**DISSENY DIGITAL BÀSIC 2016-2017**

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

L’objectiu d’aquesta pràctica és familiaritzar-se amb el programa de simulació *QuestaSim* (i *ModelSim*) i començar a utilitzar les seves propietats per realitzar funcions lògiques 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 5 entitats, que anomenareu **inversor** (la mateixa de la pràctica inicial), **and2**, **and3**, **or2** i **or3**,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. És, per tant, igual que la que es va fer servir per a la pràctica 0. En tots aquests casos feu servir com a variables d’entrada **a**, **b**, **c** (per a les portes de 3 entrades) 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;

Haureu de generar un banc de proves per a cada dispositiu al qual feu un test, de forma que pugueu comprovar que funciona correctament. Després haureu de generar un banc de proves que us permeti de 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 tots les combinacions de valors possibles. A la pràctica 0 ja s’ha mostrat una forma de fer-ho, però quan els senyals poden ser periòdics, és a dir, poden repetir-se cada un cert temps, hi ha una manera més senzilla de fer-ho. A continuació detallem com fer-ho per a 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 de variables 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

-- el programa encara no sap quins serán d’entrada i de sortida.

SIGNAL ent1, ent2, sortida: BIT;

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

-- que haurem generat i compilat amb anterioritat. Cal que el programa l’hagi

-- incorporat en el directori de treball

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

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

BEGIN

-- Aquí establim la relació entre els senyals que hem definit 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. Aquí diem que analitzarem la variació de les variables

-- ent1, ent2 i ent3 i, per aquest motiu, ho indiquem al PROCESS dient

-- la variació de quines variables volem que el programa analitzi.

PROCESS (ent1,ent2)

BEGIN

ent1<=NOT ent1 AFTER 100 ns;

ent2<=NOT ent2 AFTER 50 ns;

END PROCESS;

END test;

D’altra banda i fent servir ara el que ja vàrem aprendre a la pràctica 0, la variació dels senyals també la podem fer de forma manual, descrivint 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

ent3 <= '0';

ent2 <= '0';

ent1 <= '0';

WAIT FOR 100 ns;

ent3 <= '1';

ent2 <= '0';

ent1 <= '0';

WAIT FOR 100 ns;

ent3 <= '0';

ent2 <= '1';

ent1 <= '0';

WAIT FOR 100 ns;

ent3 <= '1';

ent2 <= '1';

ent1 <= '0';

WAIT FOR 100 ns;

ent3 <= '0';

ent2 <= '0';

ent1 <= '1';

WAIT FOR 100 ns;

ent3 <= '1';

ent2 <= '0';

ent1 <= '1';

WAIT FOR 100 ns;

ent3 <= '0';

ent2 <= '1';

ent1 <= '1';

WAIT FOR 100 ns;

ent3 <= '1';

ent2 <= '1';

ent1 <= '1';

WAIT FOR 100 ns;

ent3 <= '0';

ent2 <= '0';

ent1 <= '0';

WAIT FOR 100 ns;

END PROCESS;

Com podeu comprovar, en aquest darrer codi hem d’escriure de forma manual totes les combinacions possibles que adquireixen les entrades, el que fa que sigui molt extens pel cas de 8 entrades i encara més per a 32 entrades diferents.

**Treball a desenvolupar de forma autònoma:**

1. Genereu les entitats **and2**, **or2**, **and3**, **or3**, **and4** i **or4**, fent servir l’arquitectura logica, tal i com es mostra a l’exemple del començament del guió.
2. Genereu ara un banc de proves **bdp\_portes** (amb la seva arquitectura **vectors**) que us permeti de testejar i mostrar el comportament simultani de TOTES les portes. Per a això, feu servir quatre senyals d’entrada **ent1**, **ent2**, **ent3** i **ent4** dels quals **ent1**, **ent2** i **ent3** els fareu servir com a senyals d’entrada a les portes de dues 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**).

**Treball a desenvolupar al laboratori:**

1. A l’entitat **and2** (definida al començament d’aquesta pràctica), afegiu-li una nova arquitectura per incorporar-li un retràs de 2 ns (ara li direu **logicaretard**). Modifiqueu el banc de proves per afegir-hi aquest segon dispositiu (DUT).
2. Implementeu ara l’entitat i les arquitectures **logica** i **logicaretard** de la funció OR de 2 entrades (**or2**) i comproveu que el funcionament és el correcte. Igual que abans, a les entitats tindrem que les variables d’entrada seran **a** i **b** i la variable de sortida, **z**. De la mateixa manera que abans, genereu un banc de proves per comprovar el seu funcionament.
3. Igual que l’anterior però per les funcions AND i OR de 3 entrades i AND i OR de 4 entrades. Les entitats seran, respectivament, **and3**, **or3**, **and4** i **or4** i les arquitectures seran **logica** i **logicaretard**.

1. Genereu ara un banc de proves **bdp\_portes** (amb la seva arquitectura **vectors**) que us permeti de testejar i mostrar el comportament simultani de TOTES les portes, per les dues arquitectures que hem fet servir per a cada entitat, fent servir el mateix esquema descrit a l’apartat 2. Això us donarà un total de 14 senyals de sortida, 2 per cada tipus de porta (**inversor**, **and2**, **or2**, **and3**, **or3**, **and4** i **or4**).
2. Genereu l’entitat **xor2** (porta **XOR** de dues entrades) amb les arquitectures **logica** i **logicaretard** (retard de 2ns). Els senyals d’entrada de l’entitat seran **a** i **b**. Genereu el corresponent banc de proves **bdp\_funcio** (amb la seva arquitectura **vectors**) i amb els senyals que provarem seran **sort\_xor2\_logica**  i **sort\_xor2\_logicaretard**. Feu que els senyals d’entrada recorrin tots els valors possibles.

En els bancs de proves feu que els senyals d’entrada variïn més lentament que no pas el retràs de les portes lògiques. Per exemple, feu que variïn cada 50ns. Per a això feu que el senyal **ent1** canviï cada 200ns, **ent2** cada 100ns i **ent3** cada 50ns (tal com es mostra en el banc de proves de mostra).

Recordeu de fer la simulació per un temps prou llarg com per que es puguin veure totes les combinacions possibles de les entrades. I comproveu que el comportament és el que s’espera.

**Treball que haureu de pujar a través del campus virtual abans de la vostra corresponent sessió de pràctiques. L’heu de pujar, com a tard, 24 hores abans de la vostra sessió de pràctiques, que serà diferent per als diferents grups. 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**, **or2** i **or3**) i el banc de proves que permeti comprovar el funcionament correcte d’aquestes. El nom del fitxer serà: **Py\_xxx\_Cognom1\_Cognom2\_Nom\_1.vhd** (**y** és el número de la pràctica, en aquest cas, serà 2; **xxx** serà el vostre grup de pràctiques, per exemple, D01, A23, ..., que teniu assignat des del començament i que apareix a la 1a columna del document d’assignació de places al laboratori que està penjat al campusvirtual)
2. Un cop pujats els fitxers, no oblideu d’Enviar per avaluar.

**Recordeu que aquestes entitats, arquitectures i bancs de proves es faran servir en properes pràctiques**