



Radionica digitalnog dizajna II

Dan II

dr. sc. Jurica Kandrata

Centar za svemirsku i inovativnu tehnologiju

17. - 18. lipnja 2023.

Raspored - Dan I

17. lipnja 2023., subota	
9.00 - 12.00	• Kratko ponavljanje sadržaja prošle radionice • Instalacija lokalne podrške za Verilog jezik • Brainstorm ideja za zajednički integrirani krug MyChip
12.00 - 12.45	Pauza za ručak
12.45 - 15.00	• Uvod u dizajn procesorske jezgre - Arhitektura i instrukcijski set • Dizajn modula procesorske jezgre: ○ Registarska mapa ○ Memorijski sustav

Raspored - Dan II

18. lipnja 2023., nedjelja	
9.00 - 12.00	• Dizajn modula procesorske jezgre: ○ Upravljačka jedinica i Aritmetičko-logička jedinica ○ Brojač programa i Dekoder naredbi
12.00 - 12.45	Pauza za ručak
12.45 - 15.00	• Integracija modula procesorske jezgre • Sinteza i implementacija projekta na FPGA razvojnoj ploči • Verifikacija projekta

Dan II

Sadržaj

- Dizajn modula procesorske jezgre:
 - Brojač programa
 - Dekoder naredbi
 - Aritmetičko-logička jedinica
 - Upravljačka jedinica
- Integracija modula procesorske jezgre
- Sinteza i implementacija projekta na FPGA razvojnoj ploči
- Verifikacija projekta

Rezime prošlog dana

Brojač programa

Specifikacija - Brojač programa

- Sadrži adresu instrukcije koja se trenutno izvršava
- Na sljedeću instrukciju se prelazi kontrolnim signalom
- Podržava skok na danu instrukciju

Sučelje - Brojač programa

```
// Global
input wire CLK,
input wire RSTn,
// Module specific
input wire JUMP,
input wire NEXT_INSTRUCTION,
input wire [WIDTH-1:0] JUMP_ADDRESS,
output wire [WIDTH-1:0] INSTRUCTION_ADDRESS
```

Dekoder naredbi

Specifikacija - Dekoder naredbi

- Instrukciju učitava na kontrolni signal
- Učitano instrukciju raščlanjuje na elemente po formatu instrukcije

Sučelje - Dekoder naredbi

```
// Global
input wire CLK,
input wire RSTn,
// Module specific
input wire NEXT_INSTRUCTION,
input wire [WIDTH-1:0] INSTRUCTION, // Input instruction
output reg [7:0] OPCODE, // Output opcode
output reg [7:0] DEST_REGISTER, // Output destination register index
output reg [7:0] SOURCE_REGISTER1, // Output first source register index
output reg [7:0] SOURCE_REGISTER2 // Output second source register index
```

Aritmetičko-logička jedinica

Specifikacija - Aritmetičko-logička jedinica

- Kombinaijski sklop
- Podržava zbrajanje (opcode = 0b0000) i oduzimanje (opcode = 0b0001)
- Podržava statusnu zastavicu ZERO
 - Rezultat operacije je 0

Sučelje - Aritmetičko-logička jedinica

```
input wire [WIDTH-1:0] A,      // First operand
input wire [WIDTH-1:0] B,      // Second operand
input wire [3:0] OPCODE,       // Opcode: 00 for ADD, 01 for SUB
output reg [WIDTH-1:0] RESULT, // Result of the operation

output ZERO_FLAG
```

Upravljačka jedinica

Princip rada upravljačke jedinice (I)

- FETCH - DECODE - EXECUTE - WRITEBACK ciklus
- 1. FETCH: U fazi dohvaćanja (FETCH), procesor čita slijedeću instrukciju iz memorije koristeći adresu u brojaču programa.
- 2. DECODE: U fazi dekodiranja (DECODE), procesor tumači (dekodira) dohvaćenu instrukciju i priprema potrebne resurse (registre, podatke, itd.) za njeno izvršenje.

Princip rada upravljačke jedinice (II)

- FETCH - DECODE - EXECUTE - WRITEBACK ciklus
- EXECUTE: U fazi izvršenja (EXECUTE), procesor izvršava instrukciju, što može uključivati aritmetičke i logičke operacije, pristup memoriji, ili kontrolu toka programa.
- WRITEBACK: U fazi povratnog zapisivanja (WRITEBACK), procesor pohranjuje rezultate izvršene instrukcije natrag u odgovarajuće registre ili memoriju, kako je navedeno u instrukciji.

Specifikacija - Upravljačka jedinica

- Generira sve kontrolne signale
- Implementirati kao automat sa stanjima:
 - FETCH
 - DECODE
 - EXECUTE
 - WRITEBACK

Sučelje - Upravljačka jedinica

```
// Global
input wire CLK,
input wire RSTn, // active-low reset signal
// Module specific
input wire [7:0] OPERATION_CODE, // input instruction
output reg JUMP, // Load control signal for Program Counter
output reg WRITE_ENABLE, // Write Enable control signal for memory
output reg READ_ENABLE, // Write Enable control signal for memory
output reg NEXT_INSTRUCTION, // Write Enable control signal for memory
output reg INSTRUCTION_DATA_n, // Write Enable control signal for memory
output reg WRITE_REG_ENABLE,
output reg [3:0] ALU_OP_CODE,
input wire ZERO_FLAG,
output reg MEMORY_ALUn, // source of write reg data
output reg LOAD_STOREn
```

Implementacija - Upravljačka jedinica

- Verilog modul → <https://tinyurl.com/3t59pwmv>

Integracija modula

Središnja procesorska jedinica - CPU

- `central_processing_unit` modul instancira i povezuje sve module jezgre
- Multipleksori za:
 - adresnu sabirnicu i
 - registre zapisivanja u:
 - registarsku mapu i
 - memorijski pristup

Sustav na čipu

- `system_on_chip` modul instancira i povezuje procesorsku jezgru i memorijski sustav
- Jednostavni GPIO kontroler - upravlja radom LEDice:
 - Dekodira svoju adresu `0h0000_1000` i na `WRITE_ENABLE` zapisuje vrijednost

Implementacija - Sustav na čipu

- Cijeli projekt → <https://tinyurl.com/yrszv5pe>