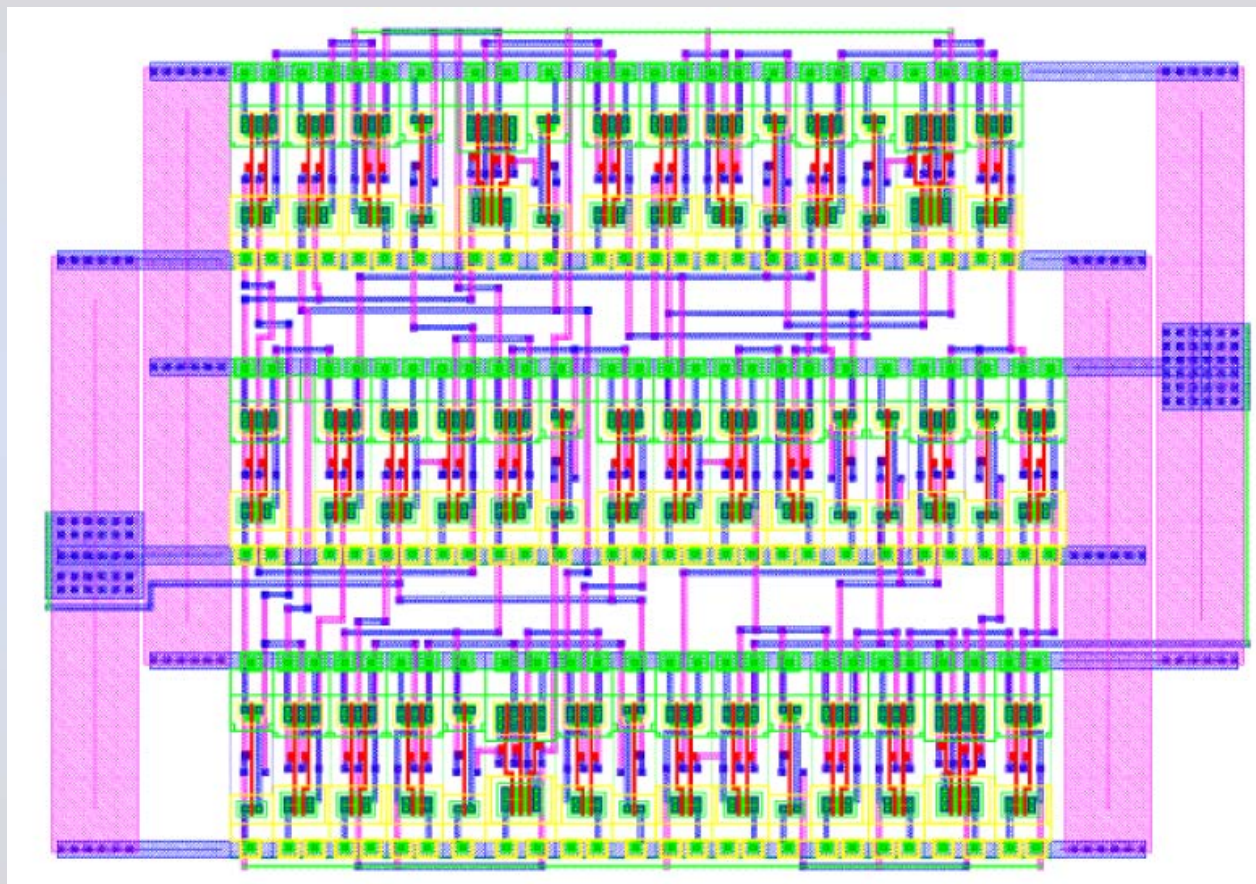


Fizički domen (kako je sistem napravljen)



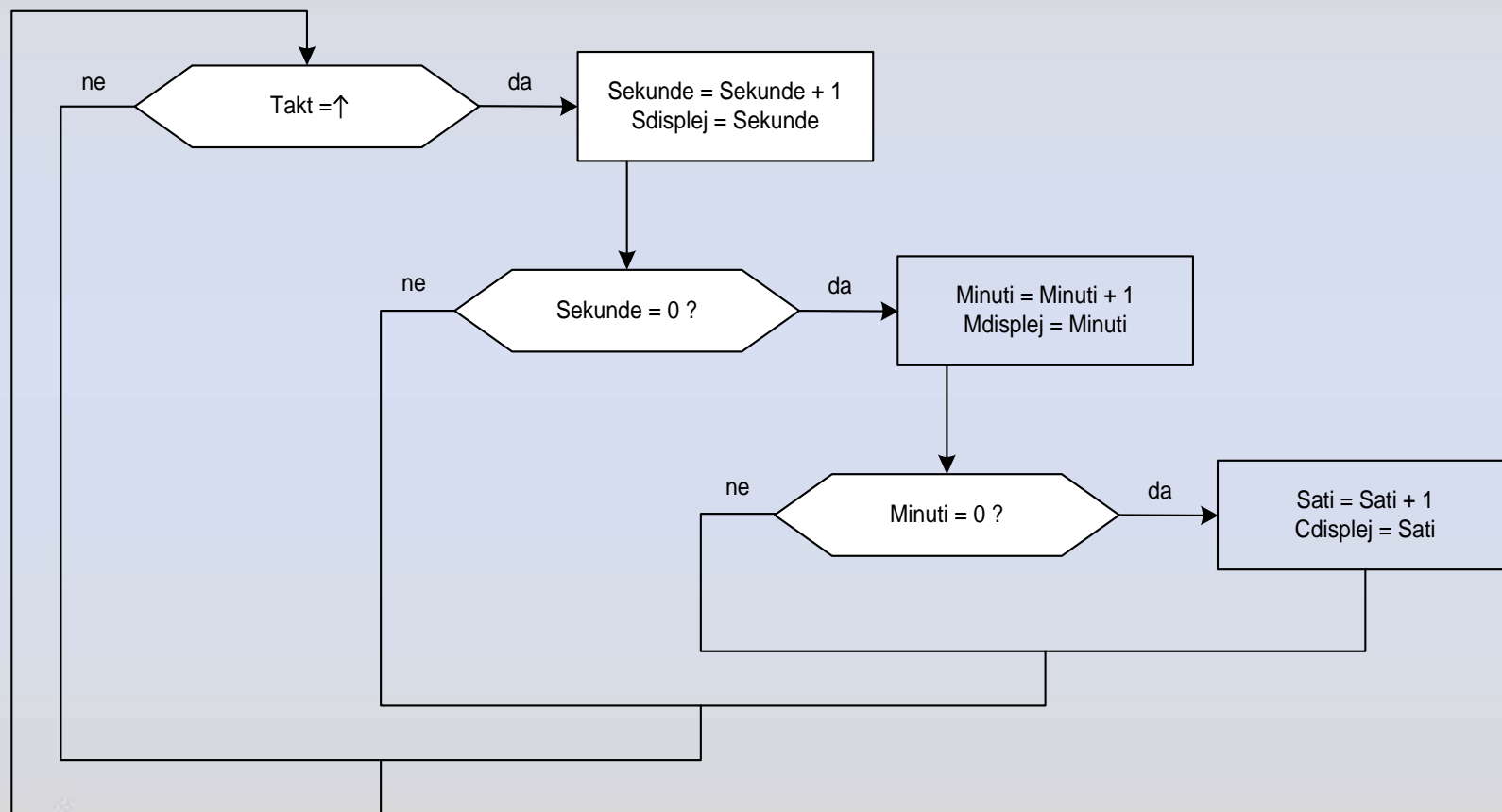
Primer: Projektovanje jednostavnog elektronskog budilnika

Problem: Razviti bihevioralnu, strukturnu i fizičku reprezentaciju projekta jednostavnog elektronskog budilnika.

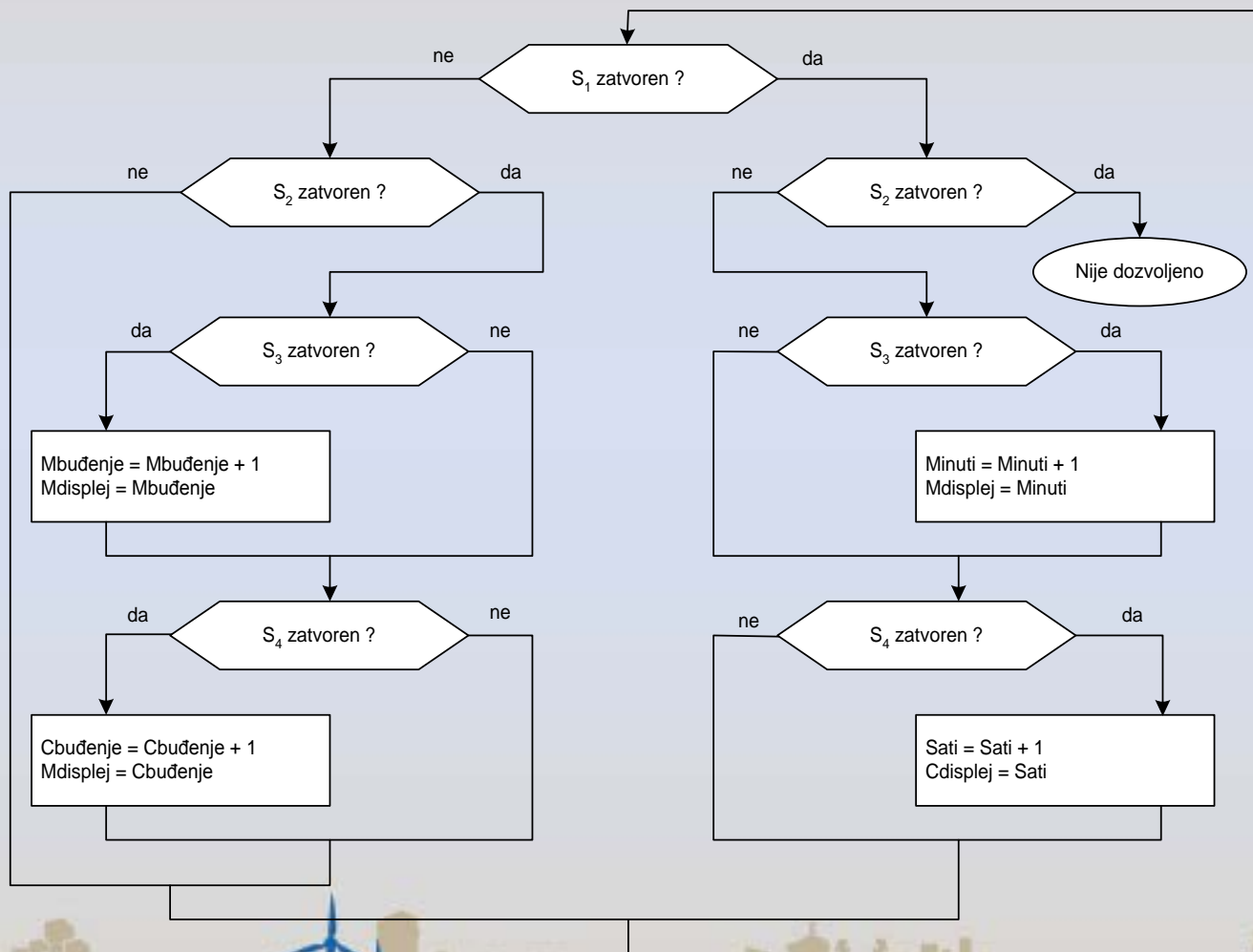
Rešenje:

- 1. Specifikacija sistema (polazni zahtevi):** Elektronski budilnik se sastoji od displeja sa tečnim kristalom (LCD) koji pokazuje sekunde, minute i sate i pet prekidača koji se koriste za: podešavanje tekućeg vremena (S1), podešavanje vremena alarma, tj. zvonjenja (S2), podešavanja minuta (S3), podešavanja sati (S4) i dozvole zvonjenja (S5). Kada je S1 zatvoren, omogućeno je podešavanje vremena pomoću prekidača S3 i S4. Svaki pritisak na S3 ili S4 uvećava minute ili sate za 1, što se može pratiti na displeju. Kada je S2 zatvoren, na sličan način pomoću prekidača S3 i S4, omogućeno je podešavanje vremena buđenja. U toku ovog podešavanja na displeju je prikazano vreme buđenja. Konačno, kada je S5 zatvoren, budilnik je aktiviran, a onda kada sat dostigne vreme buđenja emituje se jednoličan zvučni signal u trajanju od 5s.

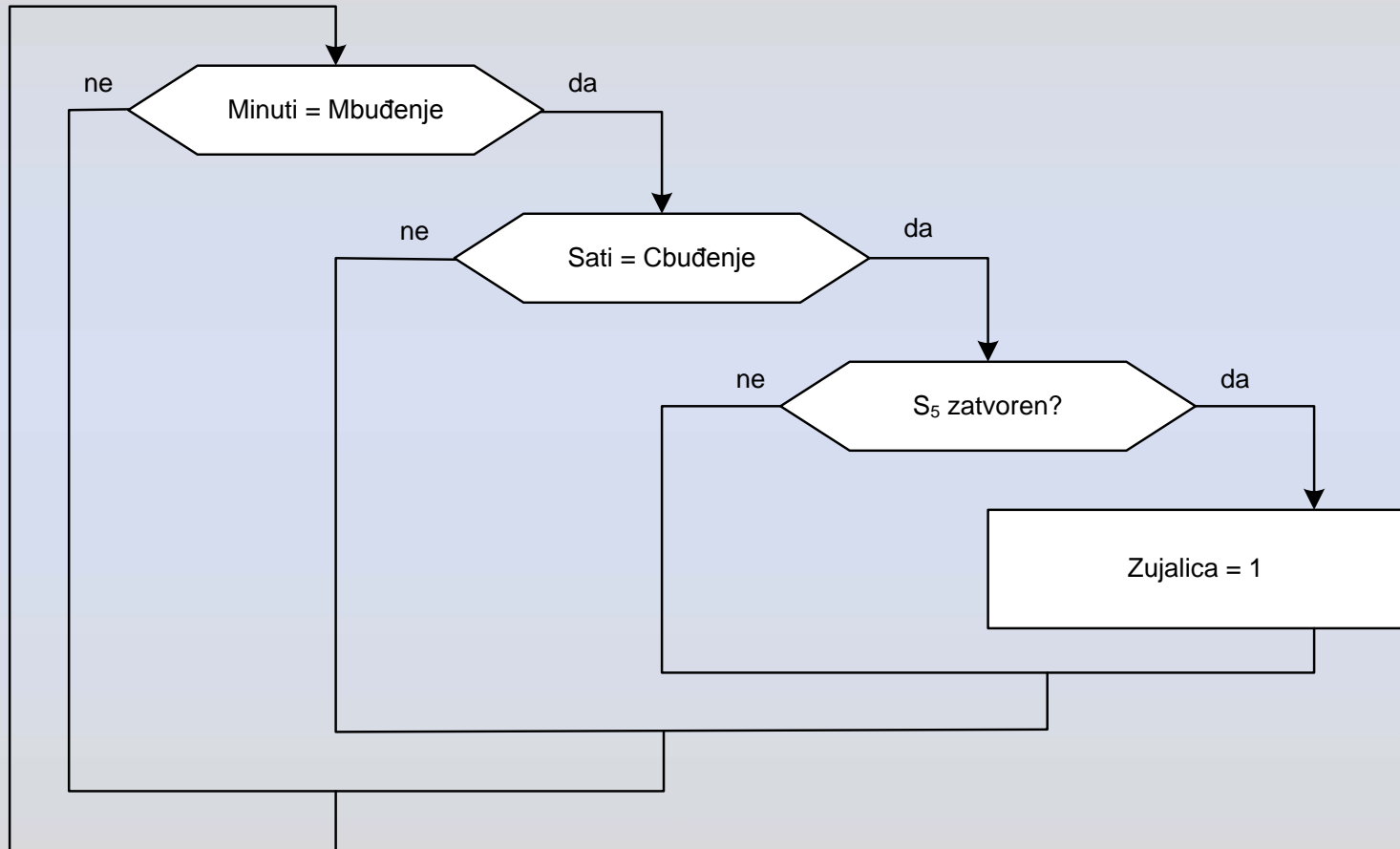
Bihevioralna reprezentacija elektronskog budilnika: proces **sat** (opisuje kako radi sat)



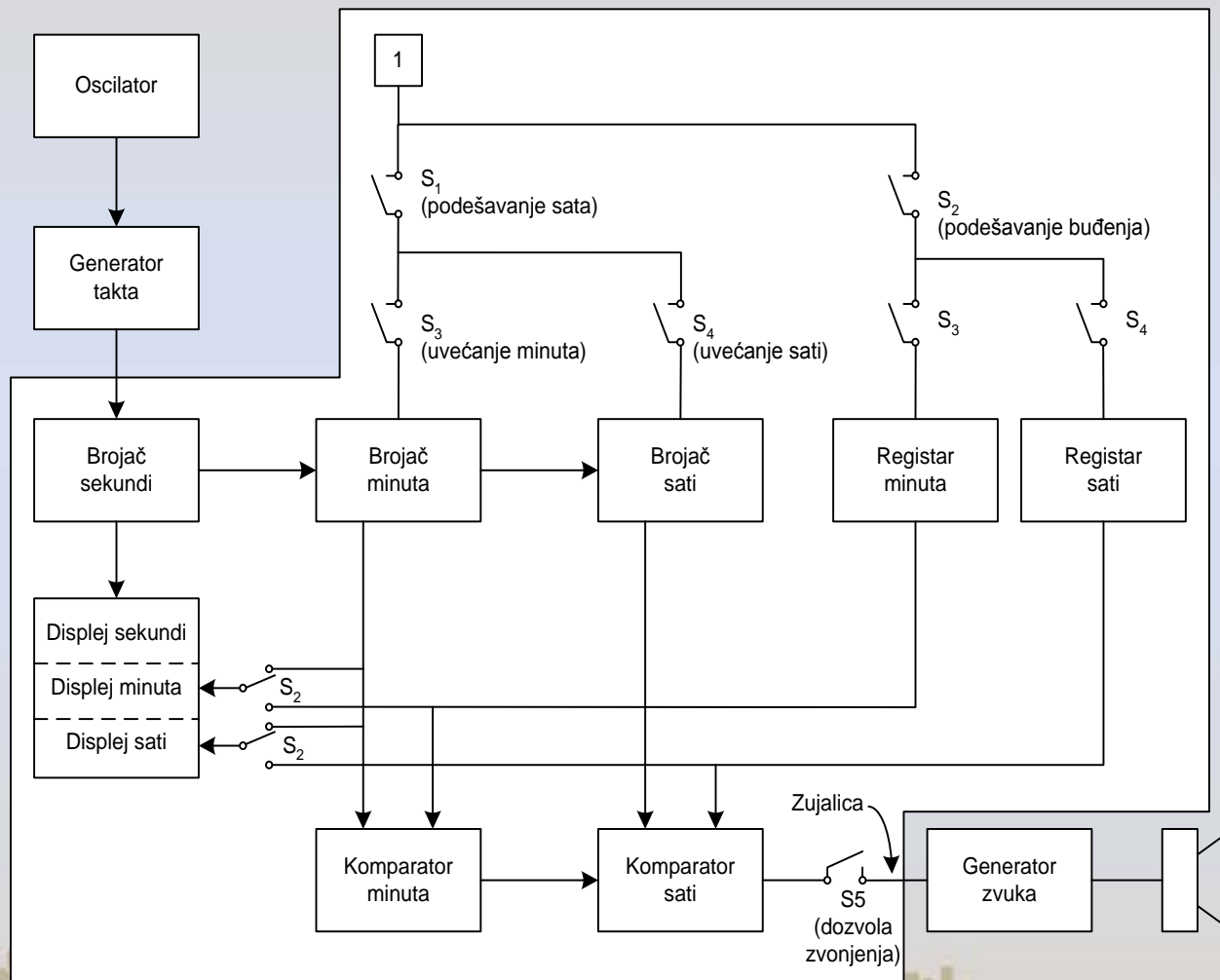
Bihevioralna reprezentacija elektronskog budilnika: proces **podešavanje** (opisuje kako se podešava vreme)



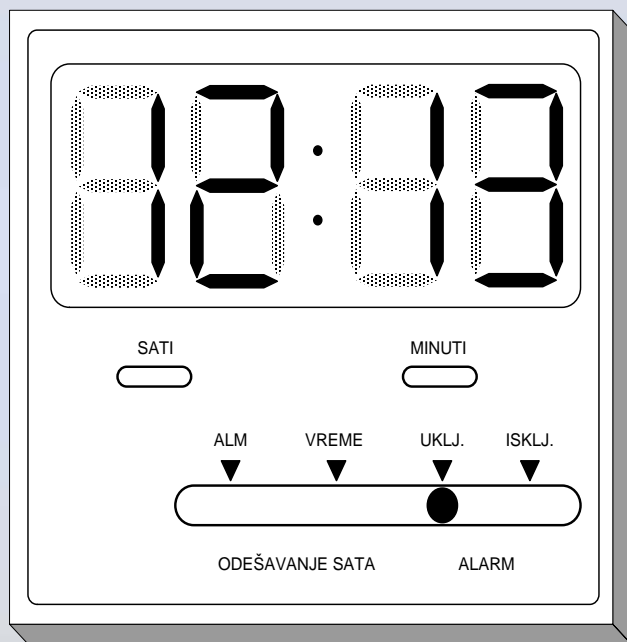
Bihevioralna reprezentacija elektronskog budilnika: proces **zvono** (opisuje uslove za aktiviranje zvona)



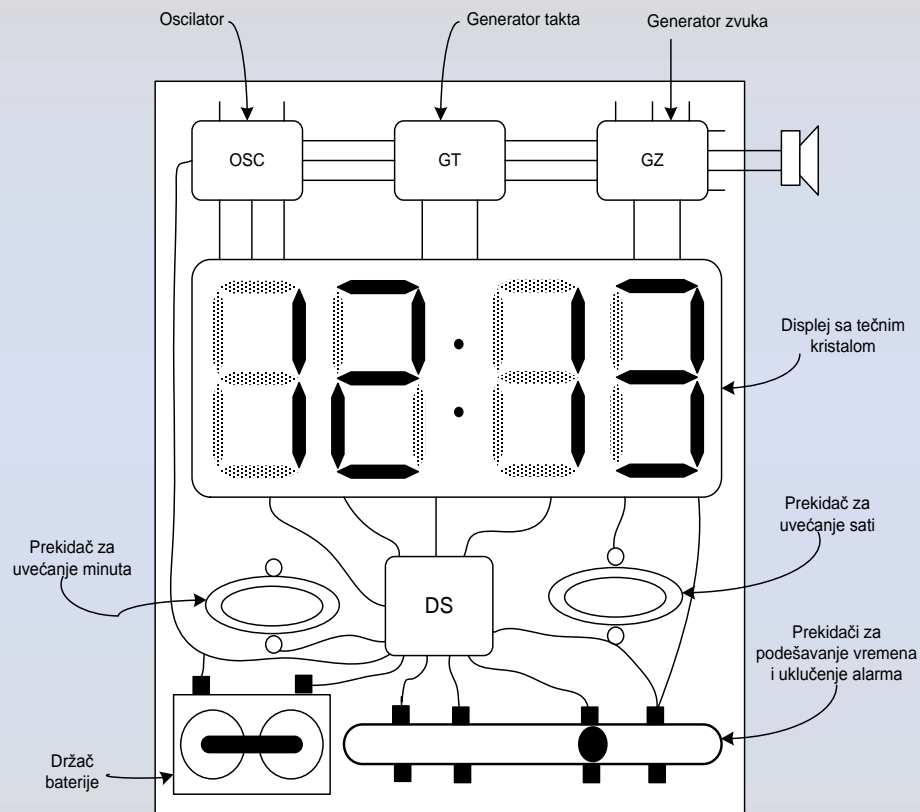
Strukturna reprezentacija elektronskog budilnika



Fizička reprezentacija elektronskog budilnika: (a) izgled prednje ploče; (b) izgled štampane ploče.

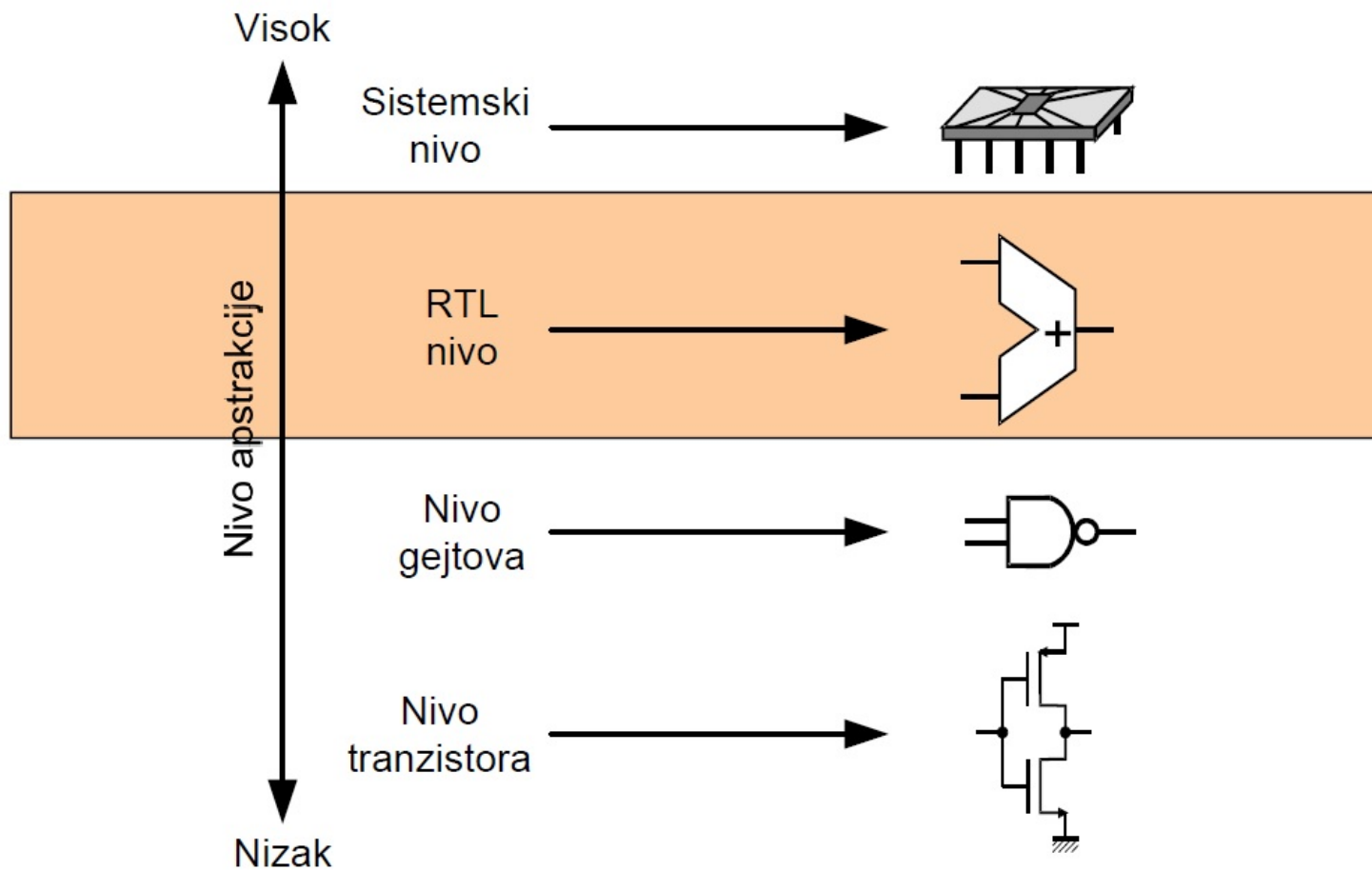


(a)

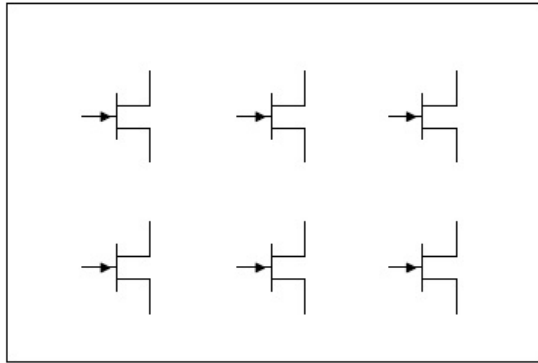


(b)

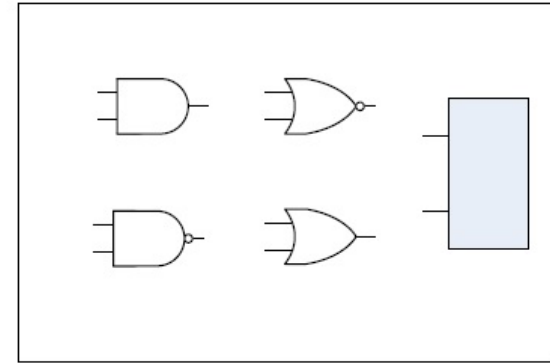
Nivoi apstrakcije



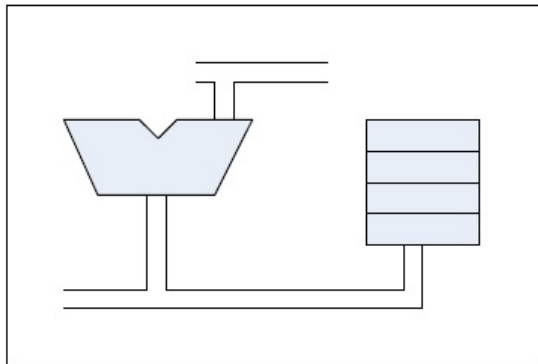
Nivoi apstrakcije



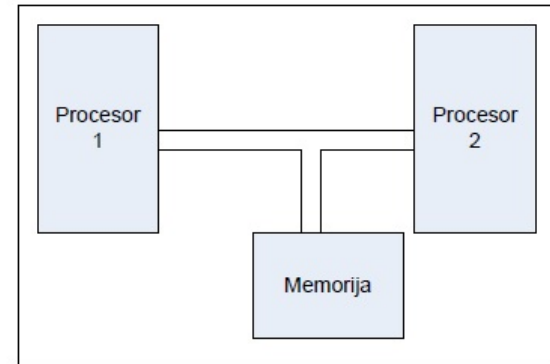
Nivo tranzistora



Nivo gejtova



Registarski (RTL) nivo

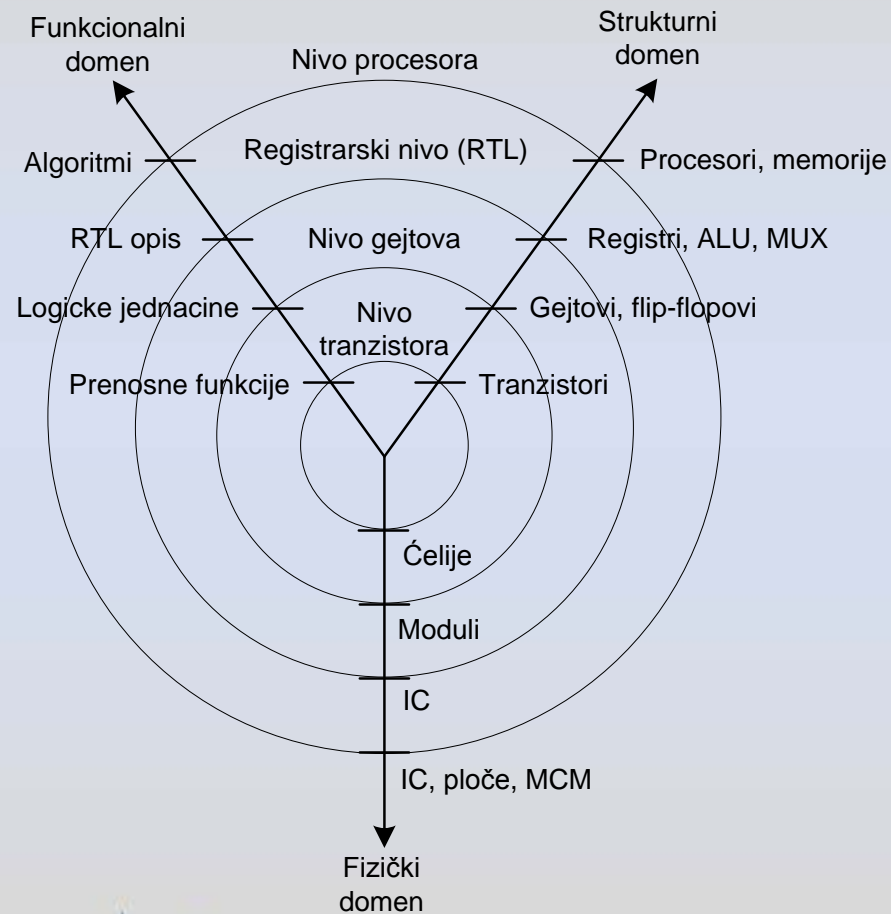


Nivo procesora

Nivoi apstrakcije

Nivo apstrakcije	Bihevioralna reprezentacija	Strukturne komponente	Fizički objekti
Tranzistorski nivo	Diferencijalne jednačine, U-I Dijagrami	Tranzistori, otpornici, kondenzatori	Analogne i digitalne ćelije
Nivo gejtova	Logičke funkcije, konačni automati	Logička kola, flip-flopovi	Moduli, funkcionalne Jedinice
Registarski nivo (RTL)	Algoritmi, dijagrami toka, ASM dijagrami	Sabirači, komparatori, registri, brojači, multiplekseri	Čipovi
Sistemska nivo	Izvršna specifikacija, programi	Procesori, kontroleri, memorije, ASIC kola	Štampane ploče, mikročip-moduli

Y dijagram



Proces projektovanja

- Sinteza
- Fizičko projektovanje
- Verifikacija
- Testiranje



Sinteza

- Proces **razrade** čiji je cilj da se uz pomoć komponenti raspoloživih na **nižem** nivou apstrakcije realizuje opis sistema **višeg** nivoa apstrakcije:

- **Sinteza visokog nivoa:**

algoritam -> funkcionalni RTL opis (VHDL)

- **RTL sinteza:**

funkcionalni RTL opis -> strukturni RTL opis

Projektant

CAD alat

- **Logička sinteza:**

strukturni RTL opis -> mreža logičkih kola

- **Automatizovano tehnološko mapiranje:**

Logička mreža -> mrežu ćelija raspoloživih u ciljnoj tehnologiji

Fizičko projektovanje

Dva aspekta:

- transformacija strukturne u fizičku reprezentaciju (*layout*)
- analiza i optimizacija električnih karakteristika kola

Aktivnosti:

- Prostorno planiranje
- Raspoređivanje
- Povezivanje



Verifikacija

- Provera da li ostvareni dizajn (projekat) zadovoljava postavljene funkcionalne zahteve i zadate performanse.
- Sprovodi se nakon svakog koraka razrade (sinteze), kao i nakon obavljenog fizičkog projektovanja.

Dva aspekta:

- **Funkcionala verifikacija:** da li sistem generiše očekivani odziv na zadatu pobudu ?
- **Vremenska verifikacija** (ili *verifikacija tajminga*) znači proveru da li sistem generiše odziv u granicama postavljenih vremenskih ograničenja.



Metodi za verifikaciju

- Simulacija
- Vremenska analiza
- Formalna verifikacija
- Hardverska emulacija



Simulacija

- Da li projektovani sistem korektno realizuje funkcije definisane polaznom specifikacijom?
- Omogućava otkrivanje grešaka pre nego što je sistem realizovan
- Nemogućnost potpune verifikacije

Xilinx ISE Design Suite 14.5

The screenshot displays the Xilinx ISE Design Suite 14.5 simulation environment. The main window shows a timing diagram for a simulation of 1,000,000 ns. The diagram plots the values of signals x, y, z, and f over time. The signals x, y, and z are shown as green waveforms, while f is a constant low signal. The time axis is marked from 0 ns to 900 ns in 100 ns increments. The console window at the bottom shows the following text:

```
ISim P.58f (signature 0x7708f090)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

The console window also includes a status bar at the bottom with the text "Sim Time: 1,000,000 ps".

Vremenska analiza

- Verifikacija fokusirana na **tajming**.
- Analiza strukture kola kako bi se odredile sve moguće putanje signala od ulaza do izlaza, proračunala propagaciona kašnjenja duž ovih putanja i odredili relevantni vremenski parametri kola, kao što su: maksimalno propagaciono kašnjenje i maksimalna frekvencija taktnog signala.



Formalna verifikacija

- Provera ekvivalentnosti: upoređivanje dve formalne reprezentacije sistema kako bi se ustanovilo da li imaju istu funkciju.
- Primenjuje se nakon sinteze kako bi se proverilo da li je funkcija sintetizovanog kola identična funkciji polaznog funkcionalnog opisa.
- Garantuje potpunu verifikaciju.
- Primena ograničena na male sistema, zbog velike kompleksnosti pratećih matematičkih izračunavanja.



Hardverska emulacija

- Fizičku realizacija prototipa koji će oponašati rad sistema koji se projektuje.
- Primer: realizacija FPGA kola koje emulira rad ASIC kola.



Testiranje

- **Testiranje nije isto što i verifikacija!**
- Testiranje - pronalaženja fizičkih defekata (kvarova) u gotovom proizvodu koji mogu nastati tokom proizvodnje.
- Sprovodi se nakon fabrikacije.
- Na ulaze kola se dovode "test-vektori", a generisani izlaz se poredi sa očekivanim.
- Generisanje test sekvenci - primenom posebnih algoritama, za dato kolo, se kreira što manji skup test-vektora kojima će kolo biti što potpunije testirano.



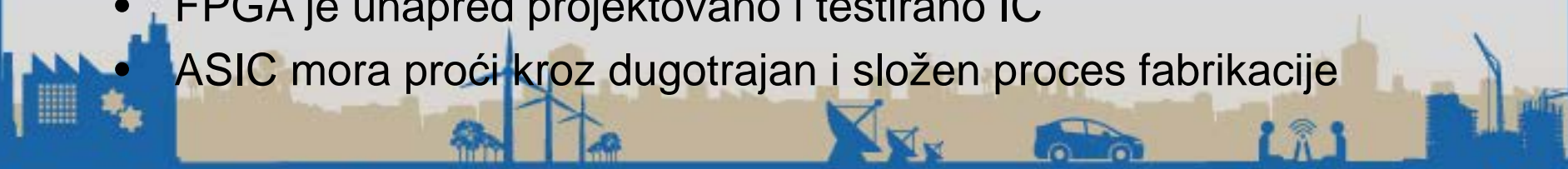
CAD alati

- CAD (***Computer-Aided Design***) alati – softverski alati za **projektovanje pomoću računara**, tj. za automatizaciju pojedinih faza projektovanja.
- Podela:
 - alati za opis i modelovanje
 - alati za sintezu
 - alati za verifikaciju i simulaciju
 - alati za raspoređivanje i povezivanje
 - alati za generisanje test sekvenci

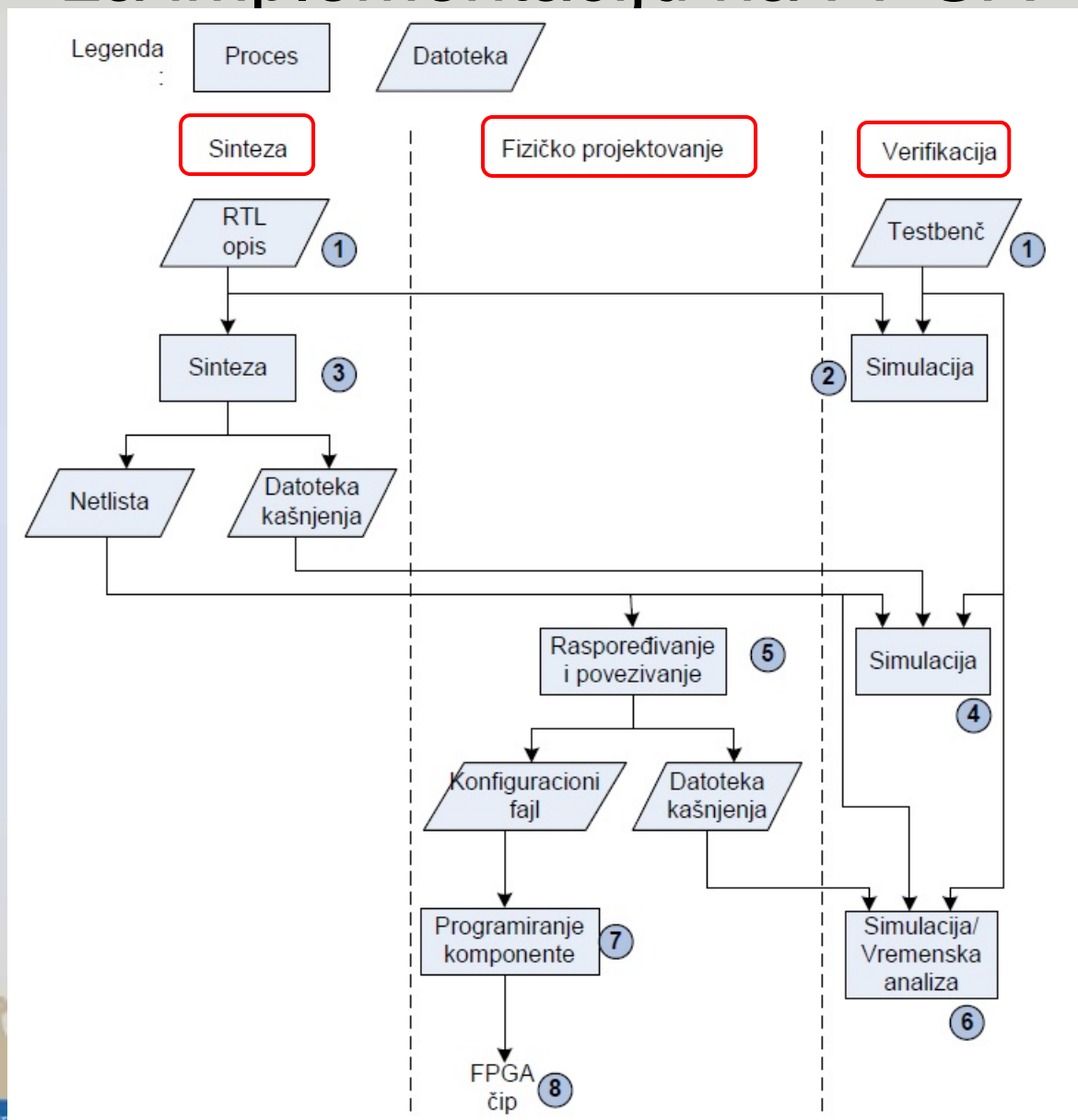


Tok projektovanja

- Projektovanje digitalnih kola i sistema je proces **razrade (sinteze)** i **verifikacije**, kojim se polazni apstraktni opis visokog nivoa postepeno transformiše u detaljni strukturni opis niskog nivoa.
- Tok projektovanja (niz aktivnosti) zavisi od:
 - Složenosti sistema koji se projektuje:
 - CAD alati za sintezu - samo za sisteme srednje složenosti (2000 - 50000 gejtova)
 - Veći sistemi se dele na manje delove koji se nezavisno sintetišu
- Ciljne implementacione tehnologije:
 - FPGA je unapred projektovano i testirano IC
 - ASIC mora proći kroz dugotrajan i složen proces fabrikacije



Tok projektovanja sistema srednje složenosti za implementaciju na FPGA



Tok projektovanja sistema srednje složenosti za implementaciju na FPGA

Tok projektovanja (**u idealnom slučaju**) podrazumeva sledeće korake:

1. **Razvoj** dizajn-datoteke i testbenča.
2. **Funkcionalna simulacija**, čiji je cilj verifikacija polaznog RTL opisa.
3. **Sinteza**. Rezultat sinteze je strukturni opis na nivou gejtova. Rezultujuća logička mreža je opisana tekstualno u obliku tzv. *netliste*.
4. Datoteka koja sadrži netlistu koristi se za simulaciju i vremensku analizu, kako bi se verifikovala funkcionalna korektnost sintetizovane logičke mreže i obavila **preliminarna provera tajminga**.
5. **Raspoređivanje i povezivanje**. Gejtovi i veze iz netliste se preslikavaju na fizičke elemente i fizičke veze u FPGA kolu.
6. **Ekstrakcija preciznih podataka o tajmingu**. Netlista, proširena ovim podacima koristi se za simulaciju i vremensku analizu u cilju verifikacije korektnost raspoređivanja i povezivanja i provere da li kolo zadovoljava postavljena vremenska ograničenja.
7. Generisanje konfiguracione datoteke i **programiranje FPGA kola**.
8. **Verifikacija rada fizičke komponente**.

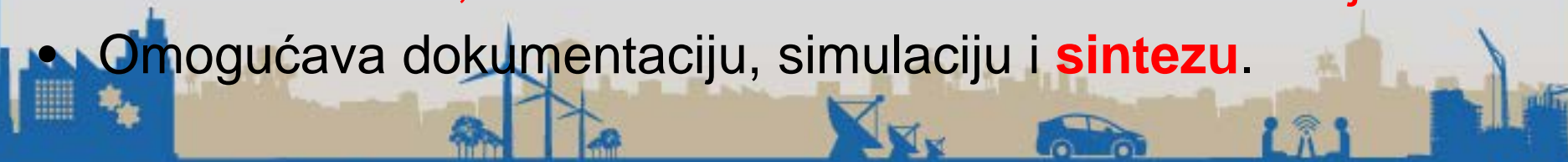
Tok projektovanja sistema srednje složenosti za implementaciju na FPGA

- Opisani tok projektovanja odgovara **idealnom scenariju**, kad polazni RTL opis uspešno prolazi funkcionalnu simulaciju i sintetizovana struktura zadovoljava unapred postavljena vremenska ograničenja.
- **U realnosti**, tok projektovanja može sadržati više iteracija radi korekcije grešaka ili problema u tajmingu. To obično zahteva reviziju polaznog RTL opisa i fino podešavanje parametara u softverima za sintezu, raspoređivanje i rutiranje.



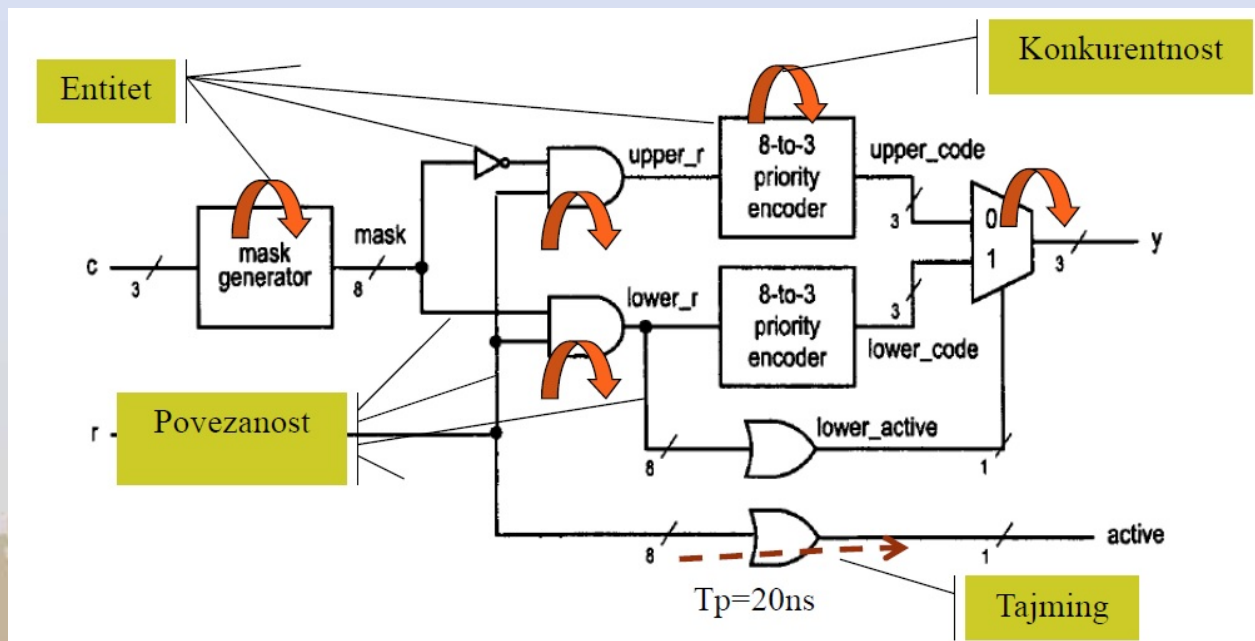
Jezici za opis hardvera (HDL)

- HDL - *Hardware description language*
- **Način za razmenu informacija između projektanta i CAD alata.**
- Treba da omogući:
- Verno i precizno modelovanje i opisivanje kola i sistema, bilo da se radi o postojećem kolu odnosno sistemu ili o kolu/sistemu koji se projektuje i to na željenom nivou apstrakcije, kako u funkcionalnom tako i u strukturnom domenu.
- **“interfejs” između projektanta i CAD alata**
- HDL nije isto što i programski jezik (Fortran, C, Java...)
- Slična sintaksa, ali različita sematika i način korišćenja.
- Omogućava dokumentaciju, simulaciju i **sintezu**.



Karakteristike savremenih HDL-ova

- Podršku za kocepte: entitet, povezanost, konkurentnost i tajming.
- Mogućnost predstavljanja propagacionih kašnjenja i informacije o tajmingu
- Mogućnost opisa strukture kola (povezanosti)
- Mogućnost funkcionalnog opisa kola
- Mogućnost modelovanja na gejtovskom i RTL nivou
- Podrška za hijerarhijsko projektovanja



HDL vs Programski jezici

Programski jezici :

- **Sekvencijalni** model izračunavanja
- Namenjeni za opis **algoritama**
- Izvršavaju se na **računaru**



HDL:

- Sekvencijalni i **konkurentni** (paralelni) model izračunavanja
- Namenjeni za opis **hardvera**
- “Izvršavaju” se na **integriranim kolima** (npr. FPGA)

HDL vs CAD vs IC

- Projektant - HDL - CAD - IC - Korisnik

VHDL

Xilinx ISE (Integrated
Software Environment)
Design Suite

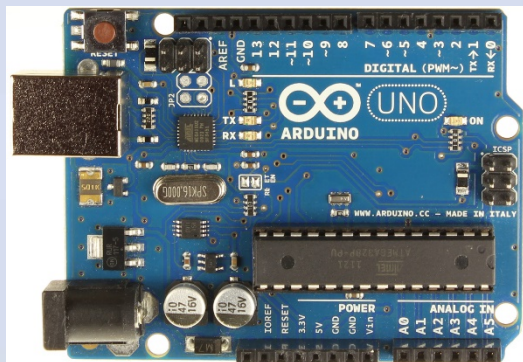
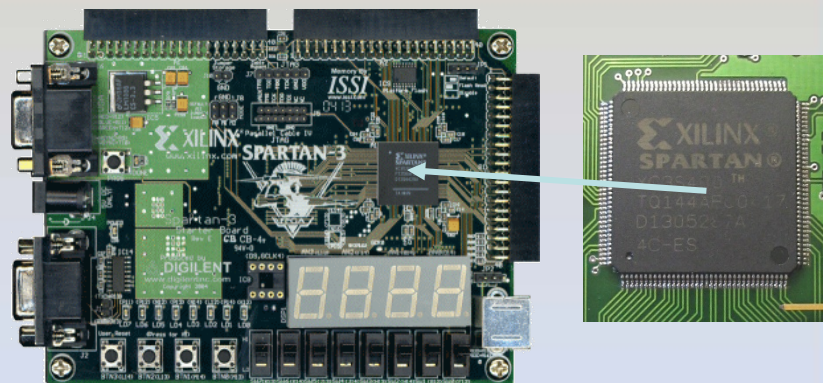
FPGA

(Basys™2 Spartan-3E
FPGA Board)

FPGA vs Arduino vs Raspberry Pi

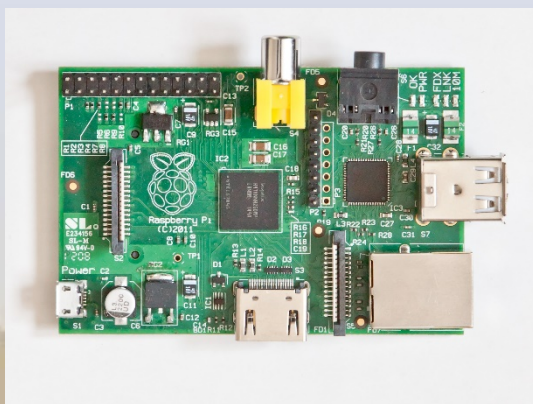
FPGA:

- **Integrirano kolo**
- Projektovanje pomoću HDL-a
- “Kontroliše” se HARDVER
- “Najbrže” (u zavisnosti od namene)



Arduino:

- **Mikrokontroler**
- Programiranje (C ili C++)
- “Kontroliše” se softver
- “Brže” (u zavisnosti od namene)



Raspberry Pi:

- **Kompijuter**
- Programiranje (Python)
- Kontroliše se softver **preko OS-a**
- “Brzo” (u zavisnosti od namene)

Oblasti primene FPGA

FPGA can be used to solve **any problem which is computable**.

Their advantage lies in that they are sometimes significantly **faster** for some applications due to their **parallel nature** and optimality in terms of the number of gates used for a certain process.

- <https://www.youtube.com/watch?v=6Y2ZTIjeeWY&list=PLgPjbEo1PPXuV81NWfYfcR0o5ibFgHUD9&index=8>

Specific applications of FPGAs include:

- digital signal processing,
- **ASIC prototyping**, (<https://www.youtube.com/watch?v=vxSvQ-lcmHM>)
- medical imaging,
- computer vision,
- speech recognition,
- cryptography,
- bioinformatics,
- computer hardware emulation,
- as a part of embedded systems, ...