

Programirljiva vezja

Andrej Trost

Katedra za elektroniko

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija

E-pošta: andrej.trost@fe.uni-lj.si

Programirljiva logična vezja predstavljajo pomembno skupino digitalnih integriranih vezij. Na začetku so nadomeščala standardne digitalne gradnike, omogočala izdelavo in testiranje prototipov namenskih vezij. Pri novih napravah nudijo prednost hitrejšega in cenejšega razvojnega cikla v primerjavi z namenski vezji, zmogljivejšo obdelavo podatkov v primerjavi s procesorji ter dobro razmerje med zmogljivostjo in porabo energije. Gradniki programirljivih vezij izvajajo naloge hkrati in so odlični za izvedbo časovno kritične obdelave podatkov. Možnost spreminjanja vezja, vgrajenega v napravi, pa nudi podobno prilagodljivost, kot jo imajo mikroprocesorski sistemi. Čeprav spadajo med dražja integrirana vezja, so programirljiva vezja nepogrešljiva v širokem spektru digitalnih naprav, posebej pa na področju razvoja novih naprav in prototipov vezij za izobraževalne namene.

Kaj so programirljiva vezja?

Programirljiva ali programabilna logična vezja so integrirana vezja, ki jih lahko po izdelavi spreminjamo tako, da izvajajo želeno funkcijo. Vezja z oznako PLD (Programmable Logic Devices) so se pojavila kot alternativa digitalnim integriranim vezjem zelo visoke integracije. Razvoj in proizvodnja namenskih integriranih vezij (ASIC – Application Specific Integrated Circuits) sta zelo zahteven in drag proces, ki je zato upravičen predvsem pri velikoserijskih izdelkih. Programirljiva vezja uporabljamo za izdelavo prototipov, manjših serij, v učne namene, preizkušanje vezij v strojni opremi in aplikacijah, kjer izkoriščamo možnost spreminjanja vezja.

Programiranje delovanja logičnega vezja izvajamo s spreminjanjem programske opreme ali oblikovanjem oz. konfiguracijo strojne opreme. Programsko opremo lahko spreminjamo v mikroprocesorskih sistemih, strojno pa v programirljivih vezjih. Novejša vezja združujejo obe možnosti za izdelavo zelo prilagodljivih digitalnih naprav in sistemov. Prednosti programirljivih integriranih vezij so:

- hiter razvoj prototipa v strojni opremi,
- občutno nižji stroški izdelave v primerjavi s proizvodnjo namenskih čipov,
- združevanje logike in manj komponent na tiskanih vezjih.

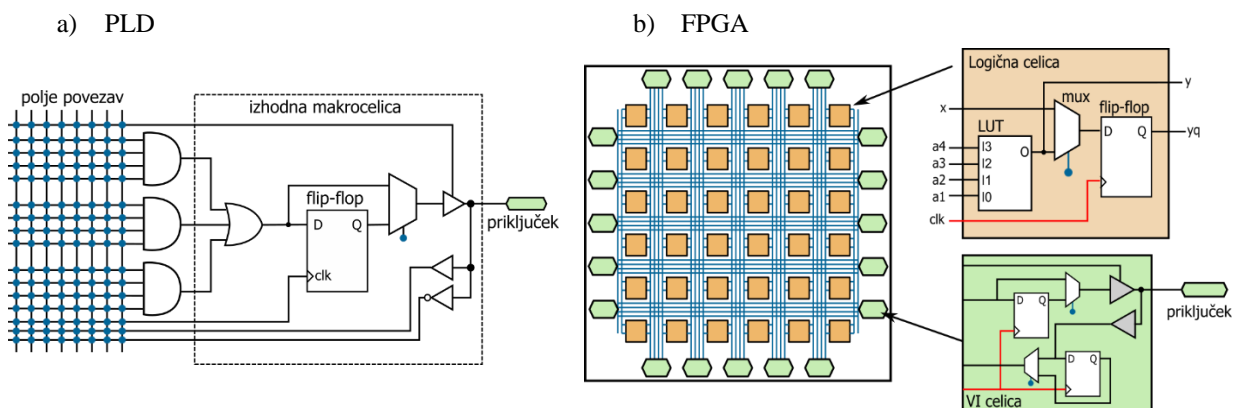
Logični gradniki, ki omogočajo oblikovanje že izdelanih programirljivih vezij, zasedejo velik del vezja in predstavljajo redundanco v končnem izdelku. Glavne slabosti teh vezij v primerjavi z namenski vezji so:

- višja cena posameznega integriranega vezja zaradi redundance v površini vezja,
- zakasnitve programirljivih elementov in povezav zaradi katerih so vezja v enaki proizvodni tehnologiji počasnejša od namenskih vezij in
- višja poraba energije.

Zgodovina programirljivih vezij

Zasnova programirljivih vezij temelji na pomnilnikih, katerih vsebino določamo med postopkom ali po postopku proizvodnje. Bralne pomnilnike (ROM, Read Only Memory) lahko izdelamo kot univerzalna integrirana vezja, ki jim določamo vsebino s spremembo mask med proizvodnjo. Programirljivi bralni pomnilniki (PROM, Programmable ROM) omogočajo spreminjanje oz. programiranje vsebine že izdelanega vezja. Ponovno programiranje dosežemo s tranzistorji, ki ohranjajo ujet električni naboj brez napajanja. Med prvimi so bili pomnilniki, ki smo jih brisali z ultravijolično svetlobo, pozneje pa električno (EEPROM). V izboljšani izvedbi s stališča hitrosti zapisovanja vsebine imajo takšni pomnilniki danes oznako Flash.

S pomnilnikom lahko posnemamo le delovanje kombinacijske logike, pa še to ne prav učinkovito. Prve učinkovitejše načine so raziskovalci našli s teorijo logičnih vezij, kjer izvedemo kombinacijske funkcije v obliki vsote produktivnih členov. Vsoto naredimo z logičnimi vrati **ali** (or), produktivni členi pa so večvhodna vrata **in** (and), na katere vezemo direktni ali negirani vhod.



Slika 1: Zgradba programirljivih vezij: a) matrika PLD in ena izhodna makrocelica, b) vezje FPGA ter poenostavljena shema logične in vhodno-izhodne (VI) celice. Celice povezuje programirljivo stikalno polje.

Programirljive matrice z oznakami PAL, PLA in GAL vsebujejo vnaprej izdelano matriko (npr. 16 vhodov in 32 produktivnih členov) in programirljive povezave na vhodih logičnih vrat. Z nastavljanjem povezav določamo njihovo kombinacijsko logično funkcijo. Za izvedbo sekvenčnih vezij dodamo flip-flope na izhode produktivnih členov. Programirljivo matriko ni mogoče učinkovito skalirati, ker narašča število elementov s kvadratom števila vhodnih signalov.

Kompleksne programirljive naprave (CPLD) vsebujejo več manjših programirljivih matrik, priključenih na stikalno povezovalno polje. Zunanje priključke vezja zagotavljajo izhodne makrocelice, ki povezujejo zunanje signale na stikalno polje in izhode produktivnih členov z opsijskim flip-flopom na zunanji priključek (slika 1a). Uporabnik vezja CPLD potrebuje programsko opremo za tehnološko preslikavo splošne logične sheme v zgradbo izbranega programirljivega vezja.

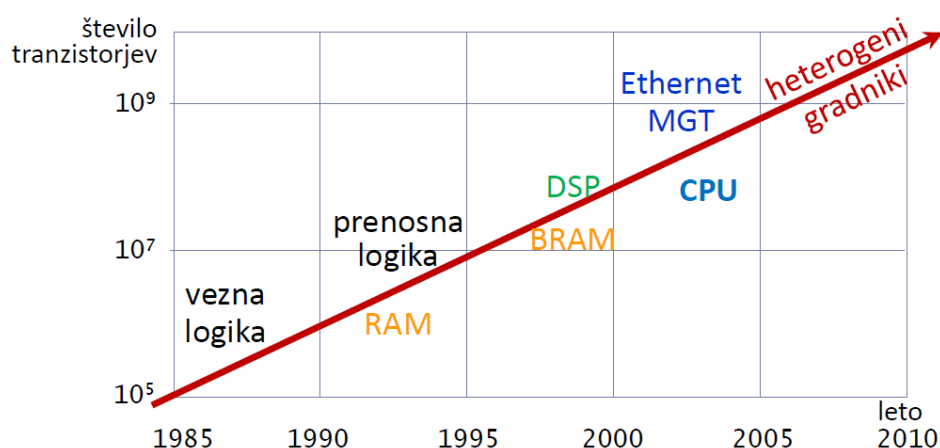
Drugačno zgradbo programirljivih vezij, osnovano na vpoglednih tabelah (LUT, look-up table), je predstavilo podjetje Xilinx leta 1984. Programirljivo polje logičnih vrat (FPGA, Field Programmable Gate Array) vsebuje v matriko razporejene logične celice, med katerimi so stikalne povezave, na robu pa so vhodno-izhodne celice z zunanjimi priključki (slika 1b). Logične celice vsebujejo eno ali več vpoglednih tabel, programirljive izbiralnike (mux) in flip-flope za izvedbo majhnega dela sekvenčnega vezja. Vpogledna tabela je narejena s statičnim pomnilnikom SRAM, ki mu vsebino določimo ob programiranju. Tudi stikalna matrika vsebuje statične pomnilne gradnike za povezovalna stikala, zato je celotno vezje lahko narejeno v tehnologiji CMOS. Prednost CMOS-izvedbe sta dostopnost in zmogljivost tehnologije, vezja lahko programiramo »na terenu« (field-programming), potrebujejo pa zunanji pomnilnik Flash, ker izgubijo vsebino ob izklopu napajanja. Vsebinsko vezja FPGA, ki je vgrajeno v neko napravo, je mogoče velikokrat spreminjati. Postopek opisuje izraz »rekonfiguracija« (ponovno oblikovanje) in s tem je povezana nova zvrst uporabe programirljivih vezij.

Programska orodja izvajajo sintezo iz strojno-opisnih jezikov v gradnike vezja FPGA in tehnološko preslikavo: dodelitev gradnikov, povezovanje ter izdelavo konfiguracijske datoteke. Povezava vsebine datoteke in zgradbe vezja v čipu FPGA je zaradi onemogočanja reverznega inženiringa skrita, zato v tem delu potrebujemo programska orodja proizvajalcev. Za učne namene in manjša vezja so dostopna brezplačno.

Razvoj tehnologije FPGA

Programirljivo matriko lahko dobro skaliramo, zato so vezja FPGA danes prevladujoča med zmogljivimi programirljivimi vezji. Posledica regularne zgradbe matrike logičnih celic je velik potencial povečevanja zmogljivosti. Proizvajalci FPGA so razvili družine vezij, znotraj katerih so integrirana vezja z enako zgradbo in tehnološko izvedbo, razlikujejo pa se po velikosti matrike, hitrosti in številu priključkov.

Povečevanje zmogljivosti je povezano z razvojem mikroelektronske tehnologije CMOS. Že od leta 1975 velja Moorov zakon, ki pravi, da se število tranzistorjev na integriranem vezju podvoji približno vsaki dve leti. V vsaki tehnološki generaciji sta se razpolovili površina in poraba moči tranzistorjev, obenem pa se je za okoli 40 % povišala frekvenca sinhronih vezij. Robert Dennard je odkril zakon, ki pravi, da ostane ob skaliranju vezij gostota moči, ki se pretvarja v toploto, konstantna. Vezja FPGA so dobro izkoriščala ta razvoj. Slika 2 prikazuje število tranzistorjev na posameznem vezju FPGA, od začetkov z nekaj sto tisoč tranzistorji do sodobnih vezij, ki vsebujejo več milijard tranzistorjev.



Slika 2: Razvoj tehnologije FPGA sledi Moorovemu zakonu.

Programirljiva vezja so bila najprej namenjena izvedbi vezne logike v napravah z namenski integriranimi vezji. Prva vezja FPGA so lahko posnemala digitalna vezja z nekaj tisoč logičnimi vrati in nekaj sto flip-flopi. Za izboljšanje učinkovitosti so vezjem dodali posebne gradnike: v logičnih celicah so strukture za hitro izvedbo prenosne logike aritmetičnih operacij, vpogledne tabele pa je mogoče uporabiti tudi kot majhne bralno-pisalne (RAM) pomnilnike. Pozneje so med matriko celic dodali nekoliko večje blokovne pomnilnike (BRAM) in bloke za digitalno obdelavo signalov (DSP, Digital Signal Processing).

Dodatni bloki v vezju FPGA so narejeni tako, da jih prilagodimo zelo različnim aplikacijam. Blokovni pomnilnik je statični pomnilnik s sinhronim vmesnikom in prilagodljivim številom vrat, naslovov in podatkovno širino. Pomnilnik BRAM kapacitete 36 Kb nastavimo kot 32K×1 bit, 16K×2 bita, 8K×4 bite, 4K×9 bitov, 2K×18 bitov 1K×36 bitov ali 512×72 bitov z eno- ali dvovratnim (dual-port) dostopom.

DSP-bloki vsebujejo gradnike za množenje, ki so pogosta operacija v digitalni obdelavi signalov. V novejših izvedbah ponujajo izbiro števila oz. velikosti množilnikov, dodatne aritmetične operacije, najzmogljivejši pa podpirajo poleg celoštevilskih tudi operacije s plavajočo vejico.

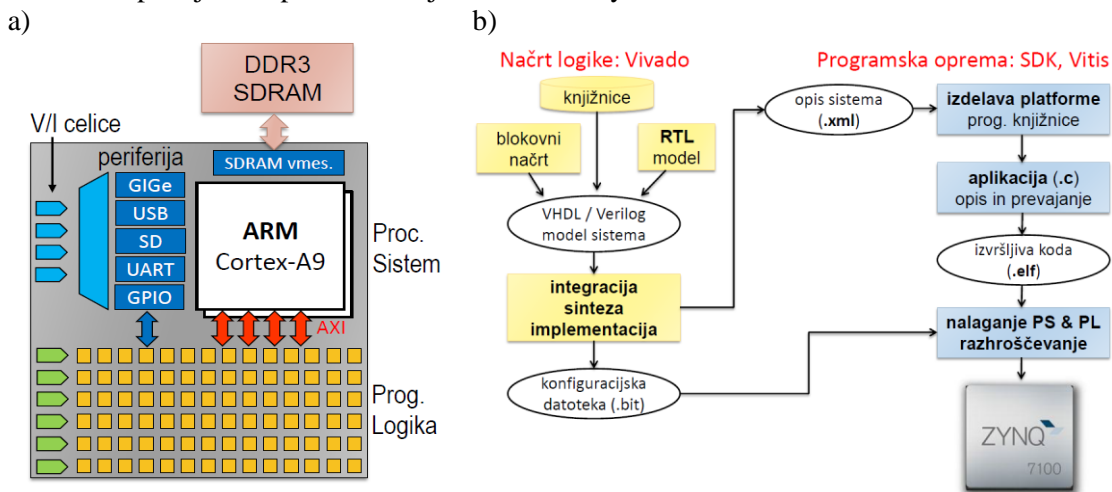
Okoli leta 2006 je Dennardovo skaliranje prenehalo, saj je zaradi majhnih dimenzij k porabi moči pomembno prispevala statična poraba oz. puščanje (leakage) tranzistorjev. Zmogljivosti ni bilo več mogoče povečevati z višanjem frekvenc, ker lahko pride do toplotnega uničenja elementov. Mikroprocesorji so razvili arhitekture z več jedri, programirljiva vezja pa vključujejo vse bolj specializirane gradnike. Poleg

matrike celic so v vezju procesorska jedra (CPU), večgigabitni zaporedni pretvorniki (MGT, Multi-Gigabit Transceiver), vmesniki za Ethernet komunikacijo, pomnilniški krmilniki ipd. Vezja s programirljivo logiko, procesorji in periferijo se imenujejo sistemi na čipu in zaslužijo posebno obravnavo.

Sistemi na čipu

Časovno manj kritična in postopkovno zahtevna opravila izvaja v digitalnih napravah vgrajen mikroprocesor oz. mikrokrmilnik. Za naprave, ki potrebujejo posebne vmesnike, strojno-pospešeno računanje ali sprotno obdelavo hitrih signalov uporabimo namenska ali programirljiva vezja. Heterogeno integrirano vezje s procesorskim jedrom, perifernimi enotami in programirljivo logiko se imenuje Sistem na čipu (SoC, System-on-Chip).

Mehkojedrni (soft-core) mikroprocesor je narejen z delom programirljive logike in ga vključimo v sistem kot komponento intelektualne lastnine (IP component). Zaradi programirljivih povezav ni tako učinkovit, kot če bil narejen s tranzistorji, zato imamo tudi družine vezij z vgrajenim procesorskim jedrom (hard-core). Slika 3a prikazuje blokovni diagram sistema na čipu Zynq proizvajalca AMD-Xilinx z dvojednim procesorjem ARM Cortex-A9 in pripadajočimi perifernimi enotami. Konkurenčni proizvajalec Intel-Altera ponuja zelo podobna vezja v družinah Cyclone V in Arria V.



Slika 3: a) Sistem na čipu AMD-Xilinx Zynq z dvojednim procesorjem ARM, perifernimi enotami in programirljivo matriko. Programska oprema je v zunanjem pomnilniku SDRAM. b) Potek načrtovanja strojne in programske opreme.

Sistemi na čipu imajo veliko povezav med programirljivo logiko, procesorjem in perifernimi enotami. SoC omogočajo visoke hitrosti prenosa podatkov in velike možnosti delitve nalog med programske in namensko strojno opremo. Zmogljivejši SoC imajo še več jeder, ki si med seboj delijo systemska in realno-časna opravila ter dodatne namenske strojne enote (npr. grafični procesor, video kodek).

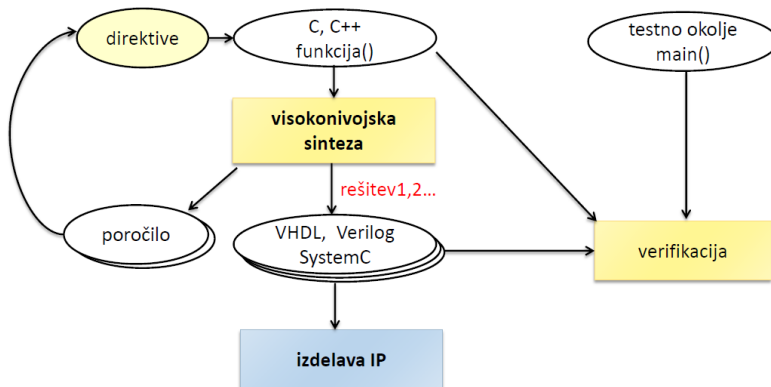
Proces razvoja naprav SoC vključuje načrtovanje logike in programske opreme (slika 3b). Načrtovanje logike poteka zelo podobno kot za vezja FPGA, s tem da vključuje orodja za povezavo s procesorjem. Periferne enote je treba priključiti na ustrezna vodila in jim dodeliti naslovni prostor. Rezultat prevajanja je poleg konfiguracijske datoteke tudi opis sistema, ki je osnova za izdelavo programske platforme. Po izdelavi in prevajanju programskih aplikacij izvedemo programiranje FPGA in nalaganje ter razhroščevanje aplikacije.

Načrtovalska orodja

Razvoj zmogljivejših programirljivih vezij in sistemov je povzročil potrebo po drugačnem in učinkovitejšem pristopu k načrtovanju logičnih gradnikov. Sinteza vezja iz strojno-opisnih jezikov je nadomestila shematsko načrtovanje že ob pojavu manjših programirljivih vezij. Opis algoritma v jeziku je bistveno bolj učinkovit kot risanje in popravljanje sheme, ki pa sta še vedno uporabna za sestavljanje sistemov iz komponent intelektualne lastnine (Intellectual Property, IP).

Zelo velika vezja FPGA in sistemi na čipu so lahko prevelik zalogaj za tradicionalne strojno-opisne jezike (VHDL, Verilog, System Verilog), ki obravnavajo vezja na ravni registrov (RTL). Pojavil se je načrtovalski razkorak (design gap) med učinkovitostjo načrtovanja in verifikacije ter zmogljivostjo ciljnega sistema. Odgovor na razkorak so orodja za visokonivojsko sintezo vezja.

Če je namen programirljive logike v SoC strojno pospeševanje algoritmov, nas bolj kot podrobnosti registrskega delovanja zanimajo lastnosti vezja ob določenih načrtovalskih odločitvah. Visokonivojska sinteza omogoča, da opis algoritma, npr. v obliki funkcije v jeziku C, avtomatsko pretvorimo v logično vezje, kot prikazuje slika 4. Glavna funkcija (main) predstavlja testno okolje za preizkušanje algoritma in verifikacijo sintetiziranega vezja.



Slika 4: Potek visokonivojske sinteze logičnega vezja iz opisa v jeziku C.

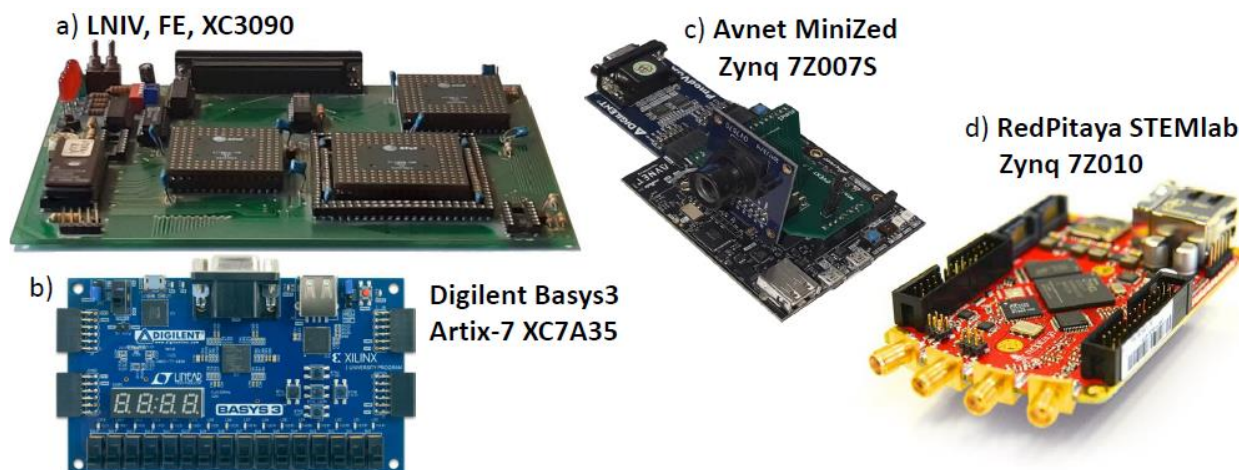
Visokonivojska sinteza potrebuje informacije o ciljni tehnologiji in direktive, s katerimi vodimo postopek sinteze logike. Z direktivami izbiramo vmesnike, nastavljamo stopnjo paralelizma, računski cevovod, delitev virov ipd. Sintetizator avtomatizira časovno delitev izvajanja algoritma in delitev logičnih virov, ki vključujejo gradnike za računanje s fiksno decimalno, plavajočo decimalno ali poljubno natančnimi celimi števili. Poročilo sinteze vsebuje oceno porabe virov in računskih zmogljivosti vezja, izraženih v številu ciklov nastavljene ure. Če sintezo ponavljamo z različnimi direktivami, raziskujemo načrtovalske odločitve in iščemo optimalno za svoje zahteve. Rezultat sinteze je komponenta IP v strojno-opisnem jeziku, ki jo integriramo v digitalni sistem in izvedemo tehnološko preslikavo.

Še en korak naprej so orodja namenjena računalniškim strokovnjakom, kjer poteka enovit razvoj strojne in programske opreme. Programska orodja so narejena za heterogene platforme, pri katerih ni potrebno načrtovalčevo poznavanje podrobnosti tehnološke izvedbe. Primer je ogrodje OpenCL, ki je napisano tako za strojne pospeševalnike v vezju FPGA kot za platforme z grafičnimi in signalnimi procesorji.

Uporaba programirljivih vezij

Vezja FPGA so se sprva uporabljala za izdelavo prototipov in posnemanje digitalnih vezij. Najbolj znan primer z začetkov razvoja emulatorjev je prototip procesorja Pentium na množici med seboj povezanih vezij FPGA, ki je bil sposoben naložiti operacijski sistem. Emulatorji so se pozneje razvili v specializirane naprave z vezji ASIC, prototipna izvedba logike v FPGA pa je še vedno pomemben korak pri razvoju in verifikaciji novih digitalnih vezij.

Prototipne razvojne plošče poleg FPGA vsebujejo še različne vmesnike za implementacijo digitalnih naprav. Slika 5 prikazuje nekaj plošč, ki jih uporabljamo v Laboratoriju za načrtovanje integriranih vezij na Fakulteti za elektrotehniko v Ljubljani (LNIV, FE). Prvo ploščo s tremi vezji FPGA kapacitete nekaj tisoč logičnih vrat smo razvili in uporabljali že leta 1995 (slika 5a), novejša pa so komercialni produkti, ki jim dodamo le specifične vmesnike.



Slika 5: Prototipne in učne razvojne plošče s programirljivimi vezji: a) emulator starejše generacije, b) novejša učna razvojna plošča, c) sistem na čipu s procesorskimi vmesniki in d) programirljivi sistem za obdelavo signalov.

Najpomembnejša aktualna področja uporabe FPGA so:

- telekomunikacije (paketna obdelava, 5G, radiodifuzija),
- industrijska (strojni vid, krmilniki, robotika, prototipi in posnemovalniki),
- avtomobilska (avtonomni asistenčni sistemi, infotainment, fuzija senzorjev),
- uporabniška elektronika, vojaška, letalska, vesoljska tehnika in
- podatkovni centri (vmesniki, strojno pospeševanje algoritmov).

Sodobna vezja FPGA glede na zmogljivost razdelimo na tri razrede. Najzmogljivejša vezja, izdelana v najnovejših tehnologijah integriranih vezij, so iz družin Virtex proizvajalca AMD-Xilinx, Stratix proizvajalca Intel-Altera in Speedster proizvajalca Ashronix. To so izredno draga vezja, ki se uporabljajo za hitro obdelavo podatkov, izdelavo zmogljivih emulatorjev in pospeševanje algoritmov v podatkovnih centrih. Podjetje Microsoft jih uporablja za pohitritev iskanja v podatkovnih centrih, Amazon pa omogoča dostop do zmogljivih strojnih pospeševalnikov v okviru storitev v oblaku.

Srednji razred predstavlja kompromis med zmogljivostjo in ceno, ki omogoča širšo uporabo na področju industrijske, avtomobilske in uporabniške elektronike. Vezja najdemo v multimedijskih sistemih, merilnih napravah, medicinskih napravah ipd. Ena izmed ključnih prednosti tehnologije FPGA je hiter vstop z izdelkom na trg ter možnost poznejših popravkov in pretvorbe načrta v namensko integrirano vezje.

Za manj kompleksna digitalna vezja, naprave z nizko porabo in učne namene so najprimernejša vezja iz nižjega razreda. Proizvajalcev FPGA je v nižjem in srednjem razredu precej več, poleg največjih omenimo še Lattice, Microchip in Microsemi (nekdanji Actel). V zadnjem času vstopa v tehnologijo FPGA tudi japonski Renesas z vezji za uporabniške in IoT-naprave ter evropski Cologne Chip z gradniki za široko potrošnjo.

Andrej Trost je doktoriral leta 2000 na Fakulteti za elektrotehniko Univerze v Ljubljani, kjer je trenutno izredni profesor na Katedri za elektroniko. Raziskovalno se ukvarja z razvojem vezij v tehnologiji FPGA in načrtovanjem digitalnih sistemov za pedagoške in industrijske aplikacije. Razvil je vrsto prototipnih sistemov za emulacijo digitalnih vezij in programska orodja za načrtovanje digitalnih vezij z visokonivojskimi jeziki. Načrtovanje digitalnih vezij poučuje pri predmetih na dodiplomskem in podiplomskem študiju. Objavil je več kot 100 znanstvenih prispevkov in je mentor pri 130 zaključnih nalogah. V letih 2018 do 2021 je vodil Slovensko sekcijo IEEE.