

FPGA – šta je, zašto se koristi i kakva je veza sa HDL kodom

Uvod u programabilni digitalni hardver

Aleksandar Peulić

Zašto nam je potreban FPGA?

Klasični digitalni sistemi imaju dva pristupa:

- Mikroprocesor (CPU)
- ASIC (Application Specific Integrated Circuit)

Zašto nam je potreban FPGA?

Mikroprocesor (CPU)

- izvršava instrukcije sekvencijalno
- fleksibilan
- ali spor za paralelne operacije

Zašto nam je potreban FPGA?

ASIC (Application Specific Integrated Circuit)

- veoma brz
- optimizovan za jednu funkciju
- ali skup i ne može se menjati

Zašto nam je potreban FPGA?

FPGA predstavlja kompromis između CPU i ASIC.

- može se rekonfigurisati
- omogućava paralelnu obradu
- nema troškova proizvodnje novog čipa

CPU ← fleksibilnost

FPGA ← kompromis

ASIC ← brzina

Šta je FPGA?

- FPGA (Field-Programmable Gate Array) je integrisano kolo koje omogućava implementaciju proizvoljnih digitalnih kola nakon proizvodnje čipa.
- Imamo proizveden čip, koji jos uvek nije definisan!

Šta je FPGA?

- FPGA je programabilni digitalni hardver
- struktura kola može se promeniti konfiguracijom
- korisnik definiše logiku i povezivanje blokova

FPGA može implementirati:

- digitalne filtere
- procesore
- komunikacione kontrolere
- AI akceleratori
- kompletne digitalne sisteme

Šta je FPGA?

- Za razliku od mikrop procesora koji izvršava program, FPGA omogućava da mi dizajniramo hardver.
- Mi zapravo definišemo kako će izgledati digitalno kolo unutar čipa.
- CPU izvršava program.
- FPGA postaje hardver koji smo opisali.

Unutrašnja struktura FPGA

FPGA se sastoji od tri osnovna dela:

- Logički blokovi (LUT – Look-Up Tables)
- Flip-flopovi
- Programabilna mreža povezivanja (Routing)

Unutrašnja struktura FPGA

Logički blokovi (LUT – Look-Up Tables)

- realizuju logičke funkcije
- implementiraju kombinaciona kola

Unutrašnja struktura FPGA

Flip-flopovi

- omogućavaju sekvencijalnu logiku
- koriste se za registre, brojače i FSM

Unutrašnja struktura FPGA

Programabilna mreža povezivanja (Routing)

- povezuje logičke blokove
- omogućava formiranje različitih digitalnih kola

Unutrašnja struktura FPGA

FPGA = mnogo malih logičkih blokova +
programabilne veze

Kako HDL kod postaje hardver u FPGA

- FPGA se konfiguriše pomoću HDL jezika (Hardware Description Language).

Najčešći HDL jezici su:

- Verilog
 - VHDL
-
- HDL kod ne opisuje program, već opisuje digitalno kolo.

Kako HDL kod postaje hardver u FPGA

Primer Verilog koda:

- `assign y = a & b;`
- Ovaj kod opisuje AND logičko kolo između ulaza a i b.

Nakon sinteze:

- alat generiše logičku mrežu
- FPGA se konfiguriše da realizuje to kolo

Kako HDL kod postaje hardver u FPGA

Kod mikroprocesora pišemo program koji procesor izvršava.

Kod FPGA mi ne pišemo program — mi opisujemo hardver koji želimo da napravimo

Proces implementacije FPGA dizajna

Razvoj FPGA sistema prolazi kroz nekoliko koraka:

Pisanje HDL koda

- Verilog ili VHDL opis digitalnog kola

Proces implementacije FPGA dizajna

Sinteza (Synthesis)

- HDL kod se prevodi u logička kola

Proces implementacije FPGA dizajna

Place and Route

- logički blokovi se raspoređuju u FPGA
- određuju se veze između blokova

Proces implementacije FPGA dizajna

Generisanje bitstream fajla

- konfiguracioni fajl za FPGA

Proces implementacije FPGA dizajna

Konfiguracija FPGA

- bitstream se učitava u FPGA čip

Zašto je FPGA paralelan i često brži od CPU

- CPU – sekvencijalna obrada
- Procesor izvršava instrukcije jednu po jednu.

Zašto je FPGA paralelan i često brži od CPU

- CPU – sekvencijalna obrada

Primer:

- operacija A
- operacija B
- operacija C

Zašto je FPGA paralelan i često brži od CPU

FPGA – paralelna obrada

U FPGA više operacija može raditi istovremeno.

Primer:

- operacija A
- operacija B
- operacija C

mogu se izvršavati u istom taktu.

Zašto je FPGA paralelan i često brži od CPU

Posledica

FPGA može biti mnogo brži u zadacima kao što su:

- digitalna obrada signala
- video obrada
- kriptografija
- AI akceleratori
- komunikacioni sistemi

Zašto je FPGA paralelan i često brži od CPU

CPU izvršava instrukcije.

FPGA izvršava hardver.

Primer implementacije digitalnog kola u FPGA

Razmotrimo 2:1 multiplexer.

Funkcija multiplexera:

$$y = s^{\bar{}} \cdot a + s \cdot b$$

assign y = s ? b : a;

Primer implementacije digitalnog kola u FPGA

Šta se dešava u FPGA?

Nakon sinteze:

- Vivado prepoznaje multiplexer
- implementira ga u LUT logičkim blokovima
- blokovi se povezuju preko routing mreže
- FPGA tako fizički realizuje multiplexer.

Primer implementacije digitalnog kola u FPGA

Iako mi napišemo samo jednu liniju Verilog koda, alat za sintezu zapravo generiše digitalno kolo koje implementira multiplexer.