

Technická zpráva



Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

Vyhodnocování růstu kolonií kvasinek *Saccharomyces cerevisia* s použitím akcelérátoru Uni1P/DX64

Jan Kloub, Jan Schier
kloub@utia.cas.cz, +420-2-6605 2502

*Prostředky pro rychlý vývoj
HW-akcelerovaných vestavěných aplikací zpracování obrazu a videa
1ET400750408*

*Department of Signal Processing
<http://zs.utia.cz>*

Obsah

1 Úvod	1
2 Program na straně PC	2
2.1 Spuštění programu	3
3 Stručný popis karty Uni1P	4
4 Modul DX64	4
5 Implementace hranového detektoru	4
5.1 Návrh hranového detektoru pomocí nástroje Synplify DSP	6
5.2 Propojení hranového detektoru a paměťových portů	6
5.3 Využití částečné dynamické rekonfigurace	7
5.4 Program pro DSP	7
6 Závěr	8
7 Poděkování	8
8 Výpis CDROM	11

Revize

Revize	Datum	Autor	Popis změn v dokumentu
0	10.9.2008	Kloub	Vytvoření dokumentu
1	10.12.2008	Schier	Rozšířený úvod, přepracovaná struktura dokumentu

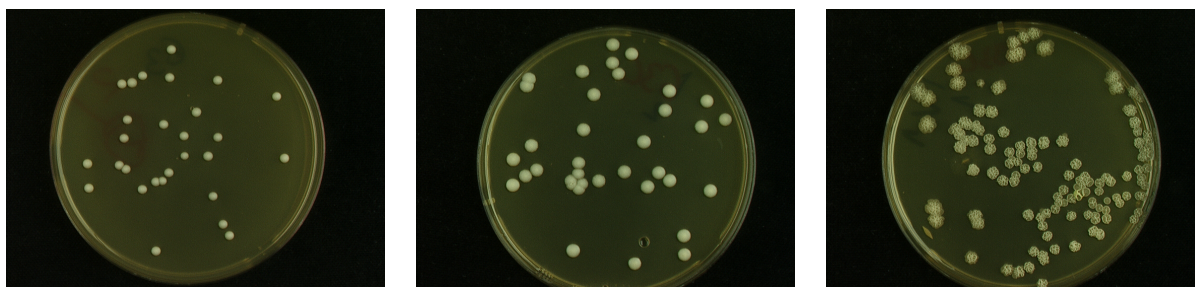
1 Úvod

Tento dokument popisuje implementaci aplikace pro předzpracování obrazu pro kvantitativní analýzu růstu kolonií kvasinek *Saccharomyces cerevisiae*, prováděnou z obrazu těchto kolonií na platformě Uni1P s rozšiřujícími moduly DX64.

Kvantitativní analýza růstu kvasinek či jiných mikroorganismů (např. bakterií) se používá v některých experimentech pro stanovení vlivu testovaných látek, obsažených v kultivačním médiu, na růst kolonií. Jedná se o látky, které nejsou v médiu přirozeně přítomné, jako jsou antibiotika, mutageny nebo testovaná léčiva. Při těchto experimentech rostou kolonie na pevném médiu (na Petriho miskách) v kultivačním boxu (termostatu). Rychlost jejich růstu je potom, kromě složení média, závislá na fázi růstu a hustotě kolonií a na teplotě a vlhkosti v kultivačním boxu.

Pro vyhodnocení experimentů je nutné stanovit relativní plochu, hmotnost a počet kolonií na misce v průběhu času. Pro vyhodnocení plochy a počtu kolonií se používá snímkování Petriho misek pomocí kamery nebo skeneru v časových odstupech daných rychlostí růstu kolonií, použitých v daném experimentu – např. v námi uvažovaném případě kvasinek *Saccharomyces cerevisiae* byly snímky pořizovány ve druhém, třetím, pátém a sedmém dni po výsevu.

Příklady snímků jsou na obrázku 1 (pořízeny Katedrou genetiky a mikrobiologie Přírodovědecké fakulty Univerzity Karlovy).



Obrázek 1: Příklady snímků Petriho misek

Vlastní vyhodnocení snímků bývá ve výzkumné praxi často prováděno ručně - na zvětšeném snímku je měřen průměr jednotlivých kolonií, ze kterého je počítána jejich plocha. Je zřejmé, že tento postup je jak velmi časově náročný (při experimentech je běžné vyhodnocovat desítky až stovky vzorků), tak náročný na soustředěnost operátora a, z podstaty metody, zatížený relativně vysokou chybou danou ručním prováděním měření.

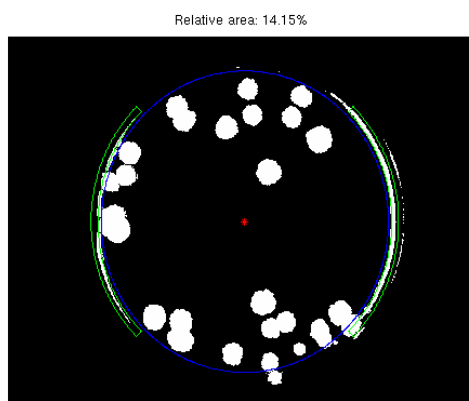
Z těchto důvodů pramení snaha ho automatizovat s použitím metod počítačového zpracování obrazu. U snímků, které měly být zpracovány námi vyvíjeným softwarem, bylo třeba respektovat následující specifika, daná postupem snímkování, použitým při pořizování výše uvedených příkladů:

- misky mohou být do snímacího zařízení (pod kameru) vkládány ručně, z toho vyplývají odchylky v jejich poloze mezi jednotlivými snímky.
- misky jsou osvětlovány dvěma postranními lineárními zdroji (zářivkami), jejich přesné nastavení není kalibrováno a může se mezi jednotlivými experimenty lehce změnit.
- výška média v miskách může být různá, snímky jsou proto doostřovány, tím se může v malých mezích měnit rozměr misky na snímku
- odchylky polohy misky na snímku mohou způsobit fluktuace intenzity osvětlení, projevující se změnou úrovně šedi na snímku.
- misky jsou při snímkování podloženy matným černým pozadím.

Z výše uvedeného je zřejmé, že nejdříve je třeba určit přesnou polohu misky na snímku a nastavit práh snímku tak, aby bylo odmaskováno jeho pozadí. Z toho vyplývá následující postup zpracování snímků:

1. najdi intenzitu pozadí, nastav práh tak, aby bylo odříznuto
2. upřesni polohu misky ve snímku
3. urči práh tak, aby bylo odříznuto pozadí kolonií v misce
4. spočítej plochu misky na snímku a plochu kolonií, obsažených na misce
5. proveď detekci hran kolonií jako přípravu pro určení počtu kolonií

Pokud jde o určení polohy misky, z příkladů snímků, uvedených na obr. 1, je zřejmé, že pro snímky jsou charakteristické vertikální srpkovité odrazy osvětlení na vnitřních bocích misky. Tyto odrazy musí být z obrazu odfiltrovány, aby nezkreslovaly celkovou spočtenou plochu kolonií. V našem řešení jsme se rozhodli využít korelace těchto odrazů s maskou obsahující 2 čtvrtkruhové segmenty pro určení přesné polohy misky ve snímku: maska je posouvána po snímku; v místě, kde překryje odrazy, leží maximum její korelace s obrazem (tato poloha je pro ilustraci zobrazena na obr. 2; maska je nakreslena zelenou barvou).



Obrázek 2: Výsledná poloha vyhledávací masky na snímku

Tento dokument stručně popisuje prototyp software pro realizaci výše zmíněných kroků zpracování s využitím hardwarové akcelerace výpočtu hranové detekce pomocí platformy Uni1P/DX64. Podrobný popis použití této platformy je obsažen v příručce[7].

2 Program na straně PC

Program na straně osobního počítače (hostitelského systému) je napsaná v jazyku C. Tento program nejprve zajistí inicializaci akcelérátoru: inicializaci karty Uni1P (sekce 3), nahrání příslušného kódu do modulu DX64 (sekce 4) a jeho spuštění. Pro komunikaci s deskou Uni1P používá knihovny WinDriver, ve kterých je podpořena komunikace s obvodem PLX9054, ovládajícím PCI rozhraní akcelerační desky.

Před nahráním programu musí být již nakonfigurovány obvody FPGA na desce Uni1p a na daném modulu DX64. Než je program nahrán, je DSP držen v resetovacím stavu a po nahrání programu přes HPI rozhraní je z tohoto stavu uvolněn.

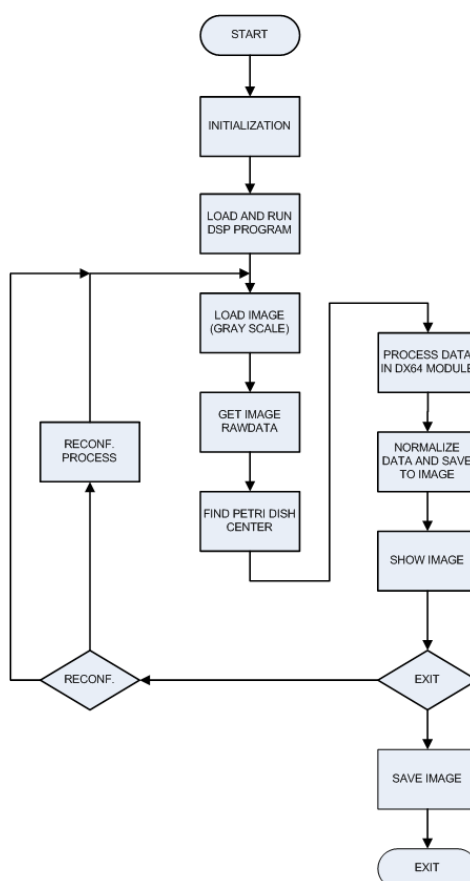
Jednotlivé šedotónové snímky jsou postupně načítány podle seznamu obrazů pomocí knihovny OpenCV. Seznam cest k obrazům je obsažen v souboru `images.lst`. Následuje vyhledání středu misky a vymaskování pozadí (mimo kolonie) černou barvou. Takto připravené snímky jsou zpracovány hranovým detektorem na modulu DX64. Výchozí implementace detektoru je navrhnutá pro 752 obrazových sloupců, což

odpovídá šířce vstupních obrazů s Petriho miskami. Obrazová data jsou nahrána do sdílené paměti DSP na daném modulu DX64 a data jsou zpracována postupem, popsáním v sekcích 5.2 a 5.4. Na obrázku 3 je zobrazen vývojový diagram programu běžícího na straně osobního počítače.

Každý obrazový bod vstupních dat je reprezentován jedním bajtem a pro výstupní data dvěmi bajty. Výstupní data Sobelova detektoru jsou na straně PC normalizována a poté jsou použita k zobrazení obrazu. Hranová detekce se provádí pouze v oblasti, kde byla detekována miska s koloniemi. Pro tento účel je implementován hranový detektor pro 512 obrazových sloupců, který odpovídá šířce Petriho misky. Hranový detektor lze za běhu programu rekonfigurovat. Pokud dojde k rekonfiguraci, je pro hranovou detekci využit pouze výřez z obrazu, jehož pozice je dána nalezeným středem misky. Ovládání aplikace je shrnuto v tabulce 1.

Klávesa	Význam klávesy
0..9	Volba bitstreamu pro rekonfiguraci.
ESC	Ukončení aplikace.

Tabulka 1: Ovládání aplikace pomocí klávesnice.



Obrázek 3: Vývojový diagram programu pro osobního počítače.

2.1 Spuštění programu

Program je spouštěn pomocí příkazu `sobe1`, který inicializuje systémové FPGA na desce Uni1P a FPGA na modulech DX64 a spustí program na straně PC. Program komunikuje s modulem M2 (třetí pozice na desce Uni1P) a spouští se následovně:

```
sobel.exe <dsp_binary_file> <image_list>
[[<bitstream_0> <width_0>] ... [<bitstream_9> <width_9>]]
```

Parametr `<dsp_binary_file>` je cesta k binárnímu souboru obsahující program pro DSP na modulu DX64, `<image_list>` je cesta k souboru se seznamem zpracovávaných obrazových souborů. Následujícími parametry může být seznam (částečných) bitstreamů filtrů a k nim příslušných šířek v obrazových bodech, pro jaké byly jednotlivé filtry implementovány. Maximální počet parametrů pro jednotlivé bitstreamy je omezen na deset. Seznam bitstreamů může obsahovat celkové nebo částečné bitstreamy. Částečné bitstreamy lze použít za předpokladu, že byl předem nahrán celkový bitstream (například v rámci inicializace desky).

3 Stručný popis karty Uni1P

Uni1P je vývojová karta, určená pro připojení na sběrnici PCI osobního počítače. Rozhraní PCI je implementováno pomocí obvodu PLX9054, který zajišťuje komunikaci se zbývajícími komponentami desky a osobního počítače.

Softwarová podpora PCI rozhraní desky je v operačním systému Windows zajištěna pomocí "WinDriver PCI for Windows Driver Development Toolkit" [1].

Uni1P obsahuje tři obvody FPGA, jeden obvod CPLD a čtyři sloty pro moduly DX64. Na obrázku 4 je znázorněna architektura Uni1P.

Funkce jednotlivých obvodů je následující:

- CPLD slouží k řízení konfiguračního rozhraní systémového FPGA (System FPGA) v režimu "Slave SelectMAP".
- "System FPGA" řídí konfigurační rozhraní zbývajících FPGA na desce Uni1P a obvodů FPGA na modulech DX64.
- "Master FPGA" je použito pro komunikaci s moduly DX64 a pro komunikaci s externím rozhraním.
- "Comm FPGA" je určeno pro obsluhu paměťového rozhraní SDRAM paměti a obsluhu rozšiřujícího konektoru pro obecné užití.

Podrobnosti lze nalézt např. v[7].

4 Modul DX64

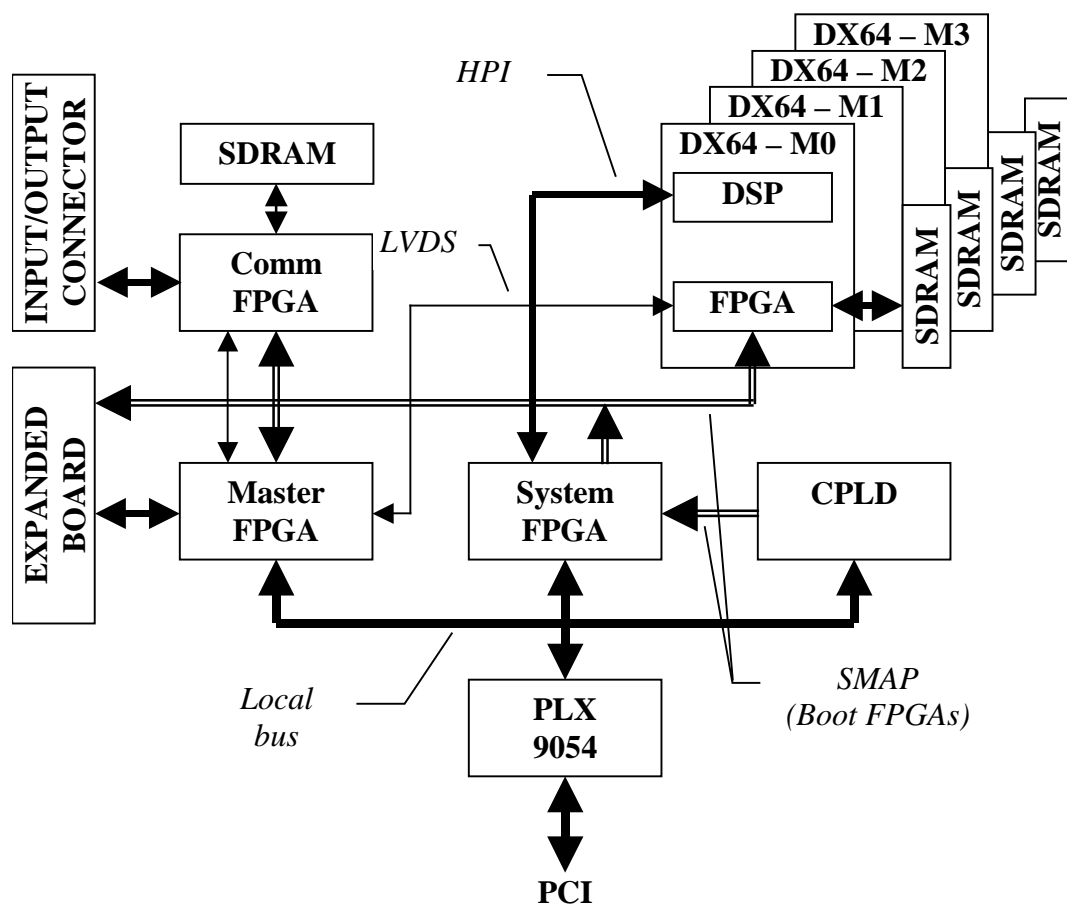
Modul DX64 je osazen signálovým procesorem firmy Texas Instruments TMS320C6416 (DSP), obvodem FPGA firmy Xilinx Virtex-II (xc2v250-4fg256 nebo xc2v1000-4fg256) a pamětí SDRAM (2 x SDRAM_8M32).

Obvod FPGA a DSP jsou spolu propojeny pomocí rozhraní EMIFB a zároveň jsou propojeny přes konektor modulu s deskou Uni1P. DSP je propojeno s Uni1P pomocí rozhraní HPI. Obvod FPGA je propojen s Uni1P pomocí tří rozhraní: rozhraní standardu LVDS, rozhraní, propojujícího sousední moduly, a konečně pomocí rozhraní pro paměť SDRAM, osazenou na Uni1P.

5 Implementace hranového detektoru

Jak bylo uvedeno již v sekci 2, byl s využitím desky Uni1P a moduly DX64 implementován obrazový filtr s podporou částečné dynamické rekonfigurace. Jako obrazový filtr byl implementován Sobelův hranový detektor.

Filtr je implementován v obvodu FPGA na modulu DX64 a obrazová data jsou předávána pomocí připojeného DSP.

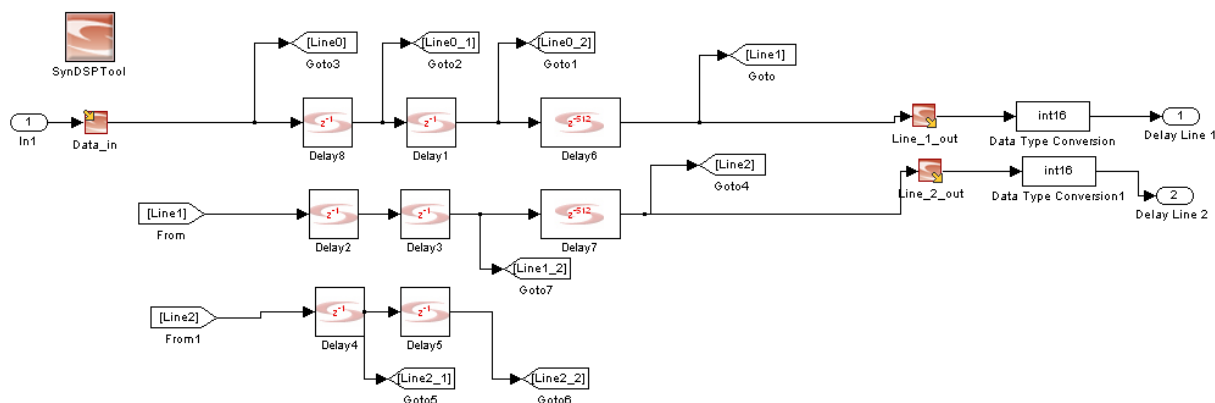


Blokové schéma Uni1P

Obrázek 4: Architektura vývojové desky Uni1P

5.1 Návrh hranového detektoru pomocí nástroje Synplify DSP

Filtr byl navržen pomocí nástroje Synplify DSP. Tento nástroj je nadstavbou prostředí Simulink firmy MathWorks. Obrazový filtr je popsán schématickým propojením jednotlivých výpočetních bloků (viz obrázek 5). Výsledné propojení lze exportovat do jazyku pro popis hardware, například do jazyku VHDL. Během tohoto exportu je provedena optimalizace vzhledem k cílové platformě (FPGA).



Obrázek 5: Příklad zapojení části hranového detektoru v Synplify DSP

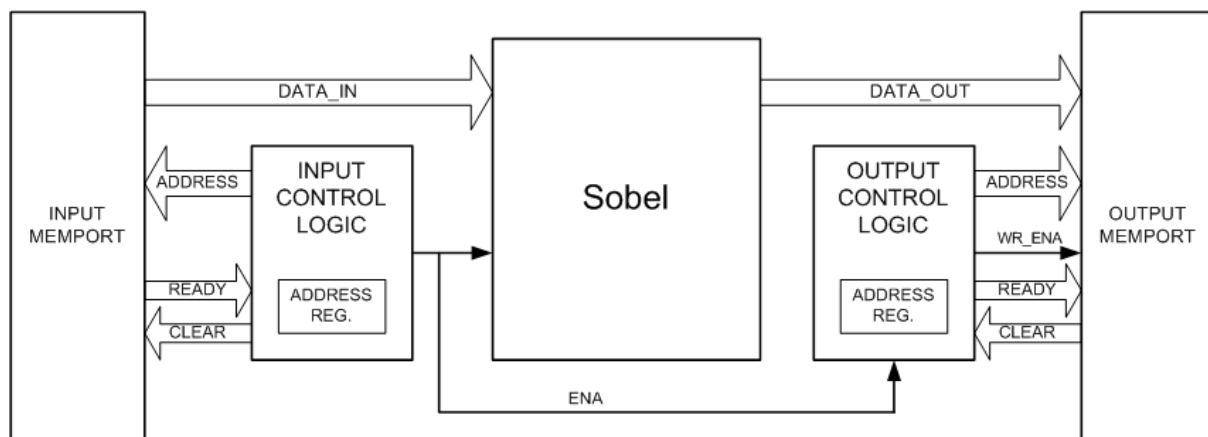
Model hranového detektoru je uložen na přiloženém médiu v adresáři Sobe1/. Po otevření modelu jsou načteny parametry přiloženého obrazu `image.jpg`. Parametry obrazu jsou použity při návrhu a generování hranového detektoru. Pro vygenerování hranového detektoru je třeba nejdříve spustit simulaci a poté detektor syntetizovat pomocí "SynDSP Tool". Po dokončení syntézy jsou vygenerovány VHDL soubory s implementací hranového detektoru, test a testovací data.

5.2 Propojení hranového detektoru a paměťových portů

Výsledkem exportu hranového detektoru je zdrojový kód komponenty detektoru ve VHDL, který bude instanciován v celkovém návrhu aplikace v FPGA. V tomto případě je třeba detektor připojit k paměťovým portům, pomocí nichž budou předávána obrazová data.

Jeden vstupní port bude využit pro vstupní data a druhý pro výstupní data. K hranovému detektoru je třeba připojit paměťová rozhraní portů a řídit jejich adresování. Adresování pamětí je určeno na základě signálů určující obsazenost sektorů paměťových portů a vlastní hodnotě čítače adres v rámci jednoho vstupního a jednoho výstupního sektoru. Každý sektor je po zpracování příslušným způsobem označen v registru `MEMPORT_READY`, a tak může být informováno DSP o průběhu zpracování dat.

Na obrázku 6 je znázorněna implementace propojení filtru s paměťovými porty, implementovanými v FPGA. Řídící logika na vstupu adresuje paměť vstupního portu od prvního (od nejnižší adresy) sektoru s platnými daty. Platnost dat v jednotlivých sektorech je určena pomocí signálu `READY`, který je odvozen od příslušných hodnot v registru `MEMPORT_READY`. Po zpracování jednoho vstupního sektoru je sektor označen jako prázdný pomocí signálu `CLEAR` a je adresován další sektor. Pokud žádný sektor není platný je detektor pozastaven pomocí signálu `ENA`. Obdobným způsobem je obsloužen výstup detektoru, který zapisuje na poslední volný sektor výstupního portu. Detektor má jistou latenci než vybaví data. Latence je kompenzována zpožděním signálu `ENA` (zpožděný signál `WR.ENA`), který povoluje zápis do paměti výstupního portu a její adresaci. Během zápisu jsou postupně označovány sektory s platnými daty. Řídící logika si udržuje informaci o všech sektorech, do kterých bylo zapsáno a předává tyto hodnoty do registru `MEMPORT_READY` pomocí signálu `READY`. Pokud dojde k zápisu do registru `MEMPORT_READY` ze strany DSP jsou tyto hodnoty smazány pomocí signálu `CLEAR`. Smazání informace o platnosti sektoru je provedeno zápisem logické 1 na příslušné pozici.



Obrázek 6: Znázornění propojení paměťových portů a hranového detektoru

5.3 Využití částečné dynamické rekonfigurace

Pro efektivnější využití obvodů FPGA na modulu DX64 se nabízí využít částečnou dynamickou rekonfiguraci a měnit obrazové filtry za běhu aplikace. Část implementace obsahující paměťové porty a rozhraní EMIFB může být použita beze změny (statická část) a měněna může být pouze část s implementovaným obrazovým filtrem (dynamická část).

Propojení mezi statickou a dynamickou částí tvoří datové rozhraní filtru, které se skládá z 8-mi bitového datového vstupu, 16-ti bitového datového výstupu, signálu určující platnost dat a resetovacího signálu.

Pokud je signál určující platnost dat aktivní, tak je každý hodinový takt načtena hodnota jednoho obrazového bodu. Platnost výstupních dat je dána (konstantní) latencí filtru.

Vstupní a výstupní data jsou předávána pomocí adresovací logiky ve statické části z/do adresového prostoru paměťových portů.

5.4 Program pro DSP

Program DSP obsluhuje předávání dat ze sdílené paměti mezi DSP a hostitelským systémem. Komunikace mezi DSP a hostitelským systémem je řízena pomocí hodnot sdílených proměnných v paměti. Předávány jsou hodnoty adres, velikostí a platnosti vstupních a výstupních dat.

Na obrázku 7 je znázorněn vývojový graf hlavního programu a obslužných rutin přerušení. Hlavní program čeká na vstupní data a postupně je zpracovává v závislosti na počtu volných vstupních sektorů. Přenos dat do vstupních sektorů je inicializován pomocí řadiče EDMA. Po dokončení přenosu dat jsou označeny všechny zapsané sektory jako platné. Konec přenosu je indikován přerušením od EDMA řadiče. Obslužná rutina přerušení zkontroluje zda šlo o přenos vstupních a nebo výstupních dat. Pokud byly přenášeny vstupní data, pak hlavní program přestane čekat na dokončení přenosu vstupních dat a pokračuje. Pokud došlo k přenosu výstupních dat, jsou přenesené sektory uvolněny. Vstupní sektory jsou zpracovány pomocí hranového detektoru a výstupní data jsou postupně zapisovány do výstupních sektorů které jsou označovány za platné. Pomocí externího přerušení EXT4 je DSP informován o platných datech ve výstupních sektorech a je provedena obslužná rutina. Obslužná rutina pro signál EXT4 určí počet přenášených sektorů a jejich adresu a inicializuje EDMA přenos. Po dokončení přenosu je generováno přerušení EDMA řadiče a postupuje se tak jak bylo popsáno výše.

Pokud byla zpracována všechna data, pak je informován nadřazený systém o dokončení zpracování jednoho obrazového snímku.

6 Závěr

V dokumentu bylo popsáno použití akcelerační desky Uni1P s moduly DX64 pro zpracování snímků kolonií kvasinek na Petriho miskách. Byla popsána implementace filtru pro hranovou detekci obrazu, jako příklad možné implementace obrazového filtru na této architektuře.

Příklad obrazového výstupu hranové detekce z obrazu 8 je na obrázku 9.

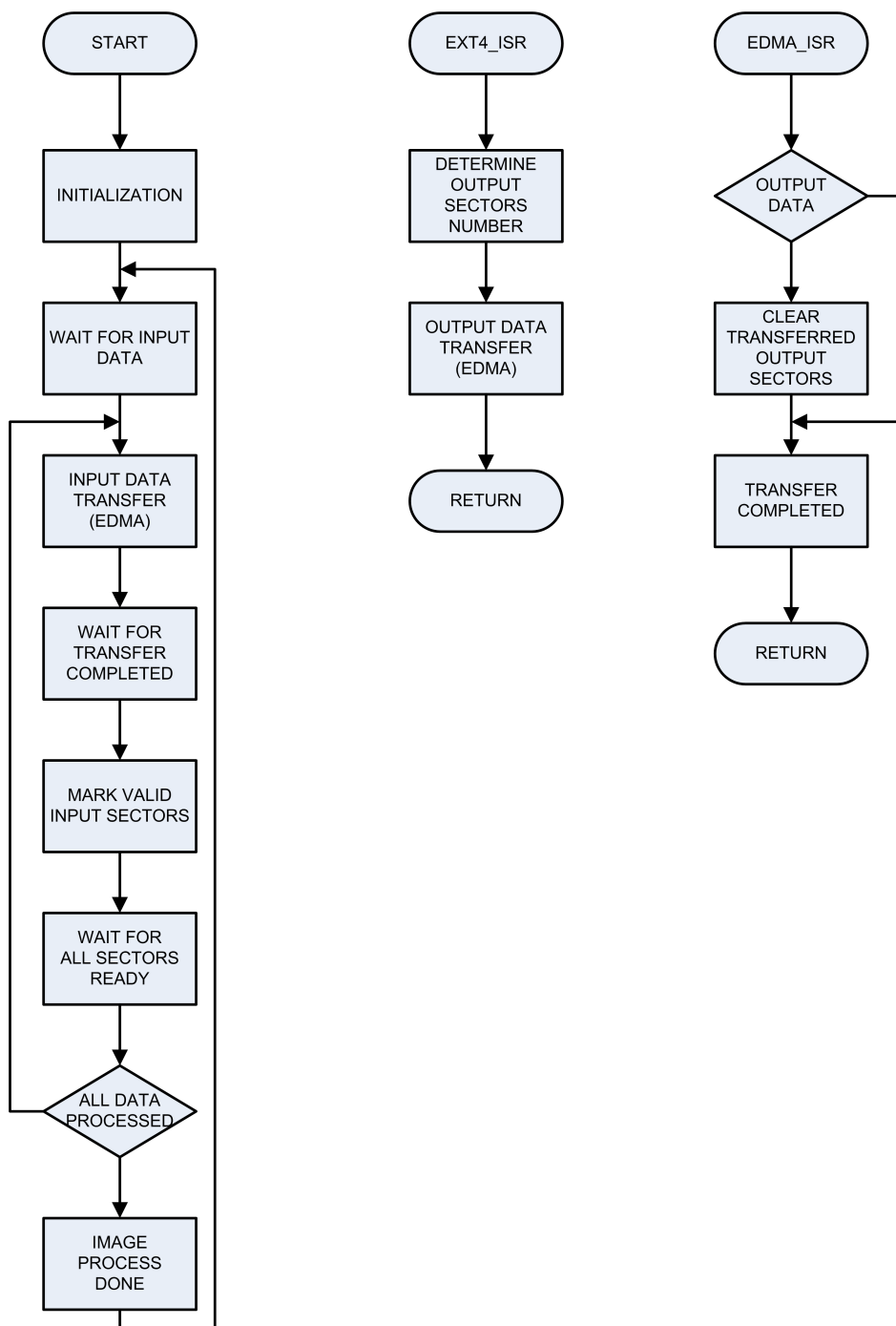
V aplikaci je využita částečná dynamická rekonfigurace, která je použita pro změnu funkce hranového detektoru. Struktura implementace hranového detektoru je závislá na šířce zpracovávaného obrazu. Vzhledem k efektivnímu využití zdrojů obvodu FPGA je hranový detektor implementován v několika modifikacích pro dané rozměry obrazu a za běhu aplikace je propojení obvodu FPGA měněno. S využitím částečné dynamické rekonfigurace je dosaženo kratší doby potřebné pro změnu funkce obvodu, velikost paměti potřebné pro uložení bitstreamu a navíc může být část funkce obvodu zachována ve statické části. Velikost bitstreamů pro jednotlivé implementace hranového detektoru a potřebná doba pro rekonfiguraci je uvedena v tabulce 2. Úspora paměti pro uložení bitstreamu i doby rekonfigurace se v našem případě pohybuje okolo 68 %.

Bitstream	Velikost bitstreamu	Doba rekonfigurace
Plný bitstream	199 204 B	7,97 ms
Částečný pro 752 px	62 600 B	2,49 ms
Částečný pro 512 px	62 252 B	2,50 ms

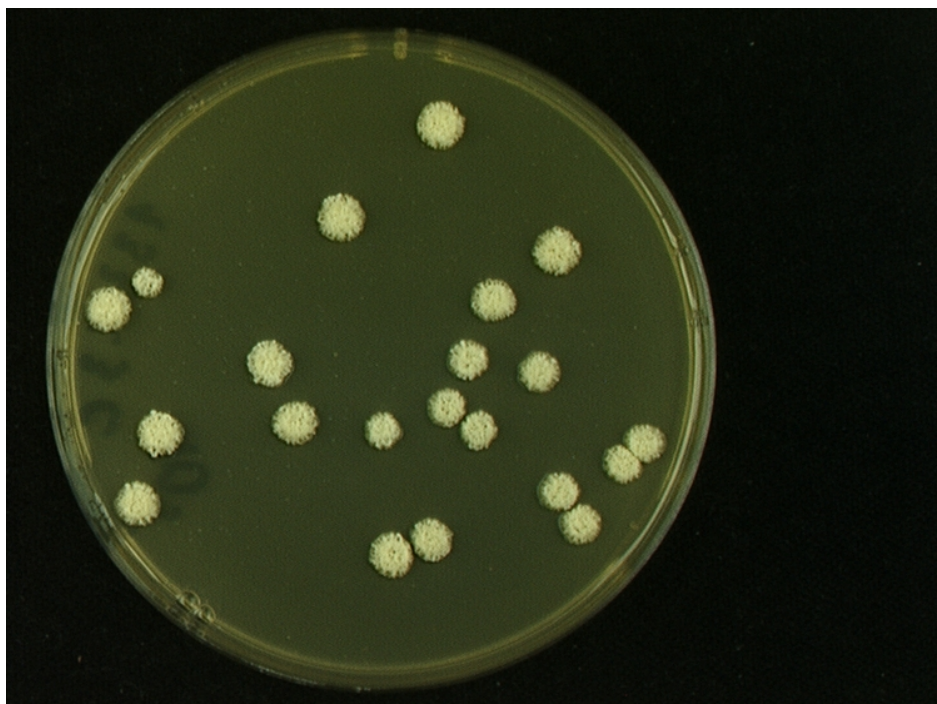
Tabulka 2: Výsledky úspor při využití částečné dynamické rekonfigurace

7 Poděkování

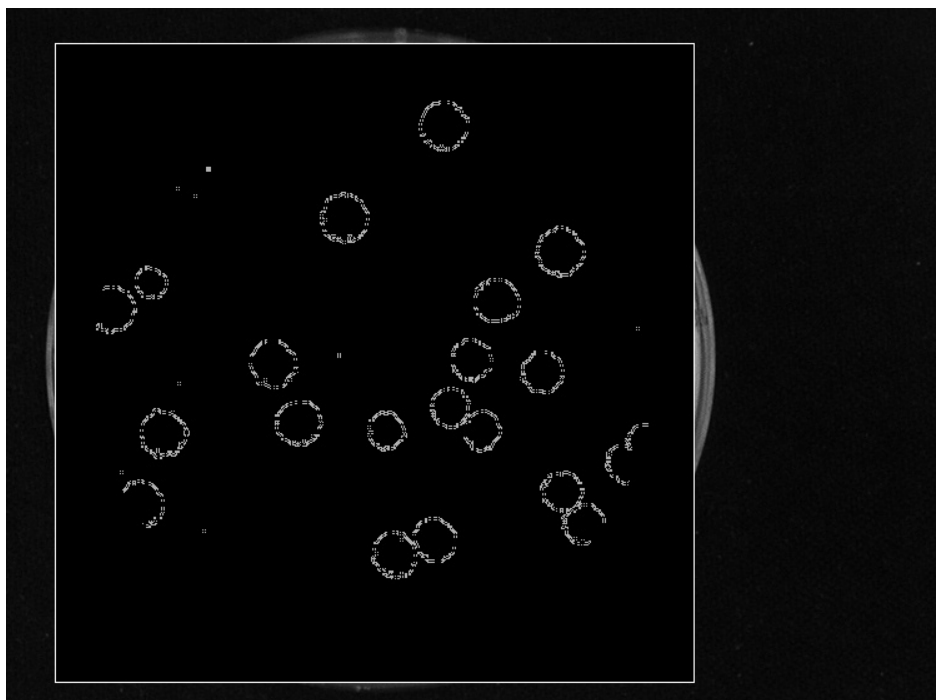
Tato zpráva vznikla za podpory grantu AVČR č. 1ET400750408 – Rapid prototyping tools for development of HW-accelerated embedded image- and video-processing applications.



Obrázek 7: Vývojový diagram programu pro DSP.



Obrázek 8: Zdrojový obraz před hranovou detekcí.



Obrázek 9: Příklad výřezu obrazu s hranovou detekcí.

8 Výpis CDROM

Na CD se nachází text dokumentu, zdrojové kódy pro programy DSP a PC, model hranového detektoru v Synplify DSP a VHDL kódy pro obvody FPGA.

Příložené CD má následující adresářovou strukturu:

```
.
|-- boot_sobel_dish/           Aplikace pro hranovou detekci
|   |-- bin/                  Bitstreamy pro obvody FPGA
|   |-- common/               Společné soubory pro program DSP a PC
|   |-- dsp/                  Zdrojové kódy programu pro DSP
|   |   '-- PR_Flow/
|   |       '-- synth/       VHDL zdrojové kódy
|   |           |-- common/   Zdrojové kódy knihoven
|   |           |-- rmodule_0/ Zdrojové kódy pro Sobel (1024 px)
|   |           |-- rmodule_1/ Zdrojové kódy pro Sobel (512 px)
|   |           |-- rmodule_2/ Zdrojové kódy pro Sobel (752 px)
|   |           |-- static/    Zdrojové kódy statické části
|   |           '-- top/      Zdrojové kódy pro nejvyšší úroveň návthu
|   |-- HPIheader/            Definice paměťového prostoru HPI
|   |-- images/               Obrazy misek s koloniemi kvasinek
|   |-- PC_part/              Program pro PC
|   |   |-- lib               Potřebné knihovny
|   |   '-- src               Zdrojové kódy programu
|   |-- utils/                Nástroje pro komunikaci s deskou Uni1P
|   '-- sobel.bat              Dávka pro spuštění aplikace
|-- doc/                      text dokumentu ve formátu PDF
|-- Sobel/                    Model hranového detektoru - Synplify DSP
'-- readme.txt
```

Reference

- [1] Jungo Ltd. *WinDriver PCI for Windows Development Tool* [online]. Dostupné na WWW:
<http://www.jungo.com/>
- [2] Xilinx. *Virtex-II Platform FPGA User Guide* [online]. Dostupné na WWW:
http://www.xilinx.com/support/documentation/user_guides/ug002.pdf
- [3] Xilinx. *Virtex FPGA Series Configuration and Readback* [online]. Dostupné na WWW:
http://www.xilinx.com/support/documentation/application_notes/xapp138.pdf
- [4] Hegar, A.: *Uni1P - Registry*, FIT VUT, Brno, 2005.
- [5] Texas Instruments. *TMS320C6414, TMS320C6415, TMS320C6416 FIXED-POINT DIGITAL SIGNAL PROCESSORS* [online]. Dostupné na WWW:
<http://focus.ti.com/lit/ds/sprs146n/sprs146n.pdf>
- [6] Texas Instruments. *TMS320C6000 Peripherals Reference Guide* [online]. Dostupné na WWW:
<http://archer.ee.nctu.edu.tw/class/dsplab/07f/..%5C07s%5Creference%5Cspru190d.pdf>
- [7] Kloub, J.: *Aplikační příručka k vývojové desce Uni1P a modulům DX64*, ÚTIA AV ČR, Praha, 2008