Registri i ALU

Cilj:
- Dizajn registra (sekvencijalna logika)
- Dizajn ALU (kombinaciona logika)
- Testiranje kroz simulaciju

Registri

Registri:
- memorijski elementi
- čuvaju podatke
- rade na clock signal

Registar pamti stanje → zato je sekvencijalna logika

Struktura registarskog modula

always @(posedge clk)

radi na rastućoj ivici clock-a
koristi:
-rst
-we
-addr

Struktura registarskog modula

- reset (rst)
- write enable (we)
- adresiranje (addr)
- skladištenje podataka

Ovo je mini memorija sa 4 registra:

```
reg [7:0] R0;
reg [7:0] R1;
reg [7:0] R2;
reg [7:0] R3;
```

ALU (Arithmetic Logic Unit)

ALU (Arithmetic Logic Unit):
- aritmetičke operacije
- logičke operacije
- shift operacije

Operacije u ALU

sel = 000 → A + B
sel = 001 → A - B
sel = 010 → AND
sel = 011 → OR
sel = 100 → XOR
sel = 101 → NOT
sel = 110 → shift left
sel = 111 → shift right

Konkatenacija

{signal1, signal2, signal3}

spajanje više signala u jedan veći vektor

Konkatenacija

{A, B}

ako je:
A = 4 bita
B = 4 bita
Dobijamo:
[A3 A2 A1 A0 B3 B2 B1 B0] → 8 bita

Konkatenacija u ALU

{carry, Y} = A + B;
Rezultat sabiranja dva 8-bitna broja je 9-bitni broj.
Koristi se konkatenacija da se razdvoji rezultat:”

najviši bit → carry
ostalo → rezultat

Konkatenacija u ALU

{carry, Y} = A + B;
Rezultat sabiranja dva 8-bitna broja je 9-bitni broj.
Koristi se konkatenacija da se razdvoji rezultat:”

najviši bit → carry
ostalo → rezultat

A + B = [carry][Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0]

Konkatenacija u ALU

{carry, Y} = A + B;
Verilog automatski radi:
carry = najviši bit
Y = donjih 8 bita

Bez konkatenacije

Y = A + B;
Gubimo carry!
255 + 1 = 256 = 1 00000000

dobijemo samo:

00000000 (carry izgubljen)

Konkatenacija = operacija spajanja ili razdvajanja bitova u jedinstveni vektor

{A, B} = 16'hABCD;

A = AB
B = CD

Zero signal

assign zero = (Y == 0);
Detekcija da li je rezultat nula!

Koristi se u:
-procesorima
-grananju (branch)

Kombinaciona vs sekvencijalna logika

Registri → sekvencijalna (clk)
ALU → kombinaciona (always @(*))