

# 1. PROJECT MANAGER

Ovo je deo za **organizaciju projekta**.

## Settings

Tu se podesava projekat:

- koji je FPGA/board izabran
- koji je top modul
- jezik (Verilog/VHDL)
- simulator
- putanje do fajlova

## Add Sources

Dodavanje fajlova u projekat:

- design source: Verilog/VHDL fajlovi
- constraints: .xdc
- simulation source: testbench fajlovi

## Language Templates

Gotovi šabloni za Verilog/VHDL konstrukcije.

Korisno kada se ne pamti sintaksa.

## IP Catalog

Biblioteka gotovih IP blokova:

- FIFO
- Block RAM
- Clocking Wizard
- AXI blokovi
- DSP jezgra itd.

Dakle, Project Manager je praktično:

**pravljenje i organizacija projekta.**

---

# 2. IP INTEGRATOR

Ovo služi za **blokovski dizajn** umesto pisanja sveg HDL koda ručno.

## Create Block Design

Pravi se grafički dizajn povezivanjem blokova.

### Open Block Design

Otvora postojeći blokovski dizajn.

### Generate Block Design

Vivado iz blokovskog prikaza generiše HDL i pomoćne fajlove.

Ovo se često koristi kada radiš:

- Zynq procesor
- AXI magistralu
- gotove IP blokove
- kompleksnije SoC sisteme

Za početne vežbe sa logičkim kolima ovo ne treba.

---

## 3. SIMULATION

Ovo je faza **provere ponašanja dizajna pre implementacije na FPGA**.

### Run Simulation

Pokreće simulaciju.

Tu se proveraa da li Verilog radi kako treba.

U simulaciji se gleda:

- ulaze i izlaze kroz vreme
- da li AND, OR, NOT rade kako treba
- da li FSM prelazi između stanja kako treba
- da li brojač broji pravilno

Važno:

simulacija proverava **funkcionalnost**, ali još ne znači da je dizajn fizički smešten u FPGA.

---

## 4. RTL ANALYSIS

RTL = **Register Transfer Level**.

Ovo je nivo na kome Vivado analizira strukturu HDL opisa.

### Run Linter

Proverava kod i upozorava na moguće probleme:

- neiskorišćeni signali
- loše deklaracije
- sumnjive konstrukcije

- potencijalne greške u sintaksi ili stilu

### **Open Elaborated Design**

Vrlo važna stavka.

Ovde Vivado “razume” kod i prikazuje kako logika izgleda na RTL nivou:

- ulazi/izlazi
- logičke veze
- hijerarhija modula
- šematski prikaz

Ovo je korak između “napisao sam kod” i “pretvaram ga u FPGA resurse”.

- **Simulation:** da li se ponaša kako želim?
  - **RTL Analysis:** kako Vivado razume moj kod kao digitalnu logiku?
- 

## **5. SYNTHESIS**

Ovo je jedna od najvažnijih faza.

### **Run Synthesis**

Sinteza prevodi HDL kod u stvarne FPGA resurse:

- LUT
- flip-flop
- block RAM
- DSP
- clock elementi

Drugim rečima:

Verilog/VHDL se pretvara u mrežu realnih hardverskih elemenata koji postoje u FPGA-u.

### **Open Synthesized Design**

Posle sinteze može da se otvori rezultat i proveriti:

- šematski prikaz
- iskorišćenje resursa
- hijerarhiju
- preliminarne informacije o logici

Sinteza odgovara na pitanje:

**Šta će od mog koda stvarno biti napravljeno u FPGA-u?**

---

## 6. IMPLEMENTATION

Ovo je sledeći korak posle sinteze.

### Run Implementation

Ovde Vivado radi fizičku realizaciju u konkretnom FPGA čipu:

- mapiranje logike na konkretne LUT-ove i FF-ove
- placement = gde će koji element biti smešten
- routing = kako će se signali fizički povezati kroz FPGA mrežu

Tu se proverava i:

- timing
- da li signali stižu na vreme
- da li postoje kritične putanje

### Open Implemented Design

Otvora konačno implementiran dizajn:

- fizički raspored na čipu
  - rutiranje
  - timing izveštaji
  - iskorišćenje resursa
  - **Synthesis** = logičko prevođenje
  - **Implementation** = fizičko smeštanje i povezivanje u FPGA
- 

## 7. PROGRAM AND DEBUG

Ovo je završna faza.

### Generate Bitstream

Pravi `.bit` fajl, odnosno konfiguracioni fajl kojim se FPGA programira.

Bez ovoga ne može da ubaci dizajn u ploču.

### Open Hardware Manager

Otvora alat za komunikaciju sa FPGA pločom:

- povezuje ploču preko JTAG
- učitava bitstream
- programira uređaj
- radi debug ako koristiš ILA/VIO

Dakle:

- **Generate Bitstream** = napravi fajl za programiranje
  - **Hardware Manager** = pošalji taj fajl na FPGA ploču
- 

## Kako ide ceo proces dizajna redom

Najjednostavniji tok je:

1. napišeš Verilog/VHDL
  2. dodaš fajlove u projekat
  3. napraviš testbench
  4. pokreneš simulaciju
  5. proveriš RTL / elaborated design
  6. pokreneš synthesis
  7. pokreneš implementation
  8. generišeš bitstream
  9. programiraš FPGA
- 

## Znacenje svake faze!

- **Project Manager** – pravljenje i podešavanje projekta
  - **IP Catalog / IP Integrator** – gotovi blokovi i blokovski dizajn
  - **Simulation** – provera ponašanja kola
  - **RTL Analysis** – pregled kako je kod shvaćen kao logika
  - **Synthesis** – prevođenje koda u FPGA resurse
  - **Implementation** – fizičko raspoređivanje i povezivanje u čipu
  - **Generate Bitstream** – pravljenje fajla za FPGA
  - **Hardware Manager** – upis dizajna u ploču i debug
- 

Zamislamo da projektujemo kuću:

- **HDL kod** = nacrt kuće
- **Simulation** = proveravaš na papiru da li raspored ima smisla
- **RTL Analysis** = vidiš grubi plan prostorija

- **Synthesis** = pretvaraš plan u listu stvarnih elemenata: zidovi, vrata, prozori
- **Implementation** = određuješ gde tačno šta stoji na placu
- **Bitstream** = završni paket za izgradnju
- **Hardware Manager** = stvarno “useliš” projekat u FPGA