



Dizajn mikrosistema

Kako su aplikacije složenije i šire, zadatak dizajnera mikrosistema je sve teži. Najbolji način je posmatrati problem sa što višeg nivoa apstrakcije i koristiti moćne softvere u cilju low-level implementacije. Na taj način, dizajner se više može posvetiti rješavanj problema na višem nivou, ne gubeći vrijeme na naporan postupak implementacije ideje. Međutim, pomoćni alati su u mogućnosti jedino da prevedu i veoma malo optimizuju ono što je dizajnirano. Ne mogu od lošeg dizajna napraviti dobar. Dakle, i ovdje je presudan ljudski faktor.

Razvoj i proizvodnja mikrosistema je veoma složen i obiman process. U sljedećim sekcijama razmatrane su tehnologije koje se koriste, analizirane reprezentacije sistema, nivoi apstrakcije, razvojni tok, kao i ograničenja softverskih alata.

Integrirana kola (ICs)

- ▶ Fabrikacija integriranih kola (ICs)
- ▶ Podjela:
 - ▶ ASIC
 - ▶ Full-custom ASIC
 - ▶ Standard-cell ASIC
 - ▶ Gate array ASIC
 - ▶ Non-ASIC
 - ▶ Complex field-programmable devices (CPLD, FPGA)
 - ▶ Simple field-programmable devices (PLD)
 - ▶ SSI/MSI

Ne postoji najbolja tehnologija. Prilikom odabira treba uzeti u obzir površinu na čipu, brzinu, disipaciju snage i naravno cijenu.

Fabrikacija integriranog kola podrazumijeva "ređanje" slojeva dopiranog silicijuma, polisilicijuma, metala, silicijumdioksida.. na tankom silicijumskom vaferu. Neki od ovih slojeva formiraju transistor, neki metalnu vezu, neki kondenzator. Osnovni korak u fabrikaciji čipa je kreiranje specifičnih oblika u procesu poznatom kao litografija. U tu svrhu se koriste odgovarajuće specifične maske koje definišu oblik. Tehnologiju karakteriše najmanja moguća dimenzija tj dužina kanala tranzistora koja se spustila do reda nanometara.

Osnovna podjela integriranih kola se može izvršiti na ona koja je potrebno fabrikovati u clean room-u, i na ona koja se kreiraju "in the field"

Full-custom ASIC su kola specifične namjene koja se projektuju do nivoa layouta samih tranzistora. Dobijeno kolo je maksimalno optimizovano i u principu ima najbolje moguće performace. Sa druge strane, ovaj process je izuzetno složen i može se primijeniti na manje kompleksena kola. Nije praktično koristiti ovaj pristup za cjelokupne sisteme koji sadrže stotine miliona tranzistora. Osnovna namjena ovog

pristupa jeste dizajn jednostavnih ali baznih blokova koje se kasnije mogu koristiti kao komponente. Druga namjena bila bi realizacija osnovne ćelije nekog sistema koja će se replicirati mnogo puta. Kod ovog pristupa čitav postupak obavlja čovjek.

Standard-cell ASIC. U ovoj tehnologiji kolo se bazira na postojećim standardnim ćelijama za koje postoji layout. Omogućava rad na nivou logičkih kapija, što značajno olakšava razvoj. Dakle, standardne ćelije se povezuju u cilju ostvarivanja određene funkcionalnosti. Kao i kod full-custom ASIC-a, kolo je jedinstveno i process fabrikacije je isti.

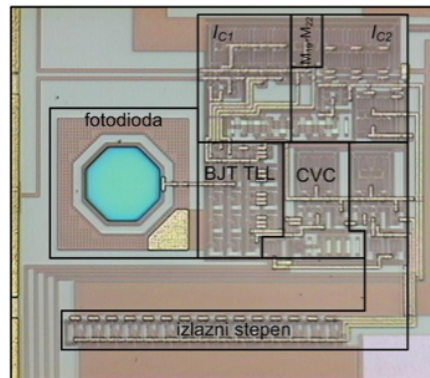
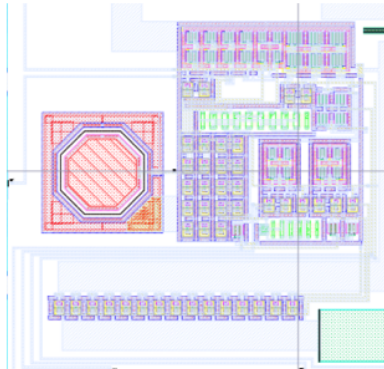
Gate-array ASIC. U ovoj tehnologiji kolo se kreira na bazi niza predefinisanih ćelija. Postoji samo jedna osnovna ćelija (za razliku od standard-cell ASIC-a). U principu, osnovna ćelija je veoma jednostavna logička kapija. Ćelije su raspoređene u obliku niza ili matrice. Kako je lokacija i tip osnovne ćelije predeterminisan, moguće je pojednostaviti process fabrikacije. Na osnovu bazne ćelije, u bibliotekama postoje i makroćelije, koje se takođe mogu koristiti u toku dizajniranja. Broj maski je značajno redukovano na svega nekoliko koliko je potrebno da se ostvare veze u novodefinisanom sistemu.

Complex field-programmable devices. U ovoj tehnologiji, uređaj se sastoji od niza generičkih logičkih ćelija (značajno složenijih nego kod gate-array ASIC-a) koje su međusobno povezane. I ćelije i veze su programabilne, i to in-field. Proces se obavlja uz pomoć jednostavnog programatora koji se povezuje sa računarom i odgovarajućeg softvera. CPLD ima sofisticiraniju logičku ćeliju, dok su veze centralizovane, sa nekoliko grupa koncentrisanih provodnih linija. Sa druge strane, logička ćelija FPGA čipa je jednostavnija, dok su veze distribuirane i značajno fleksibilnije.

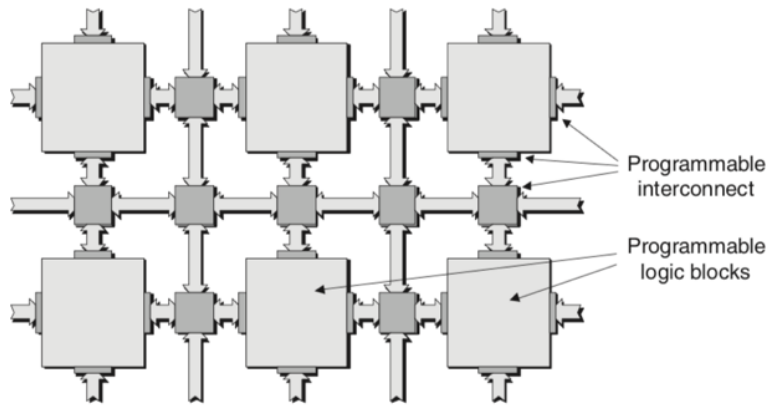
Simple field-programmable devices su programabilni uređaji sa jednostavnijom arhitekturom. To su dvodimenzioni nizovi sa and i or plane-om. Vezama se može ostaviti logička funkcija u formatu suma proizvoda. Uređaj sadrži PROM, PAL i PLA za programiranje and i or plane-a. Prevađeni su.

SSI/MSI komponente su integrisana kola koja se mogu kupiti i imaju određenu funkcionalnost. Moguće je koristiti ove komponente u izgradnji složenijeg diskretnog sistema.

Full-custom ASIC



FPGA



Poređenje tehnologija

- ▶ Parametri za poređenje tehnologija:
 - ▶ Površina
 - ▶ Brzina
 - ▶ Disipacija
 - ▶ Cijena (razvoj, proizvodnja, time-to-market)

ASIC čip će zauzeti manju površinu od FPGA čipa koji obavlja istu funkciju. Razlog je jednostavan, u cilju ostvarivanja programabilnosti potrebno je izdvojiti više prostora na čipu. Struktura je predeterminisana, i teško je očekivati da će se zahtjevi specifičnog dizajna optimalno poklopiti sa određenim standardnim čipom. Dio će svakako ostati neiskorišćen. Između standard-cell ASIC i gate array ASIC takođe postoji razlika kada je u pitanju potrebna površina, ali ta razlika je manja u odnosu na ASIC FPGA poređenje.

Brzina kola je uglavnom definisana propagacionim kašnjenjem. Brže kolo je svakako više poželjno. Na ovaj parameter prije svega utiče sam dizajn. FPGA čip je ovdje najlošiji izbor zbog svoje velike površine, kao i programabilnih veza koje imaju veoma izraženu parazitnu kapacitivnost i otpornost.

Disipacija. FPGA je opet najlošiji izbor.

Standard-cell ASIC je najbolji izbor u smislu tehničkih karakteristika. Sa druge strane, cijena i vrijeme proizvodnje su značajni nedostaci.

Poređenje FPGA - full-custom ASIC

Karakteristika	FPGA	Full-Custom
Vrijeme razvoja	Kratko	Dugo
Cijena proizvoda (u velikim serijama)	Visoka	Niska
Mogućnost izmjene nakon fabrikacije	Moguće	Nemoguće
Performanse	Srednje	Veoma visoke
Gustina pakovanja	Srednja	Veoma visoka
Potrošnja energije	Visoka	Niska
Minimalna veličina serije	Jedan	Velika
Kompleksnost dizajna	Srednja	Visoka
Kompleksnost testiranja	Srednja	Visoka
Vrijeme za ispravku greške	Sati	Mjeseci

Reprezentacija sistema

- ▶ Funkcionalna reprezentacija
- ▶ Strukturalna reprezentacija
- ▶ Fizička reprezentacija

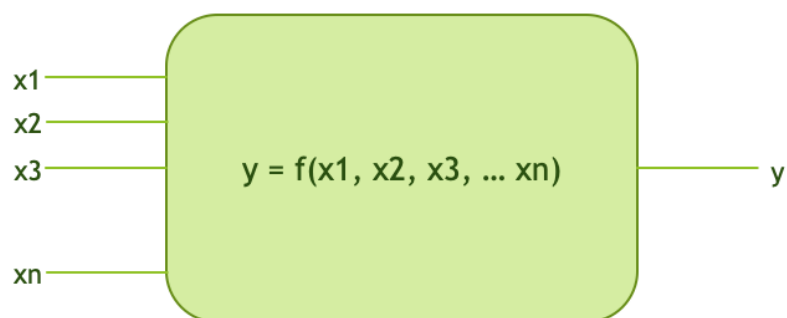
U toku razvoja sistema razlikuje se nekoliko faza: od inicijalne specifikacije do fizičke implementacije. Sistem se opisuje na više načina u zavisnosti od perspektive.

Funkcionalna reprezentacija sistema je opis sistema u smislu "šta sistem radi?". Tretira sistem kao crnu kutiju. Fokus se stavlja na vezu između ulaza i izlaza, tj posmatranje odziva za dati ulaz.

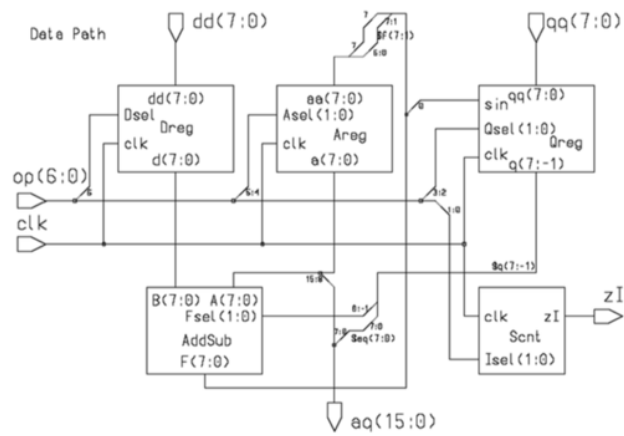
Strukturalna reprezentacija opisuje internu implementaciju (strukturu) sistema. Podrazumijeva eksplicitno specificiranje komponenti koje se koriste kao i veza među njima.

Fizička reprezentacija je fizički opis sistema i daje dodatne informacije strukturalnoj reprezentaciji. Definiše fizičke dimenzije komponenti, lokaciju na ploči ili vaferu, i fizičku vezu među njima. Primjer je layout, ili štampana ploča. Ovo je najpreciznija reprezentacija.

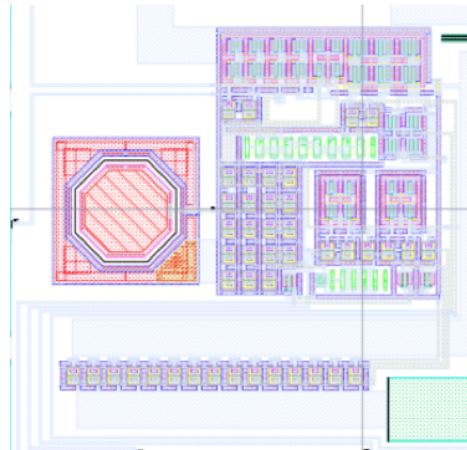
Funkcionalna reprezentacija



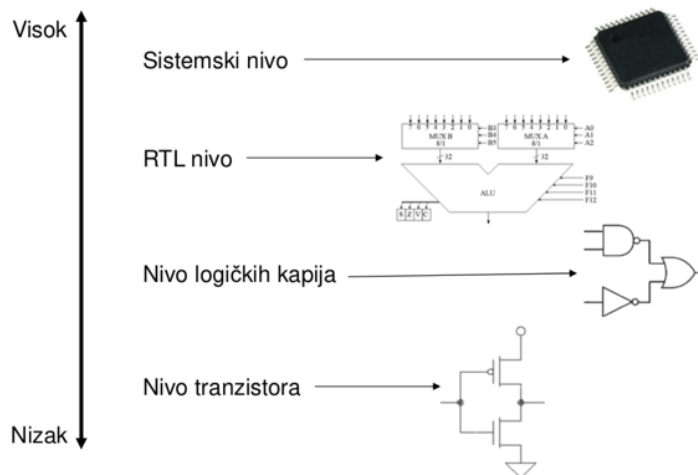
Strukturalna reprezentacija



Fizička reprezentacija



Nivoi apstrakcije



Osnovni metod za upravljanje kompleksnošću zadatka je opis sistema kroz nekoliko nivoa apstrakcije. Apstrakcija je pojednostavljen model koji uzima u obzir samo određene karakteristike, izuzimajući detalje. Viši nivo apstrakcije sadrži manje detalja, dok je niži nivo složeniji ali pouzdaniji.

Nivoi apstrakcije

Nivo apstrakcije	Tipični blokovi	Reprezentacija signala	Vremenska reprezentacija	Funkcionalni opis	Fizički opis
Nivo tranzisora	Tranzistori, otpornici, kondenzatori	Napon ili struja	Kontinulana funkcija	Diferencijalne jednačine	Tranzistor layout
Nivo logičkih kapija	And, or, xor, flip-flop	Logička 0 ili 1	Propagaciono kašnjenje	Boolean jednačine	Cell layout
RTL nivo	Adder, mux, register	Cijeli broj, stanje sistema	Takt	FSM	RT-level floor plan
Sistemski nivo	Procesor, memorija	Apstraktni tip podatka	Sekvenca događaja	Algoritam u C-u	IP-level floor plan

Razvojni zadaci

► Osnovni zadaci prilikom razvoja digitalnog sistema su:

- Sinteza
- Fizički dizajn
- Verifikacija
- Testiranje

Sistem se gradacijski razvija od abstraktnog high-level opisa do finalnog layouta. U svim fazama razvoja, potrebno je provjeravati funkcionalnost sistema u cilju usklađenosti sa postavljenim zadacima.

Sinteza

- ▶ Osnovni koraci:
 - High-level sinteza
 - RT-level sinteza
 - Gate-level sinteza
 - Mapiranje u skladu sa odabranom tehnologijom

Sinteza predstavlja opis sistema uz pomoć komponenti sa nižeg nivoa apstrakcije. Inicijalno, opis može biti na nivou funkcionalnog ili strukturalnog, dok je konačan strukturalni domen (netlist). To je dakle process koji se kreće od opisa na nivou visoke apstrakcije do opisa na nivou logičkih kapija uz upotrebu ćelija karakterističnih za izabranu tehnologiju.

High-level sinteza transformiše algoritam u RT-level funkcionalni opis (nivo register transfer operacija). Kako je ovaj process veoma složen, može se primijeniti samo na relativno jednostavne algoritme.

RT-level sinteza analizira funkcionalni RT-level opis i vrši strukturalnu implementaciju uz pomoć RT-level komponenti. Moguć je i određeni stepen optimizacije u cilju redukovanja broja komponenti.

Gate-level sinteza je slična RT-level sintezi osim što se koriste gate-level komponente. Nakon što se kreira inicijalno kolo, vrši se optimizacija u cilju ostvarivanja zahtjeva o dimenzijama kola i vremenskih ograničenja. U principu, koriste se generičke komponente, i proces ne zavisi od izabrane tehnologije.

Svaka tehnologija sadrži set primitivnih gate-level komponenti, koje mogu biti ćelije standardne biblioteke ili generičke logičke ćelije FPGA uređaja. Kako bi se implementiralo kolo na nivou logičkih kapija u određenoj tehnologiji, generičke komponente se moraju mapirati u ćelije izabrane tehnologije. Ovaj process se označava kao technology mapping.

Jasno je da čitav process ipak zavisi od izabrane tehnologije.

Fizički dizajn

- ▶ Osnovni zadaci:
 - ▶ Floor planning
 - ▶ Placement and routing
 - ▶ Circuit extraction

Fizički dizajn se može podijeliti u dvije faze. Prva je process prelaska sa strukturalnog na fizički domen, tj prevod netliste u layout. Druga faza podrazumijeva analizu električnih karakteristika kola.

Floor planning kreira layout na nivou procesora i na RT nivou. Vršiti se podjela sistema na veće funkcionalne blokove koji se smještaju na odgovarajuće lokacije u cilju vremenske i routing optimizacije.

Placement and routing kreira layout na nivou logičkih kapija. Taj layout podrazumijeva detaljan raspored ćelija kao i veza u sistemu (žica).

Nakon placement and routing faze, poznat je detaljan raspored i dužina svih veza, pa se jednostavno mogu izračunati odgovarajuće parazitne kapacitivnosti i otpornosti. Taj process se naziva circuit extraction. Dobijeni podaci se mogu iskoristiti za određivanje propagacionog kašnjenja.

Verifikacija

- ▶ Osnovni aspekti:
 - Funkcionalnost - funkcionalna verifikacija
 - Performanse - vremenska verifikacija
- ▶ Metode verifikacije:
 - Simulacija
 - Vremenska analiza
 - Formalna verifikacija
 - Hardverska emulacija

Verifikacija je process provjere da li dizajn ispunjava specificirane performanse. Ima dva aspekta: funkcionalnost i performanse.

Funkcionalna verifikacija provjerava da li sistem generiše željeni izlaz.

Performanse se uglavnom odnose na vremenska ograničenja sistema. Vremenska verifikacija provjerava da li je generisani odziv u zadatim vremenskim okvirima. Verifikacija se obavlja u različitim fazama dizajniranja sistema kao i na različitim nivoima apstrakcije.

Simulacija predstavlja kreiranje modela sistema, “izvršavanje” (simuliranje) tog modela za zadati set ulaznih parametara, i na kraju analizu dobijenih rezultata. Simulacija se može izvršiti na bilo kom nivou apstrakcije, u funkcionalnom i strukturalnom domenu. Omogućava posmatranje ponašanja sistema, bez njegove implementacije. Simulacija sistema se vrši za određeni set ulaznih podataka, što znači da ne može biti u potpunost pouzdana. Dodatni nedostatak predstavlja kompleksnost izračunavanja. Naime, operacije na nivou hardvera se izvršavaju paralelno, dok procesor instrukcije izvršava sekvencijalno.

Vremenska analiza se fokusira na vremenske aspekte kola. Izračunava maksimalnu

vrijednost propagacionog kašnjenja, kao i maksimalnu frekvenciju takt impulsa. Formalna verifikacija podrazumijeva matematičko modelovanje sistema. Obično se kreiraju dvije reprezentacije istog sistema i provjerava da li generišu jednak izlaz. Harverska emulacija podrazumijeva kreiranje prototipa. Čest primjer je kreiranje FPGA čipa koji emulira ponašanje složenog ASIC dizajna. Iako je FPGA značajno sporiji i jednostavniji, moguće je mnogo brže (nego simulacijom) ostvariti verifikaciju i povezati ga u složeniji sistem.

Testiranje

- ▶ Test pattern generation
- ▶ Design for test



Dok je verifikacija process utvrđivanja da li kolo obavlja zadatu funkciju prema utvrđenim kriterijumima i specifikaciji, testiranje je prosto testiranje specifičnog proizvedenog kola. Već je utvrđeno (kroz verifikaciju) da kolo obavlja zadatu funkciju, ali se treba uvjeriti da su pojedinačna kola fabrikovana korektno.

Nemoguće je (odnosno vrlo nepraktično) generisati sve moguće ulaze i posmatrati odziv kola. Iz tog razloga, koriste se specijalni algoritmi koji generišu set ulaznih podataka. Ovaj process se označava kao test pattern generation.

Testiranje manje složenih kola je jednostavno. Međutim, ukoliko je u pitanju složeniji sistem, potrebno je kreirati posebno kolo koje će se koristiti prilikom testiranja (design for test).

Razvojni tok za medium-sized FPGA

- ▶ Razvoj dizajn fajla i testbench-a
- ▶ Simuliranje rada sistema na osnovu dizajn fajla u cilju provjere njegove funkcionalnosti
- ▶ Sinteza
- ▶ Simulacija i vremenska analiza na osnovu netlist fajla koji je generisan nakon sinteze kako bi se provjerila korektnost sinteze i preliminarno timing
- ▶ Placement-and-routing
- ▶ Dodavanje pouzdanijih timing informacija u netlist fajl, simulacija i vremenska analiza u cilju provjere korektnosti sistema nakon placement-and-routing faze, kao i nova timing analiza
- ▶ Generisanje konfiguracionog fajla i programiranje uređaja
- ▶ Provjera funkcionalnosti fizičkog kola

Medium-sized dizajn je dizajn koji ne zahtijeva podjelu na particije prilikom projektovanja. Trenutni softveri za za sintezu i placement-and-routing mogu efektivno da obrade kolo ovog nivoa složenosti. Ova arhitektura nije trivijalna. Reda je jednostavnog procesora ili bus interfejsa. Razvojni tok je prikazan na sljedećoj slici. Dijagram počinje dizajn fajlom koji je obično RT-level opis kola. Zaseban fajl – testbench predstavlja virtualno eksperimentalno okruženje za simulaciju i verifikaciju. Sadrži kod koji generiše ulazne signale i prihvata (posmatra) odziv. Nakon što su ovi fajlovi kreirani, kolo se može konstruisati i verifikovati. Predstavljeni dijagram je idealan. U realnim uslovima, potrebno je obično nekoliko iteracija kako bi se prevazišao svaki od koraka.

