

PROGRAMIRANJE KOMUNIKACIONOG HARDVERA
– Poglavlje 1 –

1 Programabilni čipovi

Sa razvojem integriranih čipova došlo je do pojave velikog broja čipova specijalizovanih za obavljanje pojedinih funkcija poput – dekodera, multipleksera, brojača, pomeračkih registara itd. Lista osnovnih čipova iz popularne serije 7400 može da se pogleda na [wikipediji](#). Međutim, često je bilo potrebno implementirati i funkcije koje nisu bile pokrivenne osnovnim integriranim čipovima koji su bili na raspolaganju. S jedne strane nije bilo isplativo proširivati skup osnovnih čipova sa funkcionalnostima koje se ne koriste često i koje ne bi koristio veliki broj korisnika, a s druge strane trebalo je omogućiti korisnicima da za svoje potrebe mogu lako razviti i implementirati specifične funkcije koje nisu pokrivenne osnovnim skupom. Iz ovih oprečnih aspekata izrodila se ideja da se razvijaju tzv. programabilni čipovi čiju funkciju bi definisao korisnik. U prevodu, ovi čipovi nemaju namenu (funkciju) sve dok ih sam korisnik ne programira da obavljaju određenu funkciju. Programabilni čipovi su omogućili veću fleksibilnost korisnicima koji su sada mogli da definišu i svoje specifične funkcije, a opet proizvođačima se isplatila proizvodnja takvih čipova jer je baza korisnika koji bi kupovali te čipove ostala i dalje velika kao i kod integriranih čipova osnovnog skupa funkcionalnosti. Programabilni čipovi su se razvijali i unapređivali svoje mogućnosti i performanse kako se i sama tehnologija razvijala, tako da postoji više tipova programabilnih čipova, počevši od prvobitnih PAL (*Programmable Array Logic*) i PLA (*Programmable Logic Array*) čipova razvijenih 70-tih godina prošlog veka do današnjih veoma moćnih FPGA (*Field-Programmable Gate Array*) čipova.

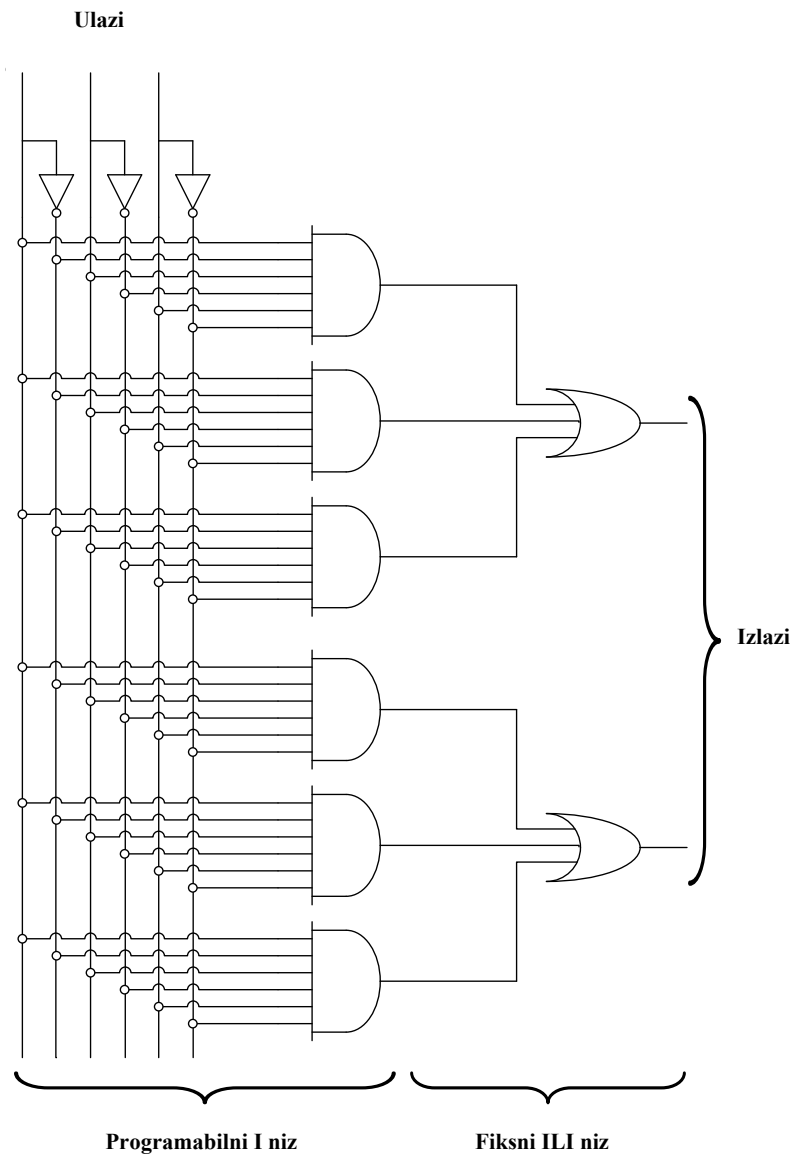
1.1 PAL

Struktura PAL čipa je prikazana na slici 1.1.1. PAL čip se sastoji iz programabilnog niza I kola za kojim sledi fiksni niz ILI kola. Ideja na kojoj se zasniva ova struktura je da se svaka kombinaciona funkcija može predstaviti kao suma proizvoda¹. Naime, kombinaciona funkcija je ona čija vrednost izlaza zavisi samo od trenutnih vrednosti ulaza. Otuda, za kombinacionu funkciju može da se napravi tabela istinitosti, a na osnovu tabele istinitosti može se kreirati predstava kombinacione funkcije kao suma proizvoda ili proizvod suma. Kao primer možemo uzeti dvoulazni multiplekser čija je tabela istinitosti data u tabeli 1.1.1. Podsetimo se da je uloga multipleksera da se na izlaz Y preslika jedan od ulaza X_i , pri čemu ulazni signali za selekciju S_i određuju koji od ulaza će biti preslikan na izlaz.

Tabela 1.1.1. – Tabela istinitosti za dvoulazni multiplekser

S_0	X_0	X_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

¹ U Prilogu ovih skripti su izloženi i definisani osnovni pojmovi poput kombinacione i sekvencijalne logike, kao i osnovnih digitalnih komponenti poput brojača, multipleksera, kodera, dekodera i dr. Studenti bi trebalo da poznaju ove pojmove na osnovu predmeta iz ranijih godina studiranja. Međutim, u prilogu su ipak izloženi osnovni pojmovi kako bi čitaoci koji nisu upoznati sa njima mogli lakše da prate ove skripte.

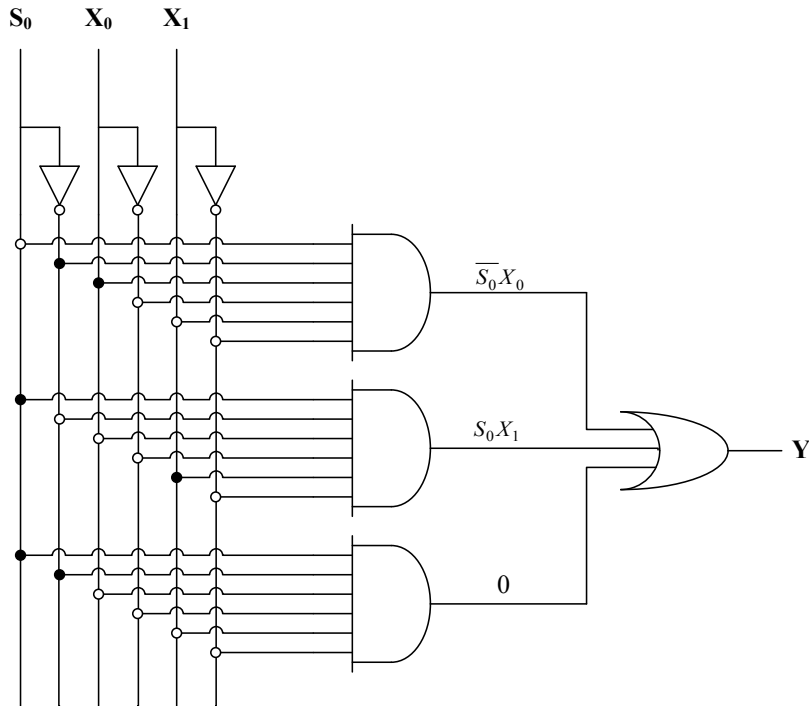


Slika 1.1.1. – PAL arhitektura

U slučaju dvoulaznog multipleksera se na izlaz Y prosleđuje signal X_0 ili X_1 u zavisnosti od vrednosti selektora S_0 kao što se vidi iz tabele 1.1.1. Na osnovu tabele istinitosti možemo da napišemo izraz za Y u obliku kompletne sume proizvoda koju zatim možemo uprostiti:

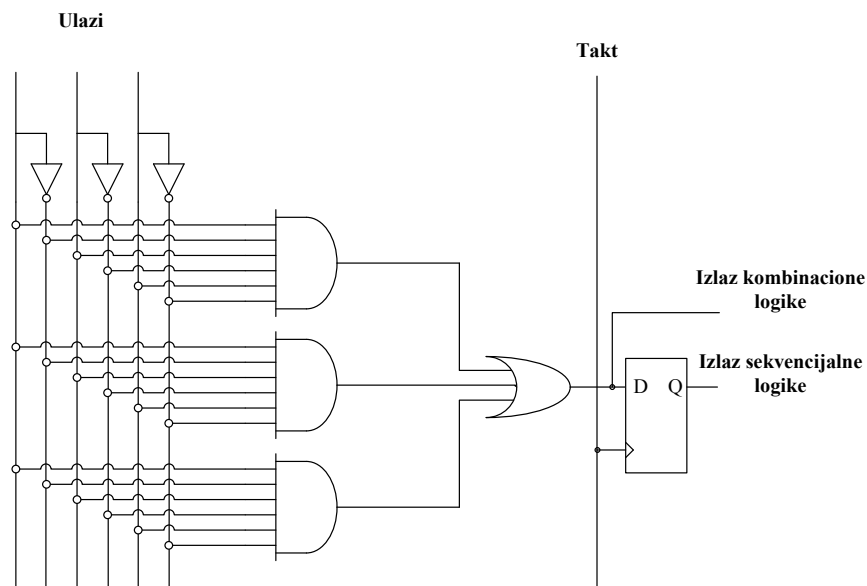
$$Y = \overline{S_0}X_0\overline{X_1} + \overline{S_0}X_0X_1 + S_0\overline{X_0}X_1 + S_0X_0X_1 = \overline{S_0}X_0 + S_0X_1$$

Poznavajući sumu proizvoda može se programirati PAL čip da izvrši funkciju dvoulaznog multipleksera i na slici 1.1.2 je prikazan deo PAL čipa koji je izvršava. Sa crnim kružićima je označena uspostavljena veza između vertikalnih i horizontalnih linija. U slučaju da nije uspostavljena veza (prazan kružić) tada je na ulazu odgovarajućeg I kola prisutna logička '1' kao neutralan element za I operaciju. Otuda se na poslednje I kolo povezuju i S_0 signal i njegov komplement jer tada na izlazu I kola imamo sigurno logičku '0' koja je neutralan element za Ili operaciju (naravno, isto bi važilo da smo tako povezali X_0 ili X_1 signal).



Slika 1.1.2. – Konfiguracija PAL čipa za obavljanje funkcije dvoulaznog multipleksera

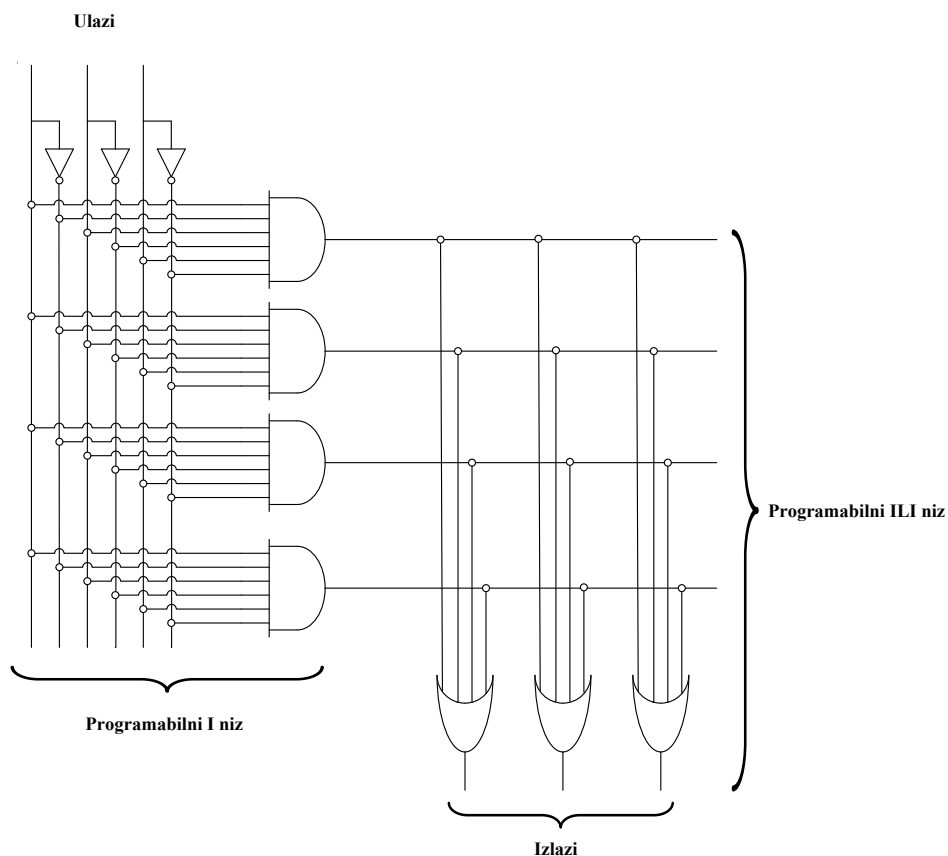
Kao što se vidi sa slike 1.1.1 PAL čip nema registarskih elemenata tj. nema memorije tako da može da se koristi samo za kreiranje kombinacionih funkcija. Iz tog razloga je u pojedinim PAL čipovima dodat D-FF na svakom izlazu (tačnije izlaz fiksnog ILI niza je vođen na ulaz D-FF) čime je omogućeno da se u PAL čipovima mogu implementirati i jednostavne sekvencijalne funkcije. Takvi čipovi su označavani kao PAL čipovi sa registrima na izlazima (*registered PALs*). Primer dela takvog čipa je dat na slici 1.1.3, pri čemu je uzet kao primer slučaj gde postoje dva izlaza - izlaz sekvencijalne logike koji prolazi kroz flip-flop i izlaz kombinacione logike koji ne prolazi kroz flip-flop.



Slika 1.1.3. – Arhitektura PAL čipa koji podržava i sekvencijalnu logiku

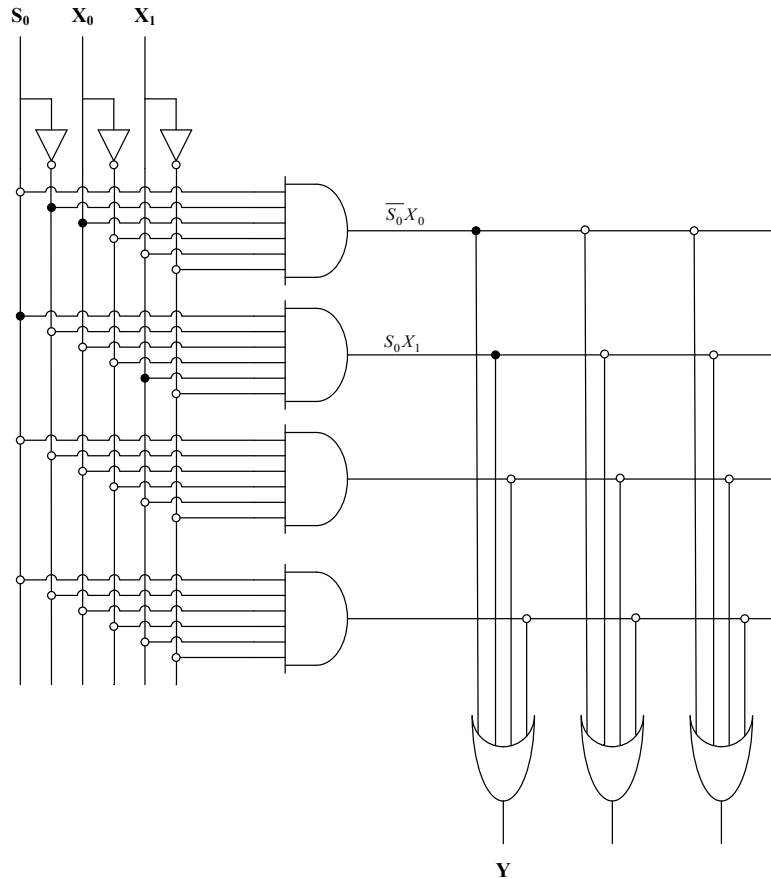
1.2 PLA

PLA čipovi su uneli dodatni stepen fleksibilnosti u odnosu na PAL čipove. Naime, kod PLA čipova je umesto fiksnog ILI niza korišćen programabilni ILI niz čime je postignut veći stepen fleksibilnosti u dizajnu. Međutim, to je povećalo kašnjenje kroz čip pa je maksimalna frekvencija na kojoj je dizajn mogao raditi bila nešto niža u odnosu na PAL čipove. PAL i PLA čipovi su bili dostupni na tržištu od sredine 70-tih godina prošlog veka, a danas se više ne proizvode jer su ih zamenili tehnološki moćniji čipovi kao što ćemo videti u narednim sekcijama. Arhitektura PLA čipa je prikazana na slici 1.2.1. Očigledno je da i PLA čipovi podržavaju samo kombinacione funkcije jer ne sadrže registerske elemente.



Slika 1.2.1. – Arhitektura PLA čipa

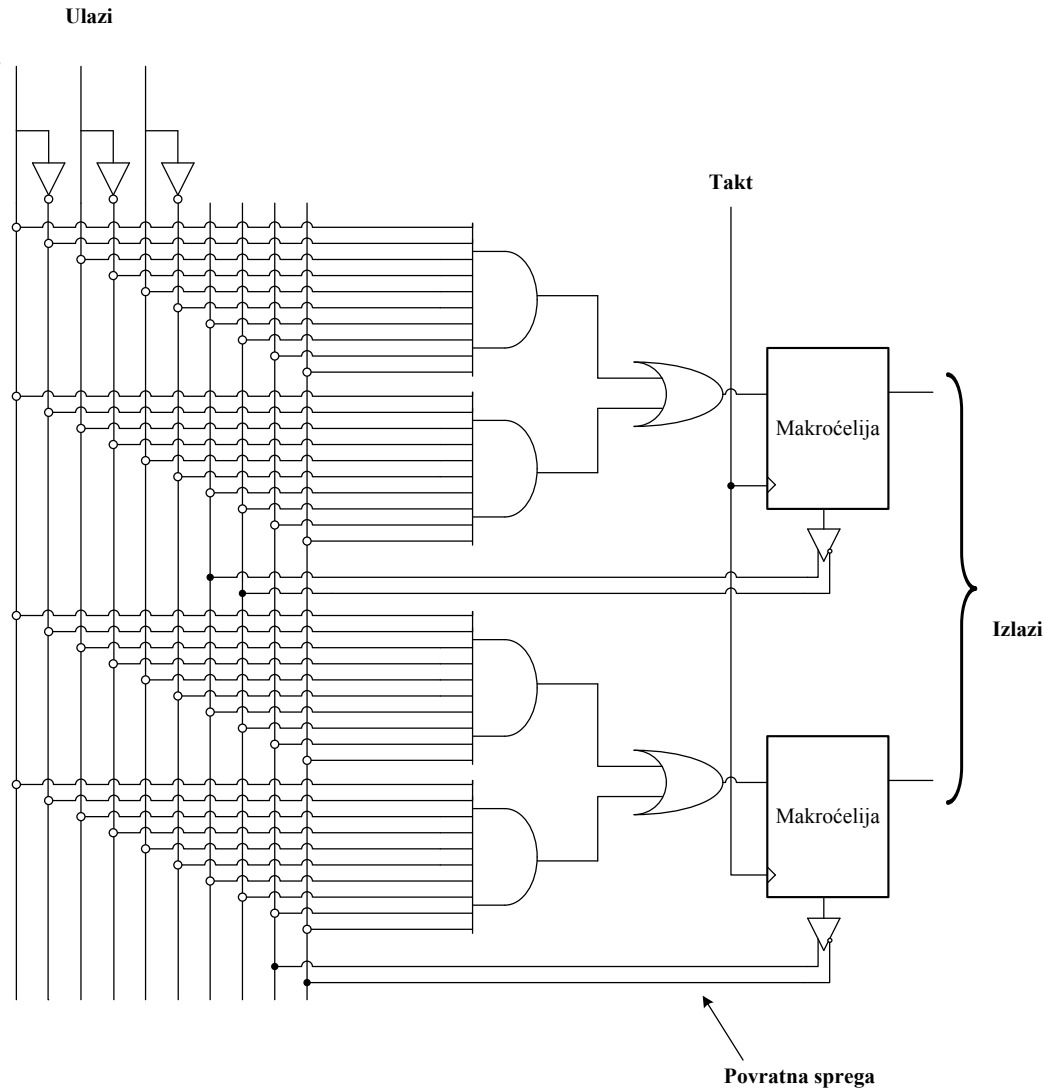
Primer programiranog dela čipa koji obavlja funkciju dvoulaznog multipleksera je prikazan na slici 1.2.2. Ukoliko ulaz u I kolo nije povezan na spoljašnji ulaz (nema spoja između vertikalne i horizontalne linije u I ravni tj. programabilnom I nizu) tada je on za I kolo neutralan element tj. ima vrednost logičke '1'. Slično važi i za ILI ravan tj. programabilan ILI niz, gde nepovezan ulaz u ILI kolo ima vrednost logičke '0' tj. neutralnog elementa za ILI kolo. Može se lako uočiti da je PLA čip fleksibilniji nego PAL čip. Naime, u PLA čipu izlaz svakog I kola se može povezati sa svakim ILI kolom, za razliku od PAL čipa. Tako je u primeru sa slike 1.1.2 jedno I kolo (I kolo na čijem je izlazu '0') praktično neiskorišćeno u PAL čipu prilikom realizacije dvoulaznog multipleksera, a u PLA čipu bi to I kolo moglo da se programira i iskoristi za neku drugu kombinacionu funkciju koja bi radila uporedo sa realizovanim dvoulaznim multiplekserom.



Slika 1.2.2. – Konfiguracija PLA čipa za obavljanje funkcije dvoulaznog multipleksera

1.3 GAL

GAL (*Generic PAL*) čipovi su se pojavili početkom osamdesetih godina prošlog veka. Ovi čipovi su doneli nekoliko unapređenja u odnosu na PAL i PLA čipove. Uvedene su programabilne makročelije koje su sadržale flip-flop, kao i logička kola i multipleksere, čime je omogućena efikasnija implementacija kombinacione i sekvencijalne logike. Pored toga, uvedena je povratna sprega tako što se jedan od izlaza iz makročelije vraćao nazad u čip i time mogao da se vodi na ulaze drugih makročelija. Takva povratna sprega je omogućavala i kreiranje složenijih kombinacionih i sekvencijalnih kola. U principu, povratna sprega je bila moguća i kod PAL i PLA čipova, ali samo ako se izlaz čipa povezivao sa ulazom čipa, što je očigledno manje efikasno, pre svega zbog uvođenja većeg kašnjenja, trošenja ulaznih i izlaznih pinova čipa za formiranje povratne sprege, kao i veće potrošnje snage. GAL čipovi se označavaju i kao SPLD (*Simple Programmable Logic Device*) čipovi. Ovi čipovi se uglavnom više ne proizvode, već se koriste kao osnovne komponente pri izgradnji CPLD (*Complex Programmable Logic Device*) čipova o kojima će biti reči u sledećoj sekciji. Na slici 1.3.1 je prikazana arhitektura GAL čipa. Može se videti velika sličnost sa arhitekturom PAL čipa sa slike 1.1.1, pošto i ovde postoji programabilni I niz i fiksni ILI niz (otuda i naziv *Generic PAL*), ali u GAL arhitekturi možemo videti da postoji i povratna sprega kao i programabilne makročelije koje omogućavaju projektovanje složenijih logičkih kola, odnosno funkcija.

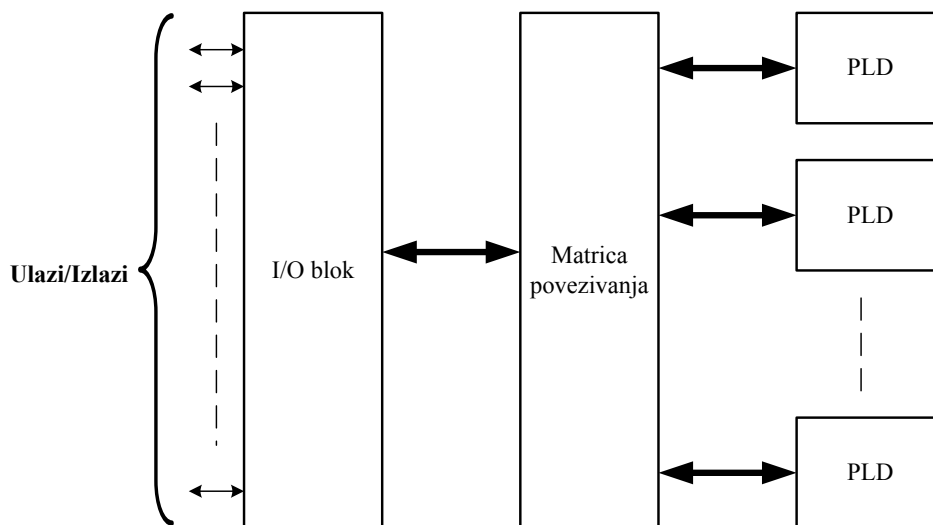


Slika 1.3.1. – Arhitektura GAL čipa

1.4 CPLD

Sa daljim razvojem tehnologije omogućena je proizvodnja složenijih čipova. Prva ideja je bila kreiranje složenog programabilnog čipa koji bi predstavljao proširenje prethodnih generacija čipova tako što bi se unutar složenog čipa napravila mreža komponenti (PLD-ova) koje bi bile zasnovane na arhitekturi prethodnih generacija čipova. Otuda i naziv CPLD (*Complex PLD*) kojim se označava da ovi čipovi predstavljaju mrežu PLD čipova (ili preciznije blokova). Generalna arhitektura CPLD čipa je data na slici 1.4.1. Kao što se vidi, CPLD čip sadrži više PLD blokova. PLD blokovi su tipično zasnovani na GAL arhitekturi, ali u pojedinim familijama CPLD čipova kompanije Xilinx, PLD blokovi su zasnovani PLA arhitekturi. PLD blokovi se međusobno mogu povezivati preko matrice povezivanja čime se omogućava formiranje veoma složenih logičkih kola. Takođe, matrica povezivanja omogućuje vezu između ulazno/izlaznog dela (I/O bloka) i PLD blokova. I/O blok u stvari sadrži pinove kojima se CPLD čip povezuje sa okolinom, pa stoga I/O blok definiše ulazno/izlazne interfejsse. Sami PLD blokovi su

programabilni, kao i matrica povezivanja, čime se omogućava kreiranje veoma složenih logičkih kola, a i isto tako na jednom čipu se može implementirati i veći broj međusobno nezavisnih složenih logičkih kola. Moderni CPLD čipovi imaju i niz dodatnih mogućnosti među kojima su podrška za veći broj različitih tehnologija I/O interfejsa (1.5V (LVCMOS15, HSTL_1), 1.8V (LVCMOS18), 2.5V (LVCMOS25, SSTL2_1), 3.3V (LVTTTL, LVCMOS33, SSTL3_1)), i podrška za JTAG (*Joint Test Action Group*) kojim se omogućava programiranje čipa, ali i testiranje dizajna na čipu tako što se prate vrednosti unutrašnjih signala u čipu tokom rada samog čipa.

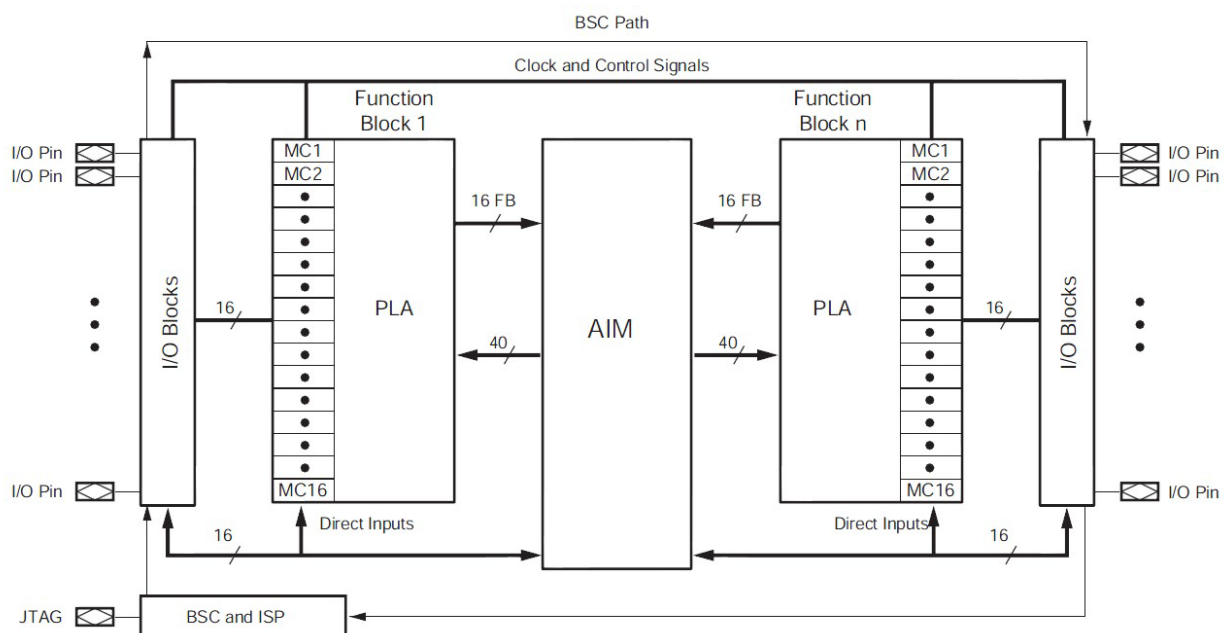


Slika 1.4.1. – Arhitektura CPLD čipa

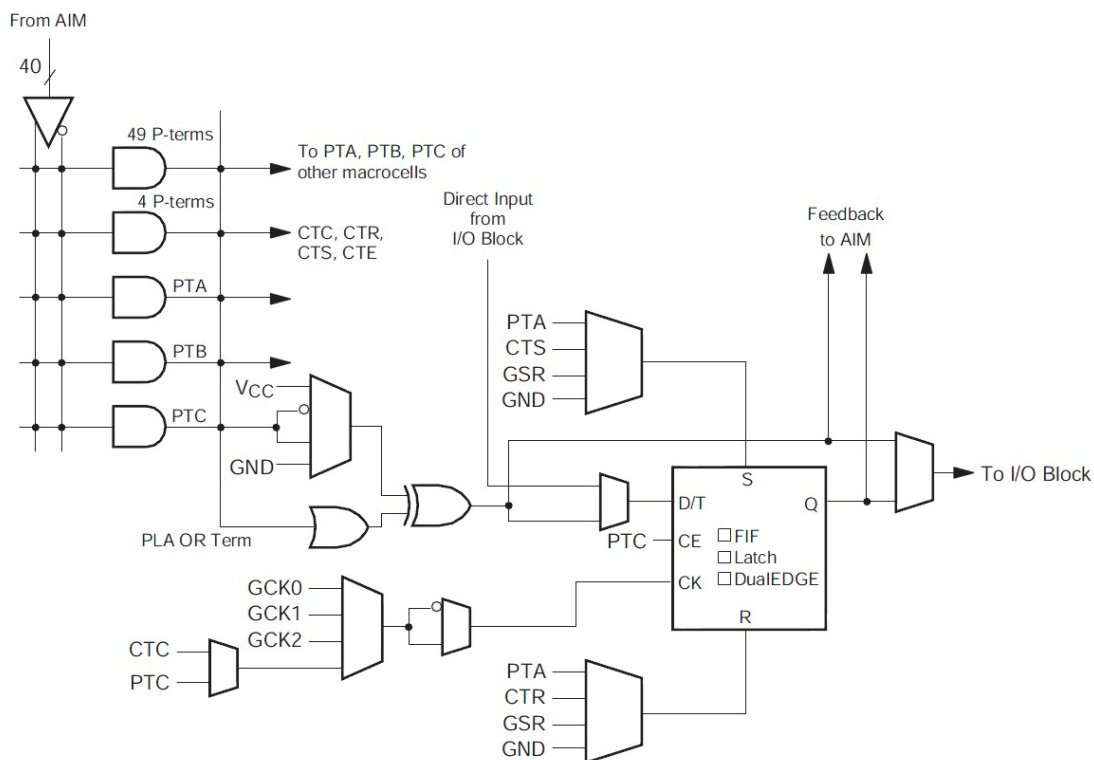
Očigledno, jedan od problema koji se sada javlja jeste kako optimalno postaviti dizajn logičkog kola na sam čip, odnosno kako distribuirati delove logičkog kola po PLD blokovima, a da se dobiju optimalne performanse koje se mogu ogledati u minimalnom kašnjenju kroz realizovano logičko kolo, minimalnom broju zauzetih PLD-ova ili maksimalnoj podržanoj frekvenciji realizovanog kola. Veoma često su zahtevi u pogledu optimalnih performansi i oprečni, pa tako ako se postigne minimalno kašnjenje neretko se zauzmu veći resursi čipa i obrnuto. Stoga se često vrši balansiranje kriterijuma (zahteva) koje znači da se kao optimalno rešenje uzima kompromisno rešenje između oprečnih zahteva. Problem postavljanja dizajna na čip postaje još složeniji kada se na čip postavlja nekoliko nezavisnih složenih logičkih kola, jer treba odrediti optimalno rešenje za sva logička kola, a pri tome se ona postavljaju na isti čip tj. konkurišu za iste resurse.

Performanse CPLD čipova iz familije CoolRunner-II poznatog proizvođača Xilinx se mogu videti u [pdf fajlu](#) dostupnom i na njihovom sajtu www.xilinx.com. Detaljnije informacije o CoolRunner-II familiji se mogu naći u odgovarajućem [pdf fajlu](#), takođe dostupnom na navedenom sajtu. Najmanji čip iz familije CoolRunner-II ima svega 32 makroćelije, dok najveći čip iz te familije ima 512 makroćelija. Broj (ulaznih/izlaznih) pinova koji su na raspolaganju korisniku za njegove funkcije se kreće od 33 do 270. Arhitektura čipova iz familije CoolRunner-II je prikazana na slici 1.4.2. Kao što vidimo postoje I/O blokovi koji predstavljaju interfejse prema spoljašnjoj okolini čipa. AIM (*Advanced Interconnect Matrix*) blok predstavlja matricu povezivanja, dok funkcijski blokovi predstavljaju PLD blokove sa slike 1.4.1. Vidimo da su funkcijski blokovi zasnovani na PLA arhitekturi. Svaki funkcijski blok ima 16 makroćelija. U jedan funkcijski blok ulazi 56 signala, od toga 40 signala dolazi iz AIM bloka, a 16 signala iz I/O

bloka. Iz jednog funkcijskog bloka izlazi 16 signala koji se vode ka AIM bloku, kao i I/O bloku (na slici 1.4.2 su prikazani kao odvojene linije, ali je izvorište tih linija isto). Takođe, na funkcijske blokove se dovode takt signali, kao i kontrolni signali set/reset.



Slika 1.4.2. – Arhitektura čipa iz familije CoolRunner-II

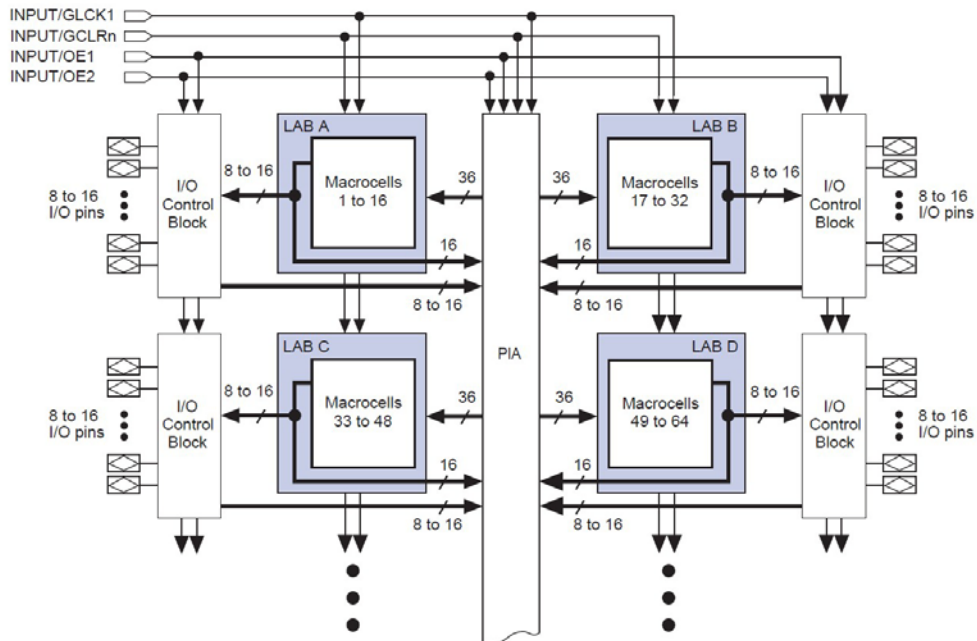


Slika 1.4.3. – Struktura jedne makroćelije u funkcijskom bloku čipa iz familije CoolRunner-II

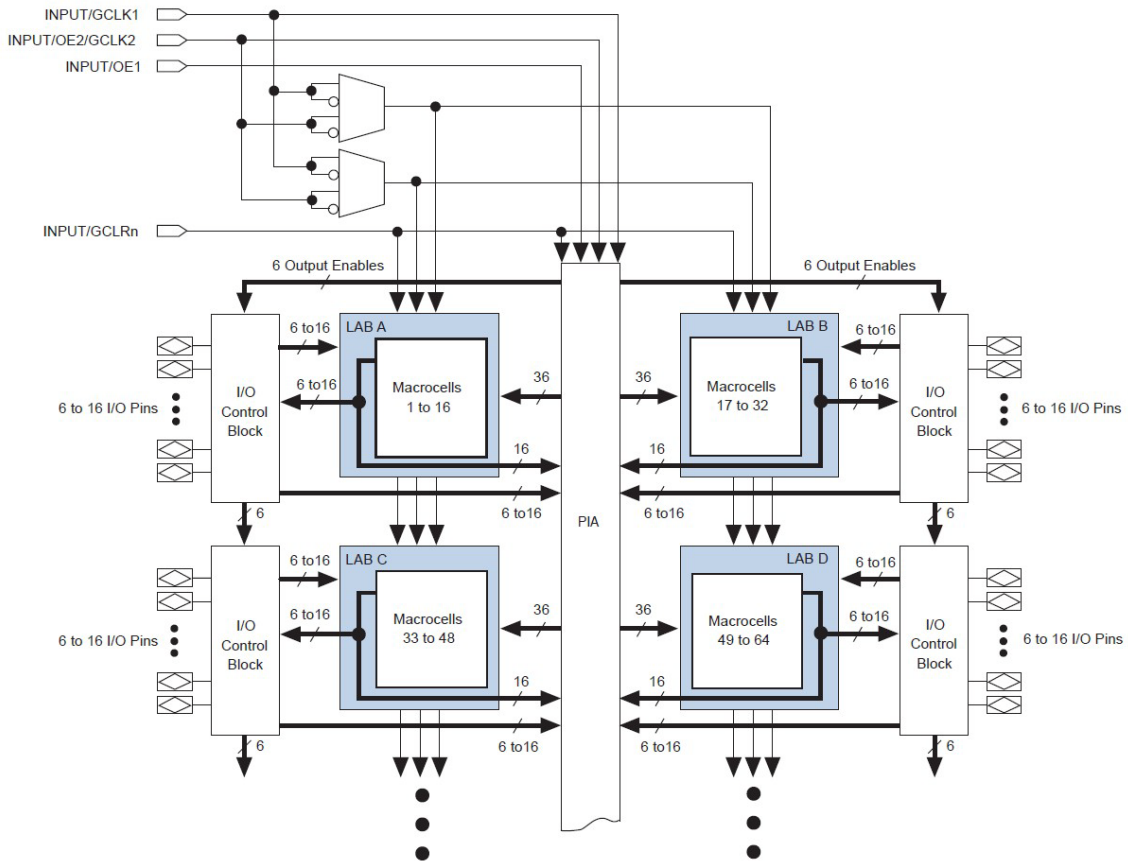
Na slici 1.4.3. je prikazana struktura jedne makroćelije u funkcijskom bloku. Može se videti da u ulaznom delu makroćelije imamo PLA strukturu sa 56 I kola (programabilni I niz) čiji se izlazi vode ka ILI kolu (deo programabilnog ILI niza). Važno je napomenuti da se programabilni I niz deli između svih 16 makroćelija unutar jednog funkcijskog bloka, odnosno na slici 1.4.3 prikazani niz od 56 I kola je zajednički za sve makroćelije u jednom funkcijskom bloku. S druge strane programabilni ILI niz je distribuiran po makroćelijama tako da se u svakoj makroćeliji nalazi po jedno ILI kolo. Produkti I niza se formiraju od 40 signala koji dolaze iz AIM bloka. Pored PLA strukture u funkcijskom bloku, u svakoj makroćeliji postoji više multipleksera čijim se konfigurisanjem definiše izbor signala koji će se koristiti unutar makroćelije. Npr. jedan od multipleksera omogućava izbor jednog od tri globalna takta (GCK0, GCK1, GCK2) koji će se koristiti unutar dotične makroćelije. Svaka makroćelija sadrži i jedan flip-flop koji se može konfigurisati da radi kao D-FF ili kao T-FF. Takođe, flip-flop se može konfigurisati da radi kao registar ili leč. Takođe se konfiguriraju i druga svojstva flip-flopa, npr. da li će flip-flop raditi na obe ivice takta (uzlaznu i silaznu) ili samo na jednu (uzlaznu ili silaznu). Ovi flip-floповi u makroćelijama su neophodni za implementaciju sekvencijalne logike. Na ulaz flip-flopa se dovodi signal sa ulaznog pina iz I/O bloka ili izlaz XOR kola, što se selektuje konfiguracijom multipleksera čiji se izlaz vodi na ulaz flip-flopa. Na ulaz XOR kola se dovodi izlaz ILI kola koji predstavlja deo programabilnog ILI niza u PLA strukturi. Takođe, na ulaz XOR kola se dovodi i izlaz multipleksera koji selektuje jedan od signala VCC, GND, PTC i invertovani PTC, čime se konfiguriraju ulaz u flip-flop ukoliko se u flip-flop ne dovodi signal sa ulaznog pina. VCC označava logičku '1', a GND označava logičku '0'. Ako se GND selektuje kao ulaz u XOR kolo, tada je izlaz XOR kola jednak izlazu ILI kola. Ako se VCC selektuje kao ulaz u XOR kolo, tada je izlaz XOR kola jednak invertovanom izlazu ILI kola. Selekcijom PTC signala ili njegove invertovane vrednosti se mogu kreirati dodatne logičke funkcije u slučaju potrebe, tj. tada je izlaz XOR kola jednak rezultatu XOR operacije nad izlazom ILI kola i PTC signala (ili njegove invertovane vrednosti). GSR signal predstavlja globalni set/reset signal kojim se može setovati/resetovati flip-flop u makroćeliji. Kao što vidimo sa slike 1.4.3, za izlaz iz makroćelije se može koristiti izlaz iz flip-flopa ili izlaz iz XOR kola. Izlaz iz XOR kola se koristi u slučaju kombinacione logike, a izlaz iz flip-flopa se koristi u slučaju sekvencijalne logike. Oba ova izlaza se vode na AIM blok, ali na I/O blok se vodi samo jedan od ovih izlaza u zavisnosti kako je multiplekser na izlazu konfigurisan (tj. programiran). Može se videti sa slike 1.4.3 i da se pojedini produkti programabilnog I niza u PLA delu vode osim na fiksno ILI kolo i na druge delove makroćelije. To su CT (*Control Terms*) signali u koje spadaju CTC, CTE, CTS i CTR, kao i PT (*Product Terms*) signali u koje spadaju PTA, PTB i PTC. Ovi signali omogućavaju veću fleksibilnost u konfigurisanju rada makroćelije. Tako se signali CTS i CTR mogu koristiti za setovanje/resetovanje flip-flopa umesto globalnog set/reset signala (GSR). CTC signal se može koristiti kao lokalni takt umesto GCKi signala koji predstavljaju globalne taktove. CTE signal se može koristiti kao signal dozvole (*Enable*). CT signali su zajednički za sve makroćelije, dok su PT signali lokalnog karaktera tj. svaka makroćelija ima svoja tri PT signala. Očigledno je da se programiranjem CPLD čipa pored programiranja povezivanja u matrici povezivanja (ovde je to AIM blok), programiraju i pojedini delovi makroćelija (konfiguriraju se multiplekseri, konfiguriraju se tip flip-flopa, konfiguriraju se PLA blok), čime se postiže velika fleksibilnost CPLD čipa.

Drugi veliki proizvođač je Altera. CPLD familije koje nudi Altera su [MAX7000](#), [MAX II](#) i [MAX V](#) familije (u nazive familija su postavljeni hiperlinkovi ka odgovarajućim pdf fajlovima koji sadrže detaljne informacije o internoj arhitekturi čipova i pregled performansi čipova u

okviru familije). Informacije o ovim familijama su dostupne i na Alterinom sajtu www.altera.com.

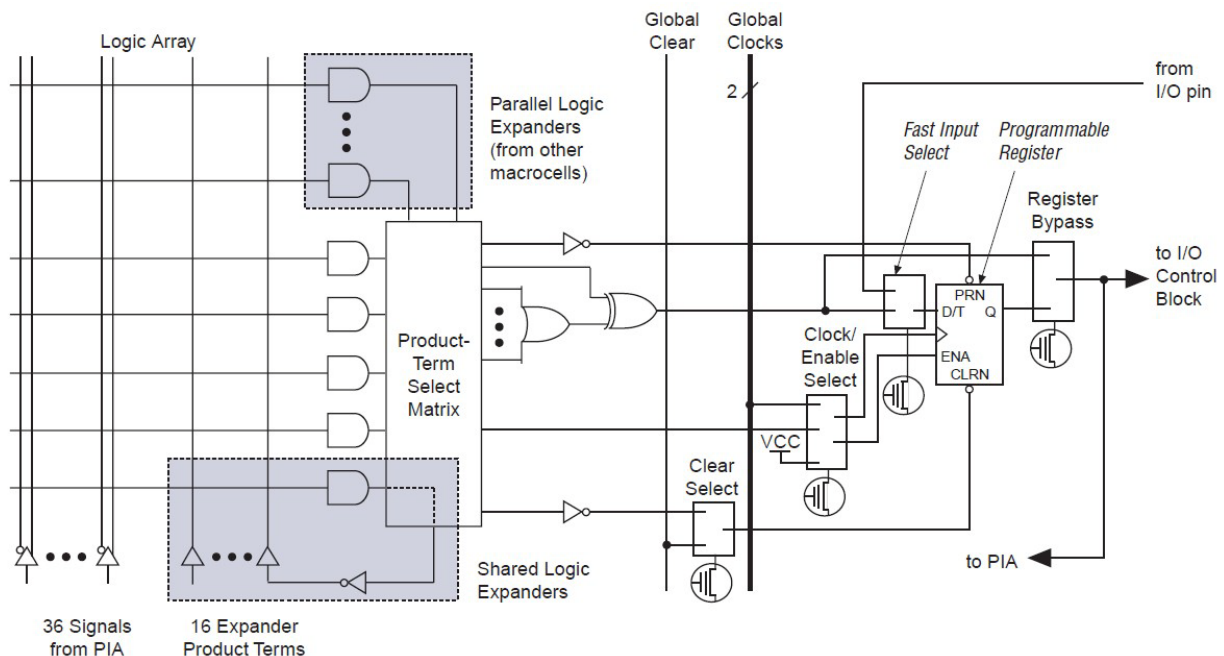


Slika 1.4.4. – Arhitektura slabijih čipova iz familije MAX7000



Slika 1.4.5. – Arhitektura jačih čipova iz familije MAX7000

Broj makroćelija u čipovima familije MAX7000 se kreće između 32 i 256, dok se broj (ulaznih/izlaznih) pinova koji su na raspolaganju korisniku kreće između 36 i 164. Arhitektura slabijih čipova iz MAX7000 familije je prikazana na slici 1.4.4. Gotovo identičnu arhitekturu imaju i jači čipovi iz MAX7000 familije i ona je prikazana na slici 1.4.5. Kao što vidimo, arhitektura sadrži I/O blokove za povezivanje čipa sa spoljašnjom okolinom, PIA (*Programmable Interconnect Array*) blok koji ima ulogu matrice povezivanja i LAB (*Logic Array Block*) blokove koji imaju ulogu PLD blokova. Svaki LAB blok se sastoji od niza 16 makroćelija. U LAB blok se vodi 36 signala iz PIA bloka (kojima se međusobno povezuju LAB blokovi, odnosno makroćelije), i globalni signali poput globalnog takta (GCLKi), globalnog reseta (GCLRn) i globalnog signala dozvole za izlaz (OEi). U slučaju jačih čipova MAX7000 familije, dovode se i signali iz I/O bloka u LAB blok (6-16 signala u zavisnosti koliko I/O pinova ima I/O blok), kao što se vidi sa slike 1.4.5. Izlaz svake makroćelije se vodi na PIA blok, pa tako iz svakog LAB bloka ide 16 signala ka PIA bloku. Takođe se izlazi makroćelija vezuju i na I/O blokove, i to 8-16 signala u slučaju slabijih čipova, odnosno 6-16 signala u slučaju jačih čipova MAX7000 familije. Očigledno, u slučaju kada I/O blok ima manje od 16 I/O pinova da se iz jednog LAB bloka ne vode izlazi iz svih makroćelija na odgovarajući I/O blok, već samo nekih makroćelija.



Slika 1.4.6. – Struktura makroćelije jačih čipova iz familije MAX7000

Struktura jedne makroćelije jačih čipova familije MAX7000 je prikazana na slici 1.4.6. Kao što vidimo i ovde imamo programabilni I niz, slično kao u CoolRunner-II familiji kompanije Xilinx. 5 I kola je dodeljeno svakoj makroćeliji. Ulaze u ta I kola čini 36 signala iz PIA bloka kao i 16 signala iz makroćelija (tzv. deljeni prošireni signali). Deljeni prošireni signal se dobija vraćanjem natrag jednog produkta koji ulazi u matricu za selekciju produkata, naravno ako se matrica konfiguriše da to uradi. Svaka makroćelija daje potencijalno jedan deljeni prošireni signal koji mogu da koriste sve makroćelije u istom LAB bloku, pa otuda ima ukupno 16 deljenih proširenih signala. Matrica za selekciju produkata vrši povezivanje ulaza i izlaza, tj. spaja ulazne produkte sa odgovarajućim izlazima. Izlazi ove matrice čine ulazi u ILI i XOR kolo

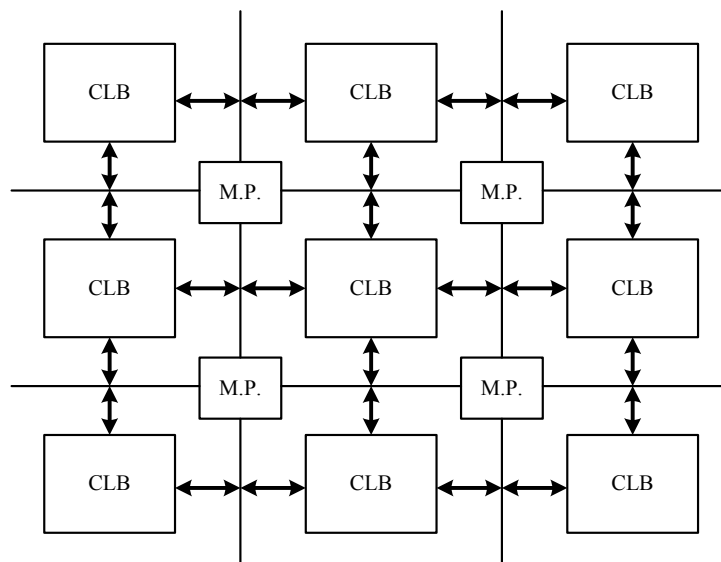
za formiranje kombinacionih funkcija, izlazi za setovanje i resetovanje flip-flopa, izlaz za takt flip-flopa (lokalni takt), kao i signal dozvole za flip-flop. Pored toga, matrica za selekciju produkata može kao ulaz da koristi i tzv. paralelne proširene signale koji čine nekorišćena I kola susednih makročelija. Do 15 paralelnih proširenih signala može da se koristi u jednoj makročeliji (broj paralelnih proširenih signala zavisi i od pozicije makročelije u nizu makročelija u okviru LAB bloka što je detaljnije objašnjeno u dokumentaciji [MAX7000](#) familije). Proširenim signalima (deljenim i paralelnim) je omogućeno kreiranje i složenijih logičkih funkcija u slučaju potrebe. Vidimo sa slike 1.4.6 da i ovde postoje multiplekseri unutar makročelija za izbor načina rada makročelije. Na primer, izlazni multiplekser (*Register Bypass*) selektuje izlaz iz flip-flopa u slučaju formiranja sekvencijalne logike, ili izlaz XOR kola u slučaju formiranja kombinacione logike. Izlaz iz makročelije se vodi u matricu povezivanja (PIA blok) i u odgovarajući I/O blok koji je povezan na LAB u kom se makročelija nalazi. Flip-flop se može programirati da radi kao D-FF, T-FF, JK-FF ili RS-FF. Multiplekser, čiji izlaz se vodi na ulaz flip-flopa, određuje da li se na flip-flop vodi signal sa ulaznog pina iz I/O bloka ili izlaz XOR kola. Kao takt flip-flopa se može koristiti globalni takt ili lokalni takt što se selektuje odgovarajućim multiplekserom. Isto važi i za reset flip-flopa, gde se može koristiti globalni reset signal ili lokalno definisan reset signal koji se dobija iz matrice za selekciju produkata.

Što se tiče familija MAX II i MAX V one su po strukturi veoma slične FPGA čipovima pa stoga neće biti obrađene u okviru ovih skripti. Čitalac, nakon čitanja sledeće sekcije u kojoj će biti izloženi FPGA čipovi, može da pogleda detaljniju dokumentaciju [MAX II](#) i [MAX V](#) familija i uvidi da su ove dve familije, u stvari, po svojoj internoj arhitekturi veoma bliske FPGA čipovima. Čak bi se moglo reći da po svojoj strukturi ove dve familije pre pripadaju FPGA čipovima nego CPLD čipovima. CPLD i FPGA čipovi drugih proizvođača neće biti obrađivani u okviru ovih skripti, već se to ostavlja čitaocu ukoliko želi da sazna strukturu programabilnih čipova drugih proizvođača poput [Lattice](#), [Actel](#) (sada Microsemi), [Achronix](#).

1.5 FPGA

Sledeći korak u razvoju programabilnih čipova je bilo unapređivanje interne arhitekture čipa. Naime, ako se pogleda interna arhitektura CPLD čipa sa slike 1.4.1 može se videti da ona nije skalabilna. U slučaju velikog broja PLD blokova, dolazi do problema ostvarivanja efikasne i ekonomične realizacije matrice povezivanja koja bi trebala da omogući međusobnu komunikaciju između bilo kojih PLD blokova. Stoga su razvijeni FPGA (*Field Programmable Gate Array*) čipovi sredinom osamdesetih godina. Opšta arhitektura FPGA čipa je prikazana na slici 1.5.1. Kao što se vidi sa slike, interna arhitektura FPGA čipa je zasnovana na rešetkastoj mreži, pri čemu se u svakom polju rešetkaste mreže nalazi CLB (*Configurable Logic Block*) blok. U suštini možemo takođe reći da je interna arhitektura FPGA čipa matična, pri čemu se u poljima matrice nalaze CLB blokovi. Rešetke predstavljaju magistrale za međusobno povezivanje CLB blokova, a u preseccima rešetki se nalaze svičevi tj. matrice povezivanja (na slici označene sa M.P.). Svaki CLB blok je povezan na susedne horizontalne i vertikalne magistrale koje služe za međusobno povezivanje CLB blokova. Svaka magistrala je sačinjena od više linija, gde svaka linija predstavlja jedan signal, a matrice povezivanja vrše međusobna spajanja delova magistrala tj. linija. Na jednu liniju može da šalje signal maksimalno jedan CLB blok, a koji CLB blok će to biti se određuje programiranjem (konfiguriranjem) FPGA čipa. S druge strane, sa iste linije više CLB blokova može da prima (isti) signal. Ovakva arhitektura je

veoma skalabilna jer matrica povezivanja više ne predstavlja usko grlo kao kod CPLD čipova, a veoma je lako povećavati broj horizontalnih i vertikalnih magistrala za formiranje FPGA čipova sa većim resursima (naravno u skladu sa postavljenim fizičkim ograničenjima poput dimenzija čipa, potrošnje čipa, kašnjenja unutar čipa i sl). Programiranjem (konfigurisanjem) matrica povezivanja se može ostvariti veza između bilo koja dva CLB bloka tako da je arhitektura FPGA čipa potpuno dostupna. Drugo poboljšanje u odnosu na CPLD čipove predstavljaju sami CLB blokovi. CLB blokovi imaju različitu strukturu u odnosu na makroćelije. Kao što smo videli u prethodnoj sekciji makroćelija se zasniva na PLA strukturi za formiranje kombinacione logike uz postojanje flip-flopa za formiranje sekvencijalne logike i nekoliko multipleksera za omogućavanje fleksibilnog programiranja makroćelije. CLB blok, s druge strane, koristi LUT (*LookUp Table*) tabele za formiranje kombinacione logike. LUT tabela je svojevrsna mala memorija u čijim se lokacijama čuva rezultat kombinacione logike, a adresa lokacije predstavlja vrednosti ulaznih signala - praktično LUT tabela predstavlja emulaciju tablice istinitosti. Prednost LUT tabele je veća fleksibilnost u realizaciji kombinacione logike jer nije zavisna od strukture sume proizvoda (ili proizvoda suma) u pogledu koliko I/ILI kola treba za realizaciju, dok je mana što je broj ulaznih podataka u LUT tabelu ograničen tj. mali (tipične LUT tabele su adresirane sa 4 ili 6 ulaznih podataka), pa je za kombinacionu logiku sa većim brojem ulaza potrebno kombinovati nekoliko LUT tabela, ali uglavnom to ne predstavlja problem. Sledeća prednost je što tipično FPGA čipovi imaju više flip-flopa u odnosu na CPLD čipove. Takođe, važna prednost je postojanje velikog broja naprednih funkcija koje CPLD čipovi nemaju. Tako da pored JTAG podrške i podrške za veliki broj standarda za I/O interfejse koju imaju i CPLD čipovi, FPGA čipovi mogu da sadrže i interne memorijske blokove koji se mogu konfigurisati kao RAM, ROM ili FIFO memorije, množače, DSP (*Digital Signal Processing*) blokove, PLL (*Phase Locked Loop*) i DLL (*Delay Locked Loop*) blokove za generisanje novih taktova iz referentnog takta (generisani takt može biti i brži i sporiji), multigigabitske primopredajnike, itd. Očigledno, namena FPGA čipova je za hardversku implementaciju visokih performansi raznovrsnih zahtevnih funkcija. Međutim, i kod FPGA čipova postoje razlike u klasi, pa postoje ekonomične klase koje su namenjene implementaciji manje zahtevnih funkcija i skuplje klase znatno viših performansi koje su namenjene implementaciji veoma zahtevnih funkcija.



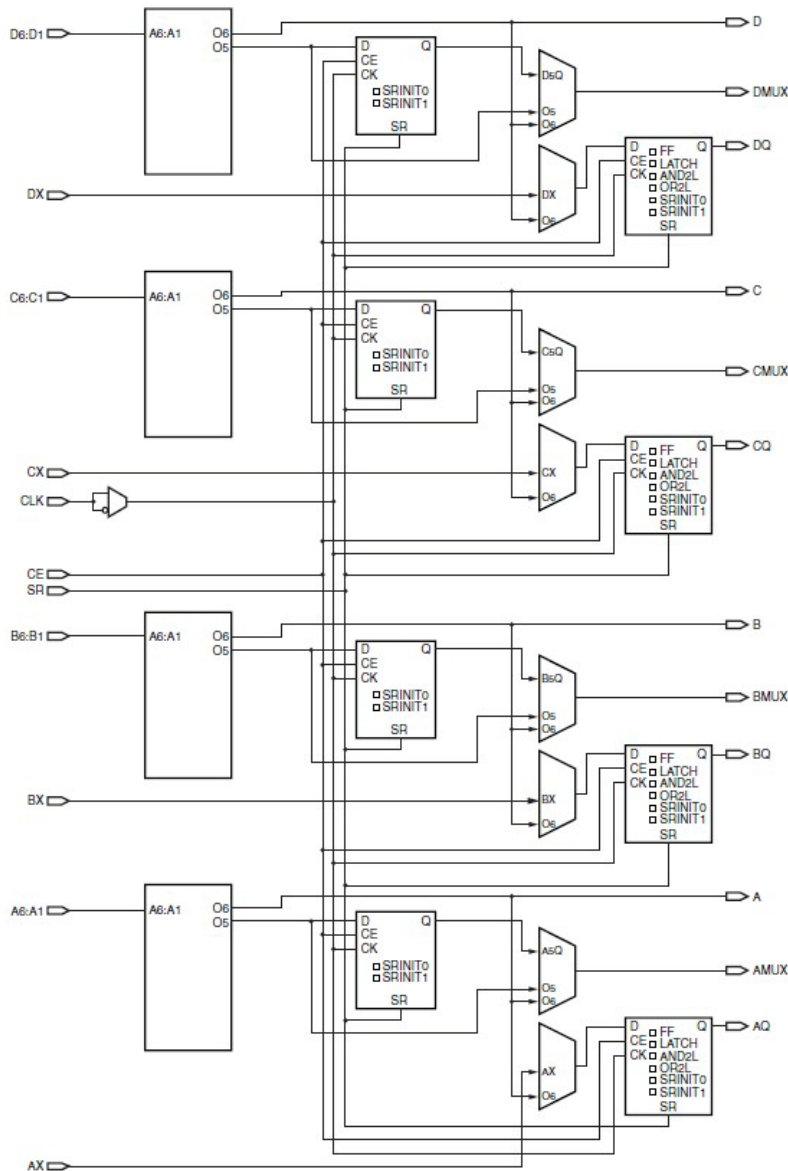
Slika 1.5.1. – Arhitektura FPGA čipa

Važno je napomenuti da i ovde postoji problem optimalnog smeštanja implementirane funkcije na sam čip pogotovo u slučaju složenih funkcija koje se postavljaju na FPGA čip sa velikom količinom resursa. Potrebno je rasporediti implementiranu funkciju po CLB blokovima koje potom treba međusobno povezati programiranjem matrica povezivanja kojih sada ima više nego kod CPLD čipa gde je bila samo jedna. Uglavnom, softver namenjen za postavljanje implementirane funkcije na sam čip nalazi suboptimalno rešenje. Kriterijum optimalnosti definiše sam korisnik, gde se za kriterijum može podesiti npr. minimalno kašnjenje, minimalni resursi, balans između više kriterijuma i dr (slično kao i kod CPLD čipa).

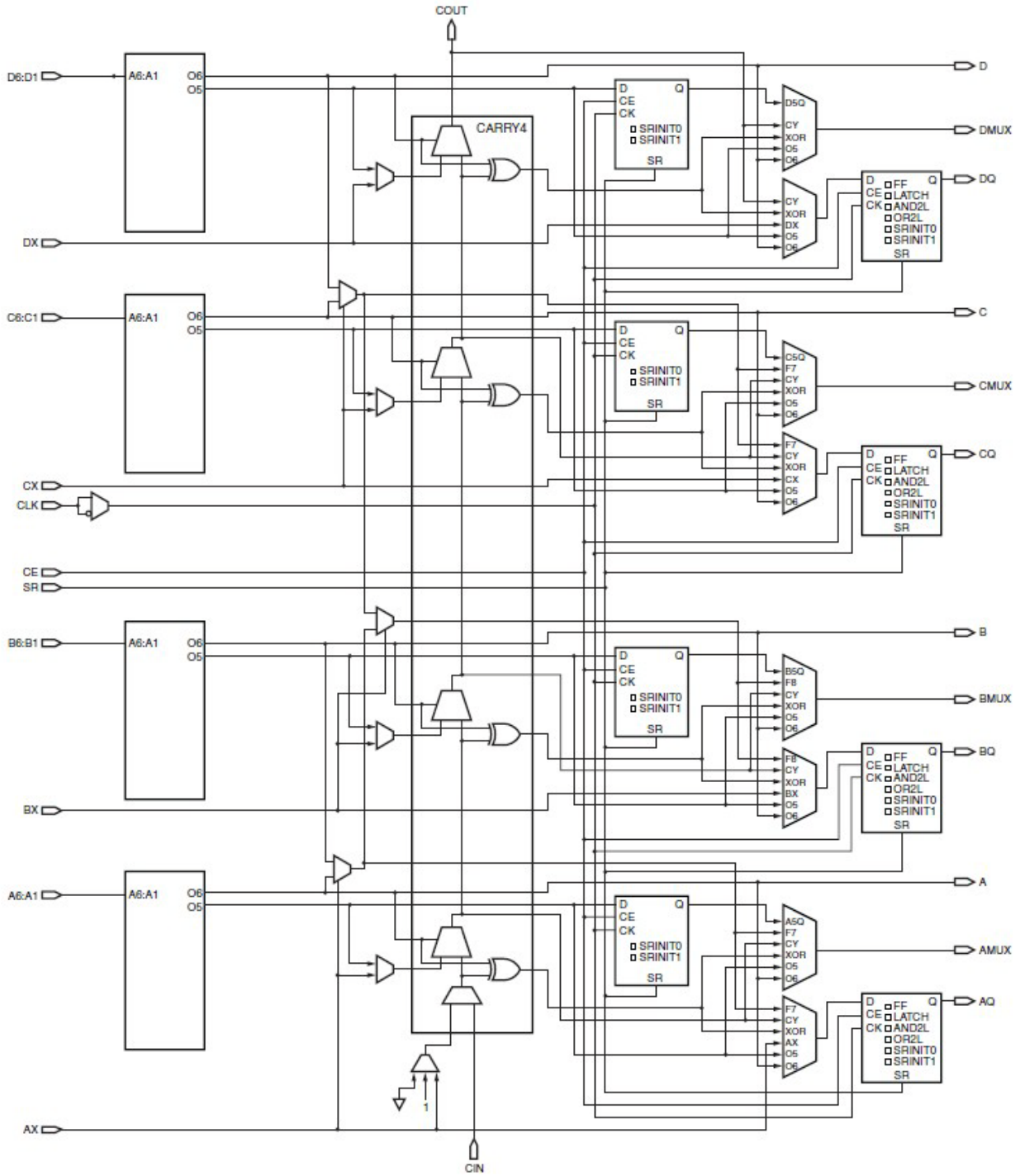
Poslednja generacija FPGA čipova kompanije Xilinx (tzv. serija 7) obuhvata tri familije: [Virtex7](#), [Kintex7](#) i [Artix7](#). Virtex7 familija ima najviše performanse, Kintex7 predstavlja srednju klasu koja ima u suštini najbolji odnos performanse/cena, a Artix7 familija ima najniže performanse, ali i najnižu cenu. Pre serije 7, Xilinx je nudio familije Spartan i Virtex (postoji više generacija ovih familija), gde su Spartan familije nudile FPGA čipove nižih performansi, ali i niže cene, dok su Virtex familije sadržale FPGA čipove visokih performansi, ali i visoke cene. Pregled čipova iz FPGA čipova prethodne generacije [Spartan6](#) i [Virtex6](#) je dat u odgovarajućim pdf fajlovima. U okviru kursa studenti će imati mogućnost rada sa čipovima iz familije [Virtex5](#). Naravno, unutar svih navedenih familija postoji širok spektar FPGA čipova koji se značajno razlikuju po svojim resursima kao što se može videti iz priloženih pdf fajlova koji daju pregled čipova po odgovarajućim familijama.

U okviru Spartan6 familije jedan CLB blok sadrži dva tzv. slajsa (*slice*), gde jedan slajs predstavlja svojevrsni ekvivalent makročelije iz CPLD čipa (pri čemu važe razlike koje su navedene između CLB bloka i makročelije navedene u prvom pasusu ove sekcije). Važno je napomenuti da dva slajsa koja se nalaze u istom CLB bloku su međusobno nezavisna, odnosno nema direktnih konekcija između njih. Slajsovi se preko svojih ulaza i izlaza povezuju na matricu povezivanja. Postoje tri vrste slajsova, SLICEX, SLICEL i SLICEM, čije su strukture prikazane na slikama 1.5.2, 1.5.3 i 1.5.4, respektivno. U svakom čipu Spartan6 familije, oko 50% slajsova je SLICEX tipa, oko 25% je SLICEL tipa i oko 25% je SLICEM tipa. SLICEX slajs je osnovni slajs koji sadrži četiri šestoulazne LUT tabele i 8 flip-flova. LUT tabela ima, kao što je rečeno, šest ulaza (signali A6 do A1) tako da se ona može koristiti za formiranje kombinacionih funkcija koje sadrže do 6 ulaznih signala. Ako kombinaciona funkcija ima više od 6 ulaznih signala, tada je neophodno kombinovanje više LUT tabela za formiranje takve kombinacione funkcije. Sa slike 1.5.2 se vidi da LUT tabela sadrži dva izlaza O5 i O6, tako da se LUT tabela može iskoristiti i za formiranje dve kombinacione funkcije pod uslovom da svaka od njih ima manje od 6 ulaznih signala. Naravno, u slučaju da je ukupan zbir ulaznih signala obe kombinacione funkcije veći od 6 onda neki od tih ulaznih signala su zajednički za obe kombinacione funkcije. Ako svaka od njih ima po 3 ili manje ulaznih signala tada skupovi ulaznih signala obe funkcije ne moraju da se preklapaju, već mogu biti u potpunosti međusobno nezavisni. Četiri flip-flova u sredini se mogu konfigurisati samo kao D-FF, a četiri flip-flova na krajevima se mogu konfigurisati ili kao D-FF ili kao leč. Ukoliko se flip-flovi na krajevima konfigurisu kao lečevi, flip-flovi u sredini se ne mogu koristiti. Razlog je veoma jednostavan, kao što se vidi sa slike 1.5.2, signal takta (CK) je zajednički za svih 8 flip-flova isto kao i signal dozvole (CE) i signal set/reset (SR). To znači da u slučaju konfigurisanja flip-flova na kraju da radi kao leč, onda svi flip-flovi na krajevima moraju da rade kao leč. Pošto signal takta (CK) je sada signal čijom vrednošću se određuje transparentnost leča, onda četiri flip-flova u sredini ne mogu raditi korektno jer signal CK nije pravi takt i samim tim se ne mogu koristiti. Slično tome, signal dozvole i signal set/reset se vode na svih 8 flip-flova pa ih samim tim

istovremeno aktiviraju. Na primer, istovremeno se resetuje svih osam flip-floпова. Tako da se može zaključiti da iako postoji osam flip-floпова, postoje dodatna ograničenja koja sprečavaju njihovo u potpunosti nezavisno korišćenje i konfigurisanje. Izlazi A, B, C i D predstavljaju izlaze kombinacione funkcije koji se vode direktno iz LUT tabela. Signali AMUX, BMUX, CMUX i DMUX predstavljaju izlaze koji mogu da se koriste kao izlazi kombinacionih funkcija ili sekvencijalnih funkcija. Svaki od tih signala predstavlja izlaz odgovarajućeg multipleksera koji multipleksira izlaz iz odgovarajućeg srednjeg flip-floпа (sekvencijalna logika), i oba izlaza odgovarajuće LUT tabele (kombinaciona logika). Signali AQ, BQ, CQ i DQ predstavljaju izlaze iz odgovarajućih flip-floпова na krajevima i samim tim sekvencijalnih funkcija. Na ulaz flip-floпа sa kraja se dovodi ili direktan ulazni signal (AX, BX, CX ili DX u zavisnosti koji flip-flop je u pitanju) ili izlaz O6 odgovarajuće LUT tabele (što se određuje konfiguracijom multipleksera čiji izlaz se direktno povezuje sa ulazom dotičnog flip-floпа).



Slika 1.5.2. – Struktura SLICEX slajsa



Slika 1.5.3. – Struktura SLICEL slajsa

slike 1.5.3 da postoji CIN ulaz i COUT izlaz koji predstavljaju tzv. *carry* signale za formiranje aritmetičkih funkcija sabiranja/oduzimanja. CLB blokovi su grupisani u kolone, a svaki CLB blok sadrži dva slajsa koja su međusobno nezavisna, znači da u jednoj koloni CLB blokova postoje dve kolone slajsova. Jedna kolona slajsova sadrži slajsove istog tipa. U slučaju slajsova tipa SLICEL i SLICEM, susjedni slajsovi u istoj koloni se međusobno povezuju navedenim CIN i COUT signalima radi ostvarivanja aritmetičkih funkcija nad dužim nizovima bita tj. vektorima.

SLICEM slajs proširuje SLICEL slajs tako što dodaje mogućnost formiranja malih RAM/ROM memorija ili pomeračkih registara, a to se postiže upotrebom složenije LUT tabele kao što se vidi sa slike 1.5.4. Lako se uočava da je LUT tabela SLICEM slajsa znatno složenija od LUT tabele koja se koristi u SLICEL i SLICEX slajsovima. Funkcije signala LUT tabele u slučaju konfigurisanja LUT tabele kao RAM/ROM memorije ili pomeračkog registra se mogu videti iz detaljnije Spartan6 [dokumentacije](#) koja daje detaljan opis SLICEM slajsa i njegovog konfigurisanja u tim slučajevima. Priložena [dokumentacija](#) daje detaljan opis CLB bloka, odnosno slajsova i njihovih mogućih konfiguracija.

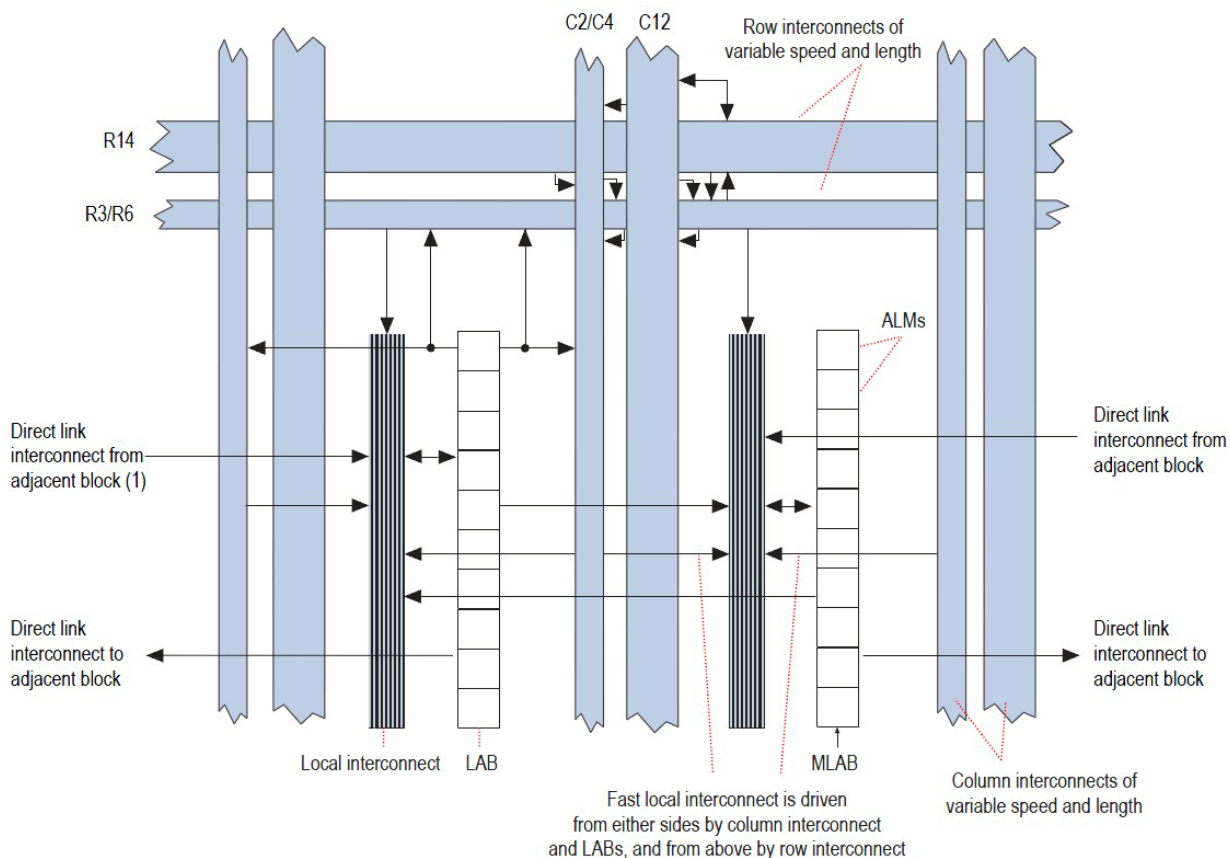
FPGA čipovi Virtex6 familije takođe koriste CLB blokove, pri čemu svaki CLB blok sadrži dva slajsa. Detalji se mogu naći u priloženoj [dokumentaciji](#). U Virtex6 familiji se koriste samo SLICEL i SLICEM slajsovi a osnovni SLICEX slajs se ne koristi, što svedoči o već navedenoj činjenici da Virtex6 familija predstavlja skuplju familiju, ali i sa boljim performansama u odnosu na Spartan6 familiju. Struktura SLICEL i SLICEM slajsova je identična strukturama ovih slajsova u Spartan6 familiji. Udeo SLICEL slajsova se kreće između 60% i 72% u čipovima Virtex6 familije (ostatak naravno čine SLICEM slajsovi).

FPGA čipovi tzv. serije 7 koriste CLB blokove identične onima korišćenim u Virtex6 familiji, što znači da svaki CLB blok ima dva slajsa, i koriste se samo SLICEL i SLICEM slajsovi. Detalji se mogu naći u priloženoj [dokumentaciji](#). Iz priložene dokumentacije se može videti da je struktura SLICEL i SLICEM slajsova ostala ista. I dalje je udeo SLICEL slajsova u granicama 60-70%. Glavno unapređenje serije 7 je gušća integracija jer se prešlo sa 40nm (Virtex6 koristi 40nm, a Spartan6 45nm tehnologiju) na 28nm tehnologiju, čime je omogućen veći broj CLB blokova (kao i drugih blokova poput npr. interne memorije) po jedinici površine u odnosu na prethodnu generaciju (Virtex6 i Spartan6) FPGA čipova. Tako da poredeći resurse [Virtex6](#) i [Virtex7](#) familije može se videti ogromna razlika u raspoloživim resursima.

Kompanija Altera nudi dve familije FPGA čipova - Stratix i Cyclone u sličnom kontekstu kao što je kompanija Xilinx nudila svoje familije Virtex i Spartan. Stratix familije predstavljaju FPGA čipove visokih performansi, ali i više cene, dok Cyclone familije predstavljaju čipove nižih performansi, ali i niže cene. Naravno, u okviru svake familije se nudi veliki raspon FPGA čipova koji se značajno razlikuju po svojim raspoloživim resursima. Detalji za nekoliko Stratix ([Stratix III](#), [Stratix IV](#), [Stratix V](#)) familija se mogu naći u priloženoj dokumentaciji, isto kao i za Cyclone familije ([Cyclone III](#), [Cyclone IV](#), [Cyclone V](#)).

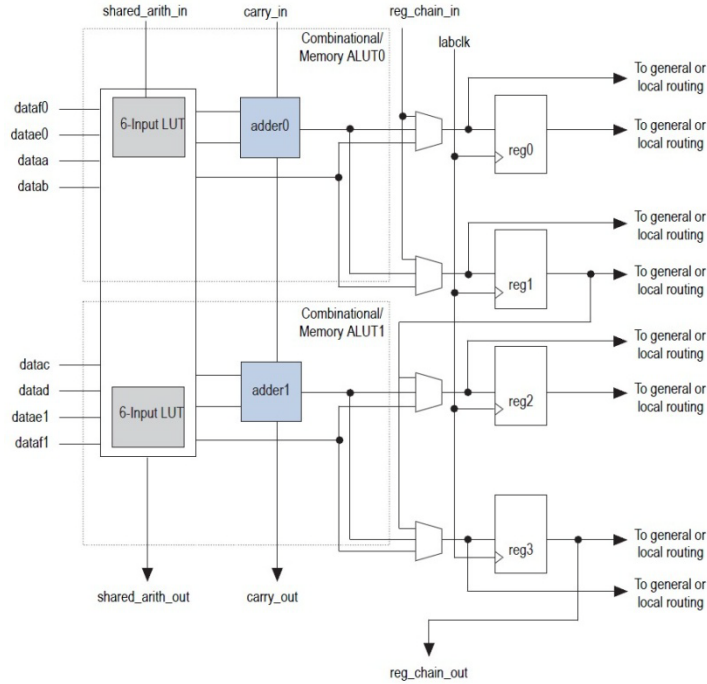
CLB blok se u Alterinim FPGA čipovima Cyclone V familije nazivaju ALM (*Adaptive Logic Module*) blokovima. ALM blokovi su organizovani u tzv. LAB (*Logic Array Block*) nizove, pri čemu se razlikuju LAB i MLAB nizovi. Jedan LAB/MLAB niz sadrži 10 ALM blokova. MLAB niz nudi sve funkcionalnosti kao i LAB niz uz dodatnu mogućnost formiranja male memorije iz ALM blokova koji ulaze u sastav MLAB niza, slično kao kod SLICEM slajsova u Xilinx čipovima. Sam LAB/MLAB niz predstavlja u suštini ekvivalent koloni slajsova u Xilinx čipovima. Postoji više magistrala koje se koriste za povezivanje LAB/MLAB nizova -

horizontalne, vertikalne i lokalne magistrale. Pri tome postoji više tipova horizontalnih i vertikalnih magistrala, koje se razlikuju po svojoj brzini i kapacitetu (u prevodu kašnjenja kroz magistrale i koliko signala može da ide preko magistrale). Način povezivanja LAB/MLAB nizova je prikazan na slici 1.5.5.

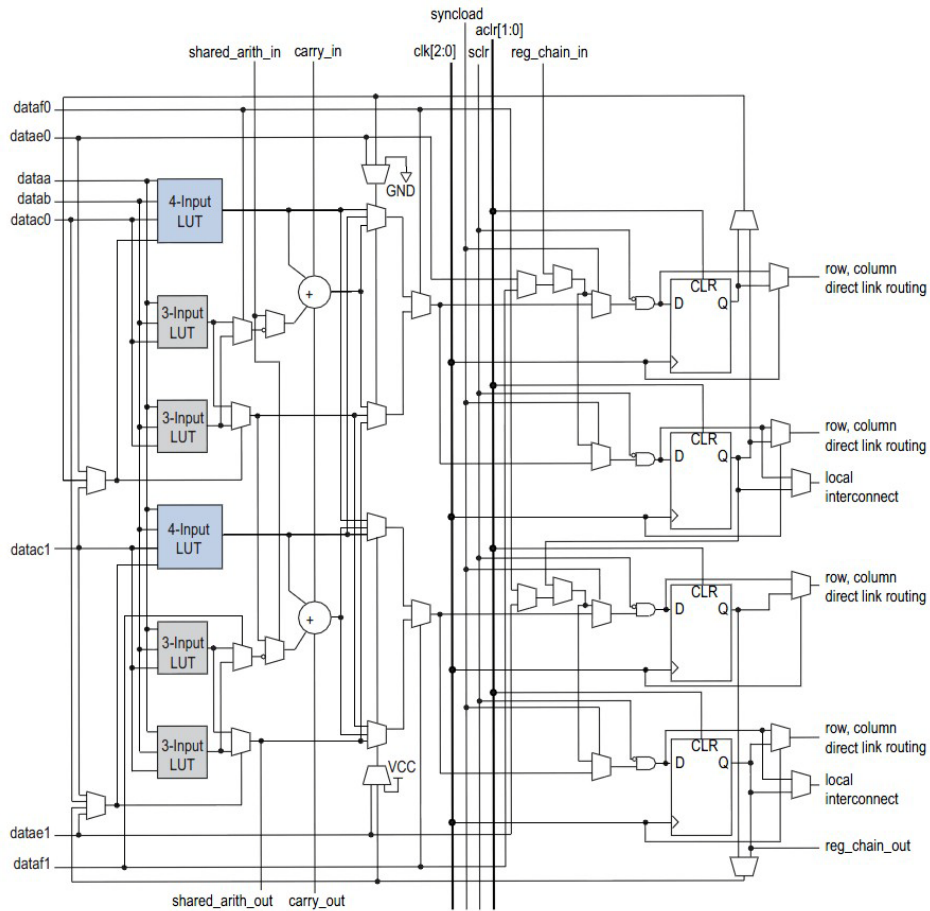


Slika 1.5.5. – Povezivanje LAB/MLAB nizova u Cyclone V familiji

Generalizovana blok-shema ALM bloka je prikazana na slici 1.5.6, a detaljna blok-shema na slici 1.5.7. Na osnovu generalizovane sheme možemo videti da se jedan ALM sastoji iz dva dela (koji nisu potpuno nezavisni kao kod Xilinx-ovih slajsova u istom CLB bloku, već postoje veze između ova dva dela), gde svaki deo sadrži jednu šestoulaznu LUT tabelu, logiku za sabiranje i dva flip-flopa. Takođe postoje signali za povezivanje sa susednim ALM blokovima (`shared_arith`, `carry` i `reg_chain` ulazni i izlazni signali). Naravno i ovde postoje multiplekseri koji omogućavaju fleksibilnije konfigurisanje rada ALM bloka. Sa detaljnije sheme (slika 1.5.7) se vidi da su šestoulazne LUT tabele kreirane iz manjih LUT tabela. Takođe, može se videti da multiplekseri na izlazu selektuju izlaz kombinacione ili sekvencijalne logike (tj. signal sa ulaza flip-flopa, odnosno izlaz flip-flopa, respektivno). Kod dva flip-flopa postoji dodatni izlazni multiplekserski blok, koji omogućava konekciju sa lokalnom magistralom za povezivanje ALM blokova iz istog ili susednih LAB/MLAB-ova. Ostala četiri izlazna multiplekserska se povezuju na horizontalnu ili vertikalnu magistralu za povezivanje sa ALM blokovima udaljenih LAB/MLAB nizova. Signali `carry_in` i `carry_out` se koriste za formiranje sabirača nad dužim nizovima bita tj. vektora. Signali `shared_arith_in` i `shared_arith_out` se koriste za formiranje sabirača sa tri ulaza (suma tri člana). Signali `reg_chain_in` i `reg_chain_out` se koriste za formiranje dužih registara (sa većim brojem bita).

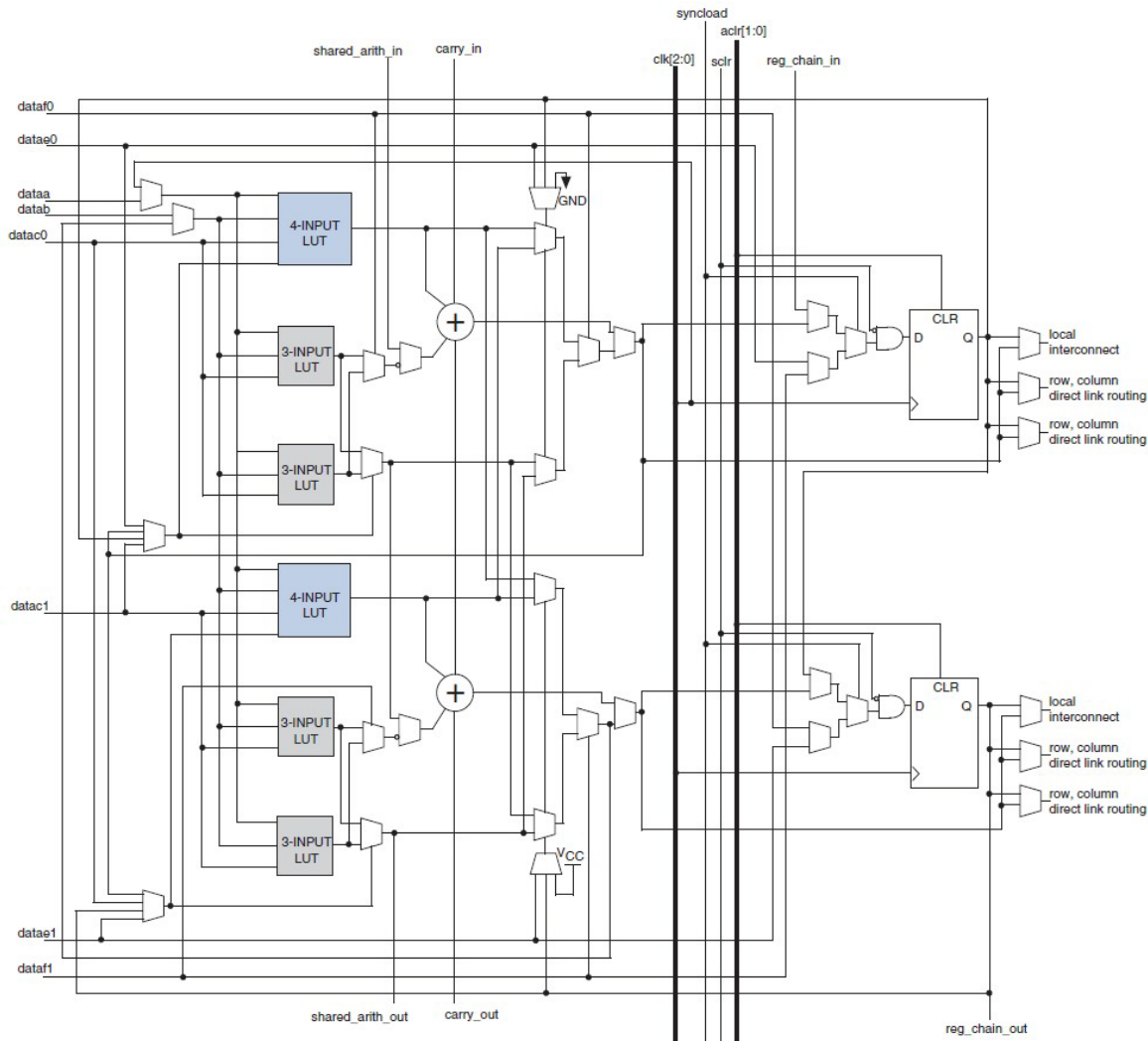


Slika 1.5.6. – Generalizovana blok-shema ALM bloka u Cyclone V familiji



Slika 1.5.7. – Detaljna blok-shema ALM bloka u Cyclone V familiji

Stratix V familija takođe koristi LAB/MLAB nizove, gde su članovi niza ALM blokovi (i ovde jedan niz sadrži 10 ALM blokova). Struktura ALM bloka je identična ALM bloku u Cyclone V familiji. Prednost Stratix V familije je što ima na raspolaganju više horizontalnih/vertikalnih magistrala boljih performansi, kao i više dodatnih resursa poput npr. gigabitskih primopredajnika, internih memorijskih blokova i sl. Inače Cyclone V familije kao interne memorijske blokove koristi tzv. M10K blokove gde jedan M10K blok ima kapacitet 10Kb, dok Stratix V familija koristi M20K blokove gde jedan M20K blok ima kapacitet 20Kb.



Slika 1.5.8. – Detaljna blok-shema ALM bloka u Stratix III i Stratix IV familijama

Starije Stratix familije - Stratix III i Stratix IV takođe koriste LAB/MLAB nizove čiji su članovi ALM blokovi. Osnovna razlika je postojanje samo dva flip-flopa u ALM bloku, pri čemu su svakom flip-flopu pridružena tri izlazna multipleksera, gde jedan služi za povezivanje na lokalnu magistralu, a druga dva za povezivanje na horizontalnu ili vertikalnu magistralu. Detaljna blok-shema ALM bloka u Stratix III i Stratix IV familijama je prikazana na slici 1.5.8. Stratix III i Stratix IV familije kao interne memorijske blokove na raspolaganju imaju dva tipa memorijskih blokova M144K (kapacitet M144K bloka je 144Kb) i M9K (kapacitet M9K bloka je 9Kb). Kompanija Xilinx je na primer u svim svojim familijama koristila princip koji je Altera koristila tek u svojoj petoj generaciji (Stratix V i Cyclone V), a to je da svi interni memorijski

1.6 Metode programiranja konekcija

U prethodnim sekcijama je dat pregled tipova programabilnih čipova. Kod najprostijih programabilnih čipova poput PAL i PLA čipova su mogla da se programiraju spajanja linija u programabilnom I nizu, odnosno kod PLA čipova i spajanja linija u programabilnom ILI nizu, a sve u cilju da se realizuje željena logička funkcija. Kod složenijih programabilnih čipova je dodata mogućnost konfigurisanja i ponašanja gradivnih blokova, npr. konfigurisanje multipleksera u CLB blokovima FPGA čipova, pri čemu je konfigurisanje fizički izvršavano slično programiranju kod spajanja linija.

Ideja je da se spojevi linija mogu programirati na dva načina - spoj se ostvaruje ili spoj se ne ostvaruje. Ovim principom se povezuju linije u programabilnom čipu (linije u programabilnim I/ILI nizovima; u matricama povezivanja), ali i gradivnim blokovima složenijih programabilnih čipova (npr. na kontrolne ulaze pojedinih multipleksera u gradivnom bloku se spaja 0 ili 1, po istom principu kao što se spajaju linije čime se konfigurirše rad multipleksera). Postoji nekoliko tehnika za programiranje programabilnih čipova, odnosno za konfigurisanje spojeva linija:

- osigurači
- antiosigurači
- SRAM (*Static Random Access Memory*)
- fleš memorija
- EPROM (*Erasable Programmable Read Only Memory*)
- EEPROM (*Electrically Erasable Programmable Read Only Memory*)

U slučaju tehnike koja koristi osigurače, osigurači se postavljaju na svakom spoju linija. Inicijalno, osigurači su zatvoreni i linije su spojene. Programiranjem čipa se određuje koji od osigurača treba da ostvari spoj, a koji ne. Tako da kada se odredi željena konfiguracija čipa koja će izvršavati željenu logičku funkciju, vrši se konfigurisanje (programiranje) osigurača. Osigurači koji ne treba da ostvare spoj se pregorevaju dovođenjem napona određene vrednosti čime se postiže željena konfiguracija čipa. Ovaj metod ne dozvoljava reprogramiranje čipa, odnosno jednom programirani čip zadržava trajno svoju funkciju. Očigledno je da u slučaju ove tehnike dizajner treba pažljivo da proveri svoju implementaciju pre programiranja čipa, a potom i da veoma detaljno testira programirani čip da bi otkrio eventualne greške pre programiranja velike serije čipova, odnosno pokretanja proizvodnje. Naravno, napomena o testiranju programiranog čipa važi nezavisno od tehnike programiranja. Prednost ove tehnike je što programabilni čip trajno zadržava svoju konfiguraciju čak i kad se isključi napajanje čipa. Ova tehnika se više ne koristi.

Tehnika antiosigurača je veoma slična tehnici osigurača. Razlika je u suprotnom ponašanju antiosigurača. Naime, antiosigurač inicijalno predstavlja otvorenu vezu (nema spoja), a dovođenjem visokog napona se uspostavlja veza, odnosno spoj. Ostale osobine su identične tehnici osigurača, uz jedinu razliku da je ova tehnika još u upotrebi.

SRAM tehnika se zasniva na upotrebi SRAM ćelija u programabilnom čipu. Ove ćelije kontrolišu tzv. 'pass' tranzistore (danas se koristi uglavnom CMOS (*Complementary Metal–Oxide–Semiconductor*) tehnologija) ili trostatičke bafere koji se nalaze u spojevima. 'Pass' tranzistor ima sličnu funkciju kao osigurač/antiosigurač, odnosno ili predstavlja kratak spoj ili otvorenu vezu. Bazu ovog tranzistora kontroliše vrednost SRAM ćelije (0 ili 1), čime se

kontroliše propusnost tranzistora. Na sličan način se radi i sa trostatičkim baferima. Pokazalo se da su 'pass' tranzistori bolji za kraće putanje gde nema velikog broja interkonekcija, a trostatički baferi su bolji u suprotnom slučaju (duže putanje i veći broj interkonekcija). SRAM ćelije ne zadržavaju svoj sadržaj nakon isključivanja napajanja čipa, pa se konfiguracija programabilnog čipa mora čuvati u eksternoj PROM (*Programmable Read Only Memory*) ili fleš memoriji. To znači da po uključanju napajanja čipa, čip se prvo konfigurira i tek nakon toga postaje operativan tj. može da izvršava svoju logičku funkciju za koju je programiran. SRAM tehnika dodaje najviše resursa u programabilni čip od svih nabrojanih tehnika, i jedina zahteva upotrebu eksterne memorije za čuvanje konfiguracije. Međutim, prednost je što je čip lako reprogramirati, pri čemu se broj reprogramiranja može izvršavati neograničen broj puta. Takođe, ova tehnika je jedna od najjeftinijih tehnika trenutno na raspolaganju. Ova tehnika je zbog ove dve navedene prednosti veoma popularna i često se koristi u programabilnim čipovima mnogih proizvođača.

Tehnike bazirane na fleš memoriji, EPROM i EEPROM su identične po ponašanju koje je veoma slično SRAM tehnici. Naime, i ovde se koristi tranzistor za ostvarivanje spoja (ili neostvarivanje spoja) pri čemu ovaj tranzistor ima dva gejta (gejt je termin koji se koristi kod CMOS tranzistora i u stvari je isto što i baza tranzistora). Tzv. plutajući (*floating*) gejt se puni nabojem ako tranzistor treba da ostvari spoj, pri čemu se ovaj naboj zadržava i po isključanju napajanja čipa, što ga čini sličnim tehnici sa osiguračem/antiosiguračem. Samim tim ovde nema potrebe za eksternim memorijskim čipom za čuvanje konfiguracije. Programabilni čip koji koristi ovu tehniku se može reprogramirati, s tim što se fleš memorija i EEPROM mogu elektronski reprogramirati, dok u slučaju EPROM memorije mora da se vrši prvo UV (ultraljubičasto) zračenje čipa (što nije brz proces, već može da traje i do nekoliko desetina minuta) radi brisanja prethodne konfiguracije pre nego što se može učitati nova konfiguracija čipa. Trenutno se danas još uvek koristi fleš memorija u pojedinim programabilnim čipovima, dok se EEPROM ne koristi jer troši više resursa od fleš memorije. EPROM se očigledno više ne koristi zbog kompleksne rekonfiguracije čipa. Važno je napomenuti da fleš memorija (isto važi i za EEPROM) zbog svojih fizičkih svojstava ograničava broj rekonfiguracija (tipično nekoliko stotina što je uglavnom dovoljno u praksi, ali ipak predstavlja ograničenje kojeg nema u SRAM tehnici).

Sa stanovišta poređenja tehnika postoji više aspekata na osnovu kojih se one mogu porediti - mogućnost rekonfiguracije čipa, dodatni resursi koje tehnike unose radi ostvarivanja konfigurabilnosti, zadržavanje konfiguracije nakon isključenja napajanja čipa i dr. Po pitanju rekonfiguracije najbolja je SRAM tehnika jer nudi najfleksibilniju rekonfiguraciju, dok je tehnika osigurača i antiosigurača najlošija po tom aspektu jer uopšte ne nudi mogućnost rekonfiguracije. Sa stanovišta dodatnih resursa je najlošija SRAM tehnika jer dodaje najviše resursa (dodaje se i eksterna memorija), dok ostale tehnike dodaju znatno manje resursa, pri čemu su najbolje po tom pitanju tehnike osigurača i antiosigurača. Nakon isključivanja napajanja čipa, sve navedene tehnike zadržavaju i dalje konfiguraciju, sem SRAM tehnike koja iz tog razloga zahteva eksternu memoriju. Trenutno se koriste tehnike bazirane na SRAM, fleš memoriji i antiosiguračima.

1.7 ASIC

ASIC (*Application Specific Integrated Circuit*) čipovi ne spadaju u programabilne čipove, međutim, u literaturi se često vrše poređenja FPGA čipova i ASIC čipova, pa će u ovoj sekciji ukratko biti objašnjeni ASIC čipovi. ASIC čipovi, kao što samo njihovo ime kaže, predstavljaju integrisane čipove specifične namene. Ovi čipovi se dizajniraju da obavljaju određenu funkciju ili skup funkcija i ne mogu se reprogramirati ili koristiti u druge svrhe (tj. obavljati druge funkcije) kao programabilni čipovi. Postoji više tehnika dizajniranja i projektovanja ASIC čipova - od tehnike koja koristi gradivne ćelije tako što se implementirana funkcija raspoređuje u gradivne ćelije koje se zatim prostorno razmeste i na osnovu tog rasporeda se pravi konačni integrisani čip do tehnike gde se na nivou tranzistora dizajnira željena funkcija koja se potom integriše u vidu integrisanog čipa. Očigledno prva tehnika ima kraći razvojni proces, ali slabije performanse u odnosu na drugu navedenu tehniku koja ima znatno duži razvojni proces, ali i bolje performanse. Prednost ASIC čipova u odnosu na programabilne čipove je što su optimizovani za funkciju koju obavljaju pa samim tim efikasnije troše resurse, nema neiskorišćenih resursa kao kod programabilnih čipova i naravno, ASIC čipovi daju bolje performanse. Međutim, nedostaci su veoma dug razvojni proces u poređenju sa programabilnim čipovima i znatno veći troškovi proizvodnje. Usled većih troškova proizvodnje ASIC čipovi se koriste samo za one funkcije za koje se zna da će biti implementirane u velikom broju uređaja tj. proizvođače se veliki broj takvih ASIC čipova čime se prevazilazi problem troškova proizvodnje. Da bi se prevazišao drugi problem obično se koriste FPGA čipovi za inicijalni razvoj, a potom se na bazi tako kreiranog rešenja formira ASIC čip tako što se u njega direktno kopira razvijeno rešenje (pri čemu se izostavljaju neiskorišćeni resursi) ili se ono dodatno optimizuje i potom formira ASIC čip. Međutim, treba napomenuti da su moderni FPGA čipovi po performansama veoma bliski ASIC čipovima tako da ASIC čipovi više ne prave veliku razliku u pogledu performansi implementacije. Ukratko, u odnosu na FPGA čipove ASIC čipovi nude bolje performanse (ta prednost ipak više nije tako izražena kao ranije), ali su mane nemogućnost reprogramiranja za drugu namenu i veća cena (ova mana ne postoji u slučaju masovne proizvodnje).