, ale snižuje také kvalitu obrazu (mohou vznikat viditelné hranice mezi oblastmi).

Třetí sada obrázků je komprimována s poněkud přehnaným kompresním poměrem 1:170. V obou obrazech vidíme silné zkreslení. Na obraze komprimovaném JPEG se v plné míře projevil problém s vytvářením souvislých barevných oblastí, vytratily se veškeré detaily a obraz působí velmi nepřirozeně. Z obrázku komprimovaného JPEG 2000 také zmizelo mnoho detailů, obraz se zdá rozmazaný, ale základní prvky obrazu jsou zřetelnější a obrázek je mnohem lépe barevně podán – obraz po JPEG 2000 má 46770 originálních barev, po JPEG 1995 (originální obrázek 48 458).

JPEG JPEG2000

Kompresní poměr 20

Kompresní poměr 70

elektrorevue
ISSN 1213 - 1539

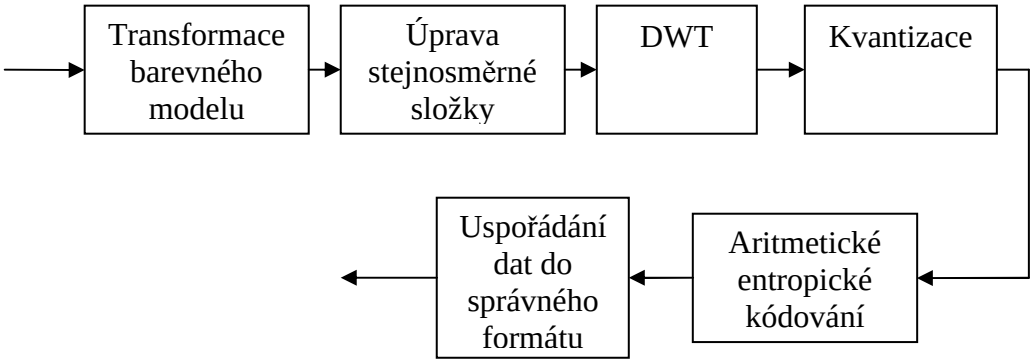
Kompresní poměr 170

Obr. 2 Srovnání kvality obrazu pro kompresi JPEG a JPEG 2000

2.2 Základní principy komprese JPEG 2000

Řetězec operací, které jsou prováděny během komprese JPEG 2000 [2], [4],[5], je naznačen na Obr. 3. Při dekompresi jsou použity inverzní transformace v přesně opačném pořadí.

Prvním krokem je transformace barevného modelu (v Tab. 1 označená MCT z anglického Multicomponent Color Transform). V tomto článku předpokládáme, že vstupní obraz je uložen v modelu RGB (barva každého pixelu je složena z červené, zelené a modré složky), což je nejčastější případ. JPEG 2000 může samozřejmě pracovat i s jinými modely, které mají např. více než tři složky – pak se tento krok neprovádí. Základní nevýhodou modelu RGB při kompresi je, že mezi složkami je velká míra korelace. Jinými slovy, velká část informace se opakuje ve všech barevných složkách. Pro kompresi se model transformuje na YUV (Y-jasová složka, U a V – barevné složky), který má míru korelace nižší. Transformační vztahy jsou velmi jednoduché. Složky YUV se vypočtou jako lineární kombinace složek RGB. Pokud se provádí bezeztrátová komprese, použije se celočíselná aproximace těchto vztahů, kde není třeba zaokrouhlovat a nevzniká tak chyba.



Obr. 3 Kompresní řetězec JPEG 2000

Úprava stejnosměrné složky je velmi jednoduchá operace, která v podstatě znamená přechod od neznaménkového vyjádření složek (obvykle 0 – 255) na znaménkové vyjádření (tedy v tomto případě -128 až 127)

elektrorevue
ISSN 1213 - 1539

Následuje diskrétní vlnková transformace (DWT), která nahradila diskrétní kosinovou transformaci v původním JPEGu. Pro ztrátovou transformaci se používá Cohen-Daubechies-Feauveau 9/7, pro bezeztrátovou kompresi se používá celočíselná varianta 5/3. Účelem této transformace je reprezentovat původní obraz tak, aby následující části komprese byly schopny co nejefektivněji odstranit redundantní informace. Transformace spočívá v použití jednoduchých filtrů, které vstupní posloupnost rozloží na dvě pásma - pásmo nízkých kmitočtů a pásmo vysokých kmitočtů. V JPEG 2000 se provádí dvojrozměrná DWT (2D DWT). Pro výpočet využijeme separability 2D DWT. Tato vlastnost umožňuje vypočítat 2D DWT ve dvou krocích pomocí jednorozměrné DWT. Nejprve se aplikuje jednorozměrná DWT na všechny řádky a pak na všechny sloupce výsledku předešlého kroku. Vzniknou tak čtyři pásma - LL (nízké kmitočty horizontálně i vertikálně), LH (nízké kmitočty horizontálně, vysoké vertikálně), HL (vysoké kmitočty horizontálně, nízké vertikálně), HH (vysoké kmitočty horizontálně, vysoké vertikálně). Na pásmo LL se může použít 2D DWT opakovaně a dostat tak další úroveň dekompozice. Úroveň dekompozice také ovlivňuje kvalitu komprese a je nastavitelná uživatelem.

V případě, že provádíme ztrátovou kompresi, provede se tzv. kvantizace. V podstatě jde o snížení přesnosti koeficientů DWT. Provádí se vydělením koeficientů kvantizačním krokem a zaokrouhlením. V tomto kroku dochází k nevratné redukci informace v obraze.

Následuje nejsložitější část komprese – aritmetické entropické kódování. Metoda, která je použita v JPEG 2000, se nazývá EBCOT (Embedded Block Coding with Optimal Truncation). Do tohoto bloku vstupují bloky koeficientů DWT (v našem případě je přednastavena hodnota 64x64 koeficientů pro jeden blok) a výstupem je komprimovaný bitstream. EBCOT je poměrně složitý adaptivní proces, který je pro kompresi velmi účinný, ovšem také výpočetně náročný. Komprese, která v něm probíhá, je bezeztrátová. Bitstream má však tu vlastnost, že je možné jej zkrátit, což má podobný efekt jako větší kvantizace provedená v předešlém kroku. Této vlastnosti se dá využít k dosažení požadované velikosti komprimovaných dat.

Poslední blok je zodpovědný za to, aby data byla uspořádána ve správném pořadí a byly k nim přidány správné hlavičky. Formát souboru je navržen pro velkou variabilitu a je proto poměrně složitý.

3 Implementace

Jak již bylo uvedeno v úvodu článku, projekt je postavený na knihovně OpenJPEG. Tato knihovna je určena pro kompresi bitmapových obrázků do formátu JPEG 2000. Pro akceleraci DSP byla vybrána jen část knihovny. Akcelerace se využívá jen v případě bezeztrátové komprese. Rozšíření na ztrátovou kompresi by znamenalo pouze implementaci jiného druhu DWT a kvantizace.

V DSP se provádí jen skutečné zpracování dat. Výpočty, které jsou potřeba k rozdělení dat na patřičné bloky, správa datových struktur a složení výsledných dat podle předepsaného formátu do výstupního souboru bylo ponecháno zcela v režii PC. Tyto úlohy zabírají zanedbatelné množství času komprese. Provádějí se jen jednou nebo v malém počtu opakování a proto není efektivní zabývat se jejich přenosem do DSP.

Nyní můžeme implementaci rozdělit na dvě části. První částí je rozhraní mezi DSP a PC. Do této části spadá jednak rozdělení projektu na část pro zpracování v PC a na část pro zpracování v DSP, určení, které řídící struktury je třeba sdílet, a úprava struktury knihovny

elektrorevue
ISSN 1213 - 1539

s ohledem na co nejmenší nutný rozsah komunikace. Dalšími problémy spadající do této části jsou PCI komunikace mezi procesy běžícími v PC v prostředí MS Windows XP, synchronizace s DSP a rychlý přenos dat z PC do DSP.

Druhou částí je implementace vlastních výkonných algoritmů komprese pro DSP, nebo přesněji úprava existující implementace pro použití v DSP. Součástí je i snaha o optimalizaci a využití prostředků, které nám DSP nabízí oproti PC

Jako akcelerátoru byla využita karta s označením FD64 [7], připojená k PC prostřednictvím sběrnice PCI a osazená DSP TMS320C6415 [8]. Tento procesor disponuje 1MB paměti SRAM na čipu, jejíž část může být použita jako cache typu L2. Na čipu jsou rovněž paměti typu cache L1, odděleně pro program a data, každá o velikosti 16kB. Jádrem procesoru jsou dvě 32 bitové datové cesty, každá složená ze čtyř výkonných jednotek a sady 32 registrů o šířce 32 bit. Hodinový kmitočet procesoru je 600 MHz. V ideálním případě může procesor dokončit v každém hodinovém cyklu jednu instrukci, tedy provést až 8 aritmetických operací a může tak dosáhnout teoretického výkonu až 4800 MIPS (milionů instrukcí za sekundu). Procesor je založen na tzv. VLIW (Very long Instruction Word) architektuře. To znamená, že v jednom instrukčním slově jsou zakódovány instrukce pro všechny výkonné jednotky a rozložení výpočtů na jednotlivé jednotky a datové cesty musí být provedeno v době překladu. Tato technologie umožňuje vysokou míru paralelismu, ale klade vysoké nároky na překladač a programátora. Pokud je program napsán tak, že není využito více výpočetních jednotek, ať již chybou programátora, nebo proto, že algoritmus nelze takto rozdělit, architektura se stává neefektivní a výpočetní výkon prudce klesá. Využití více výpočetních jednotek je náročný úkol a jen zřídka se povede plně vytížit všechny jednotky. Další silnou zbraní tohoto procesoru je tzv. EDMA (Enhaced Direct Memory Access – rozšířené DMA) subsystém, který nabízí mnoho možností, jak data přenášet a jak přenosy synchronizovat.

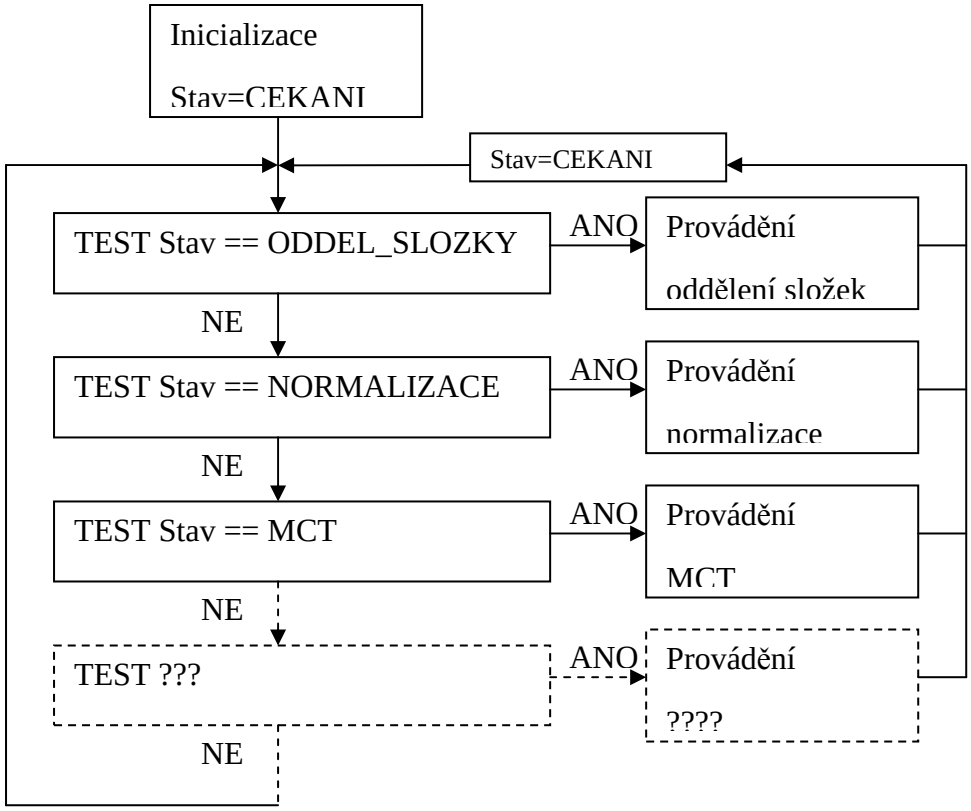
3.1 Rozhraní PC – DSP a synchronizace procesů

3.1.1 Přenos dat mezi PC a DSP

Jedním ze základních problémů systémů složených z více procesorů je přenos dat mezi částmi systému a synchronizace procesů v různých procesorech. Náš akcelerátor je připojen k PC prostřednictvím sběrnice PCI. Uživatelská aplikace je určena pro operační systém Windows XP. Tvorba ovladačů pro PCI zařízení pro systém Windows XP je poměrně složitý proces, vyžadující hluboké znalosti systému. Existuje tedy i výrazně jednodušší řešení – využít univerzální ovladač, který všechny operace na nízké úrovni skrývá před uživatelem za poměrně jednoduché rozhraní API (Application Programming Interface – sada knihovních funkcí, které zpřístupňují služby ovladače). V tomto projektu bylo použito univerzálního PCI ovladače WinDriver od firmy Jungo.

Všechny přenosy dat v systému jsou řízeny PC. DSP zde z hlediska přenosů dat pracuje v tzv. Slave módu. PC využívá přímého přístupu do paměti DSP, zapisuje do něj potřebná vstupní data, řídící struktury a čte výsledky. Každé čtení nebo zápis z/do paměti DSP vyvolá v DSP DMA přenos. Software v DSP nemusí nijak spolupracovat a tento proces je pro něj zcela skryt. Přenosová rychlost ve směru z PC do DSP se v našem systému pohybovala kolem 90 MB/s. V opačném směru, tedy přenos výsledků z DSP do PC, probíhá výrazně pomaleji – pod 10MB/s. Zvolený režim práce není z hlediska přenosu z DSP optimální. Aby bylo možné přenášet data DSP srovnatelnou rychlostí jako do něj, musely by tyto přenosy být řízeny DSP – DSP by muselo provádět přímý přístup do paměti PC.

elektrorevue
ISSN 1213 - 1539



Obr. 4 Stavový automat v DSP

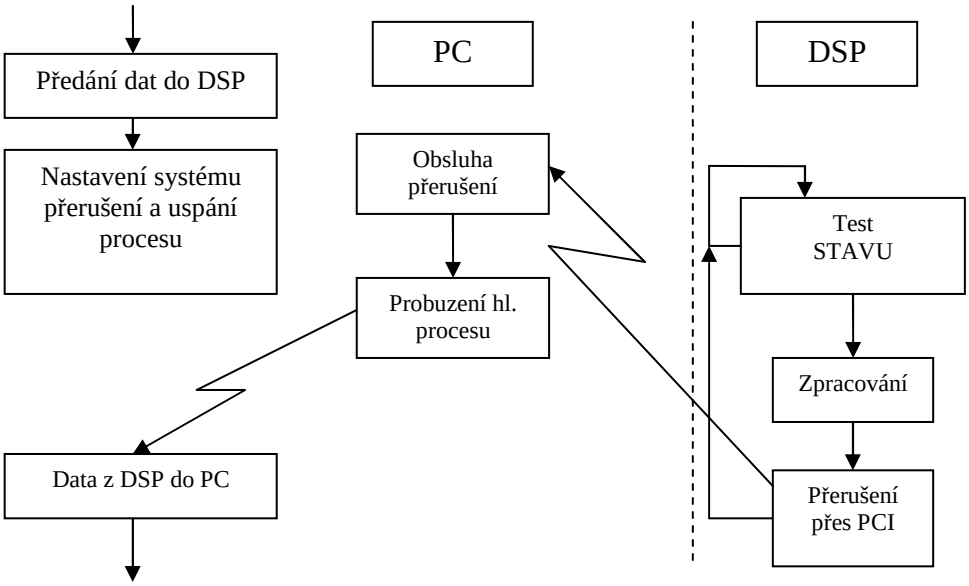
Aby DSP mohlo tímto způsobem přenášet data do PC, je již nutná spolupráce SW v DSP a situace se poněkud komplikuje. V PC je nutné: alokovat paměť, zajistit, aby byla opravdu alokována v operační paměti (aby nebyla přesunuta do odkládacího souboru), zajistit, aby byla ve fyzické paměti fixována, a zjistit fyzické adresy všech bloků této paměti (paměť nemusí být alokována jako souvislý blok ve fyzické paměti, i když se tak z pohledu programátora jeví). Seznam těchto bloků je nutné předat do DSP, které musí přenos dat rozdělit na několik samostatných DMA přenosů, podle bloků paměti.

V našem systému představují přenosy výsledků prakticky zanedbatelnou část času nutného ke zpracování, proto tento efektivnější způsob přenosu dat nebyl implementován.

3.1.2 Synchronizace PC a DSP

Vzhledem k velkému množství parametrů komprese, které mohou být teoreticky nastavovány pro různé části obrazu nebo různé úrovně dekompozice různě, je nutné spravovat poměrně složitou hierarchii datových struktur, která všechny tyto informace uchovává. Z hlediska programu jsou manipulace s těmito strukturami, jejich vyplňování a předávání do dalších úrovní poměrně složitou a co do délky kódu velmi významnou částí knihovny. Z hlediska délky výpočtu jsou však operace s nimi zcela zanedbatelné – jednoduché výpočty s malým počtem opakování. Přenos těchto úkolů do DSP by nepřinesl žádnou výhodu, pouze by zkomplikoval SW pro DSP a ztížil by správu paměti v DSP. Veškerá správa těchto struktur a řízení běhu komprese proto byly ponechány v PC a do DSP byly přeneseny pouze skutečně výkonné algoritmy, které jsou parametrovány pomocí zjednodušených datových struktur používaných v PC.

elektrorevue
ISSN 1213 - 1539



Obr. 5 Synchronizace DSP a PC pomocí přerušení

Program pro PC tedy vypadá tak, že volání funkcí pro zpracování dat je nahrazeno přenosem dat do DSP (pokud tam ještě nejsou k dispozici z předchozího kroku), vytvořením jednoduché řídící struktury a jejím přenosem do DSP. Struktura obsahuje mimo jiné i stavovou proměnnou, která DSP říká, který krok komprese má být vykonán a následně informuje PC, zda je proces v běhu, skončil v pořádku či s chybou. Program DSP pak může mít zcela jednoduchou strukturu naznačenou na Obr. 4.

Pokud bychom program pro PC implementovali nejjednodušším způsobem – předá data DSP, spustí danou část komprese, testuje stavovou proměnnou a čeká na dokončení – chovali bychom se velmi neefektivně. Procesor PC by zcela zbytečně trávil čas komunikací PCI a testováním stavu. Vznikaly by navíc zbytečné přenosy jak na PCI, tak v EDMA systému DSP. Proto bylo využito systému přerušení. Chování aplikace PC je naznačeno na Obr. 5. PC přenese data do DSP, spustí daný krok komprese, nastaví systém přerušení a uspí se. V okamžiku, kdy DSP dokončí daný krok komprese, vyvolá přes PCI přerušení, proces v PC se probudí a pokračuje v práci. V čase zpracování dat, kdy je proces v PC uspán, není kompresí procesor PC zatížen a může zpracovávat jiné úlohy.

3.2 Implementace výkonných algoritmů a jejich optimalizace

V prvním kroku byly algoritmy v co největší míře převzaty z původní implementace knihovny pro PC. Byly provedeny jen nezbytné úpravy. Taková implementace však není schopna využít možností, které architektura DSP nabízí. Při překladu jsou samozřejmě prováděny automatické optimalizace, ani ty však nemohou efektivně pracovat, pokud na ně není při psaní programu myšleno. Nejlepší optimalizace by se pravděpodobně dosáhlo ručním přepsáním algoritmů v assembleru a ručním naplánováním rozložení operací do výkonných jednotek. Toto je ovšem velmi pracné a zdlouhavé. Provedené optimalizace spočívaly zejména ve sloučení operací, které jsou prováděny nad stejnými daty, úpravě pořadí některých výpočtů, aby se zvýšila efektivita přístupů do paměti a využití systému EDMA k přesunům a třídění dat na pozadí zpracování další části dat.

elektrorevue
ISSN 1213 - 1539

Úspěšnost optimalizace samozřejmě silně závisí na chování algoritmu. Vstupními daty pro kompresi je v našem případě obraz vyjádřený v modelu RGB načtený z bitmapového obrázku (.bmp). Každá barevná složka je kódována osmi bity, takže jeden pixel zabírá v paměti prostor o velikosti tří byte. Obraz v paměti je dvourozměrné pole, kde se v každém řádku střídají barevné složky vyjádřené v osmibitovém neznaménkovém formátu. Tato reprezentace není vhodná pro další zpracování. Vstupem do DWT musí být tři dvourozměrná pole (každá barevná složka zvlášť), kde každý prvek je vyjádřen ve 32-bitovým znaménkovém formátu. Transformace barevného modelu vyžaduje pro každý plně barevný pixel dvě sčítání, dvě odčítání a dva bitové posuny. Posun stejnosměrné složky představuje jedno odčítání pro každou barvu. Z paměti je třeba načíst tři byty a zapsat do ní výsledek 12 byte – veškerá data jsou uložena ve vnitřní paměti SRAM. Výše zmíněnými optimalizacemi, provedenými na úrovni programu v jazyce C, se povedlo dosáhnout 70-ti násobného zrychlení s konečným výsledkem 11 taktů procesoru pro zpracování jednoho pixelu barevného obrazu.

Velikost obrázku	**256 x 200**	
	DSP [µs]	**PC [µs]**
Přenos dat	1 603	
Posun stejnosměrné úrovně + MCT	973	3349
DWT	8 147	20 955
EBCOT	627 829	522 578
Celkově	**638 552**	**546 882**
PC je rychlejší o	**15%**	

Tab. 1 Srovnání rychlosti komprese s podporou DSP a bez ní

Následujícím krokem je 2D DWT, která zpracovává každou barvu zvlášť. Výpočet jednorozměrné varianty je složen z aplikace dvou filtrů [4]. Pro výpočet lichých koeficientů je třeba provést dvě sčítání a jeden bitový posun a pro sudé tři sčítání a jeden bitový posun. Následně je nutné koeficienty setřídit (liché k sobě a sudé k sobě). Vstupem výpočtu jsou hodnoty tří sousedních bodů. Jak bylo řečeno, při 2D DWT je nutné provést dvakrát jednorozměrnou transformaci – jednou na řádky a potom na sloupce výsledku.

DWT se povedlo urychlit více než 20-ti násobně na 16 taktů/pixel (tentokrát ale pixel jedné barvy). Opět byly uspořádány výpočty tak, aby je optimalizační algoritmy byly schopny lépe naplánovat do výpočetních jednotek a algoritmus byl upraven tak, aby mohl zpracovávat vždy dva sloupce najednou. Pro úsporu paměti jsou výstupní data ukládána na místo vstupních dat. Toto je důvod, proč je nutné koeficienty po výpočtu setřídit. Ukládání výsledků do jiného pole v paměti by mohlo ušetřit další čas.

Posledním krokem, který byl přenesen do DSP, je proces EBCOT, tedy aritmetické entropické kódování. Tento algoritmus je výrazně složitější než předešlé kroky. Po optimalizaci představuje zpracování tohoto algoritmu přibližně 90 % času komprese. Optimalizace tohoto algoritmu nebyla příliš úspěšná. Komplikuje ji jednak složitost implementace a hlavně samotné chování algoritmu a jeho přístup k datům. Zatímco předešlé kroky přistupovaly k datům pravidelně, předvídatelně a v cyklech s velkým počtem opakování, běh EBCOT je závislý na datech samotných a možnosti paralelizace, které DSP nabízí, není možné využít.

V kodecích provedených jako ASIC (Aplication Specific Integrated Circuite – zákaznické integrované obvody) a v implementacích pro hradlová pole nabízených na trhu je obvykle několik jednotek provádějících EBCOT. Tím se dosáhne paralelizace na úrovni zpracování celých bloků dat a výrazného zrychlení.

Asi nejzajímavějším údajem zcela jistě je, jak se systém choval po optimalizaci jako celek ve srovnání s původní čistě softwarovou implementací. Použité PC bylo založeno na dvoujádrovém procesoru Intel Pentium 4 běžícím na kmitočtu 2.8 GHz.

elektrorevue
ISSN 1213 - 1539

V Tab. 1 vidíme srovnání doby komprese v případě, že bylo použito DSP, s případem, kdy celá komprese proběhla v PC. První fáze komprese jsou téměř čtyřnásobně rychleji prováděny v DSP. DWT je rychlejší dvojnásobně. Pouze EBCOT v DSP výrazně zaostává. Celkově trvala komprese s využitím DSP o 15 % déle. Musíme si ale uvědomit, že srovnáváme DSP běžící na 600 MHz s příkonem v řádu jednotek wattů s nesrovnatelně složitějším procesorem taktovaným na 2.8 GHz s příkonem pohybujícím se okolo 100 W. V době, kdy zpracovává data DSP je procesor prakticky nevytížen a může zpracovávat jiné úkoly.

4 Závěr

V článku byl nejprve stručně představen kompresní standard JPEG 2000, byly popsány základní rozdíly oproti jeho předchůdci standardu JPEG. V další kapitole byly ukázány hlavní rozdíly v degradaci obrazu při kompresi těmito metodami tak, jak je může pozorovat každý uživatel. Následující odstavce seznamují s funkcí jednotlivých částí kompresního řetězce. Zbytek článku pojednává o projektu, který si kladl za cíl upravit knihovnu pro kompresi JPEG 2000 tak, aby výpočetně náročné části komprese vykonával signálový procesor umístěný na kartě připojené přes sběrnici PCI. Jsou popsány problémy, které bylo nutné řešit a výsledky, kterých bylo dosaženo.

Měření, které byly provedeny na výsledném systému, ukázaly, že komprese bez využití DSP zůstala o 15 % rychlejší, než implementace s použitím DSP, i přes provedené optimalizace. Toto však není možné považovat čistě za neúspěch. Tato implementace je pouze prvním krokem k prakticky použitelné aplikaci srovnatelné s komerčním řešením. Ukázala, kde jsou problémy a na kterou část se zaměřit.

Na srovnání s PC se můžeme podívat také z jiného pohledu. Mnohem jednoduší DSP s výrazně nižší spotřebou energie umožňuje výrazně složitějšímu procesoru s vyšší spotřebou věnovat se v době komprese jiným úlohám.

Poměr mezi výpočetním výkonem a příkonem se stává stále častěji sledovanou veličinou. Obrovský rozvoj embedded aplikací s nároky na vysoký výpočetní výkon dává tomuto poměru velký význam – obzvláště u bateriově napájených systémů. Nízký příkon procesoru souvisí s menšími nároky na prostor, do kterého může být zařízení vestavěno, spolehlivostí systému (není potřeba aktivní chlazení, která snižuje spolehlivost) a přináší tak prospěch v mnoha ohledech.

Největší prostor pro další práci a optimalizaci je bezesporu v implementaci algoritmu EBCOT. Tento algoritmus představuje asi 90 % času celé komprese a jeho sebemenší zlepšení ovlivní výkon celé aplikace. Nejvhodnějším řešením se zdá být implementace toho algoritmu do FPGA, které je na použité PCI kartě rovněž k dispozici a která nabízí možnost paralelizace na úrovni celých funkčních blokků.

5 Poděkování

Tento článek vznikl za podpory Ministerstva školství, mládeže a tělovýchovy České republiky (Výzkumný záměr MSM0021630529)

elektrorevue
ISSN 1213 - 1539

6 Literatura

[1] Vershueren, J.: Open JPEG, 2003. Dostupný na <www.openjpeg.org> (6. 3. 2008)

[2] Oficiální stránky organizace JPEG :<www.jpeg.org>

[3] Christopoulos, Ebrahimi, Skordas: JPEG2000: The New Still Picture Compression Standard, ACM Multimedia Workshop Marina Del Rey CA USA, 2000

[4] JPEG: JPEG 2000 Image Coding System: Part 1 Final Committee Draft Version 1.0, 2000, Available on <www.jpeg.org> (1.2.2007)

[5] Impoco, G.: JPEG 2000 – A Short Tutorial, 2004. Available on <www.dmi.unict.it/~impoco/files/tutorial_JPEG2000.pdf> (1.2.2007)

[6] Kváš Marek: 2-D Signal Processing Based on TMS320Cxx structures [Diploma thesis], BUT Brno, Department of Control and Instrumentation

[7] DFC Design: Stránky výrobce <dspfpga.com>

[8] Texas Instruments: firemní literatura k rodině DSP TMS320C64x dostupná na <ti.com>