

EKSPERIMENTALNO TESTNO OKOLJE ZA DRUŽINO STANDARDOV IEEE 1149.X

Robert Sedevčič, Uroš Kač, Franc Novak

Institut Jožef Stefan, Ljubljana, Slovenija

Ključne besede: robna testna linija, IEEE 1149.1 testno vodilo, krmilnik testnega vodila, Linux gonilnik, SVF prevajalnik

Izvleček: V članku je predstavljen projekt EBS (ang. Experimental Boundary-Scan), katerega osnovni namen je vzpostaviti preprost laboratorijski testni sistem za IEEE 1149.x združljiva vezja. EBS temelji na široko dostopnih elektronskih komponentah, ki podpirajo tehniko robne testne linije, ter prosto dostopnem operacijskem sistemu Linux. Projekt sestoji iz strojnega ter programskega dela. Razvit je bil računalniški vmesnik med sistemskim vodilom ISA ter IEEE 1149.1 testnim vodilom, ki temelji na standardnem krmilniku SN74ACT8990. Vmesnik je podprt z ustreznim gonilnikom za Linux. Realiziran je bil tudi splošen prevajalnik za jezik SVF (ang. Serial Vector Format), ki sodi med standardne formate opisa testnih postopkov na osnovi robne testne linije. Prevajalnik je preko vtičnih programskih modulov povezan z gonilnikom oz. strojnimi vmesniki ter skupaj s pripadajočima grafičnima vmesnikoma predstavlja funkcionalno zaključeno testno okolje. Izdelan testni sistem je bil uspešno uporabljen za izvedbo različnih laboratorijskih eksperimentov z IEEE 1149.4 združljivimi vezji.

Experimental test environment for IEEE 1149.x standards

Key words: boundary-scan, IEEE 1149.1 test bus, test bus controller, Linux device driver, SVF compiler

Abstract: The development of complex, multi-layer PCBs, associated with the miniaturisation of electronic device packages and new assembly methods (BGA, SMT, COB,...), made physical access, required by the traditional in-circuit test methods, increasingly difficult. The IEEE 1149.1 standard test access port and boundary-scan architecture presented a solution to the limited access circuit testing problem and since its adoption in 1990 became an important design-for-testability (DFT) technique in the electronic industry.

In order to efficiently use the boundary-scan (BS) infrastructure for testing and other purposes, adequate tester hardware as well as test development software tools are required. These can greatly facilitate the generation and application of BS based test or in-system programming (ISP) procedures in most modern complex devices. Although there is a number of adequate BS test solutions available on the market, these primarily target industrial applications where robust operation in medium or high-volume production testing is required. On the other hand, purchase of expensive professional BS test equipment can represent a major obstacle for academic institutions involved in research or educational activities regarding IEEE 1149.1 and related standards. Furthermore, most commercially available systems are relatively complex and do not provide the transparency, which is required for a thorough understanding of the boundary-scan test technique. Only a fully open and custom configurable platform could provide the necessary freedom to users with specific requirements as well as an efficient educational tool for teaching the basic principles of BS testing. The absence of similar solutions motivated the implementation of our Experimental Boundary-Scan (EBS) platform, which is intended as a suite of BS test tools, based on the GNU/Linux operating system.

The EBS environment was conceived as a simple laboratory test system for IEEE 1149.x compliant circuits. The project is based on widely available hardware supporting boundary-scan test techniques and on the open source Linux operating system. The project is divided into hardware and software related parts. A simple ISA-bus PC adapter featuring the SN74ACT8990 test bus controller was implemented along with the appropriate Linux device driver. The SN74ACT8990 performs transformation of the test data supplied by the host processor and generates adequate data and control IEEE 1149.1 test bus signals (TDO, TDI, TMS, TCK). The SN74ACT8990 data, control and status registers are mapped into the host processor I/O space through the 16-bit ISA bus interface. On-board glue logic provides a configurable I/O base address and IRQ level as well as a hardwired test clock frequency divider. Although no additional processing or storage capability is available on-board, the implemented test bus adapter can represent a sufficient and cost effective solution comparable to many commercially available adapters.

A generic, plug-in based compiler for the widely supported Serial Vector Format (SVF) test description language and basic graphical user interfaces were also developed. SVF is an ASCII format used for describing test patterns that represent stimulus, expected response and mask data according to the IEEE 1149.1 standard. Along with the implemented hardware platform these tools form a functional boundary-scan test environment. Since the software tools are conceived as independent modules featuring well-defined application interfaces, support for alternative BS hardware as well as additional software utilities can be easily included into the system. So far the existing test system was successfully applied in a number of laboratory experiments concerning prototype IEEE 1149.4 mixed-signal test bus compliant devices. The complete EBS project is freely distributable and has been made available through the Sourceforge software development network service.

1 Uvod

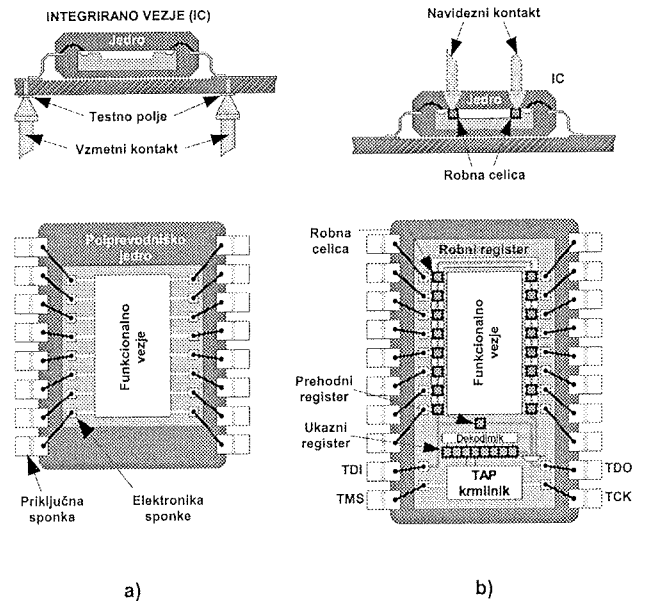
Z uvajanjem novih tehnologij izdelave se povečuje gostota integriranih vezij, večja se število priključnih sponk posameznih komponent ter manjšajo razdalje med njimi. Posledično se zmanjšujejo tudi razdalje med bakrenimi povezavami, kar skupaj z večanjem števila plasti v tiskanih vezjih in novimi načini pritrjevanja komponent (BGA, SMT,

COB, ...) izredno otežuje fizični dostop do posameznih komponent tiskanega vezja /1/ in s tem izvedbo klasičnih testnih postopkov z uporabo vmesnikov z vzmetnimi kontakti (slika 1a). Kot rešitev problema omejenega dostopa so proizvajalci komponent v osemdesetih letih razvili postopek t.i. robne testne linije (ang. boundary-scan). Ta nadomešča fizični dostop do sponk posameznih komponent vezja z dostopom preko celic pomikalnega registra,

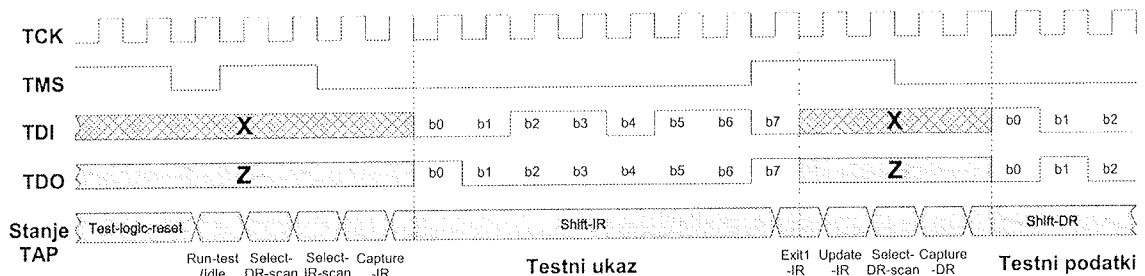
nameščenega okoli jedra integriranega vezja (slika 1b). Leta 1990 je bil postopek robne testne linije na pobudo združenja Joint Test Action Group (JTAG) sprejet kot mednarodni standard IEEE 1149.1 /2/.

Uporaba robne testne linije zahteva ustrezno strojno ter programsko opremo /3/. IEEE 1149.1 združljiva integrirana vezja uporabljajo specifičen serijski protokol (slika 2a), ki omogoča prenos testnih ukazov in podatkov preko skupnega vodila in je določen z avtomatom končnih stanj v krmilniku testnega vmesnika (TAP) komponente (slika 2b). Arhitektura tipičnih testnih platform zahteva pretvorbo testnih podatkov in ukazov v ustrezno bitno sekvenco na IEEE 1149.1 testnem vodilu, za kar lahko uporabimo poseben krmilnik testnega vodila (slika 2c).

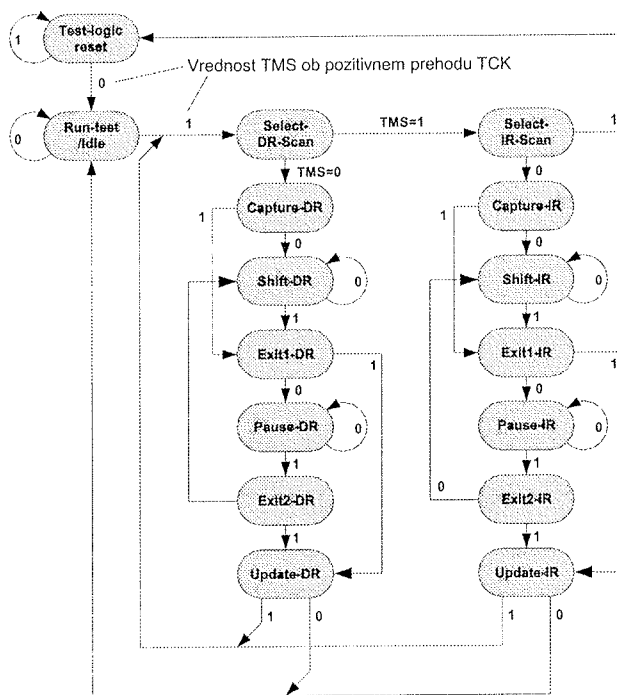
Namenska orodja nam lahko močno poenostavijo izvedbo ter skrajšajo čas, ki je potreben za realizacijo testnega postopka /4-7/. Kljub temu, da lahko na trgu srečamo več strojnih ter programskih rešitev različnih proizvajalcev, so tovrstna orodja v prvi vrsti namenjena industrijskim aplikac-



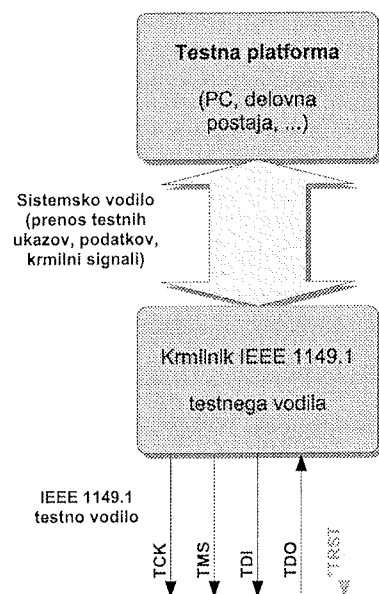
Slika 1: Osnovni koncept robne testne linije



a)



b)



c)

Slika 2: Prikaz protokola IEEE 1149.1 testnega vodila in tipične povezave s testno napravo

ijam, ki zahtevajo predvsem preprosto uporabo in robustno delovanje. Temu je primeren tudi cenovni razred teh sistemov, ki lahko za akademske in raziskovalne ustanove predstavlja precejšnjo oviro pri vzpostavitvi laboratorijskega okolja, potrebnega za preučevanje, raziskave in razvoj novih komponent in ustreznih testnih postopkov na osnovi 1149.1 infrastrukture /8-11/.

Večina komercialnih testnih sistemov je relativno kompleksnih, hkrati pa uporabniku ne nudijo potrebne transparentnosti, ki je nujna za razumevanje tehnike robne testne linije. Zahtevnejši uporabnik za realizacijo specifičnih rešitev pogosto potrebuje popolnoma odprto, prosto dostopno in nadgradljivo platformo. Pomanjkanje tovrstnih rešitev je bil eden od glavnih razlogov za nastanek projekta EBS, ki združuje vsa osnovna orodja, potrebna za izvedbo testnega postopka v skladu s standardom IEEE 1149.1.

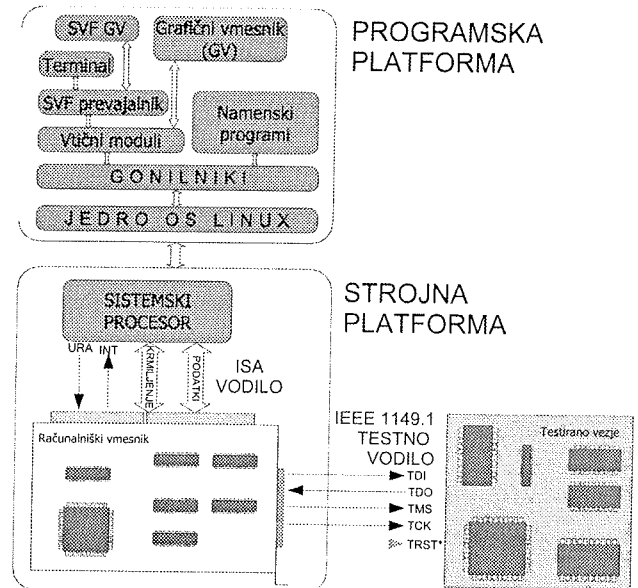
Projekt EBS nudi popolnoma odprto ter prosto razširljivo platformo, zasnovano na operacijskem sistemu (OS) GNU/Linux /12/. V nadaljevanju je predstavljen razvoj strojne in programske opreme, ki tvori preprost testni sistem. Strojni del temelji na ISA razširitveni kartici za osebni računalnik, ki omogoča nadzor IEEE 1149.1 testnega vodila. Vmesnik je ustrezno podprt z gonilnikom za OS Linux ter višje nivojskimi namenskimi programi. Programski del je zasnovan kot skupek neodvisnih modulov z dobro definiranimi medsebojnimi povezavami. To omogoča enostavno nadgradnjo ter razvoj dodatnih testnih orodij, neodvisno od nižje nivojske strojne opreme. Delo na projektu EBS še ni zaključeno, saj je v načrtu razvoj izboljšane vmesnika kot tudi nekaterih dodatnih programskih orodij. Celoten projekt je dostopen preko internetne spletne strani /13/ kot del zbirke prostega programja Sourceforge /14/.

2 Struktura projekta EBS

Projekt EBS je sestavljen iz strojnega ter programskega dela. Kot osnovna platforma je privzeta PC arhitektura i386 z OS Linux. V programskem delu želimo realizirati univerzalno platformo, ki bi podpirala krmilnike testnega vodila različnih proizvajalcev. Ti krmilniki /15/ se medsebojno razlikujejo v hitrosti delovanja (od 20 do 65 MHz), širini podatkovne besede (8/16 bitni), velikosti medpomnilnika (od 16b do 8Kb) itd. Njihova osnovna lastnost je, da bistveno poenostavijo povezavo med klasičnim vzporednim vodilom in enim ali več neodvisnimi IEEE 1149.1 testnimi vodili. Za učinkovitejši razvoj in izvedbo testnega postopka je potrebno obstoječi strojni opremi zagotoviti ustrežno programsko podporo. Načrtovano programsko platformo projekta EBS smo razdelili na več medsebojno neodvisnih modulov: gonilniki, vtični moduli, SVF prevajalnik ter grafični vmesniki. Struktura projekta EBS je shematično prikazana na sliki 3.

2.1 Strojna platforma

V okviru projekta smo implementirali preprost računalniški (PC) vmesnik za ISA vodilo, ki temelji na krmilniku testnega

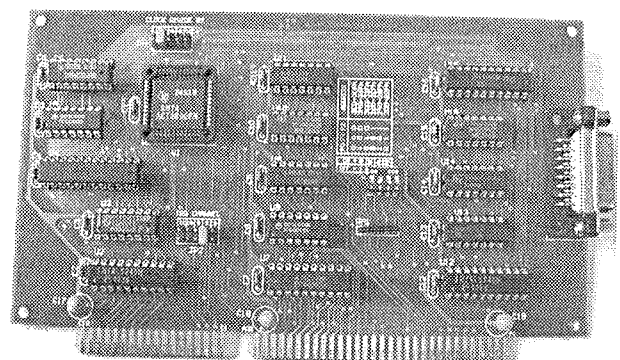


Slika 3: Strojna ter programska struktura projekta EBS

vodila SN74ACT8990 /16/ proizvajalca Texas Instruments. Njegova naloga je generiranje ustreznih signalov, ki osebnemu računalniku omogočajo dostop do testne infrastrukture v IEEE 1149.1 združljivih vezjih preko testnega vodila. Standard določa, da sestavljajo vodilo štiri obvezne linije (TMS, TCK, TDO, TDI) ter ena neobvezna linija (TRST*). Krmilni liniji TCK in TMS skrbita za pravilen prenos podatkov preko linij TDI in TDO. SN74ACT8990 omogoča vzporedno krmiljenje največ šestih IEEE 1149.1 testnih vodil, ki uporabljajo ločene TMS linije, medsebojno pa si delijo TDI, TDO in TCK linije.

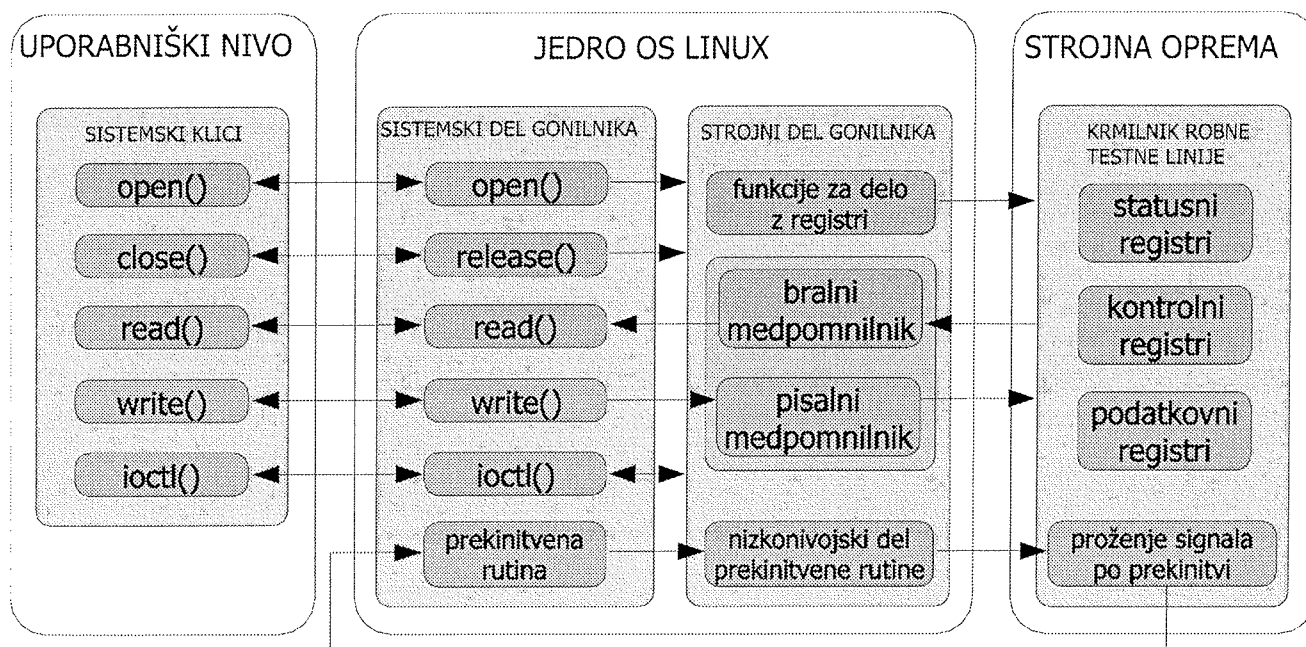
PC vmesnik, ki je prikazan na sliki 4, sestoji iz naslednjih sklopov:

- logike 16-bitnega ISA vodila;
- delilnika testne ure TCK;
- krmilnika testnega vodila SN74ACT8990;
- logike testnega vodila.



Slika 4: Računalniški vmesnik med vodilom ISA ter testnim vodilom

Podatkovni, kontrolni ter statusni registri krmilnika SN74ACT8990 so preko vodila ISA prezrcaljeni v vhod-



Slika 5: Shematičen prikaz zgradbe gonilnika za OS Linux

no/izhodni prostor sistemskega procesorja. Na razširitveni kartici je mogoče preko stikal izbrati osnovni V/I naslov ter ustrezen IRQ kanal. SN74ACT8990 deluje do frekvence 30 MHz, vendar pa obstoječa izvedba uporablja sistemsko uro vodila ISA (8,33 MHz). Frekvenco urinega takta lahko dodatno znižamo za 2x, 4x, 8x ali 16x z ustrežno nastavitvijo delilnika testne ure. Tri nivojski gonilniki, ki povezujejo krmilnik robne testne linije s testnim vodilom, so namenjeni zaščiti krmilnika ter izboljšanju kvalitete signalov na testnem vodilu. Kljub temu, da razširitvena kartica ne omogoča nobene nadaljnje obdelave oz. shranjevanja testnih podatkov, predstavlja uporabno in predvsem ceneno rešitev v primerjavi s komercialno dostopnimi razširitvenimi karticami. V prihodnosti načrtujemo realizacijo nekoliko zmogljivejšega vmesnika za vodilo PCI.

2.2 Programska struktura

Operacijski sistem Linux ima zaradi svoje odprtosti in zanesljivosti širok krog uporabnikov v raziskovalnih ter akademskih ustanovah. V postopku razvoja programske opreme, se je izkazal kot zelo solidno programsko okolje, ki uporabniku omogoča dober nadzor tudi nad najnižjimi programskimi nivoji. Programska platforma projekta EBS je razdeljena na dva osnovna modula (slika 3) in sicer:

- gonilnik krmilnika za OS Linux;
- okolje za pripravo testnih postopkov.

2.2.1 Gonilnik krmilnika za OS Linux

Gonilniki so programski moduli, ki so povezani v jedro OS z namenom, da ločeno nadzorujejo delovanje pripadajoče strojne opreme ter zakrijejo podrobnosti le-te pred uporabnikom /17/. Tako namenski programi iz uporabniškega nivoja dostopajo do jedra sistema in s tem do različnih funkcij strojne opreme s pomočjo standardnih sistemskih

klicev. Jedro OS obravnava klice teh funkcij kot posebne zahteve in jih prenaša na ustrezne funkcije-metode znotraj programske kode gonilnikov /18/. Realizirani gonilnik krmilnika SN74ACT8990 podpira pet sistemskih klicev: `open()`, `close()`, `read()`, `write()` ter `ioctl()` /19/. Uporabnik pri pisanju testnih programov posredno preko sistemskih klicev oz. metod upravlja s krmilnikom testnega vodila. Gonilnik je zasnovan tako, da podpira nastavitve vseh možnih konfiguracij krmilnika ter tako omogoča izrabo vseh njegovih zmogljivosti. SN74ACT8990 lahko deluje v prekinitvenem načinu, zato gonilnik vključuje ustrezno prekinitveno funkcijo, ki skrbi za praznjenje pisalnega ter polnjenje bralnega medpomnilnika (slika 5). Prednost takšnega delovanja je, da uporabniku na višjih nivojih programiranja ni potrebno skrbeti za sprotno prenašanje testnih podatkov med sistemskim pomnilnikom in bralnim oz. pisalnim registrom krmilnika.

Programska koda gonilnika je razdeljena na strojni ter sistemski del. Prvi dostopa neposredno do registrov krmilnika ter tako skrbi za komunikacijo med krmilnikom ter gonilnikom (programiranje na nivoju registrov), drugi del pa tvori povezavo med jedrom OS Linux ter gonilnikom (slika 5). Takšna zasnova nam omogoča, da lahko v primeru spremembe strojne opreme, uporabimo isti gonilnik, saj je potrebno prilagoditi zgolj nižje nivojski del programske kode.

2.2.2 Okolje za pripravo testnih postopkov

Testni postopek lahko izvedemo neposredno z uporabo sistemskih klicev, kar sicer ponuja popoln nadzor nad izvajanjem testne procedure, vendar pa je za testiranje kompleksnih vezij takšen postopek preveč zamuden. V ta namen smo razvili prevajalnik za jezik SVF (ang. Serial Vector Format). SVF je standardiziran jezik /20/ za opis testnih

postopkov na osnovi robne testne linije. Struktura SVF je dokaj preprosta, saj ne vključuje nobenih odločitvenih in zančnih struktur. Definira štirinajst različnih stavkov, katerih parametri so v šestnajstiškem številskem sistemu zapisane logične vrednosti vhodnih in izhodnih vektorjev ter bitnih mask. Odzive vezja na vhodne vektorje je mogoče preko izhodne maske primerjati s pričakovanimi vrednostmi. Slika 6 podaja primer testnega postopka v jeziku SVF, ki v testirano vezje najprej prenese 8 bitov dolg ukaz (41) ter nato še 32 bitov dolg vhodni testni vektor (ABCD1234). Pričakovan odziv na vhodni vektor (11112222) primerjamo z dejanskim odzivom na vsakem bitu (maska FFFFFFFF).

```

.
.
.
SIR 8 TDI (41);
SDR 32 TDI (ABCD1234) TDO (11112222) MASK (FFFFFFF);
.
.
.
    
```

Slika 6: Zapis v jeziku SVF

Prevajalnik je bil implementiran s pomočjo programskih orodij Flex ter Bison, postopek prevajanja pa združuje tri vrste analiz /21/:

- leksikalno analizo – iz vhodnih simbolov tvori nosilce pomena;
- sintaktično analizo – preveri strukturo stavkov;
- semantično analizo – določi pomen posameznim stavkom.

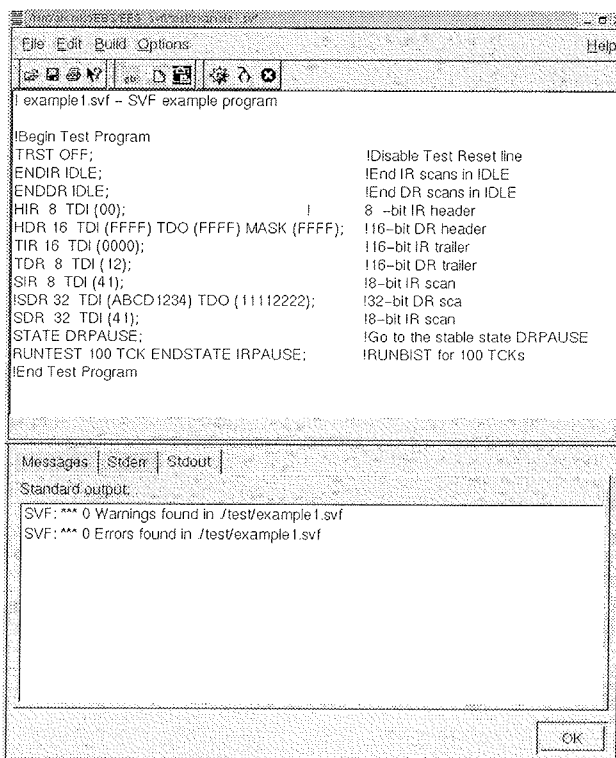
Prevajalnik deluje v t.i. interpretativnem načinu, kar pomeni da mora vsak stavek jezika SVF najprej uspešno prestati vse tri faze prevajanja, šele nato lahko temu sledi ustrezna akcija na strojni platformi. Prevajalnik je neodvisen od strojne platforme, saj so vse funkcije, ki izvajajo operacije nad specifičnim krmilnikom, vključene v posebno dinamično povezljivo knjižnico /22, 23/. Splošen SVF prevajalnik lahko tako uporabimo za različne izvedbe krmilnikov testnega vodila, pri čemer je potrebno za vsakega zagotoviti le ustrezen vtični programski modul (ang. plug-in).

Predstavljen prevajalnik je sprva deloval zgolj v terminalskem načinu, kasneje pa je bil razširjen z ustreznim grafičnim vmesnikom (GV). Ta omogoča enostavnejše upravljanje tistim uporabnikom, ki jim je grafično okolje bolj domače. Slika 7 prikazuje realizirani grafični vmesnik.

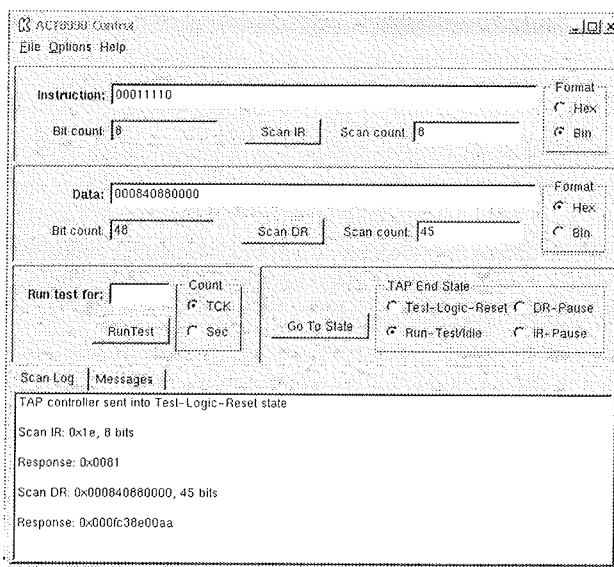
Poleg tega je bil razvit tudi preprost grafični vmesnik, s pomočjo katerega lahko uporabnik izvaja osnovne testne procedure združljive s standardom IEEE 1149.1, kot so pošiljanje posameznih ukaznih ter podatkovnih nizov ter upravljanje s testno infrastrukturo naprave. Ta grafični vmesnik (slika 8) dostopa do naprave neposredno preko sistemskih klicev.

3 Uporaba orodij ter razvoj testa

Za boljše razumevanje opisanih orodij bomo celoten testni postopek predstavili na preprostem zgledu. Vezje na sliki



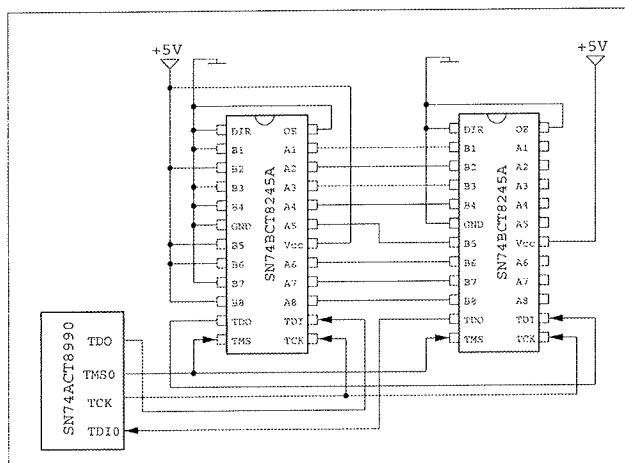
Slika 7: Grafični vmesnik za izvedbo testa napisanega s SVF jezikom



Slika 8: Preprost grafični vmesnik za izvajanje osnovnih IEEE 1149.1 procedur

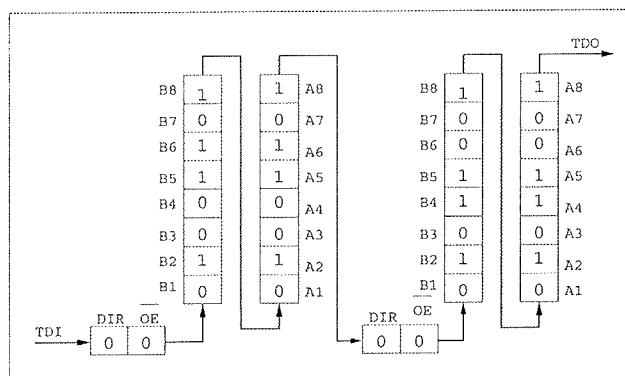
9 je sestavljeno iz dveh integriranih vezij SN74BCT8245A /24/, ki sta združljivi s standardom IEEE 1149.1. S pomočjo robne testne linije želimo odkriti morebitne napake na povezavah med priključki A1-A8 prve ter priključki B1-B8 druge komponente.

Povezave med posameznimi komponentami tiskanega vezja testiramo s pomočjo IEEE 1149.1 ukaza EXTEST. Komponente z vgrajeno robno testno linijo imajo na vseh digital-



Slika 9: Zgled preprostega vezja z vgrajeno robno testno linijo

nih vhodih in izhodih vgrajene posebne spominske celice, ki so povezane v robni pomikalni register /25/. Na vezje s slike 9 lahko gledamo kot na dve medsebojno povezani verigi spominskih celic, ki tvorita enoten pomikalni register. Testni postopek je sestavljen iz niza ukazov in testnih vektorjev. V 8-bitna ukazna registra obeh vezij je potrebno najprej naložiti kodo ukaza EXTEST, ki je v primeru SN74BCT8245A enaka vrednosti 00h (v šestnajstiškem zapisu). Vsak robni pomikalni register ima 18 celic, torej je testna linija vezja na sliki 8 dolga 36 bitov. V robna pomikalna registra vpišemo ustrezen testni vektor (001640000h), s katerim vzbujamo izhodne sponke prve komponente (A1-A8). Logično stanje posameznih spominskih celic po vpisu prvega vektorja je prikazano na sliki 10.



Slika 10: Logične vrednosti v pomikalnih registrih po vpisu prvega testnega vektorja - vzbujanja

```

!Begin Test Program
ENDR IDLE;                               !End IR scans in IDLE
ENDR IDLE;                               !End DR scans in IDLE
SIR 16 TDI (0000) TDC (8181) MASK (FFFF); !16-bit instruction scan
SDR 36 TDI (001640000);                  !36-bit data scan, first vector
SDR 36 TDI (000000000) TDO (000005959) MASK (00000FFFF); !36-bit data
!scan, second vector and response to first vector
!End Test Program

```

Slika 11: SVF zapis testnega postopka

Z vpisom naslednjega vektorja zajamemo in prenesemo odziv na vhodnih sponkah druge komponente (B1-B8) iz

celic pomikalnega registra ter ga primerjamo s pričakovano vrednostjo (000005959h). Slika 11 podaja SVF zapis opisane testne procedure.

4 Zaključek

Načrtovalci tiskanih vezij se pogosto ne zavedajo dovolj potrebe po vnaprejšnjem načrtovanju zmožnosti testiranja vezja, ki pa ob čedalje večji kompleksnosti postaja ena izmed osnovnih zahtev za uspešno realizacijo končnega proizvoda. Uporaba tehnike robne testne linije lahko občutno poenostavi odkrivanje morebitnih proizvodnih napak kot tudi vzdrževanje sistema med njegovo celotno življenjsko dobo.

Osnovni namen projekta EBS je realizacija preprostega a uporabnega orodja za izvajanje testnih postopkov v skladu s standardom IEEE 1149.1. Pomembni lastnosti predstavljenega sistema sta njegova odprtost in fleksibilnost, ki omogočata poseganje v vse nivoje delovanja. Namenjen je predvsem raziskovalnim ter akademskim ustanovam, saj omogoča enostavno nadgradnjo ter nadaljnji razvoj v skladu s specifičnimi potrebami uporabnika, obenem pa upamo, da bo naš projekt pripomogel tudi k širši uveljavitvi predstavljenih tehnik testiranja med načrtovalci in razvojnimi inženirji v domači industriji.

Literatura:

- /1/ K.P. Parker, *The Boundary-Scan Handbook*, 2nd edition, Kluwer Academic Publishers, Dordrecht, 1998, pp. 1-7.
- /2/ IEEE, Standard test access port and boundary-scan architecture, IEEE Standard 1149.1a-1993 (1993).
- /3/ R. Nelson, "Boundary-scan software aids PCB evaluation", *Test & Measurement World*, October (1999).
- /4/ R. Raina, R. Bailey, D. Belete, V. Khosa, R. Molyneaux, J. Prado, A. Razdan, "DFT advances in Motorola's next-generation 74xx PowerPC microprocessor", *Proceedings on ITC 2000*, pp. 131-140.
- /5/ F. Golshan, "Test and on-line debug capabilities of IEEE Std 1149.1 in UltraSPARC-III microprocessor", *Proceedings on ITC 2000*, pp. 141-150.
- /6/ B. Kerridge, "Flash stretches boundary-scan limits", *Test & Measurement Europe*, June (2002).
- /7/ R. Nelson, "Systems expand IEEE 1149.1 test", *Test and Measurement World*, February (2000).
- /8/ K. Lofstrom, "Early capture for boundary scan timing measurements", *Proceedings on ITC 1996*, pp. 417-422.
- /9/ M. Santo-Zarnik, F. Novak, U. Kač, S. Maček, "Experiments with IEEE 1149.4 KLIC test chip: a case study", *Proceedings on IMSTW 1999*, pp. 131-135.
- /10/ U. Kač, F. Novak, S. Maček, M. Santo-Zarnik, "Alternative test methods using IEEE 1149.4", *Proceedings on DATE 2000*, pp. 463-467.
- /11/ U. Kač, F. Novak, Florence Azad's, Pascal Nouet, Michel Renovell, "Implementation of an experimental IEEE 1149.4 mixed-signal test chip", 1st IEEE International Workshop on Board Test, October 2002.
- /12/ Free Software Foundation, spletna stran <http://www.fsf.org>.
- /13/ Spletna stran projekta EBS: <http://ebps.sourceforge.net>.
- /14/ Spletna stran projekta Sourceforge: <http://sourceforge.net>.

- /15/ U. Kač, R. Sedeščič, F. Novak, A. Biasizzo, "Linux-based experimental boundary scan environment", *Microprocessors and Microsystems*, Vol. 26, No. 5, 2002, pp. 199-206.
- /16/ Test-Bus Controller SN74ACT8990, Application Report, Texas Instruments, 2000, spletna stran <http://www.ti.com/sc/docs/jtag/jtaghome.html>.
- /17/ R. Sedeščič, Izvedba gonilnikov v OS Linux, *Monitor*, januar 2002.
- /18/ A. Rubini, J. Corbet, *Linux Device Drivers*, 2nd edition, O'Reilly & Associates, Inc., 2001.
- /19/ U. Kač, R. Sedeščič, F. Novak, A. Biasizzo, Eksperimentalno okolje za vezja z IEEE 1149.1 testno infrastrukturo, *Elektrotehniška konferenca*, Portorož, 2001.
- /20/ Serial Vector Format Specification, Revision E, Asset InterTech, 1999, spletna stran <http://www.asset-intertech.com/support/svf.pdf>.
- /21/ N. Wirth, *Compiler Construction*, Addison Wesley, 1996.
- /22/ R. Sedeščič, Programske knjižnice v Linuxu, *Monitor*, januar 2003.
- /23/ David A. Wheeler, Program Library HOWTO, spletna stran <http://www.dwheeler.com/program-library/>.
- /24/ Texas Instruments Inc., SN74BCT8245A Scan Test Device with Octal Bus Transceivers, data sheet, <http://www-s.ti.com/sc/p sheets/scbs043e/scbs043e.pdf>.
- /25/ U. Kač, R. Sedeščič, Testiranje vezij z vgrajeno IEEE 1149.1 testno infrastrukturo, *Svet elektronike*, letnik 9, januar 2002.

Robert Sedeščič
Institut "Jožef Stefan"
Jamova 39, 1001 Ljubljana, Slovenia
tel.: +386 (0)1 477 3550, fax: +386 (0)1 251 9385
email: robert.sedevcic@ijs.si

mag. Uroš Kač
Institut "Jožef Stefan"
Jamova 39, 1001 Ljubljana, Slovenia
tel.: +386 (0)1 477 3550, fax: +386 (0)1 251 9385
email: uros.kac@ijs.si

prof. dr. Franc Novak
Institut "Jožef Stefan"
Jamova 39, 1001 Ljubljana, Slovenia
tel.: +386 (0)1 477 3386, fax: +386 (0)1 251 9385
email: franc.novak@ijs.si

Prispelo (Arrived): 27.01.2003 Sprejeto (Accepted): 26.08.2003