

PROGRAMABILNA LOGIČKA KOLA

1. DIGITALNA INTEGRISANA KOLA

1.1 Stepen integracije

Složenost integrisanog kola se meri stepenom integracije, koji predstavlja broj osnovnih elemenata (tranzistora, gejtova) realizovanih (integriranih) u jednom kolu. Po stepenu integracije integrisana kola se mogu podeliti u pet grupa:

- kola malog stepena integracije SSI (*Small-Scale Integration*). Sadrže do 100 osnovnih elemenata. (logička kola, flip-flopovi)
- kola srednjeg stepena integracije, MSI (*Medium-Scale Integration*). Sadrže od 100 do 1000 osnovnih elemenata. (registri, brojači, aritmetička kola).
- kola visokog stepena integracije, LSI (*Large-Scale Integration*). Sadrže od 1000 do 10000 elemenata. (8-bitni mikroprocesori, RAM, ROM)
- kola vrlo visokog stepena integracije VLSI (*Very Large Scale Integration*), sa 10000 do 100000 elemenata. (16- i 32-bitni mikroprocesori)
- kola ultra velikog stepena integracije, ULSI (*Ultra Large Scale Integration*), sa 100000 do 1.000.000 i više osnovnih elemenata. (64-bitni mikroprocesori, memorije velikog kapaciteta (>100Mbita)).

1.2. Aplikaciono specifična integrisana kola (ASIC)

Prema metodologiji projektovanja, integrisana kola možemo podeliti na:

- (a) Standardne komponente (*Standard Integrated Circuits* - SIC) -u ovu grupu spadaju SSI, MSI, LSI, VLSI i ULSI kola, koja su, u tehnološkom smislu, proizvedena nezavisno od krajnjeg korisnika. Naime, kolo se proizvodi za nepoznatog korisnika, korisnik nema uticaj na karakteristike kola, kolo se proizvodi u velikim serijama po relativno niskoj pojedinačnoj ceni.
- (b) Aplikaciono specifična integrisana kola, ASIC (*Application Specific Integrated Circuits*) - ova kola su projektovana tako da odgovaraju konkretnoj specifičnoj nameni. Za razliku od standardnih komponenata, funkciju ASIC kola definiše korisnik. ASIC kola se proizvode po narudžbi i u relativno malim serijama što poskupljuje njihovu proizvodnju.

Razlikuju se sledeće vrste ASIC kola:

1. *full-custom* (kola potpuno po naurudžbi). Kolo se u potpunosti (do nivoa osnovnih elemenata) projektuje za tačno definisanu primenu. Ostvaruju se najbolje karakterisike, ali je pojedinačna cena visoka.
2. *semi-custom* (kola po narudžbi). Za fabrikaciju naručenog kola, proizvođač koristi pretprojektovane ili prefabrikovane (nedovršene) strukture koje doradom prilagođava korisničkim zahtevima. S obzirom da se na taj način skraćuje vreme pripreme proizvodnje, krajnja pojedinačna cena je niža, ali su performanse lošije u poređenju sa *full custom* projektovanjem.
 - *standard cell* (standardne ćelije). Proizvođač nudi veći broj standardnih ćelija (logička kola, flip-flopovi, ali i strukture SSI, MSI složenosti). Korisnik isporučuje proizvođaču šemu sačinjenu od standardnih ćelija, na osnovu koje proizvođač izrađuje čip. Pri tome proizvođač ne startuje od početka već samo uklapa i povezuje pretprojektovane standardne ćelije.
 - *gate arrays* (gejtovske matrice). Kolo se izrađuje na podlozi sačinjenoj od velikog broja prefabrikovanih jednostavnih logičkih elemenata ili MOS tranzistora koji nisu međusobno povezani. Korisnik isporučuje proizvođaču šemu do nivoa osnovnih elemenata na bazi koje proizvođač obavlja doradu, tj. metalizaciju, polaznog čipa.
3. PLD (*Programmable Logic Devices*), programabilna logička kola. To su integrisana kola koja se mogu konfigurisati (tj. programirati) od strane krajnjeg korisnika da bi ispunili zahteve koji se

tiču konkretne aplikacije. Strukturu PLD kola čini fiksni skup komponenata kakvi su logički gejtovi, ili složeniji logički blokovi, međusobno spojeni programabilnim vezama. Programiranjem ovih veza, krajnji korisnik definiše funkciju kola, a jednom programirana funkcija se može menjati u potpunosti ili delimično. Suštinska razlika u odnosu na ostale tipove ASIC kola je u tome da PLD kolo programira sam korisnik, čime je on u potpunosti nezavisan od proizvođača.

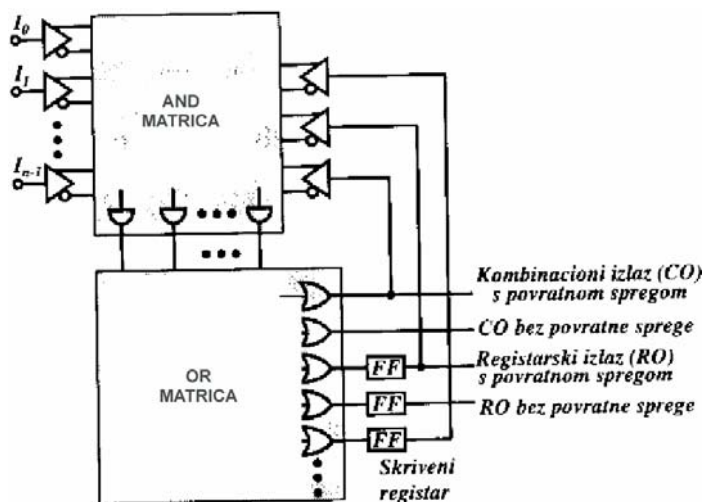
2. KLASIFIKACIJA PLD KOLA

Danas je na tržištu dostupan veliki broj tipova PLD kola, koji se razlikuju po načinu programiranja, složenosti (tj. logičkom kapacitetu, ili implementacionoj moći), unutrašnjoj strukturi, brzini rada, broju pinova. Broj dostupnih komponenata kod savremenih PLD-ova je veoma veliki tako da se sa aspekta složenosti ova kola protežu od MSI IC do VLSI IC. Raspoloživi logički kapacitet se kreće od nekoliko stotina do nekoliko stotina hiljada ekvivalentnih gejtova, a taktna učestanost od nekoliko desetina MHz do 200 MHz. PLD kola se pakuju u kućištima sa nekoliko desetina do nekoliko stotina pinova. To znači da su PLD-ovi u stanju da implementiraju širok dijapazon kombinacionih i sekvencijalnih logičkih funkcija.

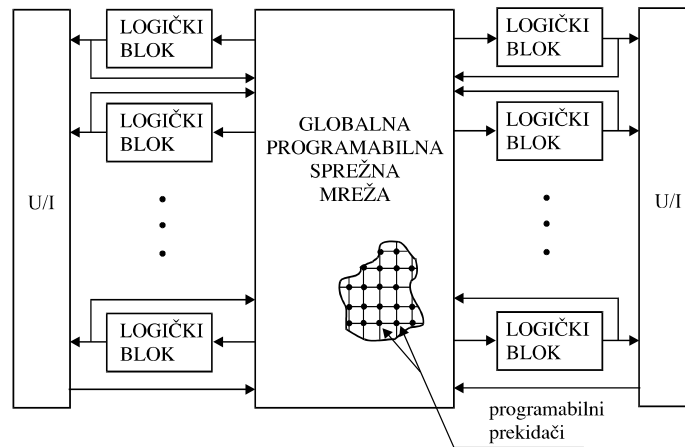
PLD kola su veoma pogodna za brzu izradu prototipa, ali i konačnih rešenja koja će se proizvoditi u malim serijama. Vreme trajanja faze projektovanja kod ovih kola je veoma kratko. U slučaju da je suštinska vrednost proizvoda kratko vreme pojavljivanja na tržištu tada izbor PLD-kola predstavlja pravo rešenje čak i sa aspekta cene.

Prema složenosti i organizaciji unutrašnje strukture, savremena PLD se mogu klasifikovati u sledeće tri kategorije:

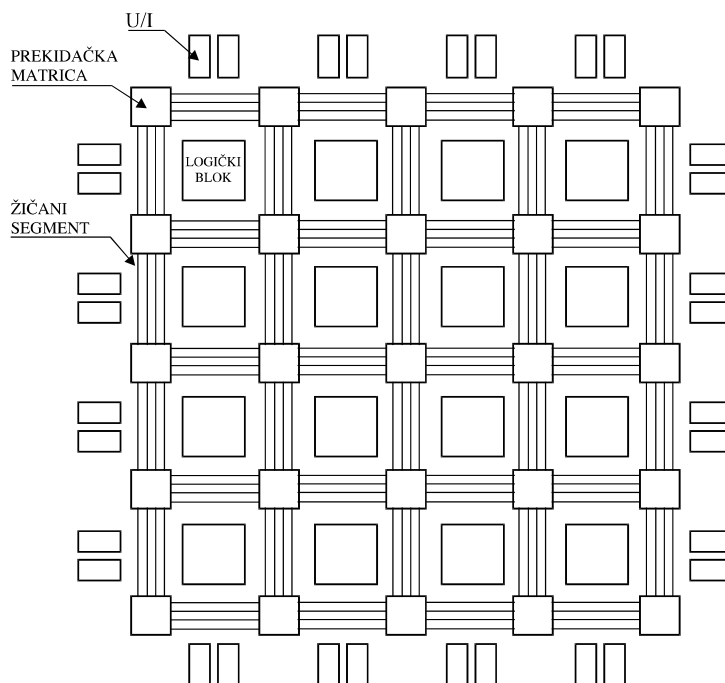
1. SPLD (*Simple PLD*). PLD kola srednjeg stepena integracije zasnovana na programabilnim AND-OR poljima, koja se koriste za implementaciju logičkih funkcija izraženih u formi "suma proizvoda" (Sl. 1).
2. CPLD (*Complex PLD*). Sastoji se od većeg broja programabilnih logičkih blokova povezanih preko centralizovane programabilne sprežne mreže (Sl. 2). Po unutrašnjoj strukturi, logički blokovi su slični tipičnom SPLD kolu.
3. FPGA (*Field Programmable Gate Array*). PLD kola visokog stepena integracije koje se izvodi kao polje velikog broja programabilnih logičkih blokova (čelija) jednostavne strukture raspoređenih na infrastrukturi koja ih međusobno povezuje. (Sl. 3).



Sl. 1 SPLD arhitektura.



SI. 2 CPLD arhitektura.



SI. 3 FPGA arhitektura.

3. TEHNOLOGIJE PROGRAMIRANJA

Konfigurabilnost PLD kola omogućena je postojanjem internih programabilnih tačka koje, u suštini, predstavljaju prekidačke elemente koji se mogu programirati tako da se ponašaju kao kratko-spojeni ili otvoreni prekidači. U fazi programiranja kola, signali koji se dovode na ulaz kola otvaraju i zatvaraju programabilne tačke (elektronske prekidače) i na taj način ostvaruju željene oblike povezivanja internih komponenta.

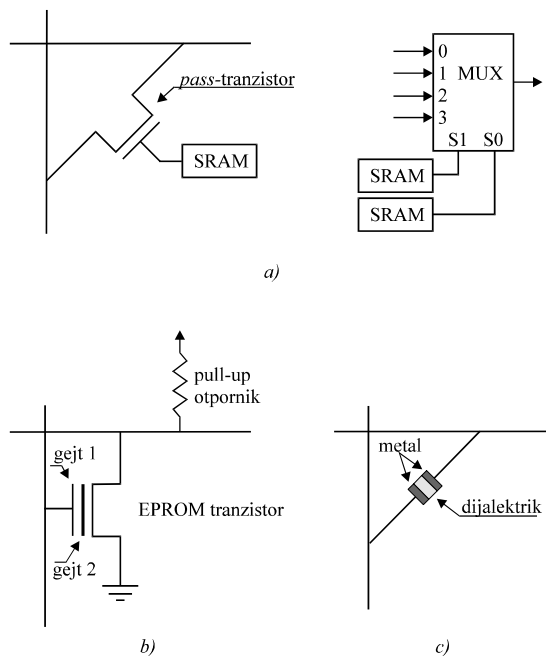
Kod prvih PLD kola za realizaciju programabilnih prekidača korišćeni su poluprovodnički osigurači. Inicijalno svi osigurači su "nesagoreni". Pobuđivanje kola nešto višim naponima od radnih uslovljava da kroz PLD protiču velike struje. Kao posledica, veze koje formiraju osigurači se raskidaju. Treba pri ovome naglasiti da ne postoji metod za rekonstrukciju (obnavljanje) stanja prekidača, tj. njegovo sagorevanje je trajno ili bespovratno. Tipičan predstavnik ovakvih kola je programabilni ROM ili PROM.

Danas, kod SPLD i CPLD kola, za realizaciju programabilnih prekidača, prevladavaju tehnologije zasnovane na tranzistorima sa izolovanim gejtom (*floating-gate*) EPROM ili EEPROM tipa, dok se kod FPGA uglavnom koriste SRAM (*Static RAM*) i *antifuse* tehnologije.

1) *SRAM tehnologija programiranja*. Kod ove tehnologije, konfigurisanje kola se ostvaruje pomoću *pass* tranzistora i multipleksera koji se upravljaju SRAM ćelijama. Par "SRAM ćelija - *pass* tranzistor" se koristi kao programabilna veza između dva žičana segmenta (Sl. 4a). Kada je u SRAM ćeliji memorisana "jedinica", *pass* tranzistor se ponaša kao zatvoren prekidač male serijske otpornosti. U suprotnom, kada je stanje SRAM ćelije "nula", *pass* tranzistor je otvoren prekidač veoma velike serijske otpornosti. Za upravljanje multiplekserom, SRAM ćelije su vezane za selekzione ulaze multipleksera (Sl. 4a). Stanje SRAM ćelija određuje koji je od ulaza multipleksera povezan sa izlazom.

S obzirom da SRAM ćelije gube memorisan sadržaj nakon isključenja napajanja, PLD kolo mora biti napunjeno konfiguracionim sadržajem pri svakom uključanju napajanja. To zahteva eksternu, premanentnu memoriju (tipa EPROM ili EEPROM) za čuvanje konfiguracionih bitova. U fazi punjenja, sve SRAM ćelije PLD kola su redno povezane u strukturu pomeračkog registra, a samo punjenje se vrši serijskim upisom konfiguracionih bitova. U zavisnosti od tipa i kapaciteta PLD kola, punjenje traje od nekoliko milisekundi do nekoliko desetina milisekundi. Glavna prednost SPRAM tehnologije, u odnosu na druge tehnologije programiranja, je mogućnost brzog reprogramiranja, koje može biti obavljeno "u sistemu". Glavni nedostaci su relativno velika površina koju na čipu zauzimaju SRAM ćelije (tipična SRAM ćelija se realizuje sa pet tranzistora) i neophodnost ugradnje eksterne memorije.

SRAM tehnologija programiranja se koristi u FPGA kolima firmi: Xilinx, Plessey, Algotronix, Concurrent Logic i Toshiba.



Sl. 4 Tehnologije programiranja PLD kola: (a) SRAM; (b) EEPROM; (c) antifuz.

2) *Floating_Gate tehnologija*. Radi se o istoj tehnologiji koja se sreće kod EPROM i EEPROM memorija. Programabilni prekidač je tranzistor sa izolovanim gejtom (EPROM tranzistor), koji se, programiranjem, može permanentno zakočiti (Sl. 4b). Ovo se postiže injektovanjem naelektrisanja na izolovani gejt tranzistora (gejt 2 na Sl. 4b). Do injektovanja naelektrisanja dolazi kada se između upravljačkog gejta (gejt 1 na Sl. 4b) i drejna tranzistora dovede visok napon. Injektovano naelektrisanje povećava napon praga tranzistora, tako da on, u normalnom režimu rada, ostaje stalno zakočen (tj. neprovodan). Injektovano naelektrisanje se odstranjuje izlaganjem izolovanog gejta dejstvu ultraljubičastog svetla. EEPROM tehnologija je slična EPROM tehnologiji, s tom razlikom što se odstranjivnje injektovanog naelektrisanja može ostvariti električnim putem, bez ultraljubičastog svetla.

EPROM tehnologija, kao i SRAM, omogućava reprogramiranje. Prednost EPROM tehnologije je što ona ne zahteva eksternu memoriju za čuvanje konfiguracionih bitova, ali je zato proces reprogramiranja duži i može se obaviti samo "izvan sistema". Pored toga, serijska otpornost provodnog EPROM tranzistora je veća (oko dva puta) od serijske otpornosti *pass* tranzistora koji se koristi kod SRAM tehnologije. Takođe, statička potrošnja kola koje koristi EPROM tranzistore je zbog "pull_up" otpornika (vidi sliku) veća od potrošnje kola koje koristi SRAM ćelije.

EPROM tehnologija programiranja se koristi kod većine CPLD kola i kod FPGA kola firmi Altera i Plus Logic. Kod FPGA kola firmi AMD i Lattice koristi se EEPROM tehnologija.

3) *Antifuse tehnologija programiranja.* *Antifuse* je komponenta sa dva kraja koja u neprogramiranom stanju poseduje veoma veliku serijsku otpornost (tj. predstavlja otvoren prekidač). *Antifuse* se sastoji od tri sloja. Krajnji slojevi su provodni, a sloj u sredini je dielektrik. Antifuse se postavlja između dva žičana segmenta, kao što je to prikazano na Sl. 4c. Neprogramiran, dielektrik izoluje provodne slojeve; programiran, on postaje permanentni spoj male otpornosti. Programiranje se vrši dovođenjem visokog napona (od 11 do 20V, što zavisi od tipa antifusa) na krajeve *antifuse*-a. Dovođenje napona za programiranje se vrši preko dodatnih *pass* tranzistora, koji moraju da imaju širok kanal, kao bi propustili relativno veliku struju programiranja (oko 5mA).

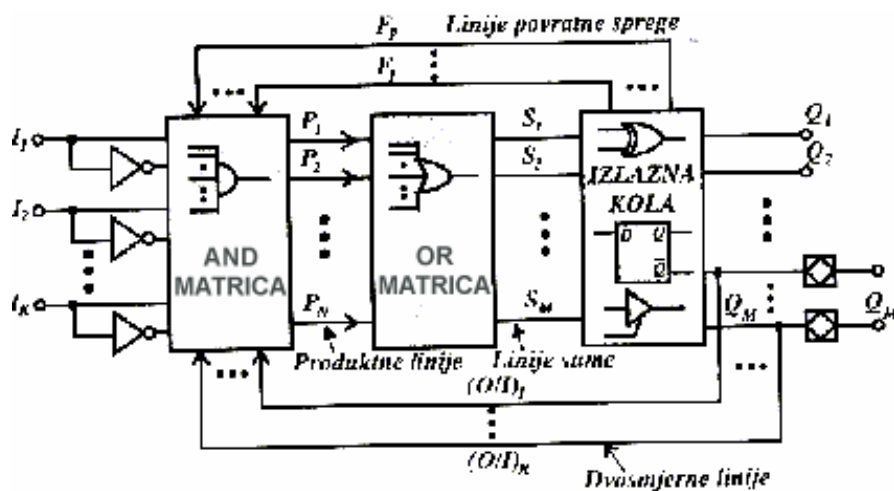
Glavna prednost *antifuse*-a su njegove male dimenzije. Ova prednost je donekle redukovana neizbežnim *pass* tranzistorima za programiranje. U poređenju sa drugim tehnologijama, *antifuse* u programiranom stanju ima relativno malu serijsku otpornost i parazitnu kapacitivnost. *Antifuse* je normalno "otvoren" prekidač, pa se prilikom konfigurisanja kola programiraju samo oni *antifuse*-ovi koje treba "zatvoriti". S obzirom da je kod tipičnih aplikacija broj "zatvorenih" prekidača mnogo manji od broja "otvorenih" (tipično broj "zatvorenih" prekidača ne prelazi 2% od ukupnog broja prekidača u kolu), to programiranje kola koje koristi *antifuse* tehnologiju traje kraće od programiranja kola koje koristi EPROM tehnologiju. Glavni nedostatak *antifuse* tehnologije je nemogućnost reprogramiranja.

Antifuse tehnologiju se sreće kod FPGA kola firmi: Actel, Quark Logic i Crosspoint.

4. PLD ARHITEKTURE

4.1. SPLD kola

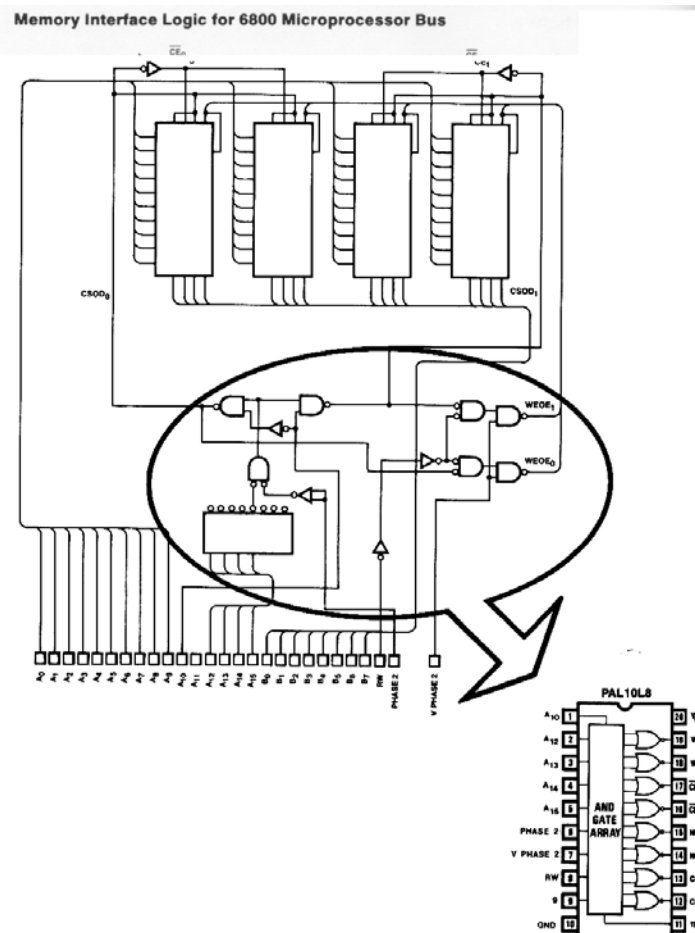
Na Sl. 5 je prikazan opšti model SPLD arhitekture. Glavni deo ove arhitekture čine dve programabilne logičke matrice: (a) AND matrica za formiranje logičkih proizvoda i (b) OR matrica za sumiranje logičkih proizvoda. Pored AND-OR polja pojedine PLD arhitekture ovog tipa poseduju izlazni stepen koji tipično obezbeđuje: (1) povratne veze sa izlaza na ulaz, (2) mogućnost promene polariteta izlaznog signala, (3) mogućnost da se pojedini eksterni priključci koriste bilo kao ulazi bilo kao izlazi i (4) memorijske elemente za memorisanje stanja izlaznih signala, što omogućuje sintezu sekvencijalnih digitalnih kola.



Sl. 5 Model SPLD kola

Uobičajeni naziv za SPLD kolo kod koga se obe logičke matrice mogu programirati je PLA (*Programmable Logic Array*). Pored PLA strukture u upotrebi su i AND-OR polja kod kojih je jedna od logičkih matrica fiksna, a druga programabilna. Konfiguracija fiksne matrice je permanentna, izvedena u toku fabrikacije kola i ne može se menjati od strane krajnjeg korisnika. AND-OR polje kod koga je AND matrica programabilna, a OR matrica fiksna se zove PAL (*Programmable Array Logic*). AND-OR polje sa fiksnom AND matricom i programabilnom OR matricom se zove ROM (*Read Only Memory*).

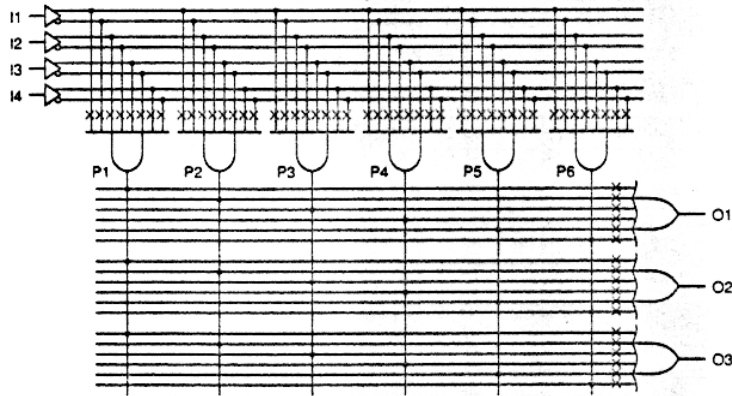
U poređenju sa ostalim tipovima PLD kola (CPLD i FPGA), SPLD kola se odlikuju relativno niskim logičkim kapacitetom (do nekoliko stotina ekvivalentnih gejtova). Glavna namena SPLD kola je zamena standardnih digitalnih kola niskog i srednjeg stepena integracije (Sl. 6) čime se postiže ušteda prostora na štampanoj ploči, pojeftinjuje proizvodnja i povećava pouzdanost u radu. Brzina rada SPLD kola je velika. Propagaciono kašnjenje “od pina do pina” je fiksno (ne zavisi od implementirane funkcije i iznosi do nekoliko nanosekundi).



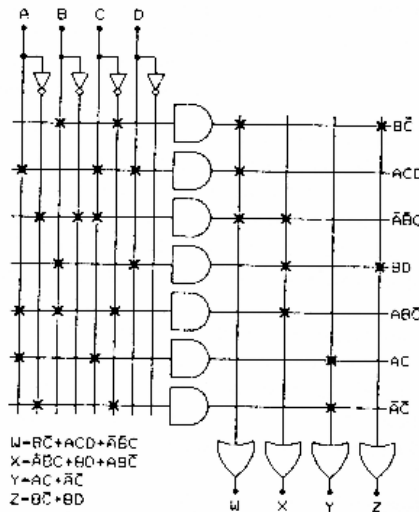
Sl. 6 Primena SPLD kola.

4.1.1 PLA

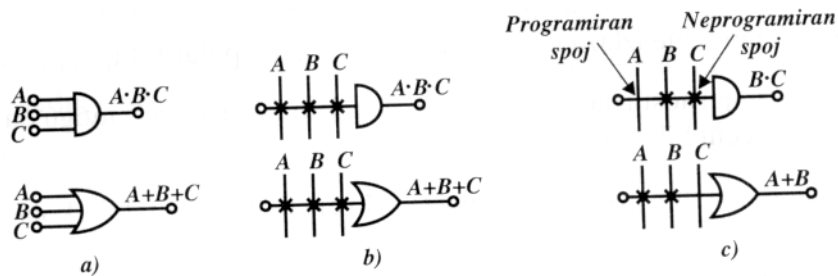
Struktura PLA kola prikazana je na Sl. 7. Programiranjem odgovarajućih prekidača (predstavljenih simbolom X na Sl. 7), svako AND kolo iz AND matrice se može povezati sa bilo kojim podskupom primarnih ulaza i njihovih komplekata formirajući tako jedan logički produkta. Slično, svako OR kolo iz programabilne OR matrice se može povezati sa bilo kojim podskupom izlaza AND matrice, formirajući tako jednu logičku funkciju. U opštem slučaju, PLA sa n ulaza, m izlaza i p produktnih članova (AND kola u AND matrici) može da realizuje proizvoljan sistem od m logičkih funkcija od n promenljivih kod koga je ukupan broj različitih produkata manji ili jednak p . Na Sl. 8, prikazan je primer realizacije sistema od 4 logičke funkcije od 4 promenljive pomoću PLA kola sa 4 ulaza, 4 izlaza i 7 logičkih produkta. Na Sl. 8, PLA struktura je predstavljena u pojednostavljenoj šematskoj formi. Način predstavljanja AND i OR logičkih kola u PLD strukturi prikazan je na Sl. 9.



Sl. 7 PLA



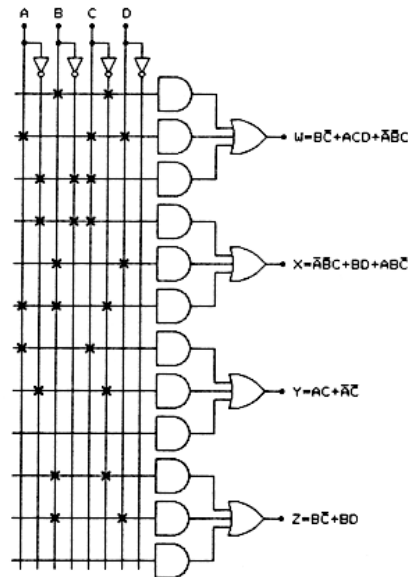
Sl. 8 Realizacija logičkih funkcija pomoću PLA



Sl. 9 a) standardni simboli; b) i c) PLD simboli.

4.1.2. PAL

PAL kola su dvonivovske AND-OR kombinacije mreže kod kojih je AND matrica programabilan, a OR matrica fiksna. Svaki produkt realizovan u AND matrici u permanentnom je spoju sa tačno jednim izlaznim OR kolom. Na Sl. 10 je prikazana struktura PAL kola sa 4 ulaza i 4 izlaza kod koga AND matrica može da realizuje 12 logičkih produkta. Na istoj slici prikazan je načina realizacije jednog konkretnog sistema logičkih funkcija. Produkti su podeljeni na grupe od po 3 produkta, gde se svaka grupa sumira uz pomoć jednog izlaznog OR kola. U poređenju sa PLA strukturom, fiksna struktura OR matrice PAL kola nameće dodatna ograničenja u pogledu mogućnosti realizacije logičkih funkcija. Naime PAL kolo sa m ulaza i n izlaza i homogenom raspodelom produkata sa p produkta po grupi može da realizuje sistem od n logičkih funkcija od m promenljivih pri čemu svaka logička funkcija poseduje najviše p produkta.

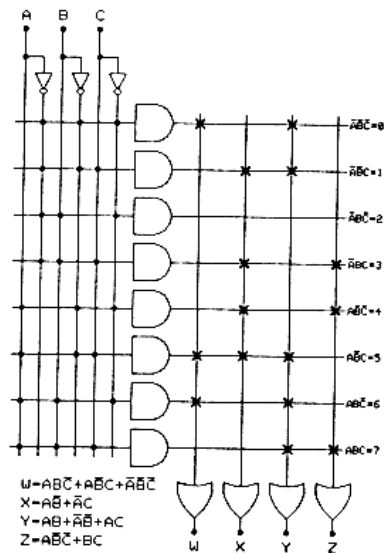


Sl. 10 PAL sa 4 ulaza i 4 izlaza

4.1.3. ROM

ROM je programabilno AND-OR polje kod koga je AND matrica fiksna, a OR matrica programabilna. Realizacija logičkih funkcija pomoću ROM-a zasnovana je na poznatom stavu iz *Boole*-ove da se svaka logička funkcija može predstaviti u vidu sume potpunih produkta (tj. minterma). Kod ROM-a sa n ulaza, fiksna AND matrica generiše sve moguće minterme od n promenljivih, dok se u OR matrici, programiranjem prekidača, sumiraju samo oni mintermi koji ulaze u izraz za sumu minterma konkretne logičke funkcije. Na taj način, ROM sa n ulaza i m izlaza može da realizuje bilo koji sistem od m logičkih funkcija od n promenljivih.

Struktura ROM-a sa 3 ulaza i 2 izlaza prikazana je na Sl. 11. Fiksni spojevi u AND matrici naznačeni su tačkama. U opštem slučaju, kod ROM-a sa n ulaza i m izlaza fiksna AND matrica se sastoji od 2^n n -ulaznih AND kola, a OR matrica od m 2^n -ulaznih OR kola. Svako AND kolo u AND matrici je u fiksnom spoju sa jednom kombinacijom ulaza i njihovih komplementa, formirajući tako jedan minterm. Drugim rečima, AND matrica ima funkciju $n/2^n$ dekodera.



Sl. 11 Realizacija logičkih funkcija pomoću ROM-a

4.1.4. Poređenje PLA, PAL i ROM

Za razliku od PLA i PAL, ROM ne postavlja ograničenja u pogledu mogućnosti realizacije logičkih funkcija. U tom smislu ROM se može smatrati univerzalnim kombinacionim blokom. Međutim, kako je kapacitet ROM-a (tj. broj programabilnih prekidača) eksponencijalna funkcija broja ulaza, ROM postaje izrazito neekonomično rešenje kada je broj ulaza veliki (tipično, veći od 16). Takođe, glomazne AND i OR matrice kod ROM-a sa velikim brojem ulaz povećavaju propagaciono kašnjenje signala što smanjuje brzinu rada kola. Iz tog razloga ROM se koristi za realizaciju kombinacionih kola u primenama gde se sistem logičkih funkcija karakteriše neregularnom strukturom u smislu da se ne može predstaviti kompaktnom analitičkom prezentacijom. Tipičan primer ovakvih kola su konvertori kodova. S druge strane, PLA i PAL su pogodni za realizaciju kombinacionih mreža regularne strukture koje se mogu opisati jednostavnim logičkim funkcijama u obliku sume logičkih produkta, kako što su multiplekseri, dekoderi, komparatori i sl. Takođe, PLD kola PLA i PAL strukture tipično poseduju memorijske elemente tipa flipflop koji omogućavaju direktnu realizaciju sekvencijalnih kola (tipa brojač, pomerački registar, konačni automat). PLA i PAL kola se koriste isključivo kao PLD komponente, dok je glavna primena ROM-a smeštanje programa u mikroprocesorskim sistemima.

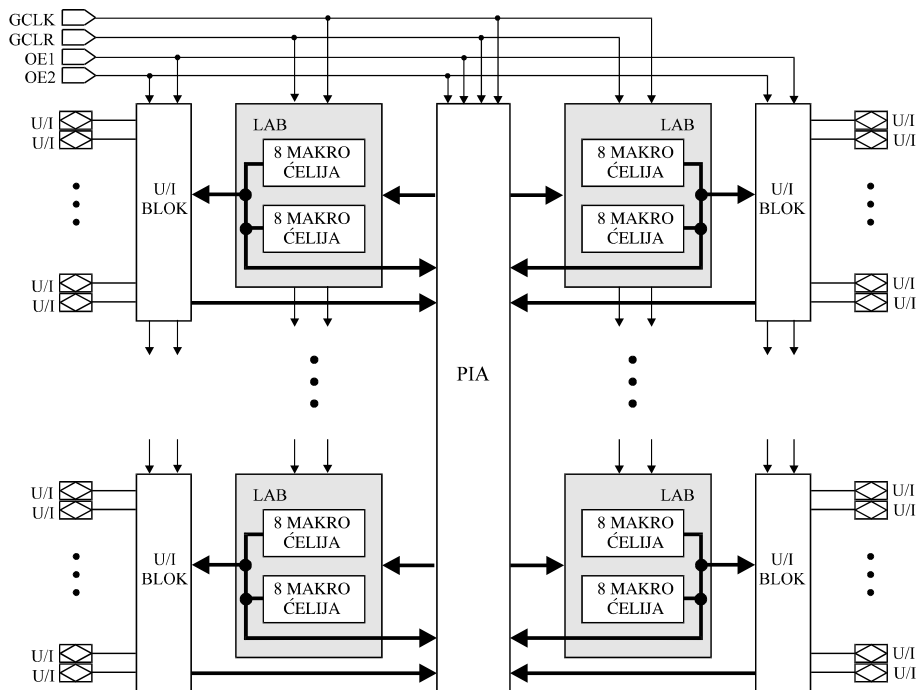
4.2. CPLD ARHITEKTURE

4.2.1. Struktura logičkog bloka

Kao što je već rečeno, CPLD kola se sastoje od više logičkih blokova SPLD strukture povezanih globalnom programabilnom sprežnom mrežom (Sl. 2). Međutim, čak i na nivou logičkih blokova, CPLD kola su obično značajno složenija od tipičnih SPLD kola. U većini slučajeva, logički blokovi koji se koriste kod CPLD kola se mogu smatrati nekom formom *proširenog* PAL-a, tj. PAL sa elementima PLA arhitekture. Proširenje PAL se vrši u cilju ekonomičnijeg korišćenja logičkih proizvoda programabilne AND matrice. Kod klasičnih PAL-ova, raspodela logičkih proizvoda je fiksna u smislu da je svaki logički proizvod privatan za jedno izlazno AND kolo (tj. makročeliju). Privatni logički proizvodi koji ostaju neiskorišćeni ne mogu se pridružiti nekoj drugoj makročeliji. Takođe, makročelije ne mogu da imaju zajedničke logičke proizvode, pa u slučajevima kada se isti proizvod koristi za formiranje logičkih funkcija u više makročelija, taj proizvod mora da se formira u svakoj makročeliji u kojoj se koristi. Kod savremenih CPLD kola, korišćenjem različitih tehnika, ovi nedostaci su u velikoj meri otklonjeni.

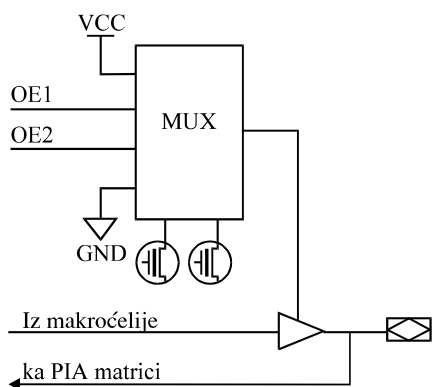
4.2.2. Altera Max

Firma Altera je razvila tri serije CPLD kola: Max 5000, 7000 i 9000. Sve tri serije imaju klasičnu CPLD arhitekturu () koja se sastoji od skupa logičkih blokova, tj LAB blokova (*Logic Array Block - LAB*) međusobno povezanih globalnom programabilnom prekidačkom matricom, tj. PIA matricom (*Programmable Interconnect Matrix - PIA*). U nastavku će biti opisana arhitektura kola iz serije Max 7000.

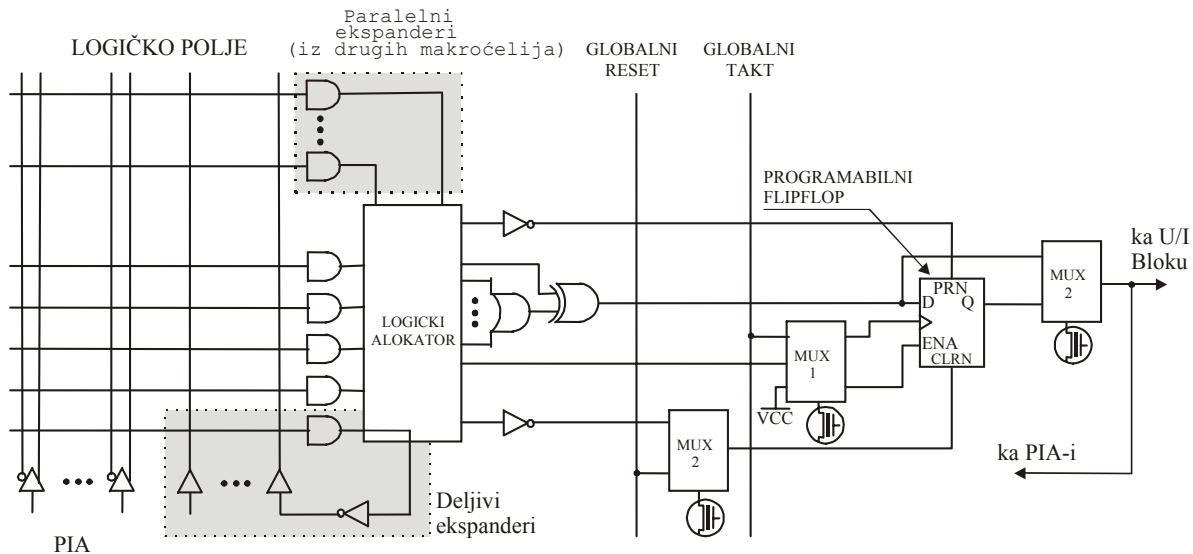


Sl. 12 Arhitektura CPLD kola Altera Max 7000.

Arhitektura serije Max 7000 je prikazana na Sl. 12. Broj LAB blokova se kreće od 2 do 16, što zavisi od tipa kola. LAB blokovi imaju strukturu proširenog PAL-a i sadrže po 16 makroćelija svrstanih u dve grupe od po 8 makroćelija. Kolo poseduje četiri pina posebne namene (globalni takt - GCLK, globalni reset - GCLR, dozvola izlaza OE1 i OE2). Preostali pinovi su bidirekcionni (U/I) i po potrebi se mogu konfigurisati bilo kao ulazi bilo kao izlazi. Svaki U/I pin je preko U/I bloka povezana sa jednom makroćelijom, kao što je to prikazano na Sl. 13. Kada se pin koristi kao izlaz, kontrola izlaznog tro-statičkog bafera se ostvaruje pomoću jednog od dva globalna signala OE1 ili OE2. Kada se U/I pin koristi kao ulaz, izlaz tro-statičkog bafera je permanento postavljen u stanje visoke impedanse, a signal doveden na pin se vodi u PIA matricu, preko koje se može proslediti bilo kom LAB bloku. S obzirom da se izlaz svake makroćelije direktno vraća u PIA matricu, u slučajevima kada se U/I pin koristi kao ulaz odgovarajuća makroćelija ne ostaje neiskorišćene već se može upotrebiti za formiranje neke "interne" funkcije.



Sl. 13 U/I blok serije Max 7000



Sl. 14 Makroćelija serije Max 7000

Struktura jedne makroćelije LAB bloka je prikazana na Sl. 14. Logičko polje je programabilna AND matrica, globalna na nivou LAB bloka, koja se koristi za formiranje logičkih proizvoda. Za svaku makroćeliju, u logičkom polju se formira pet privatnih logičkih proizvoda. Mada se u praktičnim primenama većina potrebnih kombinacionih funkcija može realizovati sa pet logičkih proizvoda, neke kombinacione funkcije su složenije i zahtevaju veći broj logičkih proizvoda. Da bi se u takvim slučajevima obezbedilo ekonomično korišćenje raspoloživog logičkog kapaciteta LAB bloka, koriste se paralelni ekspanderi (*parallel expanders*) i deljivi ekspanderi (*shared expanders*). Paralelni ekspanderi predstavljaju varijantu koncepta preusmeravanja logičkih proizvoda kod koje makroćelija i može da "pozajmi" svoje logičke proizvode makroćeliji $i+1$ (ali ne i makroćeliji $i-1$). Dodatno ograničenje je da se logički proizvodi mogu preusmeravati samo unutar grupe od 8 makroćelija, ali ne i između ovih grupa ili između LAB blokova. Deljivi ekspanderi se formiraju tako što se po jedan privatni logički proizvod iz svake makroćelije invertuje i vraća u logičko polje, tako da se može pridodati bilo kom logičkim proizvodu koji se formira u LAB bloku. Takođe, unakrsnim povezivanjem deljivih ekspandera mogu se formirati dodatni lečevi i flipflopovi. Makroćelija se sastoji od: logičkog alokoatora, OR kola za sumiranje logičkih proizvoda i programabilnog flipflopa. Posredstvom logičkog alokatora, na ulaze OR kola je moguće priključiti bilo koji podskup ulaznih logičkih proizvoda. Takođe, bilo koji od ulaznih logičkih proizvoda se može koristiti za upravljanje flipflopom (taktovanje, asinhrono resetovani i setovanje) i kontrolu polariteta kombinacione funkcije formirane na izlazu OR kola. Flipflop se može konfigurisati tako da funkcioniše kao D, T, JK ili SR flipflop. Za taktovanje i resetovanje flipflopa je moguće koristiti i globalne signale za takt i reset, što se reguliše multiplekserima 1 i 2. Makroćelija se može konfigurisati i da obavlja samo kombinacionu funkciju, tako što se multiplekserom 3 na U/I blok direktno priključuje izlaz EXOR kola.

4.2.3. Primena CPLD kola

Zahvaljujući relativno velikoj brzini rada i širokom asortimanu, CPLD kola nalaze široku primenu, počev od implementacije jednostavne sprežne logike, do realizacije prototipova jednostavnijih ASIC kola. Važan razlog za sve veću primenu CPLD kola je redizajn postojećih sistema baziranih na SPLD kolima, gde se veći broj SPLD kola zamenjuje manjim brojem CPLD kola. Sistemi koji se sastoje od više međusobno povezanih funkcionalnih modula mogu se efikasno realizovati pomoću CPLD kola, tako što se svaki modul realizuje jedim logičkim blokom. Opšte pravilo je da su za realizaciju u CPLD tehnologiji pogodni sistemi koji zahtevaju složenu logiku sa malim brojem flipflopova. Dobar primer takve klase kola su konačni automati. Sva komercijalna CPLD kola su reprogramabilna, što omogućava jednostavnu i brzu izmenu dizajna. CPLD kola koja su reprogramabilna "u sistemu" omogućavaju rekonfiguraciju hardvera (npr. izmena protokola kod kola za komunikaciju) bez isključenja napajanja. Zahvaljujući strukturi sprežne mreže, tajming se može predvideti pre nego što je sistem realizovan, što predstavlja veoma bitnu prednost CPLD kola u odnosu na FPGA kola.

4.3. FGPA ARHITEKTURE

FGPA se izvodi kao polje velikog broja logičkih blokova (ćelija) jednostavne strukture zasnovanih na multiplekserima ili *look-up* tabelama sa malim brojem ulaza, koji kao takvi mogu da realizuju samo relativno jednostavne logičke funkcije, dok se složenije funkcije realizuju povezivanjem više logičkih blokova (Sl. 3). Logički blok se programira upisom konfiguracionih podataka (niza bitova). Najveći broj logičkih blokova sadrži i registre. FGPA se može programirati na tri različita nivoa: (1) na nivou funkcije logičkih ćelija; (2) na nivou sprege između ćelija; i (3) na nivou ulaza i izlaza. Sva tri nivoa se konfiguriraju preko niza bitova koji se preko eksternog izvora pune u kolo.

U zavisnosti od načina konfiguracije, razlikujemo četiri tipa FGPA kola. Konfigurabilna kola (prvi tip) se mogu konfigurirati od strane korisnika samo jedanput nakon čega dodatne promene konfiguracije nisu izvodljive. (Iz ovog razloga ova kola se nazivaju *one-time programmable*, ili OTP). Rekonfigurabilna kola se mogu konfigurirati nekoliko puta i dele se na dve klase: statička i dinamička. Kod statičkih kola (drugi tip) konfiguracioni niz se puni jedanput, na početku rada, i dok je kolo operativno ne menja se. Nakon resetiranja sistema u FGPA je moguće napuniti novi niz. Kod dinamički rekonfigurabilnih kola moguće je vršiti promene u konfiguraciji u bilo kom trenutku rada kola. Dinamički rekonfigurabilna kola se dalje mogu podeliti na potpuno rekonfigurabilna (treći tip), pri svakom novom punjenju rekonfiguriraju se celo kolo, i parcijalno rekonfigurabilna (četvrti tip) moguće je rekonfiguracija samo jednog dela kola.

4.3.1. Logički blokovi FGPA kola

Logički blokovi savremenih, komercijalnih FGPA kola se značajno međusobno razlikuju po veličini i implementacionoj moći. Dvo-tranzistorski logički blok FGPA kola firme Crosspoint može da realizuje samo jedan invertor, ali je zato veoma malih dimenzija. Sa druge strane, logički blok FGPA kola iz serije 4000 firme Xilinx sadrže tri tabele pretraživanja (*Look Up Table - LUT*) i može da realizuje bilo koju funkciju od pet promeljivih, ali je zato znatno složeniji. Da bi se istakle razlike u veličini i složenosti logičkih blokova, uvodi se pojam *granularnost logičkog bloka*, a sve savremene FGPA arhitekture se neformalno klasifikuju samo u dve široke grupe: logički blokovi fine granularnosti i logički blokovi grube granularnosti.

Logički blokovi fine granularnosti su veoma slični ćelijama gejtovskih matrica i tipično se sastoje od nekoliko tranzistora ili jednostavnih logičkih elemenata. Takvi blokovi poseduju malo interno propagaciono kašnjenje i s obzirom da zauzimaju malu površinu njihov broj na jednom čipu može biti veoma veliki. Jednostavna struktura logičkih blokova obezbeđuje fleksibilnost prilikom implementacije logičkih funkcija. Obično se data logička funkcija može realizovati na više različitih načina, što omogućava ostvarivanje željenog kompromisa između ukupne zauzete površine (tj. broja upotrebljenih logičkih blokova) i brzine rada. Međutim, što su logički blokovi jednostavniji to je neophodno obezbediti veći broj programabilnih veza u sprežnoj mreži. Prenos signala preko programabilnih veza je sporiji u poređenju sa prenosom unutar logičkog bloka. Tako, sa povećanjem "finoće" FGPA arhitekture, sprežna mreža zauzima proporcijalno sve veći deo ukupne površine čipa, a rad kola postaje sporiji.

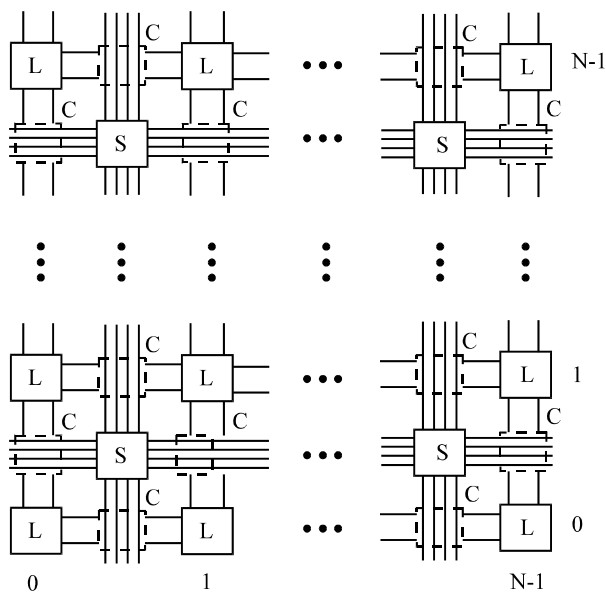
Logički blokovi grube granularnosti su najčešće zasnovani na LUT tabelama i tipično sadrže dodatne logičke elemente, kao što su multiplekseri i flipflopovi. Takvi logički blokovi mogu da realizuju složenije kombinacione i sekvencijalne logičke funkcije, pa su potrebe za povezivanje logičkih blokova programabilnim vezama manje. Složeni logički blokovi često, pored programabilnih struktura za realizaciju logike opšte namene (npr. LUT tabele), sadrže i logičke elemente za posebne namene, kao što su flipflopovi, generatori prenosa, SRAM blokovi. Kod nekih serija FGPA kola postoji mogućnost da se svaki logički blok nezavisno, po potrebi, postavi u jedan od nekoliko raspoloživih režima rada. U svakom od režima rada resursi logičkog bloka se koriste na drugačiji način. Kombinacija logičkih elemenata posebne namene i režima rada omogućava kompaktnu realizaciju logičkih struktura kao što su sabirači, komparatori, brojači. Međutim, složeni logički blokovi često su nepotpuno iskorišćeni, a ekonomičnost implementacije zavisi od nivoa usklađenosti strukture sistema koji se projektuje i FGPA arhitekture. To ima za posledicu da je logička gustina sistema realizovanih u FGPA arhitekturi grube granularnosti obično manja od one koja je deklarirana za konkretno FGPA kolo.

Granularnost logičkih blokova i tehnologija programiranja su u bliskoj vezi. Sprežne mreže realizovane na bazi *antifuse* tehnologije poseduju veću gustinu (broj programabilnih veza po jedinici površine) i manje propagaciono kašnjenje od sprežnih mreža kod kojih se koristi EPROM ili SRAM tehnologija programiranja. Zbog toga kod FGPA arhitekture fine granularnosti po pravilu se koristi *antifuse* tehnologija programiranja. Međutim, rekonfigurabilnost, koju nije moguće ostvariti pomoću *antifuse* tehnologije, je poželjna karakteristika FGPA arhitekture. Da bi smanjena gustina i povećano

propagaciono kašnjenje sprežnih mreža realizovanih na bazi EPROM ili SRAM tehnologije programiranja mogli da se tolerišu, logički blokovi trebaju biti složeniji. Zbog toga rekonfigurabilne FPGA arhitekture koriste logičke blokove grube granularnosti.

4.3.2. Sprežna mreža FPGA kola

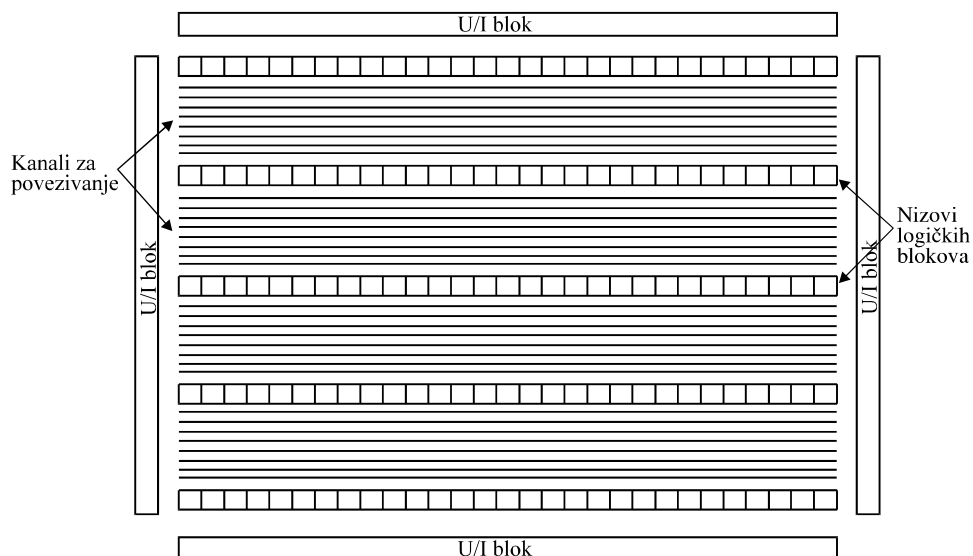
Sprežna mreža je bitna komponenta FPGA arhitekture. Na Sl. 15 je prikazan opšti model sprežne mreže jedne tipične FPGA arhitekture. Bazični elementi sprežne mreže su: žičani segmenti i programabilni prekidači. Programabilni prekidači su grupisani u blokove za povezivanje (*Connection Blocks - C*) i prekidačke blokove (*Switching Blocks - S*). Logički blok (označeni na Sl. 15 sa L) ima određeni broj priključaka sa svake strane i povezuje se sa sprežnom mrežom pomoću C bloka. Žičani segmenti prolaze neprekinuti kroz C blok, a svaki pin L bloka se može povezati sa fiksnim brojem žičanih segmenata koji prolaze kroz C blok. S blokovi omogućavaju spajanje horizontalnih i vertikalnih žičanih segmenata. Svaki žičani segment koji utiče u S blok može se povezati sa ograničenim brojem drugih žičanih segmenata koji sa druge strana utiču u isti S blok.



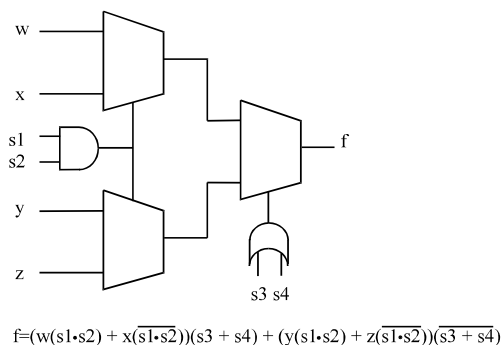
Sl. 15 Opšti model FPGA arhitekture.

4.3.3. Actel FPGA

Firma Actel u svom proizvodnom programu ima tri serije FPGA kola: Act 1, Act 2 i Act 3. Sve tri serije imaju istu bazičnu arhitekturu koja se sastoji od više nizova logičkih blokova razdvojenih horizontalnim kanalima za povezivanje (Sl. 16). Kanali za povezivanje su ispunjeni žičanim segmentima različitih dužina. U ovoj arhitekturi postoje i vertikalni žičani segmenti, ali je njihov broj daleko manji od horizontalnih. (Vertikalni žičani segmenti nisu prikazani na Sl. 16) Za realizaciju programabilnih prekidača koristi se *antifuse* tehnologija. Na Sl. 17 je prikazan logički blok iz serije Act 2. Logički blok je baziran na multiplekserima i ima osam ulaza i jedan izlaz. Mada logički blok ima relativno jednostavnu strukturu, dovođenjem na ulaze različitih kombinacija logičkih konstanti i promenljivih, samo pomoću jednog logičkog bloka moguće je realizovati 766 različitih logičkih funkcija. Upravo ovaj visok nivo funkcionalnosti ostvaren sa relativno malim brojem tranzistora predstavlja glavnu prednost logičkih blokova baziranih na multiplekserima. Međutim, funkcionalnost se postiže po cenu povećanja broja ulaza, zbog čega je neophodno da sprežna mreža bude složena.



Sl. 16 Actel FPGA



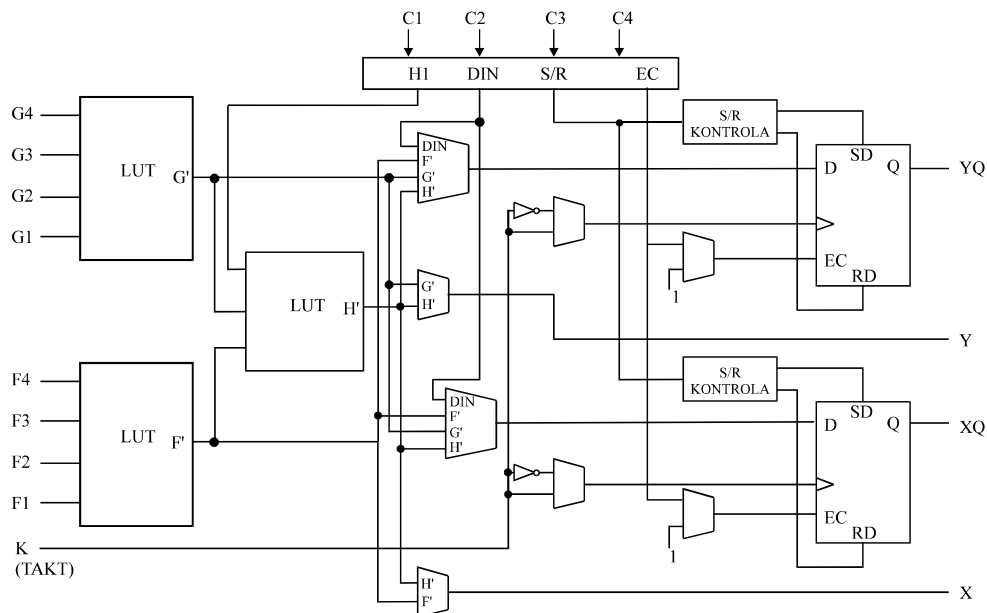
Sl. 17 ACT 2 logički blok.

Svaki ulaz logičkog bloka je u vezi sa posebnim vertikalnim žičanim segmentom, koji preseca sve horizontalne žičane segmente iz susjednog horizontalnog kanala. Izlaz logičkog bloka je spojen sa vertikalim žičanim segmentom koji preseca dva kanala iznad i dva kanala ispod logičkog bloka. Pored ovih, postoji i određeni broj neraspoređenih vertikalnih žičanih segmenti različitih dužina, koji se koriste za ostvarivanje globalnih veza. U svaki presek vertikalnog i horizontalnog žičanog segmenta ugrađen je programabilni prekidač. Horizontalni kanal se sastoji od 22 paralelne metalne veze koje su podeljene na određeni broj žičanih segmenata različitih dužina međusobno izolovanih *antifuse*-om (najkraći žičani segmenti su dužine dva logička bloka, a najduži dužine kanala). Programiranjem *antifuse*-ova mogu se formirati horizontalne veze željene dužine. Tajming Actel-ovih kola nije u potpunosti predvidljiv jer zavisi od toga na koji način će softver za povezivanje dodeliti žičane segmente u toku implementacije kola. Međutim, bogat izbor žičanih segmenata različitih dužina omogućava da se koristi algoritam koji će garantovati striktna ograničenja u broju *antifuse*-va koji se mogu koristiti za formiranje svake point-to-point veze.

4.3.4. Xilinx FPGA

Prva FPGA kola firme Xilinx, serija XC2000, su se pojavila na tržištu 1985. godine. Do danas Xilinx je razvio više novih serija od kojih su najznačajnije: XC3000, XC4000, XC5000, Spartan, Vertex. Sva Xilinx-ova FPGA kola u osnovi imaju istu arhitekturu, identičnu onoj prikazanoj na slici 4. Arhitekturu čine tri osnovna tipa gradivnih elementata: (1) logički blokovi (*Configuration Logic Blocks - CLB*); (2) ulazno-izlazni blokovi (*Input/Output Blocks - IOB*) i (3) sprežna mreža. CLB blokovi obezbeđuju funkcionalne elemente za realizaciju korisničkih funkcija, IOB blokovi ostvaruju interfejs

između pinova čipa i sprežne mreže, dok sprežna mreža obezbeđuje programabilnu vezu između CLB i IOB blokova. U nastavku će sa više detalja biti opisana arhitektura serije XC4000.



Sl. 18 XC4000 logički blok

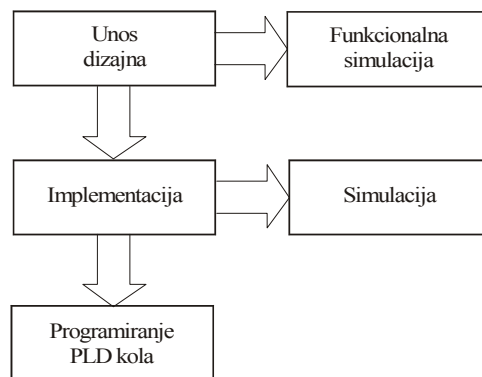
Struktura CLB blokova iz serije XC4000 je prikazana na Sl. 18. CLB blok ima 9 ulaza i 4 izlaz i može da realizuje kako kombinacione tako i sekvencijalne funkcije. Za realizaciju kombinacioni funkcija koriste se tabele pretraživanja (*Look-Up Table - LUT*). LUT tabele se realizuju kao SRAM memorije kapaciteta $2^K \times 1$. Promenljive se dovode na adresne ulaze memorije, a na izlazu memorije se generiše vrednost funkcije za datu kombinaciju ulaznih promenljivih. Direktnim upisom tabele istinitosti u memoriju, korisnik može da realizuje bilo koju kombinacionu funkciju od K promenljivih, dok se složenije funkcije mogu realizovati povezivanjem više LUT tabela. CLB blokovi iz serije XC4000 sadrže tri LUT tabele, dve četvoroulazne (F i G) i jednu troulaznu (H). Promenljive za LUT tabele F i G se dovode sa odgovarajućih ulaza CLB bloka, a njihovi izlazi se direktno povezuju sa ulazima LUT tabele H, kako je to prikazano na Sl. 18. Takva organizacija omogućava da se u CLB bloku može formirati širok spektar kombinacionih funkcija. Neke od varijanti su: bilo koje dve funkcije od 4 promenljive, jedna bilo koja funkcija od 5 promenljivih, jedna bilo koja funkcija od 4 promenljive zajedno sa još jednom funkcijom od 5 promenljivih, neke od funkcija od 9 promenljivih, i td. Mogućnost realizacije kombinacionih funkcija sa većim brojem promenljivih u jednom CLB bloku je značajna, jer se tako smanjuje broj iskorišćenih CLB blokova i ukupno propagaciono kašnjenje.

LUT tabelama F i G je pridružena logika za brzo generisanje prenosa i pozajemice (*Carry Logic*), što bitno poboljšava performanse aritmetičkih kola kao što su: sabrač, odzimači i komparatori. *Carry Logic* (nije prikazana na Sl. 18) se aktivira odgovarajućom konfiguracijom CLB bloka. U takvoj konfiguraciji, LUT tabele F i G funkcionišu kao 2-bitni sabirači sa ugrađenim generatorom paralelnog prenosa. Povezivanjem više CLB blokova moguće je formirati sabirače sa većim brojem bitova, bez značajnog povećanja propagacionog kašnjenja. Takođe, svaki CLB blok se može konfigurisati kao SRAM blok kapaciteta 16×2 ili 32×1 . Dostupnost RAM memorije u samom FPGA kolu omogućava ekonomičnu realizaciju memorijskih struktura kao što su registarska polja, FIFO baferi, LIFO stekovi i sl..

Za realizaciju sekvencijalnih funkcija, CLB blok sadrži dva D flipflop sa zajedničkim signalom takta, K , i signalom za dozvolu taktovanja, EC . Treći zajednički signal je S/R koji može biti programiran nezavisno za svaki flipflop, kao signal za asinhrono setovanje ili resetovanje. Ulazni podaci flipflopova se formiraju preko multipleksera i to mogu biti ili podaci sa ulaza CLB bloka ili izlazne vrednosti LUT tabela.

5. PROJEKTOVANJE POMOĆU PLD

Na današnjem nivou tehnologije PLD-ovi su postali kompleksna kola koja su u stanju da implementiraju veoma složene logičke funkcije. U fazi projektovanja složenih digitalnih kola i sistema koriste se brojne tehnike i sredstva za automatsko projektovanje. CAD alati, jezici za opis hardvera (HDL), kao i specijalizovani softveri za logičku sintezu, logičku simulaciju, strukturnu analizu i projektovanje su od ključne važnosti za efikasno projektovanje digitalnih sistema. Faze projektovanja pomoću PLD kola prikazane su na Sl. 19. U početnoj fazi, projektant opisuje dizajn (tj. unosi dizajn) korišćenjem neke od standardnih metoda (šematski opis, HDL,...). Nakon izvršenog unosa dizajna, a u cilju verifikacije logičke ispravnosti dizajna obavlja se funkcionalna (ili logička) simulacija. U sledećoj fazi vrši se implementacija dizajna, tako što se opis dizajna na visokom nivou preslikava na strukturu ciljnog PLD kola. Ovo preslikavanje obuhvata: (1) logičku sintezu (konvertovanje opisa dizajna u logičke jednačine i njihovu minimizaciju), (2) prilagođenje ciljnoj PLD arhitekturi (*technology mapping*) - realizacija logičkih jednačina uz pomoć raspoloživih makro-čelija (programabilne struktura kao što su *look-up* tabele, multiplekseri, CPLD logički blokovi); (3) razmeštaj (*placement*) - dodela makro-čelija fizičkim logičkim blokovima PLD kola; (4) povezivanje (*routing*). Rezultat svih ovih aktivnosti je implementacioni opis dizajna koji sadrže sve detalji vezane za internu implementaciju dizajna. Korišćenjem implementacionog opisa, moguće izvršiti veoma preciznu simulaciju kako u pogledu funkcionalnosti tako i u pogledu tajminga. U poslednjoj fazi, na bazi implementacionog opisa dizajna generiše se konfiguracioni niz koji se programira u PLD kolo.



Sl. 19 Faze projektovanja pomoću PLD kola.

5.1. UNOS DIZAJNA

Za unos dizajna PLD kola (vidi sliku 1.9) koriste se četiri metode od kojih su tri tekstualno zasnovane metode ("sagorljivi osigurač", HDL i VHDL) i jedna grafička (unos šeme kola - *schematic capture*). Neka od razvojnih okruženja kombinuje dva ili tri različita metoda projektovanja, pri čemu se za unos odgovarajućeg gradivnog bloka najčešće bira onaj metod koji je najadekvatniji.

Programiranje zasnovano na sagorevanju osigurača

Svako kolo tipa SPLD, CPLD i FPGA ima određeni broj programabilnih tačaka pomoću kojih se može implementirati pojedina logička funkcija. Sve metode unošenja dizajna u ovom slučaju, rezultiraju u opis pomoću koga se ukazuje koje od tačaka ostvaruju vezu, a koje ne. Opis mape prekidača kod SPLD-ova je standardizovan i naziva se JEDEC standard; to je tekstualni metod za opis programabilnih tačaka koji naznačava da li su one izvedene kao kratki spojevi ili pregoreli osigurači. Ovaj postupak je dugotrajan i podložan velikom broju grešaka, a primenljiv je samo kod SPLD kola. CPLD/FPGA kola se odlikuju složenom arhitekturom i poseduju veoma veliki broj interno-programabilnih tačaka (tipično, od nekoliko desetina hiljada do nekoliko miliona). Iz tog razloga unos dizajna na nivou direktne specifikacije stanja programabilnih tačaka nije praktično izvodljiv.

Unos dizajna pomoću HDL-a

HDL (*Hardware Description Language* - jezik za opis hardvera) omogućava projektantu da dizajn opiše na simbolički način, u vidu sistema logičkih funkcija, poštujući pri tome sintaksu konkretnog HDL jezika. Nakon toga, HDL obavlja automatsku konverziju simboličkog opisa u oblik mape pregornjivih osigurača za dato PLD kolo. Prve verzije HDL jezika su bile jako jednostavne, a osnovno što su obezbeđivali bila je konverzija *Bool*-ovih jednačina koje su opisivale implementaciju logičkog

sistema, u korektnu mapu osigurača. Savremeni HDL jezici obezbeđuju viši nivo apstraktnosti opsa, mada treba naglasiti da svaki proizvođač PLD kola ima svoj HDL kompilator. Primer HDL opisa 4-bitnog binarnog brojača:

```
brojac_0 = / brojac_0 ;
brojac_1 = / brojac_1 % brojac_0 ;
brojac_2 = / brojac_2 % (brojac_1 * brojac_0);
brojac_3 = / brojac_3 % (brojac_2 * brojac_1 * brojac_0);
```

gde “/” predstavlja logičku NOT operaciju, “%” predstavlja logičku XOR, a “*” logičku AND operaciju. Bitske promenljive brojac_0 do brojac_3 odgovaraju filip-flopovima brojača, dok četiri jednačine opisuju logičke funkcije koje definišu nova stanja flip-floпова u zavisnosti od tekućeg stanja brojača.

Šematski unos dizajna

Napredak PLD tehnologije, a posebno fabrikacije FPGA kola, uslovio je da se na tržištu pojave softverska sredstva za šematski unos dizajna koja se najčešće koriste kod projektovanja ASIC kola. Šematski unos ima određene prednosti koje se ogledaju u sledećem: Efikasno podržavaju hijerarhijske strukture što znači da se *top-down* projektovanje može uspešno primeniti (koriste se šeme na nivou podsistema). Korišćenje makroa, u obliku standardnih logičkih funkcija kakvi su brojači, flip-floповi, multiplekseri, itd. čini da šematski unos bude zaista moćan pristup.

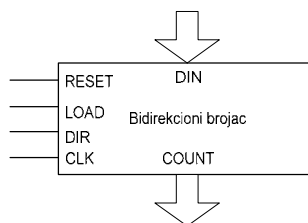
Unos dizajna na nivou VHDL-a

VHDL je HDL visokog-nivoa koji je namenjen za opis i simulaciju hardverskih sistema. Kao IEEE standard postao je široko prihvaćen od strane projekatana logičkih sistema jer je nezavisan od proizvođača PLD kola. Ova mogućnost obezbeđuje preusmeravanje dizajna na različite CPLD/FPGA proizvođače (preusmeravanje nije uvek direktno (automatsko) jer za svako FPGA kolo koriste se specifična biblioteka kola). Glavna novina VHDL-a u odnosu na HDL je mogućnost *behavioral* (algoritamskog) opisa dizajna. *Behavioral* opis 4-bitnog brojača je oblika:

```
if(brojac = 15) brojac = 0 ; else brojac = brojac + 1;
```

gde se '+' odnosi na aritmetičku operaciju sabiranja. U ovom slučaju, logičke funkcije nisu eksplicitno navedene, već sam HDL kompajler ima sposobnost da prepozna operaciju sabiranja i automatski generiše odgovarajuću logiku. VHDL omogućava da se na lak način, opišu digitalna kola složenog ponašanja. Ispod je prikazana pogradska sekvenca u VHDL-u koja opisuje sinhroni bidirekcionni brojač sa asinhronim resetom i sinronim paralelnim upisom. COUNT je izlaz brojača, a DIN paralelni ulaz. Smer brojanja se reguliše signalom DIR, RESET je signal za resetovanje, a LOAD za paralelni upis.

```
process (CLK, RESET)
begin
if RESET='1' then
COUNT <= "0000";
elsif CLK='1' and CLK'event then
if CE='1' then
if LOAD='1' then
COUNT <= DIN;
else
if DIR='1' then
COUNT <= COUNT + 1;
else
COUNT <= COUNT - 1;
end if;
end if;
end if;
end if;
end process;
```



5.2. SINTEZA

Sinteza predstavlja proces pomoću koga se uneti dizajn konvertuje u specifičnu implementaciju na gejtovskom nivou. Ovaj proces uključuje mapiranje (uklapanje) logičke funkcije na makro-ćelije ili u

logičke blokove CPLD/FPGA kola, a zatim rutiranje svih veza. U suštini proces sinteze čine sledeća dva koraka: (a) mapiranje logike na makro-čelije; i (b) rutiranje veza između makro-čelija.

Kod najvećeg broja CPLD/FPGA-ova postoje dva tipa makro-čelija: U/I makro-čelije i skrivene (*buried*) makro-čelije. U/I makro-čelije imaju direktnu fizičku vezu sa spoljnim pinovima CPLD/FPGA kola, dok su skrivene makro-čelije one koje nemaju fizičku vezu sa U/I pinovima. Iz razloga što se između broja U/I pinova sa jedne, logičkog kapaciteta sa druge, i implementiranih logičkih funkcija sa treće strane ne može uvek naći dobar kompromis, iskorišćenost dostupnih resursa CPLD/FPGA-a kod većine praktičnih dizajna nije veća od 75%. Druge osobine koje ima uticaj na maksimalno iskorišćenje čipa predstavljaju rutiranje puteva i razmeštaj čelija u okviru kola. Postoje tri metodologije za rutiranje i razmeštaj: 1) automatski; (2) ručni; (3) interaktivni. Automatsko razmeštanje i rutiranje u potpunosti obavlja specijalizovani softver. Ovakav pristup je dobar izbor kada projektant želi da se oslobodi obavljanja tih zadataka, ali ima nedostatak što je iskorišćenost CPLD/FPGA niska (obično oko 50%). Visoka iskorišćenost CPLD/FPGA-ovih resursa se ostvaruje kada se razmeštaj i rutiranje izvode ručno (kreće se oko 95%), ali se za sprovođenje ovog metoda zahteva dosta dug vremenski period kao i dobra obučenost i snalažljivost projektanta. Kod interaktivnog metoda za deo dizajna (85% do 90%) rutiranje i razmeštaj se vrši prvo automatski, a zatim se ostatak kompletira ručno.

U sledećoj tabeli prikazan je odnos potrebnog vremena da se obavi rutiranje i razmeštaj kod sva tri metoda za jedan prosečan dizajn.

Metod razmeštaja i rutiranja	Vremenski period	Iskorišćenost CPLD/FPGA resursa
automatski	2-3 časa	50%
interaktivni	1 dan	75%
ručni	1 nedelja	95%

5.3. SIMULACIJA

Zadatak simulatora je sledeći: Obavi verifikaciju dizajna, tj. proveri da li se za specificiranu pobudu (ulaz) dobija očekivani odziv (izlaz), ali da se pri tome ne izvrši programiranje PLD kola. Simulacijom se mogu detektovati greške u dizajnu, proceniti karakteristike PLD kola koja se odnose na kašnjenje signala, potrošnju energije, maksimalnu radnu frekvenciju i dr., tj. proceniti karakteristike koje je veoma teško odrediti analitički. Tačnost simulacije zavisi od tačnosti modela koji se koristi za opis sastavnih delova PLD kola.

Simulacija se može sprovesti na: (a) funkcionalnom nivou; (b) gejt nivou.

Funkcionalna simulacija tiče se implementacije na visokom nivou, kakvi su VHDL opis ili šematski prikaz, a koristi se da verifikuje logički rad dizajna. Simulacija se obavlja softverski pri čemu simulator "radi" shodno zadatoj logičkoj specifikaciji, a putem interakcije sa projektantom omogućava otkrivanje, identifikaciju i ispravljanje grešaka u dizajnu. Nakon što je proces funkcionalne simulacije verifikovao da dizajn obavlja željenu funkciju, sledi sinteza dizajna za konkretnu programabilnu arhitekturu (specificira se proizvođač i tip CPLD/FPGA komponente pomoću koje želimo da implementiramo logičku funkciju). U toku sinteze vrši se konverzija logičke specifikacije u gejt-nivovski ekvivalenat uz obavezno rutiranje i razmeštaj. Kao posledica ne baš uvek korektnih i tačnih modela komponenata sa kojima se manipuliše u toku procesa sinteze, proces sinteze implementiran na gejt-nivou može da generiše rešenje dizajna koje nema identičnu funkcionalnost kao i rešenje dizajna na funkcionalnom nivou. Zbog ovog neslaganja proces simulacije mora da se ponovi, ali sada na gejt-nivou. U principu, realno je očekivati da razlike između simulacija na funkcionalnom i gejt-nivou ne budu tako drastrične. Ipak, te razlike su izgleda neminovne, a javljaju se pre svega zbog nepredvidljivih vremenskih kašnjenja signala uzrokovanih razmeštajem makro-čelija i rutiranja signala kroz PLD kolo. Obično, korisnik (projektant) interveniše manuelno na nivou gejt-*layout*-a da bi korigovao (usklađio) te vremenske razlike. U slučaju kada korekcije, prvenstveno zbog visokog stepena iskorišćenja gejtova, nisu moguće, kao izlaz iz ove situacije se koriste sledeća dva rešenja: (a) bira se CPLD/FPGA kolo većeg kapaciteta, a postupak sinteze sprovodi iznova; i (b) vrši redizajn na funkcionalnom nivou, a time se na indirektni način sugerise na korišćenje alternativne logike za sintezu dizajna (opisuje rad kola na jedan alternativan način).