

Osnove Verilog HDL jezika

NAREDBE GRANANJA

NAREDBE GRANANJA

Naredba grananja je if-else naredba, cija je sintaksa:

```
if(<uslov>)  
  naredba  
else  
  naredba
```

pri cemu se deo sa else moze izostaviti.

- Ako zelimo vise naredbi, stavljamo ih u begin-end blok

Naredba case je analogna switch naredbi u C-u i ima sledecu sintaksu:

```
case(<izraz>)
```

```
<lista_vrednosti>: <naredba>
```

```
<lista_vrednosti>: <naredba>
```

```
...
```

```
default: <naredba>
```

```
endcase
```

- Izraz može biti proizvoljnog tipa i broja bitova.
- Lista vrednosti senavodi razdvojena zarezima nakon čega sledi naredba (koja može biti i begin-end blok) koja se izvršava ako je izraz jednak nekoj od datih vrednosti.
- default opcija se može izostaviti, a izvršava se ako izraz nije jednak ni jednoj od ponudjenih vrednosti.

- Za razliku od C-a, ne navodi se nista nalik break naredbi nakon svake od opcija.

PETLJE U VERILOGU

- Petlje while i for imaju analognu sintaksu i semantiku kao i u C-u:

```
while(<uslov>) <naredba>
```

```
for(<init>; <uslov> ; <inc>) <naredba>
```

PETLJE U VERILOGU

- Petlja forever izvršava neku naredbu beskonacno mnogo puta.
- Neophodno je navesti odgovarajuću vremensku kontrolu, inace bi se uslo u beskonacnu petlju.

PETLJE U VERILOGU

Na primer:

```
initial
```

```
forever @(posedge clk) q <= ~q;
```

je identicno kao i:

```
always @(posedge clk)
```

```
q <= ~q;
```

PETLJE U VERILOGU

- Petlja `repeat` ponavlja neku naredbu konacno mnogo puta:

`repeat(<konstanta>) <naredba>`

- gde je `<konstanta>` ceo broj ponavljanja naredbe, npr:

`repeat(5) @(posedge clk) ;`

PETLJE U VERILOGU

- U ovom slučaju imamo praznu naredbu (;) koja se izvršava kada nastupi pozitivna ivica časovnika.
- Ovo se ponavlja 5 puta.
- Efekat ove naredbe je da se sačeka 5 ciklusa časovnika pre nego što se nastavi izvršavanje bloka.

UGRADJENE NAREDBE

- Ugradjene naredbe (ili ugradjeni poslovi) u verilog-u pocinji simbolom
- \$, po cemu se razlikuju od korisnicki definisanih identifikatora.

UGRADJENE NAREDBE

Neke najvažnije su:

\$time: vraca tekuce vreme (izrazeno u broju vremenskih jedinica od pocetka simulacije)

UGRADJENE NAREDBE

`$display()`: nalik `printf`-u, ispisuje poruku na ekran.

Na primer:

```
$display($time, ": x: %b, y: %b", x, y);
```

prikazuje poruku oblika:

```
10: x: 1, y: 0
```

UGRADJENE NAREDBE

`$monitor()`: slično kao i `$display`, jedino što se time registruje da želimo da se poruka prikazuje svaki put kada se promeni neka od vrednosti koje se prikazuju.

Obično se to uradi u nekom initial bloku na početku.

U svakom trenutku može biti aktivan samo jedan monitor.

On se može uključivati i isključivati naredbama `$monitoron` i `$monitoroff`.

UGRADJENE NAREDBE

\$finish: ovom naredbom se zaustavlja simulacija.

Ako imamo neki beskonacni proces, onda se u nekom initial procesu moze staviti

#100 \$finish;

tako da se nakon 100 vremenskih jedinica ipak zaustavi simulacija.

UGRADJENE NAREDBE

`$dumpfile()`: ovim se specificira naziv fajla u koji ce biti smestene

informacije koje su pogodne za graficki prikaz simulacije (u

verilogovom VCD (value-change-dump) formatu). Na primer:

```
$dumpfile("data.vcd");
```

UGRADJENE NAREDBE

\$dumpvars(): ovom naredbom se definise koje varijable zelimo da nam

budu prikazane u VCD fajlu. Na primer:

```
$dumpvars(1, my_module);
```

kaze da se prikazuju sve varijable deklarisanе u okviru modula

my_module.

UGRADJENE NAREDBE

Da je prvi argument bio 0, tada bi se prikazivale

i sve varijable unutar podmodula koji su instancirani u modulu

my_module. Može se navesti i lista više modula, kao i konkretnih

varijabli, npr:

```
$dumpvars(1, my_module1, my_module2,  
my_var1, my_var2):
```

DIREKTIVE PRETPROCESORA

Direktive pretprocesora pocinju simbolom `.

Pored ranije opisane

`timescale direktive, tu su jos i `define,
`include, `ifdef, `else

`endif i sl. koje su potpuno analogne
odgovarajucim direktivama

u C-u.