**DISSENY DIGITAL BÀSIC 2019-2020**

***PRÀCTICA 1: Implementació de funcions lògiques***

***(dies 21, 22, 23, 24 i 25 d’octubre)***

L’objectiu d’aquesta pràctica és familiaritzar-se amb el programa de simulació *QuestaSim* (i *ModelSim*) i començar a fer servir les seves propietats per realitzar funcions lògiques tal com les descrites a les classes de teoria. En concret volem realitzar les funcions bàsiques:

* Inversió (ja fet a la pràctica 0)
* Producte lògic (AND)
* Suma lògica (OR)

A més, volem realitzar també

* Implementació de funcions lògiques

En aquesta pràctica heu de generar l’entitat, les arquitectures i el banc de proves corresponents a les portes AND i OR de dues i tres entrades. Caldrà, doncs, que genereu 7 entitats, que anomenareu **inversor** (la mateixa de la pràctica inicial), **and2**, **and3**, **and4**, **or2**, **or3** i **or4**,cadascuna amb la seva arquitectura **logica**. Recordeu que el nom de l’arquitectura fa referència sempre a una entitat concreta, que utilitzarà les definicions de les funcions lògiques AND, OR i NOT que s’han donat a teoria i que podeu trobar a qualsevol dels llibres de l’assignatura.

Feu servir com a variables d’entrada **a**, **b** (per a les portes de 3 entrades); **a**, **b**, **c** (per a les portes de 3 entrades) i **a**, **b**, **c** i **d** (per a les de 4). La sortida serà **z** per a cadascuna de les portes. A mode d’exemple, us presentem l’entitat i l’arquitectura de la porta **and2**:

ENTITY and2 IS

PORT(a, b: IN BIT; z: OUT BIT);

END and2;

ARCHITECTURE logica OF and2 IS

BEGIN

Z <= a AND b;

END logica;

Genereu un banc de proves per testejar cada dispositiu a testejar, i així comprovar el el seu correcte funcionament. Després generau un banc de proves que us permeti testejar totes les entitats simultàniament. Un aspecte molt important serà que el banc de proves haurà de realitzar-se de manera que es contemplin tots els casos possibles de les variables d’entrada, és a dir, que les variables d’entrada recorrin totes les combinacions de valors possibles. A la pràctica 0 ja s’ha mostrat una forma de fer-ho, però en el cas de senyals periòdics, és a dir, que tornen a repetir-se amb durada determinada, hi ha una manera més compacta de fer-ho. A continuació detallem com fer-ho en el cas de la porta **and2**, fent servir l’entitat i l’arquitectura de dalt:

ENTITY banc\_de\_proves IS

END banc\_de\_proves;

ARCHITECTURE test OF banc\_de\_proves IS

-- Aquí definim els components el comportament dels quals volem comprovar.

-- Tot i que encara no hem relacionat amb l’entitat que utilitzarem,

-- el **component** ha de tenir **EXACTAMENT** els mateixos noms dels senyals d’entrada i

-- sortida que l’entitat a la que lligarem.

COMPONENT la\_porta\_and2

PORT (a,b: IN BIT; z: OUT BIT);

END COMPONENT;

-- Declarem els senyals externs que hi han i de quin tipus són. Recordeu que

-- encara no es determina quins seràn d’entrada i quins de sortida.

SIGNAL ent1, ent2, sortida: BIT;

-- Relacionem el component que volem testejar amb una entitat i arquitectura

-- que abans ja haurem generat i compilat. Cal que ambdues estiguin en un fitxer .VHD

-- en el directori de treball, si no estan incorporades al mateix fitxer que estem

-- generant.

FOR DUT1: la\_porta\_and2 USE ENTITY WORK.and2(logica);

-- Hem acabat la declaració i ara comença el cos de l’arquitectura

BEGIN

-- Aquí establim la relació entre els senyals abans definits i els

-- terminals del component a testejar. El programa els relaciona

-- **en el mateix ordre** que estan a l’entitat: el 1r senyal extern serà el 1r

-- senyal de l’entitat

DUT1: la\_porta\_and2 PORT MAP (ent1,ent2,sortida);

-- Comença l’anàlisi de la variació de les variables, que cal especificar

-- explícitament. Indiquem al PROCESS la variació de totes les variables

-- que volem que el programa analitzi, aquí les d’entrada ent1 i ent2.

PROCESS (ent1,ent2)

BEGIN

ent1<=NOT ent1 AFTER 50 ns;

ent2<=NOT ent2 AFTER 100 ns;

END PROCESS;

END test;

Alternativament, i fent servir ara el que ja vàrem aprendre a la pràctica 0, la variació dels senyals també la podríem fer de forma manual, indicant tots els canvis. Per tant, podríem escriure al codi del banc de proves el següent, tenint en compte que cal substituir el text en **verd** al codi anterior pel següent:

PROCESS

BEGIN

ent1 <= '0';

ent2 <= '0';

WAIT FOR 50 ns;

ent1 <= '1';

ent2 <= '0';

WAIT FOR 50 ns;

ent2 <= '0';

ent1 <= '1';

WAIT FOR 50 ns;

ent1 <= '1';

ent2 <= '1';

WAIT FOR 50 ns;

ent1 <= '0';

ent2 <= '0';

WAIT FOR 50 ns;

END PROCESS;

Queda evident que en aquest darrer codi s’han d’escriure de forma manual totes les combinacions possibles que adquireixen les entrades, fent tot molt extens i feixuc pel cas de 8 entrades, i encara més ho seria en el cas de 32 entrades diferents.

**Treball a desenvolupar de forma autònoma:**

1. En un fitxer .vhd genereu les 7 entitats **inversor**, **and2**, **or2**, **and3**, **or3**, **and4** i **or4**, fent servir l’arquitectura **logica**, tal i com es mostra a l’exemple al principi del guió: senyals d’entrada **a**, **b** (per a les portes de 3 entrades); **a**, **b**, **c** (per a les portes de 3 entrades) i **a**, **b**, **c** i **d** (per a les de 4). La sortida serà **z** per a cadascuna de les portes.
2. En un segon fitxer .vhd genereu ara un banc de proves **bdp\_portes** (amb la seva arquitectura **test**) que us permeti de testejar i mostrar el comportament simultani de TOTES les portes. Per fer-ho, utilitzeu quatre senyals **ent1**, **ent2**, **ent3** i **ent4** per a les entrades, dels quals **ent1**, **ent2** i **ent3** els fareu servir com a senyals d’entrada a les portes de tres entrades, **ent1** i **ent2** els fareu servir com a senyals d’entrada a les portes de dues entrades i **ent1**, a l’inversor. Els senyals de sortida seran **sort\_and2\_logica** per a la porta AND de 2 entrades, realitzada amb l’arquitectura **logica**; **sort\_or2\_logica** per a la porta OR de 2 entrades, realitzada amb l’arquitectura **logica**; ... i així successivament. Això us donarà un total de 7 senyals de sortida, un per cada tipus de porta (**inversor**, **and2**, **or2**, **and3**, **or3**, **and4** i **or4**).

NOTA: els senyals d’entrada de les entitats i els senyals del banc de proves no tenen perquè tenir el mateix nom.

En el banc de proves és important que els senyals d’entrada variïn més lentament que no pas el retard que més endavant anirem introduint en les portes lògiques. Per exemple, el temps mínim de variació periòdica es pot fixar en uns 50ns. Per a això feu que el senyal **ent1** canviï cada 50ns, **ent2** cada 100ns, **ent3** cada 200ns, etc... (de manera semblant al exemple del banc de proves de mostra).

Recordeu de fer córrer la simulació un temps prou llarg per tal que es puguin veure totes les combinacions possibles de les entrades. Amb l’ajuda del cursor, comproveu que el comportament és el que s’espera.

**Aquest és el treball que haureu de pujar a través del campus virtual abans de la vostra corresponent sessió de pràctiques. El termini d’aquesta tasca és 48 hores abans de la vostra sessió de pràctiques, que serà diferent per als diferents grups. Un cop passat aquest temps ja no serà possible pujar els fitxers.**

Els fitxers que haureu de pujar a través del campusvirtual2, tal com us ho hem ensenyat a la pràctica 0, seran 2:

1. Un fitxer amb totes les entitats i arquitectures de les portes **inversor**, **and2**, **and3**, **and4**, **or2**, **or3** i **or4**). El nom del fitxer serà:

**Py\_Cognom1\_Cognom2\_Nom\_01.vhd** (**y** és el número de la pràctica, en aquest cas, serà 1). En el segon fitxer poseu el banc de proves d’aquestes portes lògiques, amb el nom **Py\_Cognom1\_Cognom2\_Nom\_02.vhd**.

1. Un cop pujats els fitxers, no oblideu de picar “Enviar per avaluar”.