Osnove Verilog HDL jezika

Leksicka svojstva

Osnove Verilog HDL jezika

- Komentari su kao u C/C++-u, /* // */
- Beline se preskacu i ignorisu (kao i u C-u).
- Jezik razlikuje mala i velika slova (case-sensitive)
- Celobrojne konstante se mogu zadavati u dekadnom, binarnom, oktalnom i heksadekadnom formatu

Osnove Verilog HDL jezika

- Format: <velicina>'<osnova><vrednost>

Osnove Verilog HDL jezika

- Format: <velicina>'<osnova><vrednost>
- Velicina je broj bitova, osnova je b, d, h ili o (mogu i velika slova B,D,H,O), dok je vrednost sam broj zapisan u datoj osnovi.
- Ako se velicina izostavi, podrazumeva se 32.
- Ako se I osnova izostavi, podrazumeva se dekadni zapis.
- Heksadekadne cifre a, b, c, d, e, f se mogu pisati i kao velika i kao mala slova.

Osnove Verilog HDL jezika

- Vrednost cifre u svim osnovama (osim dekadne) takodje moze biti z ili x.
- Prva vrednost znaci "visoka impedansa" (eng. floating), sto u stvari predstavlja "otkacenu zicu", dok x predstavlja nedefinisanu vrednost.

Osnove Verilog HDL jezika

- Sve celobrojne konstante su podrazumevano neoznacene.
- Ako se ispred celobrojne konstante stavi znak -, tada se ta konstanta smatra oznacenom vrednoscu koja se interno cuva u ***potpunom komplementu***.

Osnove Verilog HDL jezika

- 55, 13, 35, 484 // dekadne neoznacene konstante //(sirine 32 bita)

- 'h5f23, 'o4317, 'b01101101, 'd593 // konstante sa // eksplicitno // zadatom osnovom (sirine 32 bita)

- 16'hffff, 12'd59, 3'b101 // konstante sa //ekplicitno zadatom // sirinom (u bitovima)

- -53, -12, -16'h54ff // oznacene (negativne) // konstante

OSNOVNI TIPOVI PODATAKA U VERILOGU

- Svaki bit signala moze uzimati sledece vrednosti:

- 0: logicka nula
- 1: logicka jedinica
- x: nedefinisana vrednost
- z: vrednost visoke impedanse

OSNOVNI TIPOVI PODATAKA U VERILOGU

- Dva osnovna tipa podataka (signala) u verilogu su:

- -- zice (eng. "net types" ili "wires")
- -- registri (eng. registers)

OSNOVNI TIPOVI PODATAKA U VERILOGU

- Tip wire ima osobinu da moze da ima vise drajvera, u kom slucaju se vrednost signala dobija na sledeci nacin:
  ako je bar jedan od drajvera nedefinisan (x) takva je vrednost i wire signala.
- Ako drajveri imaju definisanu vrednost (0 ili 1), ali su im vrednosti razlicite, tada je rezultat opet nedefinisan.
- Ako su svi drajveri 0 ili svi 1, tada je takav i rezultat.
- Ako je neki od drajvera jednak z, tada on ne utice na vrednost signala.
- Ako su svi drajveri z, tada je i rezultat z.

OSNOVNI TIPOVI PODATAKA U VERILOGU

- Registarski tipovi konceptualno predstavljaju signal koji ima memorijsko svojstvo, tj. u njega se moze "upisati" vrednost koju on onda cuva dok se ta vrednost ne promeni.
- Obicno ove signale zamisljamo kao izlaze iz nekih registara, mada treba napomenuti da u praksi prilikom sinteze kola ne mora zaista postojati registar koji cuva vrednost ovog signala.

OSNOVNI TIPOVI PODATAKA U VERILOGU

- Najjednostavniji registarski tip je reg.

Na primer:

- reg x; // deklarise jednobitni podatak tipa reg
- reg [7:0] y; // deklarise 8-bitni podatak tipa reg

OSNOVNI TIPOVI PODATAKA U VERILOGU

- integer -- predstavlja 32-bitni reg tip, s tim sto se vrednost ovog tipa tumaci kao oznacen broj u potpunom komplementu.
  Obicno se koristi za brojace i slicne promenljive, a redje za delove koda koje treba sintetizovati u stvarno kolo.

OSNOVNI TIPOVI PODATAKA U VERILOGU

- Verilog omogucava i kreiranje nizova podataka. Vektor je jedan podatak koji sesastoji iz vise bitova.
- Niz je sekvenca vise podataka (potencijalno visebitnih).

OSNOVNI TIPOVI PODATAKA U VERILOGU

- wire x[0:99]; // definise niz od 100 jednobitnih signala. Indeksi u nizu su od 0 do 99.
- reg [7:0] y[0:255] // definise niz od 256 8-bitnih podataka.

OSNOVNI TIPOVI PODATAKA U VERILOGU

- Zapis y[0] predstavlja element u nizu sa indeksom 0.
- Zapis y[0][2] predstavlja bit na indeksu 2 u elementu niza sa indeksom 0.
- Nizovi su narocito korisni za predstavljanje memorija (npr. gore deklarisani podatak y je u stvari memorija koja se sastoji od 256 bajtova).

IZRAZI I OPERATORI U VERILOGU

- U slucaju binarnih operatora,ako operandi nisu iste sirine, tada se uzi tip prosiruje na broj bitova sireg tipa.
- Vecina operatora se jednostavno moze sintetisati,tj. napraviti konkretno hardversko kolo koje definise njihovo ponasanje.
- Izuzetak su slozeni aritmeticki operatori, poput deljenja i ostatka po modulu.

IZRAZI I OPERATORI U VERILOGU

- Operator indeksiranja ([]):
- koristi se za pristup bitovima u vektorima, tj. izdvajanje podsignala u okviru vektorskog signala.

IZRAZI I OPERATORI U VERILOGU

- Operator za pristup ugnjezdenim imenima (.)
- slicno kao u C-u, kada se pristupa clanovima struktura.

IZRAZI I OPERATORI U VERILOGU

- Aritmeticki operatori: +, -, *, /, %, kao i unarni + i -, su slicni kao i u C-u.

IZRAZI I OPERATORI U VERILOGU

- rezultat sabiranja dva n-bitna podatka zapravo n+1-bitni podatak (najvisi bit je bit prekoracenja)

IZRAZI I OPERATORI U VERILOGU

- Proizvod dva n-bitna podatka je 2n-bitni podatak.
- Zato ako rezultat ovakvih operacija dodelimo podatku koji takodje ima samo n bitova, visi bitovi ce biti izgubljeni.

IZRAZI I OPERATORI U VERILOGU

- Treba naglasiti da se operatori / i % tesko mogu sintetizovati (jer zahtevaju veoma slozenu logiku), tako da se vise koriste u testiranju i simulaciji, dok ih treba izbegavati u kolima koja zelimo da u praksi sintetizujemo, tj. preslikamo na stvarni hardver.

IZRAZI I OPERATORI U VERILOGU

- Logicki operatori: !, &&, ||.
- Ovi operatori rade isto kao i u C-u.
- Visebitni signal se smatra logicki tacnim ako je bar jedan njegov bit jednak 1.
- Rezultat ovih operatora je jednobitni signal sa odgovarajucom vrednoscu.

IZRAZI I OPERATORI U VERILOGU

- Relacioni operatori: >, <, >=, <=, ==, !=, ===, !===.
- Ovi operatori funkcionisu kao u C-u, a daju za rezultat jednobitnu odgovarajucu vrednost.

IZRAZI I OPERATORI U VERILOGU

- Poslednja dva operatora su zanimljiva: uobicajeni relacioni operatori daju rezultat x kad god je bar neki od bitova u bilo kom od operanda jednak x ili z (tj. rade samo za potpuno definisane vrednosti).
- Operatori === i !=== mogu da uporedjuju i bitove sa ovim specijalnim vrednostima, npr.

3'b01z == 3'b01z daje

x, dok 3'b01z === 3'b01z daje 1.

IZRAZI I OPERATORI U VERILOGU

- Operatori pomeranja: <<, >>, <<<, >>>.
- Operatori su slicni kao u C-u, s tim sto su prva dva logicko, a druga dva aritmeticko pomeranje (naravno, aritmeticko pomeranje u levo je isto kao i logicko, tj. << i <<< rade identicno).

IZRAZI I OPERATORI U VERILOGU

- Operatori redukcije (&, ~&, |, ~|, ^, ~^): ovo su unarni operatori koji obavljaju odgovarajucu logicku operaciju nad svim bitovima jednog signala i kao rezultat daju jednobitni signal sa odgovarajucom vrednoscu.
- Na primer &x daje jedan bit koji nastaje konjunkcijom svih bitova u signalu x.
- Operatori sa ~ ispred predstavljaju odgovarajucu negaciju (tj. NAND, NOR, NXOR).

IZRAZI I OPERATORI U VERILOGU

- Operatori grupisanja ({}, {{}}):
- u pitanju su dva operatora koji omogucavaju kreiranje novih signala grupisanjem postojecih (na neki nacin, ovi operatori su inverz indeksnog operatora koji izdvaja podsignale iz visebitnih signala).

IZRAZI I OPERATORI U VERILOGU

- Na primer:

```
wire [7:0] x;
wire [3:0] y;
```

IZRAZI I OPERATORI U VERILOGU

```
{ x[2:0], y } // dobijamo 7-bitni signal koji se
                 //  sastoji iz
                // bitova x[2], x[1], x[0], y[3],
                y[2], y[1], y[0]
```

IZRAZI I OPERATORI U VERILOGU

{ x[3], x[2], x[1] } // isto sto i x[3:1]

{ 1'b0, x } // dopisuje jednu vodecu nulu na signal x

IZRAZI I OPERATORI U VERILOGU

- Operator replikacije, koji omogucava da se isti signal ponovi vise puta, npr:

```
{4{x[0]}}      // isto sto i {x[0], x[0], x[0], x[0]}
{2{y[1:0]}}  // isto sto i {y[1:0], y[1:0]}
```

Broj ponavljanja mora biti konstanta.