**الجلسة السادسة:**

**تصميم الدارات الالكترونية بلغة VHDL / 4/**

**البنية التسلسلية للغة VHDL**

**1 -التعليمة: Process**

إن تعليمة Process من التعليمات التفرعية في لغة VHDL فهي كتعليمة تشير إلى أن كل ما يكتب ضمنها يتم تنفيذه بصورة تتابعية وعادة ما تكتب التعليمات IF, WAIT, CASE, LOOP ضمن هذه التعليمة وهذه التعليمات كلها من النمط التتابعي الذي ينفذ بصورة تسلسلية من الأعلى إلى الأسفل وبصورة دورية مادامت تعليمة Process فعالة .

تزود تعليمة Process بعدد من المتحولات أو الشروط المنطقية تسمى قائمة الحساسية sensitivity list وهي مجموعة من المتحولات إذا ذكرت ضمن هذه اللائحة لا يتم الدخول إلى كتلة التعليمة Process إلا عند حدوث تغيير على هذه المتحولات وإلا فإن مجموعة التعليمات ضمن Process لا تنفذ ولا يتم الدخول إلى هذه الكتلة أصلاً .

تكتب تعليمة Process في الكود الرئيسي وتنفذ كل مرة يتغير فيها احد متحولات الحساسية أو الشروط المنطقية الصيغة العامة لهذه التعليمة هي :

ملاحظة : كل مايكتب ضمن قوسين متوسطي الحجم [] فهو اختياري .

[label:] PROCESS (sensitivity list)

[VARIABLE name type [range] [:= initial\_value;]]

BEGIN

(sequential code)

END PROCESS [label];

يتم في القسم الأول منها قبل Begin التصريح عن متحولات خاصة إذا دعت الحاجة لها لتستعمل كوسيط ضمن التعليمات الواردة في جسم تعليمة PROCESS ,كما يمكن لهذه المتحولات أن تلقن بقيمة ابتدائية ما لأغراض المحاكاة.

**تطبيق 1 :** توصيف قلاب من النوع D

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

--------------------------------------

ENTITY dff IS

PORT (d, clk, rst: IN STD\_LOGIC;

q: OUT STD\_LOGIC);

END dff;

--------------------------------------

ARCHITECTURE behavior OF dff IS

BEGIN

**PROCESS** (clk, rst)

**BEGIN**

IF (rst='1') THEN

q <= '0';

ELSIF (clk'EVENT AND clk='1') THEN q <= d;

END IF;

**END PROCESS;**

END behavior;



**2 -التعليمة: IF** وهي من التعليمات التتابعية التي تستخدم في حالات التفريع المشروط للبرنامج ، تستخدم ضمن تعليمة Process **.**

الصيغة العامة لها من الشكل :

IF conditions THEN assignments;

ELSIF conditions THEN assignments;

..

ELSE assignments;

END IF;

**مثال :**

IF (x<y) THEN temp := "11111111" ;

ELSIF (x=y AND w='0') THEN temp:= "11110000" ;

ELSE temp:=(OTHERS =>'0');

**تطبيق 2:**

توصيف عداد رقمي صاعد حلقي من القيمة 0 إلى القيمة 9 وذلك بحيث يزداد خرجه عدة واحدة عند كل جبهة صاعدة لنبضات الساعة clk .



LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

---------------------------------------------

ENTITY counter IS

PORT (clk : IN STD\_LOGIC;

digit : OUT INTEGER RANGE 0 TO 9);

END counter;

---------------------------------------------

ARCHITECTURE counter OF counter IS

BEGIN

count: PROCESS(clk)

VARIABLE temp : INTEGER RANGE 0 TO 10;

BEGIN

IF (clk'EVENT AND clk='1') THEN

temp := temp + 1;

IF (temp=10) THEN temp := 0;

END IF;

END IF;

digit <= temp;

END PROCESS count;

END counter;



3**- التعليمة *Wait :***

وهي من التعليمات التتابعية التي لها عدة أشكال تستخدم لضبط عمليات التنفيذ للتعليمات ضمن كتلة Process , وبوجودها تفقد لائحة الحساسية لتعليمة Process فعاليتها , وبالتالي لا يعود لذكرها داع . الصيغة العامة لهذه التعليمة لها عدة أشكال:

**الشكل الأول :** وهو الانتظار حتى تحقق شرط منطقي ما متعلق بإشارة واحدة فقط , هذا الشكل يستعمل كبديل عن لائحة الحساسية ويجب ذكره في بداية كتلة Process , وبالتالي فإن كتلة التعليمات في Processتنفذ في كل مرة يتحقق فيها الشرط .

**مثال :**

-- 8-bit register with synchronous reset.

**PROCESS** -- no sensitivity list

BEGIN

WAIT UNTIL (clk'EVENT AND clk='1');

IF (rst='1') THEN

output <= "00000000";

ELSIF (clk'EVENT AND clk='1') THEN

output <= input;

END IF;

**END PROCESS;**

**الشكل الثاني :** وهو الانتظار حتى يطرأ تغيير ما على قيمة إحدى الإشارات لتنفذ التعليمات التالية .

**مثال :**

-- 8-bit register with asynchronous reset.

**PROCESS**

BEGIN

WAIT ON clk, rst;

IF (rst='1') THEN

output <= "00000000";

ELSIF (clk'EVENT AND clk='1') THEN

output <= input;

END IF;

**END PROCESS;**

**الشكل الثالث :** يستخدم فقط لأغراض المحاكاة , وهو الانتظار لفترة زمنية ما .

**مثال :**

WAIT FOR 5ns ;

**تطبيق 3:**

توصيف قلاب من النوع D باستخدام تعليمة WAIT

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

--------------------------------------

ENTITY dff IS

PORT (d, clk, rst: IN STD\_LOGIC;

q: OUT STD\_LOGIC);

END dff;

--------------------------------------

ARCHITECTURE dff OF dff IS

BEGIN

PROCESS

BEGIN

WAIT ON rst, clk;

IF (rst='1') THEN

q <= '0';

ELSIF (clk'EVENT AND clk='1') THEN

q <= d;

END IF;

END PROCESS;

END dff;

4**- تعليمة *Case:***

وهي من التعليمات التتابعية التي تستخدم ضمن كتلة Process تقوم هذه التعليمة بعمل مشابه للتعليمة التي سبق و تعرفنا عليها, وهي تعليمة When التفرعية ، مهمة هذه التعليمة إسناد قيمة ما لمتحول بحسب شرط ما .

**مثال :**

CASE control IS

WHEN "00" => x<=a; y<=b;

WHEN "01" => x<=b; y<=c;

WHEN OTHERS => x<="0000"; y<="ZZZZ";

END CASE;

كما شاهدنا من خلال المثال ، يمكن لهذه التعليمة أن تسند قيماً لعدة إشارات معاً على خلاف التعليمة المقابلة لها في المجال التفرعي التعليمةWhen التي لا تسمح بالتعامل إلا مع متحول وحيد .

**تطبيق 4:**

توصيف قلاب من النوع D باستخدام تعليمة CASE 

ENTITY dff IS

PORT (d, clk, rst: IN BIT;

q: OUT BIT);

END dff;

----------------------------------------------

ARCHITECTURE dff3 OF dff IS

BEGIN

**PROCESS** (clk, rst)

BEGIN

**CASE** rst **IS**

WHEN '1' => q<='0';

WHEN '0' =>

***IF (clk'EVENT AND clk='1') THEN***

***q <= d ;***

***END IF;***

WHEN OTHERS => NULL; -- Unnecessary, rst is of type -- BIT

**END CASE;**

**END PROCESS;**

END dff3;

----------------------------------------------

ملاحظة :

تستخدم العبارة Null إذا أردنا عدم تغيير قيمة المتحول بعد تجاهل حالة الشرط .

**5- تعليمة** **Loop** :

وهي من التعليمات التتابعية التي تستخدم ضمن كتلة Process , تقوم هذه التعليمة بتكرار تنفيذ قسم من البرنامج عدة مرات دون تكرار كتابة هذا الكود بعدد المرات التي من المطلوب تنفيذها . لهذه التعليمة عدة أشكال سنتعرف عليها من خلال الأمثلة .

**ملاحظة :** تستخدم تعليمة Exitمن داخل الحلقة للخروج من داخل حلقة التكرار عند تحقق شرط ما, تستخدم تعليمة Next من داخل الحلقة لتجاوز بعض التعليمات من داخل حلقة التكرار عند تحقق شرط ما .

**أمثلة :**

FOR i IN 0 TO 5 LOOP

x(i) <= enable AND w(i+2);

y(i) <= w(i);

END LOOP;

------------------------------------------------

WHILE (i < 10) LOOP

WAIT UNTIL clk'EVENT AND clk='1';

(other statements)

END LOOP;

-------------------------------------

FOR i IN data'RANGE LOOP

CASE data(i) IS

WHEN '0' => count:=count+1;

WHEN OTHERS => EXIT;

END CASE;

END LOOP;

------------------------------------------------

FOR i IN 0 TO 15 LOOP

NEXT WHEN i=skip; -- jumps to next iteration

(...)

END LOOP;

**تطبيق 5:**

توصيف نظام يقوم بإحصاء عدد الأصفار اليسارية الموجودة في إشارة الدخل Data , على سبيل المثال إذا كانت قيمة الدخل00000001يظهر البرنامج القيمة 7 على الخرج وهكذا ...



LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

--------------------------------------------

ENTITY LeadingZeros IS

PORT ( data: IN STD\_LOGIC\_VECTOR (7 DOWNTO 0);

zeros: OUT INTEGER RANGE 0 TO 8);

END LeadingZeros;

--------------------------------------------

ARCHITECTURE behavior OF LeadingZeros IS

BEGIN

PROCESS (data)

VARIABLE count: INTEGER RANGE 0 TO 8;

BEGIN

count := 0;

FOR i IN data'RANGE LOOP

CASE data(i) IS

WHEN '0' => count := count + 1;

WHEN OTHERS => EXIT;

END CASE;

END LOOP;

zeros <= count;

END PROCESS;

END behavior;

--------------------------------------------