



**Universiteti i Prishtinës**

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Arkitektura e Kompjuterëve, 2022/2023

Prof. Valon Raça & Synim Selimi

Anila Luta, 210757100022

Anjeza Sfishta, 210756100016

Arlind Nimani, 200718100020

Rinesa Imeri, 200714100145

## Detyra e dytë

Dizajnimi i një CPU 24-bitëshe (Single-Cycle)

### 1. Hyrje

**ALU 24 bit** (Arithmetical Logic Unit)- është një qark digjital i cili mundëson kryerjen e veprimeve aritmetike dhe logjike në numrat binarë 24 bitësh. Komponentët kryesore të kësaj ALU janë:

- **Register File**- numri i regjistrave duhet të jetë 16 dhe ata duhet të jenë të gjerë 24 bit. Register File ka tre hyrje katër bitëshe për përcaktimin e regjistrave RS, RT, RD. Register file ka një hyrje 24 bitëshe për të shkruar në regjistrin RD. Register File ka dy dalje 24 bitëshe për të lexuar të dhënat nga regjistrat e përcaktuar në RS dhe RT.

```
module RegisterFile(  
    input wire[3:0] RS,  
    input wire [3:0] RT,  
    input wire [3:0] RD,  
    input wire [23:0] WriteData,  
    output wire [23:0] ReadRS,  
    output wire [23:0] ReadRT,  
    input wire RegWrite,  
    input wire Clock);  
  
reg[23:0] Registers[15:0];
```

Figura 1. Register File

- **Memoria** – Instruction dhe Data Memory kanë secila nga 128 bajta. Instruction Memory është vetëm Read Only me hyrje dhe dalje nga 24 bita. Data Memory është Read-Write. Hyrje 24 bitëshe për fjalën që shkruhet, dhe dalje 24 bitëshe për fjalën që lexohet.

```
module InstructionMemory(  
    input [23:0] PC,  
    output [23:0] Instruction  
);  
reg [7:0] iMem[127:0];  
initial $readmemb("InstructionMemory.mem", iMem);  
assign Instruction={iMem[PC], iMem[PC+1], iMem[PC+2]};  
endmodule
```

Figura 2. Instruction Memory

```

module DataMemory(
    input [23:0] Adresa,
    input [23:0] WriteData,
    input MemWrite,
    input MemRead,
    input Clock,
    output [23:0] ReadData
);

```

```

    $writememh("DataMemory.mem", dMem);
end

assign ReadData = {dMem[Adresa], dMem[Adresa+1], dMem[Adresa+2]};

```

Figura 3. Data Memory

- **Instruksionet**- Instruksionet e perdorura janë instruksione te formatit R dhe I. Instruksionet e formatit R janë: AND, OR, ADD, SUB, SLL, MUL, XOR, SLT, kurse instruksionet e formatit I janë: ADI, SS, LS, BEQ.
- **ALU Control**- Hyrjet e ALU Control-it do të kenë 4 bit nga fusha FUNCT e instruksionit dhe 2 bit nga ALUOp. Daljet do të kenë një linjë për invertimin e hyrjes B në ALU - BNegate. BNegate bëhet 1 vetëm në rast të zbritjes dhe shërben për invertimin e B-së dhe për furnizimin e CarryIn me 1. Tre bitat tjerë shërbejnë për multiplekserin që zgjedh operacionin.

```

module ALUControl(
    input [1:0] ALUOp,
    input [3:0] Funct,
    output reg [3:0] ALUCtrl
);

```

Figura 4. ALU Control

- **Njësia e kontrollit**- Daljet e njesisë së kontrollit janë: RegDst, ALUSrc, MemToReg, RegWrite, MemRead, MemWrite, ALUOp, Branch.

## 2. Dizajni

- **ALU 24 bit** - Fillimisht e kemi krijuar një ALU 1-bitëshe e cila i përkrah veprimet ADD, OR, XOR, etj., pastaj janë lidhur 24 ALU-ja 1 bitëshe për kompletim të ALU-së 24 bitëshe. Brenda kësaj ALU-je ne kemi një multiplekser 2 me 1 ku ai 8 me 1 është përgjegjës për zgjedhjen e veprimit (për njërit nga veprimet lartë).
- **Mux\_2ne1** – përmban modulën Mux\_2ne1() Hyrjet: Hyrja0, Hyrja1, S dhe Daljen: Dalja.
- **Mux\_4ne1** – përmban modulën Mux\_4ne1() Hyrjet: Hyrja0, Hyrja1, Hyrja2, Hyrja3, S, dhe Daljen: Dalja.
- **Mux\_8ne1** – përmban modulën Mux\_8ne1() Hyrjet: Hyrja0, Hyrja1, Hyrja2, Hyrja3, Hyrja4, Hyrja5, Hyrja6, Hyrja7, S dhe Daljen: Dalja.
- **Mulreg** - është regjistër 48 bitësh special i ndare nga Register File. Shërben për të ruajtur rezultatet e shumëzimit me numrave 24 bitësh.

## 3. Ekzekutimi

Disa nga testimet që I kemi ekzekutuar janë:

- **Testimi I Register File**- Në kohën 0 Clock-u është 0, pas 5ns Clock-u është 1 vlera e RD është 6 dhe në WriteData shenohet vlera 3, gjithashtu e njëjta vlenë edhe për pjesën e dytë. Gjatë leximit të regjistrit 6 dhe 7 në dalje janë 3 dhe 5.

```
initial
begin
#0 Clock=1'b0;
#5 RD=4'd6; WriteData = 23'd3; RegWrite=1'b1;
#5 Clock=1'b1;
#5 Clock=1'b0; RegWrite = 1'b1;
#5 RD=4'd7; WriteData = 23'd5; RegWrite=1'b1;
#5 Clock=1'b1;
#5 Clock=1'b0; RegWrite=0;
#5 RS=4'd6; RT=4'd7;
#5 $stop;
```

Figura 5. Register File Test code

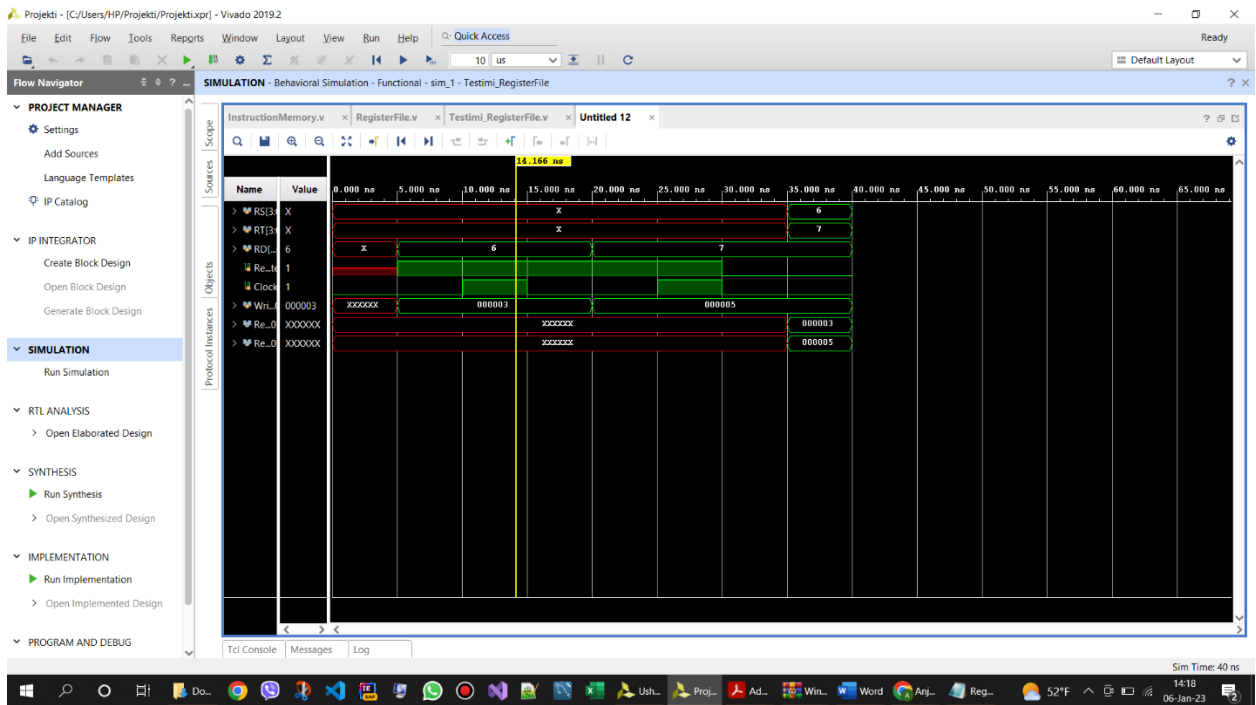


Figura 6. Register File Test

- **Testimi I Instruction Memory-Në** vijim është paraqitur ekzekutimi i testimit të Instruction Memory, ku kemi marrë instruksionet prej adresës 9 pasi që 8 të parat janë të rezervuara.

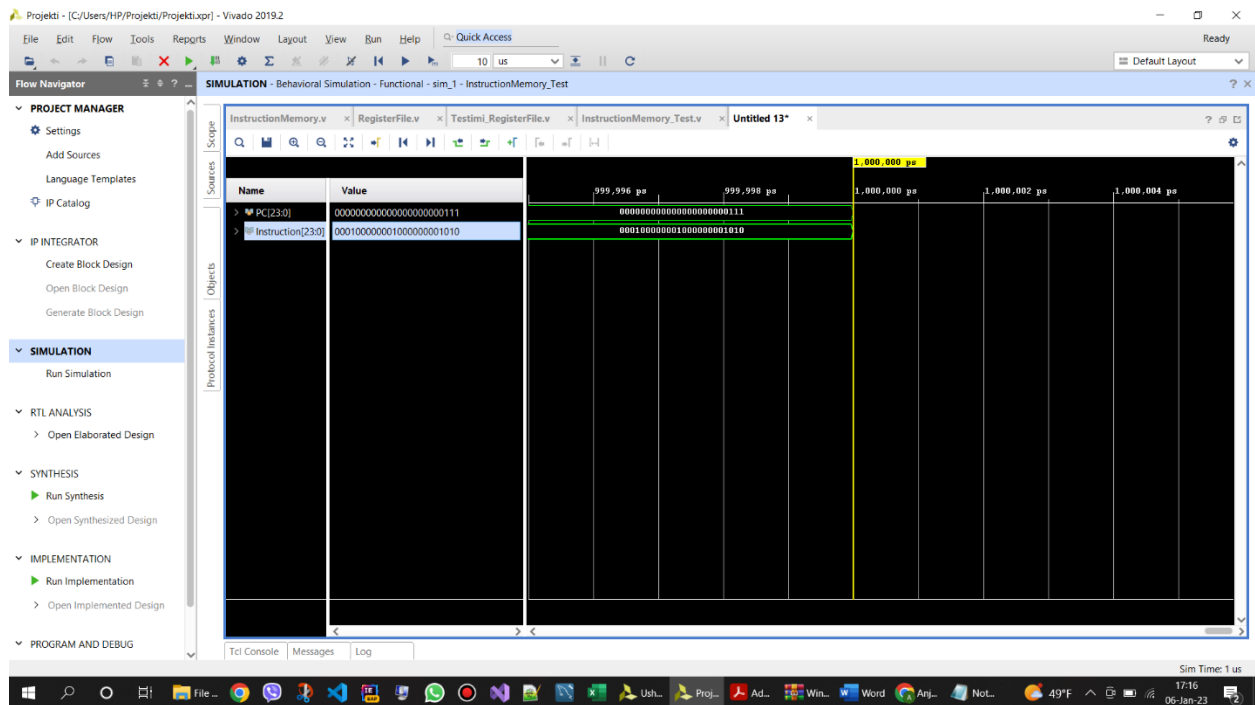


Figura 7. Instruction Memory Test

- **Testimi I Data Memory** – përfshin shkrimin e një numri në memorie dhe pas ekzekutimit shihet se numri është ruajtur në memorie.

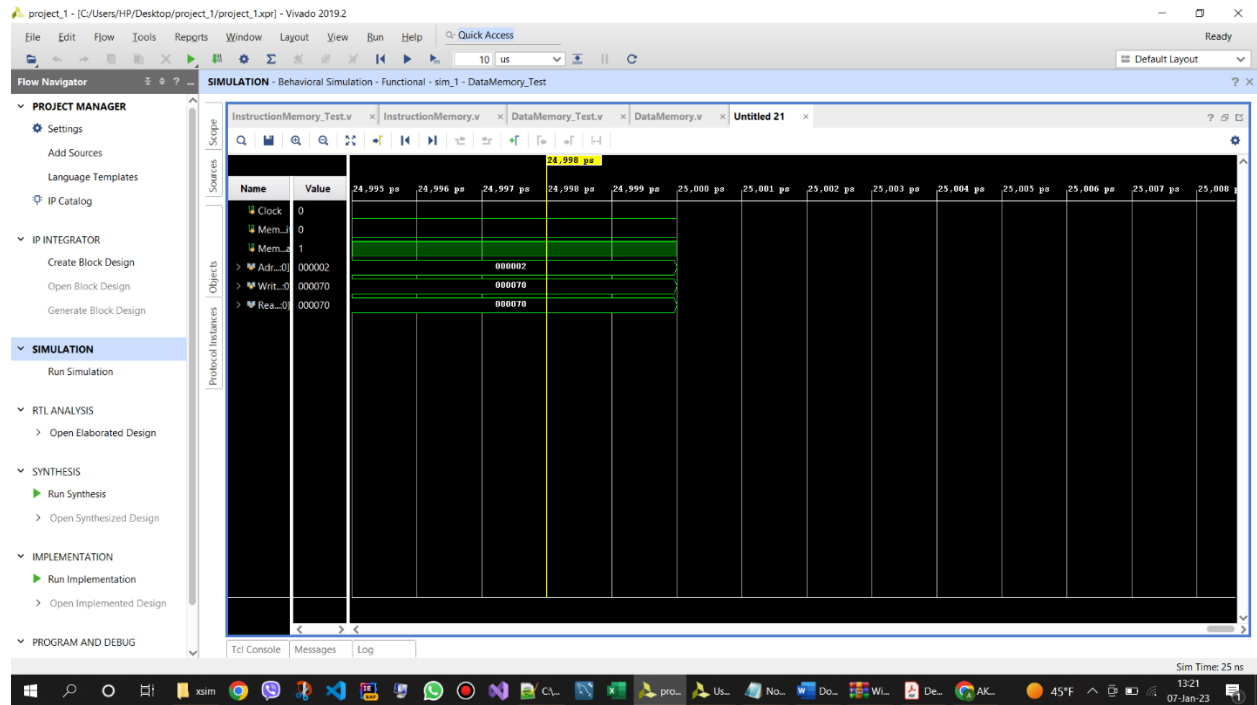


Figura 8. Data Memory Test

- **Testimi I CPU-së** – përfshin ndërrimin e Clock-ut pas 5ns.

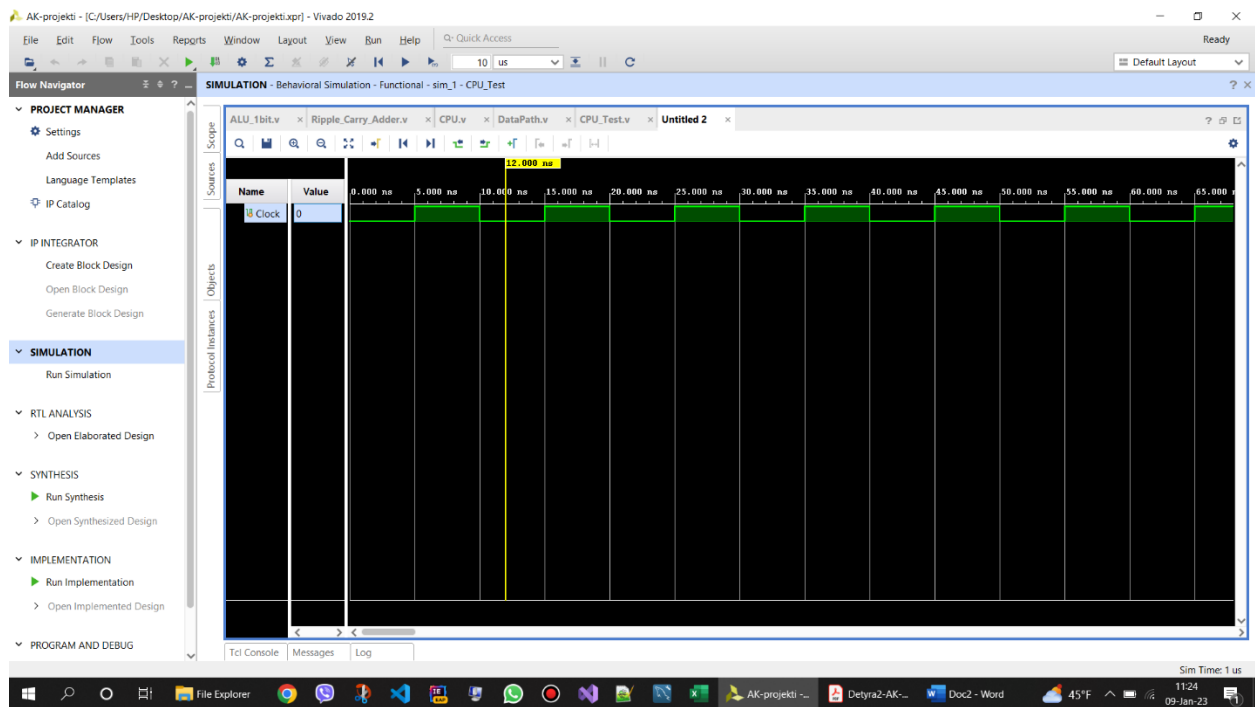


Figura 9. CPU test

## 4. Përfundimi

Pra një ALU 24-bitësh është një qark digjital i fuqishëm dhe i gjithanshëm që mund të trajtojë sasi të mëdha të dhënash dhe të kryejë një gamë të gjerë operacionesh matematikore dhe logjike. Përdoret zakonisht në mikroprocesorë dhe sisteme të tjera dixhitale që kërkojnë llogaritje me saktësi të lartë dhe një gamë të madhe vlerash.