Elektrotehniški vestnik 69(1): 1–6, 2002
Electrotechnical Review, Ljubljana, Slovenija

Modularni FPGA prototipni sistem za obdelavo slik

Andrej Trost, Andrej Žemva, Baldomir Zajc

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: andrej.trost@fe.uni-lj.si

Povzetek. Prototipni sistemi za razvoj in testiranje digitalnih vezij danes temeljijo na vezjih, ki jih je mogoče programirati. Ta vezja z oznako FPGA (Field Programmable Gate Array) imajo matrično strukturo, ki omogoča izdelavo kompleksnih logičnih vezij. Zmogljivost prototipnih sistemov povečamo z uporabo več programirljivih vezij in gradnikov, ki jih z vezji FPGA ne moremo učinkovito narediti (npr. pomnilniki). Glede na namen uporabe delimo prototipne sisteme na univerzalne sisteme in sisteme, ki so specializirani za določeno aplikacijo. V članku bomo predstavili specializiran prototipni sistem za obdelavo slik v realnem času. Prototipni sistem je zgrajen modularno, tako da ga lahko nadgrajujemo z razvojem tehnologije vezij FPGA. Topologija sistema je prirejena za izdelavo vezij v cevovodni arhitekturi. Takšna zgradba omogoča vzporedno izvajanje posameznih korakov obdelave slik nad podatki, ki so shranjeni v lokalnih pomnilnikih. Tudi postopek načrtovanja in testiranja vezij na našem prototipnem sistemu je prirejen za vezja s področja obdelave slik. Delovanje vezij lahko preskušamo po korakih ali pa opazujemo rezultat obdelave slike v realnem času. Za preskušanje vezij po korakih smo naredili vmesnik, s katerim prenašamo testne vektorje in opazujemo odzive posameznega vezja na računalniku. Predstavili bomo tudi nekaj aplikacij s področja obdelave slik, ki smo jih realizirali na prototipnem sistemu.

Ključne besede: prototipni sistemi, FPGA, obdelava slik v realnem času

Modular FPGA Based Prototyping System for Image processing

Extended abstract. Rapid prototyping systems for design and implementation of digital circuits are today based on programmable devices. Field Programmable Gate Arrays (FPGA) are programmable devices used for implementation of complex digital circuits. In order to further increase the logic and memory capacity of the prototyping systems, they consist of more FPGA devices with additional memory modules. Rapid prototyping systems can be divided into general purpose and application specific. Real time image processing is a typical application for the FPGA based prototyping systems. Many image processing algorithms can be efficiently implemented with parallel architectures. This is a preferred solution when real time constraints and energy efficiency are to be coped with.

In the paper, we present a novel prototyping system for real time image processing. The prototyping system consists of a motherboard and up to six modules with FPGA devices, as presented in Figure 1. The FPGA modules are connected to a segmented data bus, which enables parallel communication between modules with less wiring resources compared to direct interconnection. This topology is optimized for implementation of data processing circuits in a pipeline architecture.

We built FPGA modules based on different generations of FPGA devices. A typical module is composed of one or two FPGA devices, additional static memory and external connectors. Figure 2 presents an arhitecture of the module containing a Xilinx Spartan FPGA device. The FPGA device has a logic capacity of up to 40.000 equivalent logic gates. Additional memory is used for local data storage in image processing circuits. The contemporary FPGA devices have even increased the logic capacity (up to a few million gates) and internal memory for a few image lines. With these devices we will be able to extend our prototyping system.

The circuits implemented on the FPGA module can be tested before the modules on the prototyping system are integrated. The module is for this purpose connected to a hardware verification interface. The interface is composed of a programmable CPLD device and is used for configuration of FPGA modules and hardware verification of the implemented circuit.

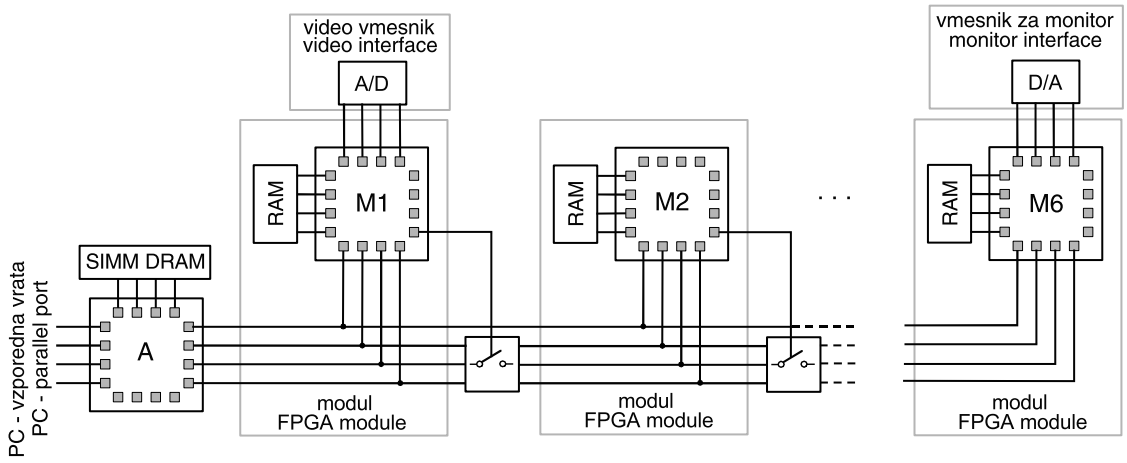
The modular prototyping system contains also some analog interfaces used in real-time operation of the system. One interface is connected to the first FPGA module and used for AD conversion of the video signal. The other interface converts the output digital stream to the RGB signals driving a computer monitor.

Figure 3 presents the design cycle on the prototyping system. First, the algorithm is described in a high level language and tested on a sample images. Then the desired circuit architecture is derived and the circuit in a VHDL environment designed. The circuit can be simulated with the same set of sample images which are converted to the test vector format. After the circuit is synthesized and compiled, the test vectors are used for hardware verification. We designed a software environment, presented in Figure 4, which provides user friendly hardware verification of the designs implemented in the FPGA module.

In the last chapter, some image processing algorithms implemented in the prototyping system are presented. We designed image filtering circuits, Canny edge detector, as presented in Figure 5, and image rotation algorithm. Results of technology mapping to the second generation FPGA modules are presented in Table 1.

Key words: rapid prototyping systems, FPGA, real time image processing

Prejet 30. november, 2001
Odobren 21. januar, 2002



Slika 1. Programirljivi prototipni sistem za obdelavo slik
Figure 1. Programmable prototyping system for image processing

1 Uvod

Digitalni sistemi za obdelavo slik so narejeni kot računalniški sistemi ali pa v obliki specializiranih digitalnih vezij. Digitalna vezja za obdelavo slik so zasnovana tako, da izvajajo več operacij hkrati, v nasprotju z računalniškimi sistemi, kjer mikroprocesor izvaja ukaze zaporedno. Hkratno izvajanje operacij pospeši izvedbo celotnega algoritma, kar omogoča obdelavo slik v realnem času ter delovanje vezij pri nižjih frekvencah in s tem povezani nižji porabi energije [1].

Načrtovanje specializiranih vezij za obdelavo slik je tesno povezano z razvojem algoritmov za obdelavo digitalnih slik. Ob spremembi algoritma lahko pride do spremembe zgradbe (topologije) elektronskega vezja na ravni podatkovnih poti ali pa na ravni operatorjev. Prototipni sistem, ki ga uporabljamo pri razvoju digitalnih vezij, mora biti dovolj fleksibilen, da omogoča vse te spremembe.

Gradniki prototipnih sistemov so danes programirljiva vezja, predvsem vezja FPGA (Field Programmable Gate Array), ki omogočajo takojšnjo izdelavo in spreminjanje vezij na ciljnem sistemu. Zmogljivost programirljivih vezij se vsako leto povečuje, tako da posamezna vezja že omogočajo izdelavo digitalnih vezij z več kot milijon logičnimi vrati [2]. Programirljivi prototipni sistemi so zgrajeni iz enega ali več programirljivih vezij in so običajno povezani z osebnim računalnikom, na katerem poteka načrtovanje in testiranje vezij [3]. S stališča obdelave slik, jih lahko razdelimo na univerzalne prototipne sisteme [4], ter specializirane sisteme za obdelavo digitalnih slik. Specializirani prototipni sistemi vsebujejo vmesnike za prenos slikovnih podatkov iz kamere ali osebnega računalnika ter imajo tudi programsko podporo za načrtovanje in testiranje vezij za obdelavo slik [5].

V članku bomo predstavili nov prototipni sistem za obdelavo slik, ki je zgrajen modularno. Sistem je prirejen za vezja v cevovodni arhitekturi in omogoča preskus delovanja vezij v realnem času (z videokamero in monitorjem) [6]. Prikazana bo zgradba posameznih modulov s programirljivimi vezji ter njihova uporaba pri izvedbi algoritmov za obdelavo slik. Predstavili bomo tudi potek načrtovanja in testiranja vezij za obdelavo slik, ki je podprt s specifično programsko opremo za naš prototipni sistem.

2 Modularni prototipni sistem

Prototipni sistem za obdelavo slik je zgrajen modularno, kar omogoča preprosto nadgradnjo in prilagoditev sistema ciljni aplikaciji. Sistem je sestavljen iz matične plošče in modulov s programirljivimi vezji, kot prikazuje slika 1. Matična plošča trenutno podpira priključitev šestih modulov. Operacije obdelave slike se izvajajo na posameznih modulih, ki vsebujejo lokalne pomnilnike za hranjenje slikovnih točk. Podatki se med moduli prenašajo prek vodila, ki je razdeljeno na lokalne segmente, tako da lahko komunikacja poteka med več pari modulov hkrati.

Topologija vodila, razdeljenega na segmente, je glavna značilnost našega sistema v primerjavi z znanimi prototipnimi sistemi, ki so sestavljeni iz več programirljivih vezij [5,7,8,9]. Segmentirano vodilo nam omogoča hitro vzporedno komunikacijo med sosednjimi moduli ob pol manjšem številu podatkovnih signalov na posameznem vezju FPGA. S tem nam ostane več prostih priključkov za lokalni pomnilnik in dodatne vmesnike, ki jih potrebujemo za obdelavo videosignala.

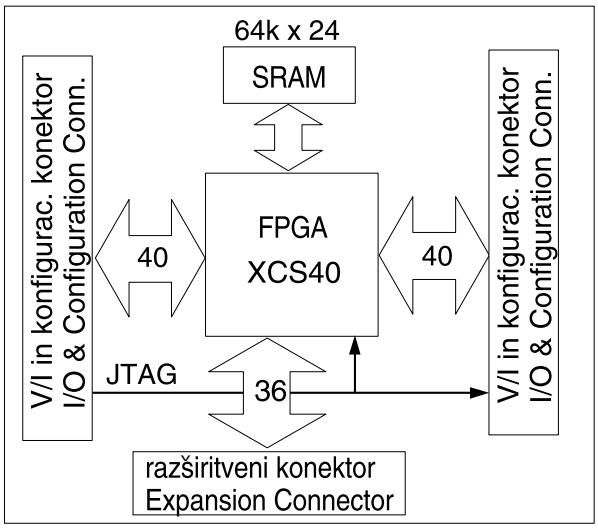
Prototipni sistem je povezan z osebnim računalnikom prek vzporednih vrat in vezja FPGA na matični plošči. Z osebnim računalnikom opravljamo konfiguriranje posameznih modulov in strojno verifikacijo celotnega sistema. Na matični plošči je tudi podnožje za dinamični

pomnilniški modul, ki ga uporabimo kot globalni pomnilnik za celotno vezje.

2.1 Zgradba modulov

Moduli s programirljivimi vezji vsebujejo priključek na univerzalno 64 polno vodilo, prek katerega poteka prenos podatkov ter konfiguriranje in testiranje vezij FPGA. Za prenos podatkovnih in krmilnih signalov med modulom in matično ploščo je na voljo 40 vhodno- izhodnih povezav. Konfiguriranje programirljivih vezij poteka prek serijskega protokola, za testiranje vezij pa uporabljamo verigo JTAG. Izdelali smo module z različnimi generacijami vezij FPGA proizvajalca Xilinx, ki se razlikujejo po logični zmogljivosti, hitrosti delovanja in razširitvenih možnostih.

Moduli prve generacije vsebujejo vezja iz družine Xilinx XC4000E. Posamezni modul vsebuje med seboj povezani vezji XC4008E ter XC4010E, ki si delita skupni statični pomnilnik v velikosti 64k $\times$ 16 bitov. Statični pomnilnik s hitrim dostopnim časom se glede na zahteve algoritma uporablja v vezjih za obdelavo slik za sprotno shranjevanje posameznih vrstic ali celotne slike. Univerzalni 64 polni konektor je vezan na prvo vezje FPGA, obe vezji pa imata še dodatna 40 polna razširitvena priključka, namenjena za dodatne pomnilniške module ali dodatne signalne povezave s sosednjimi moduli.



Slika 2. Moduli druge generacije
Figure2. Second generation FPGA modules

Moduli druge generacije so zgrajeni okoli vezja FPGA XCS40 iz družine Spartan. Poleg vezja FPGA vsebujejo še 64k $\times$ 24 bitov hitrega statičnega pomnilnika, razširitveni konektor ter dva univerzalna 64 polna konektorja, kot prikazuje slika 2.

Prvi 64 polni konektor je prirejen za priključitev modula na matično ploščo, drugi pa je namenjen za kaskadno povezavo več modulov ali pa za priključitev dodatnih vezij in vmesnikov na modul.

Programirljivi moduli tretje generacije vsebujejo vezja FPGA iz novejših Xilinxovih družin. Značilnost teh vezij je večja logična zmogljivost (zmogljivost posameznih vezij se je z nekaj 10.000 zvišala na nekaj 100.000 ali celo milijon logičnih vrat), višja hitrost delovanja ter vgrajeni pomnilniški bloki, ki so zelo primerni za shranjevanje posameznih vrstic pri obdelavi slik.

2.2 Vmesnik za testiranje modulov

Za testiranje vezij na posameznem modulu smo zgradili vmesnik, ki omogoča prenos testnih vektorjev z osebnega računalnika na vhode vezja ter prenos izhodnih signalov na računalnik. Prek vmesnika poteka tudi konfiguriranje vezij FPGA na prototipnih modulih. Vmesnik je povezan z osebnim računalnikom prek vzporednih vrat, vsa logika pa je narejena z enim vezjem CPLD XC95108.

Testiranje vezij poteka prek univerzalnega 64 polnega vodila, s katerim sta povezana vmesnik in prototipni modul. Na vodilu je na voljo 40 signalnih povezav, ki jih vezje vmesnika razdeli na pet 8-bitnih blokov. Pri inicializaciji vmesnika posamezen blok signalov definiramo kot vhodni ali pa kot izhodni blok. Testiranje vezij poteka s programom v osebnem računalniku, ki razdeli testne vektorje v 8-bitne skupine signalov ter jih pošlje v izhodne registre vmesnika. Ko je celotni testni vektor vpisan v izhodne registre, se prenese na vhode testiranega vezja. V naslednjem ciklu vmesnik vzorči izhode vezja ter jih v skupinah po 8 bitov prenese v osebni računalnik.

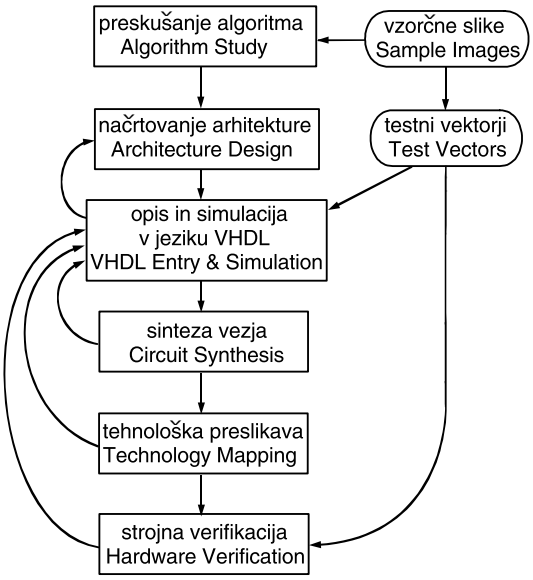
2.3 Analogni vmesniki

Na prototipnem sistemu za obdelavo slik potrebujemo tudi analogne vmesnike za vzorčenje videosignala ter prikaz slike na monitorju. Videosignal vzorčimo s hitrim analogno-digitalnim pretvornikom, pri dekodiranju pa si pomagamo še z analognim separatorjem sinhronizacijskih signalov. Vmesnik je priključen na prosti 64 polni konektor na modulu s Spartan vezjem. Celoten postopek dekodiranja videosignala in pretvorbe v digitalni tok podatkov krmili logika v vezju FPGA.

Drugi analogni vmesnik se uporablja za prikaz obdelane slike na računalniškem monitorju. Vmesnik sestavljajo trije digitalno-analogni pretvorniki za posamezne barvne komponente. Logika za krmiljenje pretvornikov, generiranje sinhronizacijskih signalov in krmiljenje medpomnilnika za izhodno sliko je narejena z vezjem FPGA iz družine Spartan. Medpomnilnik uporabljamo zaradi različnih frekvenc osveževanja slike — pri videosignalu se slika osvežuje s 25 Hz, na monitorju pa jo osvežujemo s frekvenco 50 Hz.

3 Postopek načrtovanja in testiranja vezij

Načrtovanje vezij za obdelavo slik začnemo z iskanjem in preskušanjem algoritma, kot prikazuje slika 3. Algoritem za obdelavo slike lahko napišemo v obliki programa v univerzalnem programskem jeziku (npr. C, Java) ali pa v matematičnem programskem okolju (npr. MatLab). Delovanje algoritma preskušamo na testnih slikah. Naslednji korak je načrtovanje zgradbe vezja, ki običajno poteka na shematski ravni. Izvajanja ukazov v algoritmu in pretok podatkov ponazorimo z grafi ali blokovnimi shemami. Pri načrtovanju zgradbe vezja iščemo strukture v cevovodni ali vzporedni obliki, ki bi bile najprimernejše za izvedbo algoritma.



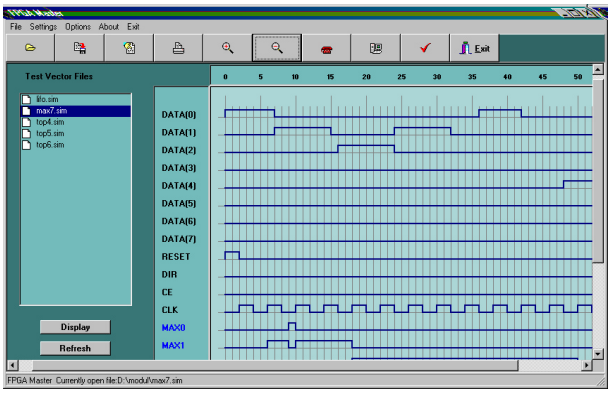
Slika 3. Postopek načrtovanja vezij za obdelavo slik
Figure 3. Design cycle for image processing circuits

Vnos vezja v računalnik danes poteka predvsem z visokonivojskimi jeziki za opis digitalnih vezij (VHDL, Verilog). Na tej stopnji se opravi simulacija vezja, kjer opazujemo delovanje posameznih operacij, ki sestavljajo algoritem, in celotnega vezja. Za simulacijo vezij, opisanih v jeziku VHDL, uporabljamo simulator Aldec Active-HDL [10], ki vsebuje tudi shematski urejevalnik za povezovanje posameznih komponent vezja ter pripomočke za izdelavo testnih programov. Vezja za obdelavo slik simuliramo v testnem okolju, napisanem v obliki VHDL programa (Testbench), ki iz datoteke s slikami bere točke in jih pošilja na vhode vezja, rezultate simulacije pa shranjuje v izhodno slikovno datoteko.

Slikovne datoteke so v obliki toka podatkov, ki gredo v vezje za obdelavo slik ali iz njega in vsebujejo poleg slikovnih točk tudi kontrolne informacije. Te datoteke uporabljamo tudi pri strojni verifikaciji vezij na posameznih modulih. Pregledovanje slik in pretvorbo v standardne slikovne formate delamo s programom, ki je del našega testnega okolja.

Naslednji postopki prevajanja so sinteza veza s programom Synopsis FPGA Express ter tehnološka preslikava in izdelava konfiguracijske datoteke, ki poteka v okolju Xilinx Foundation [11]. Postopki prevajanja so avtomatizirani in zahtevajo od uporabnika le nastavitev lokacij posameznih signalov na vezju FPGA in specifičnih uporabniških zahtev (npr. frekvenca delovanja prevedenega vezja). Če nastanejo napake pri sintezi vezja zaradi nepravilne uporabe sintakse jezika VHDL ali pa če pri tehnološki preslikavi ne moremo izpolniti uporabniških zahtev, moramo popraviti opis ali spremeniti zgradbo vezja.

Za strojno verifikacijo vezij na programirljivih modulih smo napisali programsko opremo na osebnem računalniku, s katero nastavljamo testne vektorje in opazujemo rezultate testiranja. Slika 4 prikazuje glavno okno programa za strojno verifikacijo. Program je namenjen za verifikacijo vezja na modulu, ki je priključen na osebni računalnik prek vmesnika. S programom najprej konfiguriramo vezje FPGA, nato pa poženemo postopek strojne verifikacije. V oknu programa lahko opazujemo časovni potek signalov, podobno kot pri simulaciji. Izvajanje strojne verifikacije je pri kompleksnih vezjih precej hitrejše kot izvajanje simulacije, kar nam omogoča testiranje z večjim številom testnih vektorjev.



Slika 4. Program za strojno verifikacijo vezij
Figure 4. Hardware verification software

Preskus delovanja vezij za obdelavo slik najbolje naredimo s testnimi slikami. Pri tem uporabimo enake testne vektorje kot pri simulaciji vezja in programsko opremo, ki izvaja komunikacijske cikle med modulom in vmesnikom. V tem primeru nas ne zanima časovni potek vseh signalov, temveč le končni rezultat, ki ga opazujemo v obliki izhodnih slik.

Po opravljeni strojni verifikaciji vezij na posameznih modulih, jih priključimo na matično ploščo in sestavimo verigo za obdelavo slik. Na začetku verige opravljamo vzorčenje videosignala, ki se izvaja na prvem modulu. Na naslednjih modulih so posamezni deli vezja, ki izvajajo algoritem obdelave slik. Zadnji modul pa izvaja pretvorbo toka podatkov v obliko, ki je primerna za prikaz

na računalniškem monitorju.

Tudi pri preskušanju vezja z videosignalom imamo možnost opazovati signale v posameznih modulih. Signale lahko opazujemo preko testne verige (JTAG), prek katere imamo dostop do vseh vezij FPGA. V vezje, kjer želimo opazovati ali nastavljati posamezen signal, med načrtovanjem vgradimo analizatorski modul, ki vzorči signale in jih prek testne verige pošilja v osebni računalnik [12].

Kombinacija vseh prikazanih načinov strojne verifikacije omogoča učinkovito odkrivanje napak in testiranje vezij za obdelavo slik.

4 Vezja za obdelavo slik

Predstavili bomo nekaj vezij za obdelavo slik v realnem času, ki smo jih naredili in preskusili na modularnem prototipnem sistemu. Specializirana vezja za obdelavo slik so najučinkovitejša v začetnih fazah obdelave, ko imamo ponavljajoče se operacije nad velikim številom podatkov. Vezje na posameznem modulu lahko izvaja operacije nad lokalno množico točk ali pa nad celo sliko, ki je shranjena v lokalnem pomnilniku.

4.1 Filtriranje slike

Operatorji za filtriranje slike spadajo med lokalne operatorje, ki jih zelo učinkovito naredimo z vezji v cevovodni arhitekturi. Z nizkopasovnim sitom odstra- nimo zrnati šum iz digitalne slike, z visokopasovnim sitom pa poudarimo robove objektov na sliki.

Filtriranje slike običajno izvajamo z dvodimenzionalno konvolucijo. Vezje za dvodimenzionalno konvolucijo je sestavljeno iz vrstičnih pomnilnikov FIFO, s katerimi dobimo na vhodu vezja lokalno množico točk, in operatorjev za množenje s koeficienti ter seštevanje delnih rezultatov.

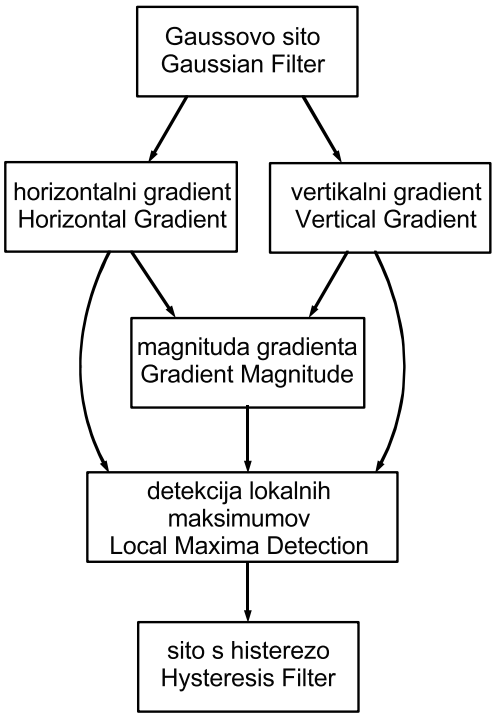
Kompleksnost vezja je odvisna od velikosti konvolucijske matrike in koeficientov. Pri preprostem povprečevalnem situ so vsi koeficienti enaki, pri boljšem, Gaussovem situ, pa množimo s koeficienti iz Gaussove matrike.

4.2 Detekcija robov

Tipičen algoritem za detekcijo robov je Cannyjev algoritem [13], ki temelji na računanju gradientov na digitalnih slikah in detekciji lokalnih maksimumov gradienta. Slika 5 prikazuje osnovne korake detekcije robov s Cannyjevim algoritmom.

V prvem koraku uporabimo Gaussovo sito za izločanje šuma na sliki. Naslednji korak je izračun horizontalnega in vertikalnega gradienta ter gradientne magnitude. Iz teh podatkov izračunamo lokalne maksimume gradienta.

Vse točke, ki niso lokalni maksimumi, postavimo na vrednost 0. V zadnjem koraku uporabimo pragovno sito za določanje robnih točk in točk ozadja. Najboljši rezultat dobimo s pragovnim sitom s histerezo, ki ima dva praga: nižjega in višjega. Točke z magnitudo, višjo od zgornjega praga, postanejo robne točke. Točke, ki imajo magnitudo nižjo od spodnjega praga, so točke ozadja. Točke z vmesno magnitudo pa spremenimo v robne točke, če se dotikajo kakšne točke, ki je že robna točka.



Slika 5. Cannyjev algoritem za detekcijo robov
Figure 5. Canny edge detection algorithm

4.3 Rotacija slike

Rotacijo slike dosežemo s transformacijo koordinat posameznih točk. Ker prihajajo v vezje za obdelavo slik zaporedne točke slike, moramo najprej v lokalnem pomnilniku sestaviti celotno sliko, narediti preslikavo koordinat in iz vezja poslati točke v ustreznem zaporedju. Rotacija slike je globalna operacija, ker potrebujemo za izračun izhodnih točk celotno sliko. Vezje deluje tako, da za vsako izhodno točko izračuna z inverzno preslikavo koordinate točke na vhodni sliki. Preslikava je narejena z množenjem koordinat s transformacijsko matriko, ki jo izračunamo za vsak kot posebej.

5 Rezultati in razprava

Tabela 1 prikazuje rezultate tehnološke preslikave vezij za obdelavo slik v vezja XCS40-4, ki jih imamo na modulih druge generacije. Uporabljena vezja vsebujejo 784

konfiguracijskih logičnih blokov (CLB).

Glede na zasedenost vezja FPGA za posamezne operacije bi lahko naredili celoten Cannyjev algoritem v enem vezju, vendar smo omejeni s količino notranjega pomnilnika v tej družini vezij. Pri posameznih operacijah potrebujemo pomnilnik za eno ali več vrstic slike, ki ga naredimo s počasnejšim in manj fleksibilnim zunanjim pomnilnikom na modulih. V prihodnje bomo za izvedbo algoritmov uporabili novejša vezja FPGA, ki že imajo dovolj notranjega pomnilnika za več vrstic slike. Pri globalnih operatorjih, kot je rotiranje slike, bomo morali še vedno uporabiti zunanji pomnilnik.

vezje **Circuit**	**število CLB** **CLB Count**	**frekvenca (MHz)** **Frequency (MHz)**
Gaussovo sito Gaussian filter	38	39
gradient Gradient	209	9.0
lokalni maks. Local max.	40	11
sito s histerezo Hysteresis filter	186	20
rotacija Rotation	279	21

Tabela 1. Rezultati tehnološke preslikave
Table 1. Rezults of technology mapping

Vezja so narejena za obdelavo 8-bitnih sivinskih slik. Zaporedne točke slike prihajajo v vezje s frekvenco 5MHz pri ločljivosti 256×256 oz. 10MHz pri ločljivosti slike 512×512 točk. Kot vidimo iz tabele, frekvenca delovanja večine vezij ustreza frekvenci za obdelavo slik pri najvišji ločljivosti.

6 Sklep

V članku smo prikazali modularni prototipni sistem, ki omogoča izdelavo in testiranje vezij za obdelavo slik v realnem času. Sistem je zgrajen iz modulov s programirljivimi vezji različnih generacij in ga lahko preprosto širimo z dodajanjem novejših družin vezij FPGA. Sistem je podprt s programsko opremo za načrtovanje in strojno verifikacijo implementiranih vezij. Prikazali smo tudi nekaj rezultatov pri izvedbi vezij za obdelavo slik na našem sistemu.

V nadaljevanju dela bomo na sistemu preskušali nove algoritme in arhitekture vezij za obdelavo slik v realnem času.

7 Literatura

[1] A. DeHon, "The Density Advantage of Configurable Computing", *IEEE Computer*, 33(4), str. 41-49, april 2000.

[2] C. Collins, "The New Virtex FPGA Family", *Xcell*, 31, str. 4-5, 1999.

[3] B. Radunovič, V. Milutinovič, "A Survey of Reconfigurable Architectures", *Field Programmable Logic and Applications FPL'98*, Tallin, 1998.

[4] P. M. Athanas, A. L. Abbott, "Real-Time Image Processing on a Custom Computing Platform", *IEEE Computer*, pp. 16-24, 1995.

[5] S. D. Haynes, P. Y. K. Cheung, W. Luk, J. Stone, "SONIC — A Plug-In Architecture for Video Processing", *Proc. of the 9th Workshop on Field-Programmable Logic and Applications*, str. 21-30, 1999.

[6] A. Trost, *"Rekonfiguracijske arhitekture vezij za obdelavo slik v realnem času"*, doktorska naloga, Fakulteta za elektrotehniko, Ljubljana, 2000.

[7] M. Gokhale, et al, "Building and Using a Highly Parallel Programmable Logic Array", IEEE Computer, str. 81-89, januar 1991.

[8] S. Hauck, G. Borriello, C. Ebeling, "Springbok: A Rapid- Prototyping System for Board-Level Designs", *ACM/SIGDA 2nd International Workshop on Field-Programmable Gate Arrays*, februar 1994.

[9] A. Lawrence, A. Kay, W. Luk, T. Nomura, I. Page, "User Reconfigurable Hardware to Speed up Product Development and Performance", *Proc. of the 5th Workshop on Field-Programmable Logic and Applications*, str. 111-118, 1995.

[10] K. Sobolewski, M. Ossysek, *"Active-VHDL Series Getting Started Guide"*, Aldec Inc., 1998.

[11] *"Foundation Series Quick Start Guide 1.5"*, Xilinx Inc., 1998.

[12] A. Trost, B. Zajc, "Strojna verifikacija vezij s sistemom za obrobno testiranje", *Zbornik devete Elektrotehniške in računalniške konference ERK 2000*, str. 45-48, Portorož, september 2000.

[13] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), str. 679-698, november 1986.

Andrej Trost je diplomiral leta 1995, magistriral leta 1998 in doktoriral leta 2000 na Fakulteti za elektrotehniko Univerze v Ljubljani. Izvoljen je bil v naziv docent in izvaja vaje na Katedri za elektroniko. Njegova raziskovalna dejavnost je povezana s programirljivimi vezji, prototipnimi sistemi in strojno verifikacijo vezij ter izdelavo vezij za obdelavo slik v realnem času.

Andrej Žemva je diplomiral leta 1989, magistriral leta 1993 in doktoriral leta 1996 na Fakulteti za elektrotehniko Univerze v Ljubljani, kjer je zaposlen kot docent. Njegovo raziskovalno področje obsega algoritme za načrtovanje digitalnih sistemov v strojni in programski opremi.

Baldomir Zajc je diplomiral leta 1960, magistriral leta 1965 in doktoriral leta 1975 na Fakulteti za elektrotehniko v Ljubljani. Zaposlen je kot profesor na Fakulteti za elektrotehniko Univerze v Ljubljani. Njegova raziskovalna zanimanja vključujejo integrirana vezja in polprevodniške tehnologije.