CURS 3
LIMBAJUL VHDL – Partea 2
Tipuri de date. Domeniul secvential

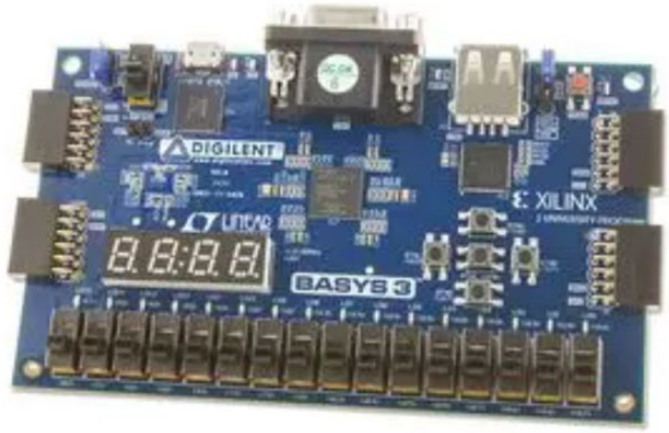
S.l. Ing. Vlad-Cristian Miclea

Universitatea Tehnica din Cluj-Napoca

Departamentul Calculatoare

Saptamanile trecute

- FPGA
- Structura unui FPGA
- Limbajul VHDL
  - Generalitati
  - Domenii de aplicare – specificare, simulare, sinte
- Structura unui program VHDL
  - Structura ierarhica
  - Entitate
  - Arhitectura
  - Structurala vs comportamentala vs flux-de-date
- Obiecte in VHDL
  - Semnale
    - Externe vs interne
    - Driver – pereche timp/valoare
    - Asignarea semnalelor
  - Parametrii generic
  - Constante




```
a <= not b after 10 ns;
```

a	0	1	
	now	10	

CUPRINS

1) Introducere
2) Tipuri de date
   - Scalare
   - Compuse
3) Operatori. Atribute
4) Domeniul secvential
   - Procese
   - Instructiunea Wait. Lista de sensibilitate
   - Variabile
5) Concluzii

TIPURI DE DATE

Generalităţi

- VHDL puternic **tipizat**
  - fiecare semnal, variabilă, constantă are un tip (definit înainte de utilizare)
  - parametrii procedurilor şi funcţiilor şi rezultatul returnat de funcţii - au obligatoriu un tip
- tipizarea obiectelor - protejează instrucţiunile de atribuire
- nivel de abstractizare relativ la implementarea structurilor de date → prin asociere de reprezentare simbolică independentă de partea hardware

TIPURI DE DATE

Tipuri de date

- 4 tipuri de date:
  - scalare - valoarea constituită dintr-un element
  - compuse - valoarea constituită din mai multe elemente
  - acces (pointeri)
  - fişier

TIPURI DE DATE

Clase de obiecte - familii de tipuri

	Constante	Variabile	Semnale	Fişiere
Scalare	DA	DA	DA	NU
Compuse	DA	DA	DA	NU
Acces	NU	DA	NU	NU
Fişier	NU	NU	NU	DA

TIPURI DE DATE

Tipuri scalare

- 4 tipuri: enumerate, întregi, flotante, fizice
- ordonate (pot fi comparate)
- **interval de validitate** - restrânge valorile posibile
  - **range** expresie1 **to** expresie2;
  - **range** expresie3 **downto** expresie4;

TIPURI DE DATE

Tipuri scalare enumerate

- **type** Nume **is** (valoare_simbolică1, valoare simbolică2, ...);
- simbolurile: identificatori sau caractere
- predefinite în pachetul Standard:
  - **type** Boolean **is** (False, True);
  - **type** Bit **is** (‘0’, ‘1’);
  - **type** Severity_Level **is** (Note, Warning, Error, Failure);
  - Character - toate caracterele admise în VHDL
- poziţia - induce relaţia de ordine între elemente

TIPURI DE DATE

Tipuri scalare enumerate

- **2 tipuri de baza:**
  - **BIT**
    - Definit in pachetul standard
    - Contine doar valorile 0 si 1
  - **STD_LOGIC**
    - Definit in pachetul std_logic_1164
    - Contine si alte valori pe langa 0 si 1
    - X – unknown
    - Z – high impedance
    - U - uninitialized

TIPURI DE DATE

Tipuri scalare întregi si flotante

- sunt definiţi operatorii aritmetici
- în declaraţie se indică domeniul (**range**)
  - exemplu: **type** Nume **is range**
- tipul **Integer** predefinit în pachetul Standard
- tipurile **Positive** şi **Natural** - subtipuri ale Integer
- Tip flotant: **Real**

TIPURI DE DATE

Tipuri scalare fizice

- **Time** – principalul tip fizic utilizat
  - unitatea de bază - femptosecunda (10^{-15} sec)
  - definit de un interval cu capetele exprimate pe 64 biţi
  - unităţi: fs, ps, ns, us, ms, sec, min, hr
  - Foarte util pentru simulare

TIPURI DE DATE

Tipuri compuse

- 2 tipuri compuse:
  - tablouri - colecţie de obiecte de acelaşi tip
  - articole - colecţie de obiecte de tipuri diferite

TIPURI DE DATE

Tipuri compuse

- tablouri
  - structuri omogene
  - elementele accesibile pe baza unor indecşi
  - definirea:
    - specificarea tipului
    - indicarea numărului indecşilor (tip discret)
    - specificarea tipului elementelor (fără tipul fişier)
  - vector - tablou cu un singur index

TIPURI DE DATE

Tipuri compuse

- tablouri
  - constrânse
    - intervalul de variaţie al indecşilor cunoscut cu anticipaţie
    - sensul de variaţie a indecşilor - specificat cu **to** şi **downto**
    - exemple:
      - **type** Funcţii_ALU **is** (dezactivată, adunare, scădere, înmulţire, împărţire);
      - **type** Adresă **is array** (0 **to** 15) **of** Bit;
      - **type** Word **is array** (31 **downto** 0) **of** Bit;
      - **type** Memory **is array** (Adresă) **of** Word;

TIPURI DE DATE

Tipuri compuse

- tablouri
  - neconstrânse
    - intervalul de variaţie al indecşilor cunoscut numai în timpul simulării
    - **<>** (box) amână definirea intervalului de indexare şi a direcţiei de variaţie
    - 2 tablouri neconstrânse în pachetul Standard:
      - **type** Bit_vector **is array** (Natural **range** <>) **of** Bit;
      - **type** String **is array** (Positive **range** <>) **of** Character;

TIPURI DE DATE

Tipuri compuse

- articole
  - elemente (câmpuri) diferite
  - enumerare câmpuri între **record** … **end record**
  - selectarea prin notaţia cu punct
- exemplu:

```
type instruction is
                                record
        op_code : processor_op;
        address_mode : mode;
        operand1, operand2 : integer range 0 to 15;
end record;
```

TIPURI DE DATE

Tipuri acces si fisier

- Mai putin utilizate
- Utilizare in simulari

TIPURI DE DATE

Conversii de tip

- conversia - **foarte restrictivă**
- posibilă în 3 cazuri:
  - tipuri cu reprezentare întreagă sau flotantă
  - pentru tablouri
    - aceleaşi dimensiuni
    - aceleaşi tipuri de elemente
    - indecşii trebuie să fie convertibili
  - conversia unui tip în el însuşi

OPERATORI

Clase şi priorităţi

- 7 clase, cu prioritate crescătoare de la 1 la 7:
  - 1. operatori logici: and, or, nand, nor, xor, xnor
  - 2. operatori relaţionali: =, /=, <, <=, >, >=
  - 3. operatori de deplasare: sll, srl, sla, sra, rol, ror
  - 4. operatori de adunare: +, -, &
  - 5. operatori de semn: +, -
  - 6. operatori de înmulţire: *, /, mod, rem
  - 7. operatori diverşi: **, abs, not
- ATENTIE: Sunt definiti doar pentru anumite tipuri!!
  - Ex: sll merge doar pt integer, nu pt std_logic_vector

OPERATORI

Caracteristici

- operatorii logici
  - **predefiniţi** pentru realizarea operaţiilor logice: ŞI, SAU, ŞI-NU, SAU-NU, SAU-EXCLUSIV, COINCIDENŢĂ
  - operanzi de tip Boolean (False, True), Bit (‘0’, ‘1’) si Std_logic
  - funcţionali pe vectori de elemente de tip Boolean, Bit, Std_logic, dacă au aceeaşi lungime

OPERATORI

Caracteristici

- operatorii relaţionali
  - rezultat de tip Boolean (False, True)
  - = şi /=
  - la tipurile enumerate, primul element este considerat cel mai mic
    - exemplu: la tipul Boolean, False e mai mic decât True

OPERATORI

Caracteristici

- operatorii de deplasare - binari
  - operează pe vectori cu elemente de tip Bit sau Boolean
- operatorii de adunare - binari
  - & - definit pe vectori (tipul String = vector de caractere)
- operatorii de semn - unari
  - au prioritate mai mică decât înmulţirea → utilizarea parantezelor pt. evitarea erorilor

OPERATORI

Caracteristici

- operatori de înmulţire - binari
- operatori diverşi
  - ** - ridicare la putere
    - operandul din stânga de tip întreg sau flotant
    - puterea - obligatoriu tip întreg
  - abs - pe orice tip numeric
  - not
    - operator logic, unar
    - operează pe obiecte de tip Boolean, Bit, Std_logic şi pe vectori de astfel de elemente

ATRIBUTE

Generalităţi

- caracteristică asociată unui tip sau unui obiect, care poate fi cunoscută în mod dinamic, în timpul rulării
- notaţie - adăugarea unui **apostrof** după numele tipului sau obiectului
- atribute:
  - predefinite
  - definite de proiectant

ATRIBUTE

Atribute predefinite

- pot returna: valori (tipizate), funcţii, tipuri, intervale de variaţie
- se aplică unor prefixe care pot fi: valori, tipuri, etichete
- simplifică scrierea
- apar în funcţii utilitare

ATRIBUTE

Atribute predefinite

- atribute definite pe tipuri
  - T desemnează un tip (obiectul asupra căruia acţionează atributul), prefix al atributului
  - obiect’nume_atribut(parametri)

ATRIBUTE

Atribute predefinite

- atribute definite pe tipuri
  - tip sau subtip **scalar** T; X = tip scalar
    - **Atribut** — **Rezultat**

T'left	Limita la stânga a lui T
T'right	Limita la dreapta a lui T
T'low	Limita inferioară a lui T
T'high	Limita superioară a lui T
T'base	Tipul de bază a lui T
T'image(X)	Şirul X
T'value(X)	Valoare de tip T

ATRIBUTE

Atribute predefinite

- atribute definite pe tipuri
  - tip sau subtip **discret** sau **fizic** T; X membru al lui T; N număr întreg

Atribut	Rezultat
T'pos(X)	Numărul poziţiei lui X în T
T'val(N)	Valoarea la poziţia N în T
T'leftof(X)	Valoarea în T, cu o poziţie în stânga lui X
T'rightof(X)	Valoarea în T, cu o poziţie în dreapta lui X
T'pred(X)	Valoarea în T, cu o poziţie mai mică decât X
T'succ(X)	Valoarea în T, cu o poziţie mai mare decât X
T'ascending	Valoare booleană pt. interval crescător sau descrescator

ATRIBUTE

Atribute predefinite

- atribute definite pe tipuri sau subtipuri **tablou**
  - A = tablou; N = număr întreg între 1 şi numărul dimensiunilor lui A
    - **Atribut** **Rezultat**

A'left(N)	Limita stânga a domeniului indicelui dimensiunii N a lui A
A'right(N)	Limita dreapta a domeniului indicelui dimensiunii N a lui A
A'low(N)	Limita inferioară a domeniului indicelui dimensiunii N a lui A
A'high(N)	Limita superioară a domeniului indicelui dimensiunii N a lui A
A'range(N)	Domeniul indicelui dimensiunii N a lui A
A'reverse_range(N)	Inversul domeniului indicelui dimensiunii N a lui A
A'length(N)	Lungimea domeniului indicelui dimensiunii N a lui A
A'ascending(N)	Valoare booleană pentru direcţia indicelui dimensiunii N a lui A

ATRIBUTE

Atribute predefinite

- atribute definite pe **semnale** S
  - atribute semnal

Atribut	Rezultat
S'delayed(T)	Semnal S întârziat cu T unităţi de timp
S'stable(T)	Valoare booleană True dacă S e fără evenimente în T
S'quiet(T)	Valoare booleană True dacă S e inactiv în T
S'transaction	Modificare valoare Bit de câte ori S este activ

ATRIBUTE

Atribute predefinite

- atribute definite pe **semnale** S
  - atribute funcţie
    - **Atribut** **Rezultat**

S'event	Valoare booleană True pt. eveniment pe S
S'active	Valoare booleană True dacă S e activ
S'last_event	Timpul trecut de la ultimul eveniment pe S (valoare Time)
S'last_active	Timpul trecut de la ultima activare a lui S (valoare Time)
S'last_value	S imediat înainte de ultima modificare
S'driving_value	Permite o operaţie de atribuire
S'driving	Valoare booleană dacă S nu este deconectat

ATRIBUTE

- Atribute predefinite
  - atribute definite pe obiecte în sens larg X
    - utilizate pentru elaborare de mesaje

Atribut	**Rezultat**
X'simple_name	Numele X
X'path_name	Numele X şi etichetele de revenire la X
X'instance_name	Numele X, etichetele de revenire la X, informaţii de configurare

ATRIBUTE

Atribute definite de utilizator

- declarare atribut
  - **attribute** nume_atribut: tip_atribut;
- specificare atribut (primeşte valoare)
  - **attribute** nume_atribut **of** obiect **is** expresie;
  - nu pot fi decât constante, deci sunt statice
- utilizare atribut - cu notaţia cu apostrof
  - obiect’nume_atribut

DOMENIUL SECVENTIAL - PROCESE

Definiţii

- unitatea de bază pentru descrierea de tip comportamental (funcţional)
- **procesul** = o serie de operaţii secvenţiale care în timpul simulării constituie o singură acţiune
- **procesul** = obiectul fundamental manipulat de simulator → orice descriere VHDL = un set de procese caracterizate de:
  - semnalele la care sunt sensibile (active)
  - operaţiile secvenţiale executate de fiecare

PROCESE

Sintaxa

{etichetă:} {**postponed**} **process** {listă_de_sensibilitate}

...

... Zona de declaraţii locale procesului

...

begin

...

... Instrucţiuni secvenţiale

...

end {**postponed**} **process** {etichetă};

PROCESE

Sintaxa

- în partea declarativă:
  - este interzisă declararea semnalelor
  - se pot declara variabile
  - se pot declara subprograme interne

PROCESE

Execuţia

- un proces există nedefinit - este global
- durata de viaţă a unui proces este cea a simulării
- timpul de execuţie al unui proces este zero
- procesele se execută în paralel, în mod concurent

PROCESE

Execuţia

- un **sistem real** îşi execută secvenţa de activităţi specifice, pentru care a fost construit, în **buclă infinită**
- orice **instrucţiune concurentă** poate fi transcrisă în termenii unui proces = **procesul echivalent**
- un **proces** nu se termină niciodată - el execută în buclă lista de instrucţiuni secvenţiale - este **ciclic**

PROCESE

Suspendarea şi reactivarea

- funcţionarea dispozitivelor electronice:
  - operează în buclă infinită
  - execută operaţiile specifice
  - îşi **suspendă** funcţionarea
  - aşteaptă îndeplinirea unor **condiţii de reactivare**
  - **reiau** operaţiile
- un proces se execută până se întâlneşte o instrucţiune **wait**

PROCESE

Instrucţiunea wait

- scop: emularea funcţionării reale
- suspendă procesul când operaţiile secvenţiale prevăzute au fost efectuate
- Toate celelalte procese/instructiuni functioneaza in mod normal
- reactivează procesul când sunt îndeplinite condiţiile specificate → mai multe tipuri de instrucţiuni **wait**, pentru a asigura varietatea de condiţii reale

PROCESE

Instrucţiunea wait

- 3 tipuri de instrucţiuni
  - **wait for** expresie de tip Time - se aşteaptă trecerea unui interval de timp
  - **wait until** condiţie de tip Boolean - se aşteaptă până condiţia devine True în urma unei modificări
  - **wait on** listă de sensibilitate - se aşteaptă până un semnal din listă îşi modifică valoarea
- sintaxa completă:

wait {**on** listă_de_semnale} {**until** condiţie_booleană} {**for** timp};

PROCESE

Instrucţiunea wait

- localizarea
  - **wait** poate apărea oriunde în proces
  - într-un proces pot exista mai multe instrucţiuni **wait**
  - lista de sensibilitate poate apărea după **process** şi e echivalentă cu "**wait on** listă de sensibilitate" aflată la sfârşitul procesului

PROCESE

Procese cu lista de sensibilitate

- Pot exista procese fara instructiunea wait
- Suspendarea procesului se face la sfarsitul lui
- Procesul functioneaza ca si cum ar exista o instructiune wait on lista_de_sensibilitate la sfarsit
- Aceasta nu mai este necesara

PROCESE

Instrucţiunea wait

- restricţii
  - semnalele din lista de sensibilitate - să fie statice
  - **wait on** nu poate fi utilizată în proces când există listă de sensibilitate
  - **wait** nu poate fi utilizată în procedurile apelate de proces

Instrucţiunea wait – exemplu (cu si fara)

```
entity mux2la1En is
    port (i0, i1, s: in Std_Logic;
          en: in Std_Logic;
           f: out Std_Logic);
end mux2la1En;
```

PROCESE

Instrucţiunea wait – exemplu cu

```
architecture BehaviorEn of mux2la1En is  begin
    process
    begin
        if en = '1' then
            if s = '0' then f <= i0;
            else f <= i1;
            end if;
        else f <= '1';
        end if;
wait on i0, i1, s;
wait until en = '1';
wait for 20 ns;  end
process;
end BehaviorEn
```

PROCESE

Semnale în procese

- **restricţii de utilizare** a semnalelor în procese:
  - în procese nu se pot declara semnale
  - orice asignare a unei valori unui semnal are efect doar când procesul se suspendă - până atunci se păstrează valorile anterioare
  - la suspendarea procesului ultima asignare a unei valori unui semnal este luată în considerare
- dacă semnalul este pe lista de sensibilitate a procesului, modificarea semnalului reactivează procesul

PROCESE

Procese cu semnale in lista de sensibilitate

- Exemplu – numarator simplu
  - Vivado
  - Se poate vedea si utilitatea atributelor

PROCESE

Variabile în procese

- variabila permite stocarea temporară a datelor
- se poate defini în cadrul procesului - cuvânt cheie **variable**
- utilizare - la descriere de algoritmi în procese
- **OBS**: VARIABILELE NU SE POT VEDEA IN SIMULARI!!

PROCESE

Variabile în procese

- asignarea de valori:
  - cu simbolul :=
  - instantanee
  - de câte ori este necesar
- poate avea orice tip sau subtip posibil, constrâns sau neconstrâns
- valoarea inițială - expresie statică, de același tip cu tipul variabilei

PROCESE

Variabile în procese - exemple

```
variable MEM is array (NATURAL range<>,NATURAL range<>)  of
    Std_Logic;
variable RAM1: MEM (0 to 1023, 0 to 7);
variable timp: TIME;
variable Condiţie: Boolean := True;
variable X: Integer := 7;
```

Concluzii

- Tipuri de date
  - Scalare – enumerate, intregi, fizice
  - Compuse – vectori
  - Exemple
- Operatori
  - Exemple
- Atribute
- Domeniul secvential
- Procese
  - Structura unui process
  - Instructiunea wait
  - Lista de sensibilitate
  - Variabile
  - Exemple