

Limbaje de descriere hardware  
-curs 8-

4.12.2024

## Descrierea sequentială

Desc. sequentială se realizează cu ajutorul instrucțiunilor specifice sequentiale, iar acestea se tratează una după alta în domeniul timpului.

Specificațiile sequentiale sunt diferențiate de cele concurente chiar dacă au funcții similare.

Specif. sequentiale sunt planate în interiorul proceselor în timp ce cele concurente sunt planate în portul de descriere concurență a arhitecturii.

Procesele sunt destinate pt ca în interiorul acestora să se realizeze descrieri sequentiale, dar ele sunt concurente între ele și sunt declarate în zonă de descriere concurență a arhitecturii.

.....  
ARCHITECTURE ex-~~am~~ OF ....

BEGIN

-zona descriere concurență

PROCESS(...)

zona declarativă

BEGIN

zona descriere sequentială

END PROCESS;

Procesul se declară după urm.

sintaxă:

eticheta PROCESS (listă de semnale)

- zonă declarativă

BEGIN

- zonă descriere secvențială

END PROCESS;

Liste de semnale constă într-o listă cu semnalele care vor activa procesul.

Activarea procesului se realizează la apariția a cel puțin a unui eveniment pe un semnal ce se află în lista de semnale a procesului.

Prim eveniment înțelegem trecerea semnalului dintr-o stare logică în altă stare logică a semnalului.

Zona declarativă conține doar declarații de variabile, nu conține semnale ~~semnale~~ și pt că sunt considerate concurente, iar aceste sunt declarate în zonă de declarativă a arhitecturii.

Variabilele reprezintă un elem. de salvare a informației care implică registrui (locații de memorie) și nu lire. Aceste sunt utilizate la simplificarea calcului sau operațiilor logice într-o zonă secvențială.

Declinarea variabilelor se face prim:

VARIABLE nume-var1, ..., nume-varn : tip := val ini

PROCESS(a, b)

VARIABLE c: std\_logic;

BEGIN

c := a;

c := b;

$d \leq c$

$d \leq a$

$d \leq b$

END PROCESS;

Operatorul de atribuire a unei variabile este " := ".

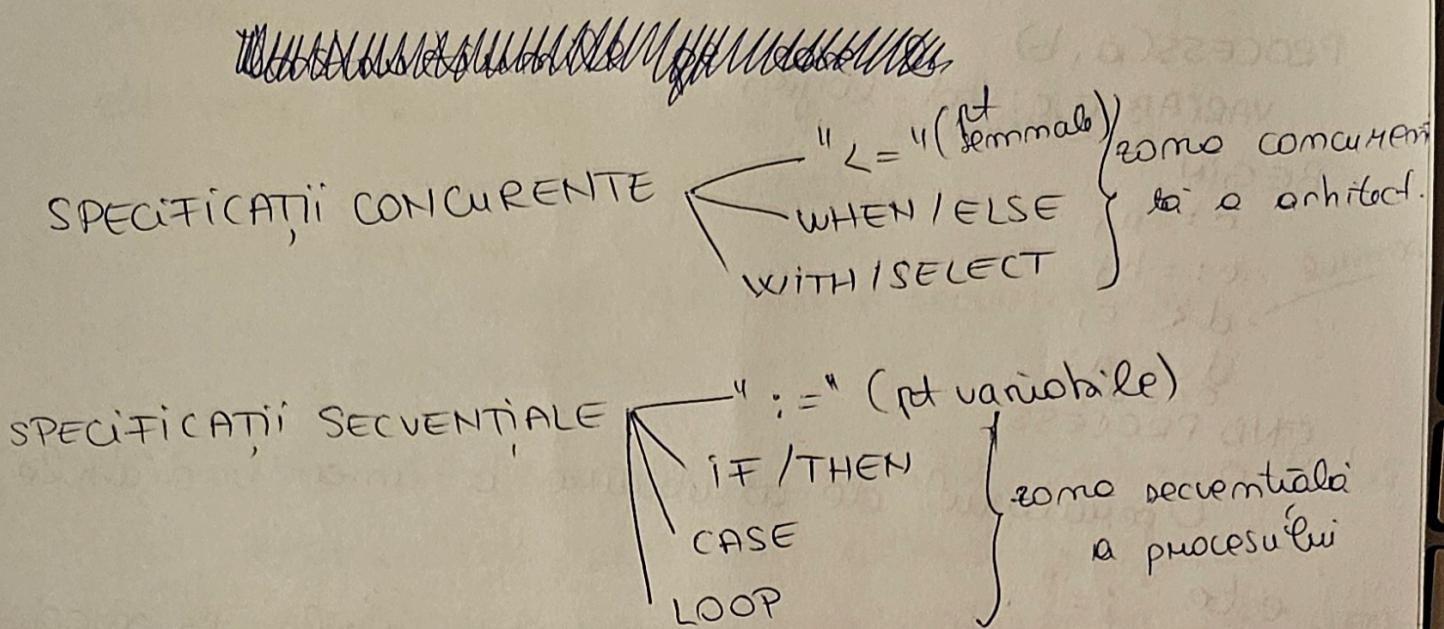
În cadrul unui proces, variabila și modifică valoarea de fiecare dată când se face o atribuire asupra ei, în timp ce semnalul și modifică valoarea numai atunci când se va ieși din proces.

Semnalul și va păstra valoarea initială de la începerea execuției procesului până la finalizarea acestuia.

Dacă s-au făcut mai multe atribuiri asupra unui semnal în cadrul procesului, la ieșire din acesta va fi luate în considerare ultima atribuire realizată după semnalului.

Variabila e locală doar procesului în care a fost declarata.

Schimbul de informații între procente se face prin semnale.



### Specificația IF / THEN

Prin acestă specificație se permite realizarea condițională a unor grupuri de remarcări care se întâmplă în cadrul procesului.

#### SINTAXĂ:

if condiție-booleană THEN

- zonă specificații secvențiale;

ELSE

- zonă specificații secvențiale;

ENDIF,

Această specificație se mai poate folosi și în forma următoare:

IF condiție - boolean<sub>1</sub> THEN  
- zomă specif. secu.

ELSE IF condiție - boolean<sub>2</sub> THEN  
- - - - -

ELSE

- zomă specif. secu.

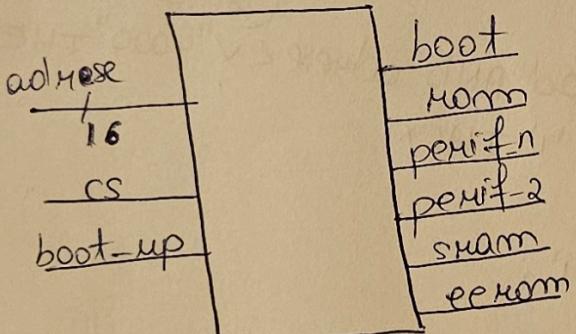
END IF;

În acest caz prima condiție are prioritățe maximă.

#### APLICATIE:

Să ne descrie în lb. VHDL un decodor de adrese pe 16 biti utilizând specif. if / THEN.

Memoria ROM, RAM, static RAM ...



FFFF4
eeprom
SHAM
perif-2
perif-n
ROM/BOOT

Address values:

- eeprom: C0004
- SHAM: 8004
- perif-2: 40104
- perif-n: 40084
- ROM/BOOT: 40004
- Address 00004 is shared by ROM and BOOT.

Memoria "boot" e selectată atunci când semnalul "boot-up" e activ.

Memoria "boot" împarte același spațiu de adreseare cu memoria ROM.

Semnalul "cs" activeaza decodonul de adrese atunci cand acesta e in 1 logic.

In funct. de adresa selectata se activeaza 1 din porturile de ieșire ale decodonului.

LIBRARY IEEE;

USE IEEE std-logic-1164.all;

ENTITY dec-16b IS

PORT (adresa :IN std-logic-vector(15 DOWNTO 0);  
cs,boot-up :IN std-logic;  
boot,nam,penif-1,penif-2,snam,eenam :OUT std-logic);

END dec-16b;

ARCHITECTURE desc OF dec-16b IS

BEGIN

PROCESS (adresa, boot-up, cs)

BEGIN

IF cs = '1' THEN

IF adresa <= x"0000" AND adresa > x"4000" THEN .

IF boot-up = '1' THEN

boot <= 1;

ELSE

nam <= "1";

END IF;

IF adresa >= x"4000" AND adresa < x"4008" THEN .

penif-1 <= 1;

ELSE adresa >= x"4008" AND adresa < x"4010" THEN .

penif-2 <= 1;

ELSE IF adresa >= x"8000" AND adresa < x"COOO" THEN .

snam <= 1;

ELSE adresa >= x"COOO" AND adresa < x"FFFF" THEN .

eenam <= 1;

{ boot = '0';  
nam = '0';  
penif-1 = '0';  
penif-2 = '0';  
snam = '0';  
eenam = '0';

END IF;  
END PROCESS;  
END descn;

Tema: !!

Să se implementeze acest ex. folosind specificație concurentă WHEN / ELSE.