

**PRÀCTICA 3:**

**DISSENY BANC REGISTRES I PC**

**Sergi Carol Bosch i Enric Lenard Uró**

Grau en Enginyeria de Sistemés TIC

**Arquitectura de Computadors**

Curs 2013-14, Grup 10, G12 de pràctiques

Realització de la pràctica: 07/04/2014

Lliurament del treball: 28/04/2014

Realització pràctica

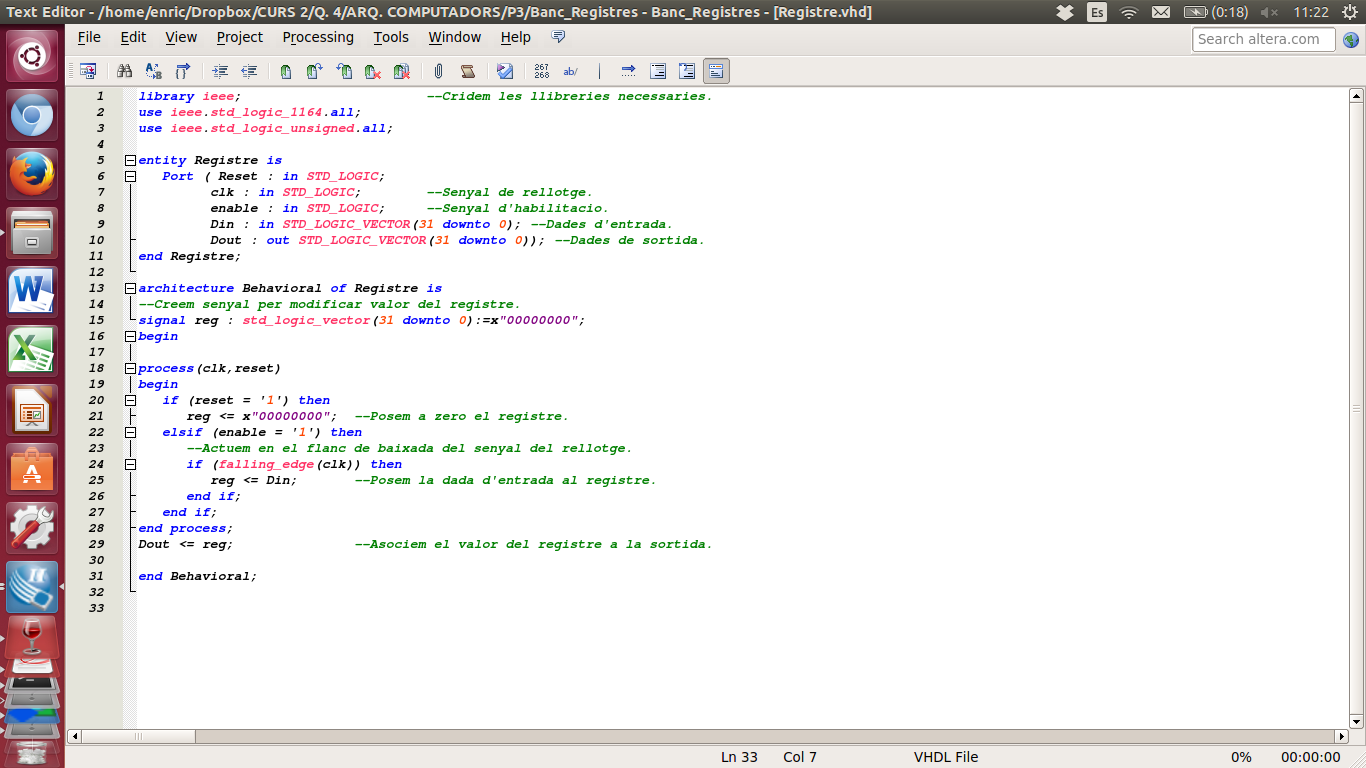
• L’objectiu d’aquesta practica és crear el banc de registres i el comptador de programa descrits a classe, per aquest motiu anirem realitzant un disseny incremental.

En primer lloc, començarem pel banc de registres, partint d'un senzill registre de 32 bits que anirem ampliant en la seva funcionalitat fins a la construcció del banc complet. Finalment, implementarem el comptador de programa i el mòduls necessaris per el correcte funcionament d’aquest.

Per tant, primer de tot dissenyem el modul Registre que tindrà les següents condicions:

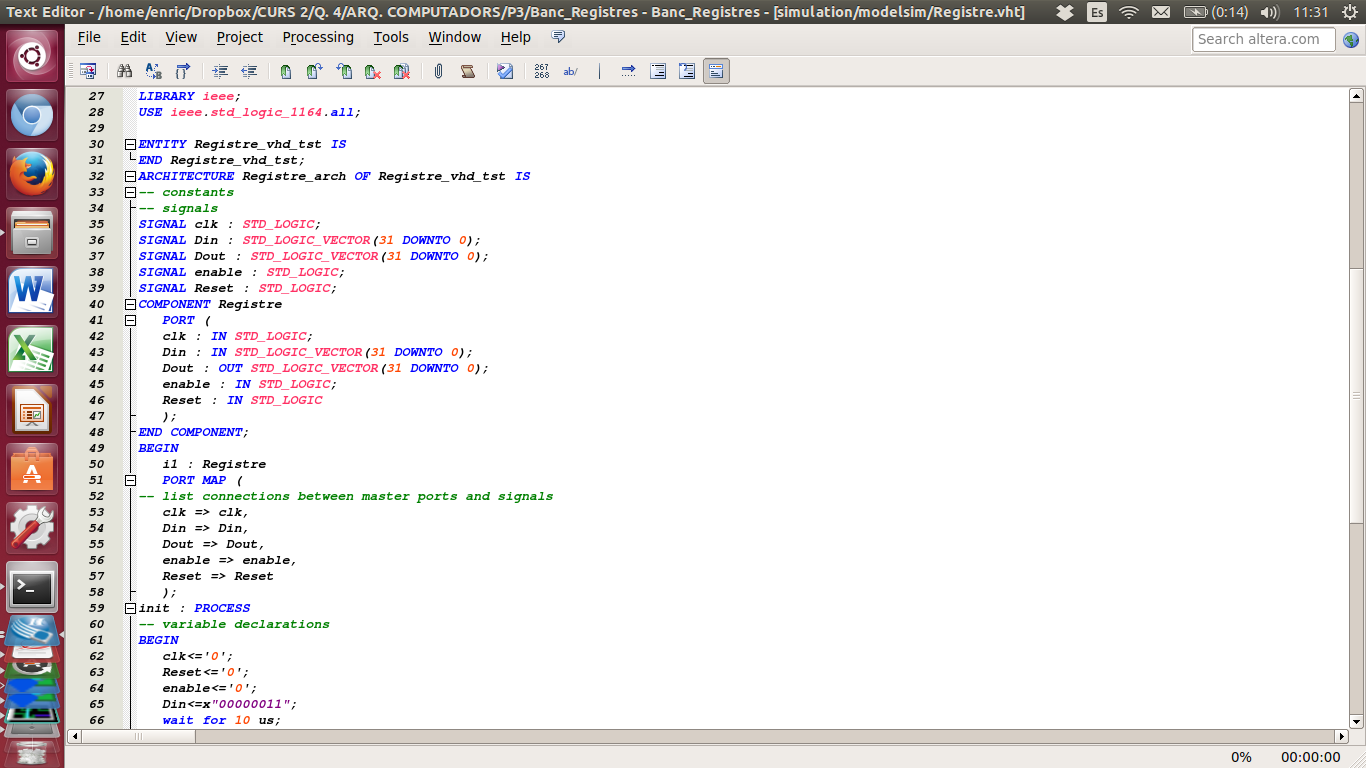
* Un senyal d'habilitació o escriptura (enable) actiu a nivell alt.
* Un senyal de rellotge (clk) per determinar el moment de l'actualització que s'efectuarà amb el flanc de baixada del senyal de rellotge del processador
* Un senyal de reset asíncron i actiu a nivell alt.
* I finalment, les dades d'entrada (Din) i les de sortida (Dout).

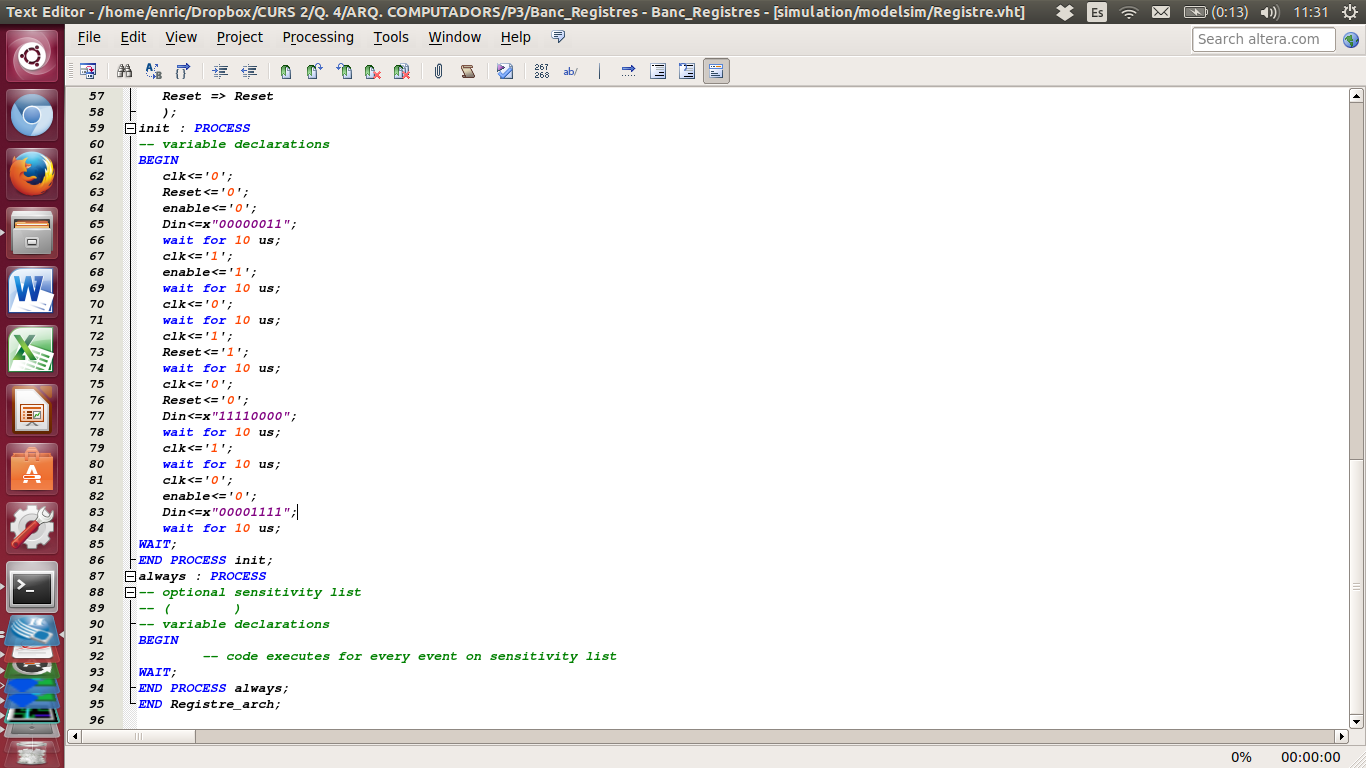
**REGISTRE**



Un cop tenim el disseny del registre de 32 bits acabat realitzem una prova del circuit aplicant uns valors determinats per poder comprovar el seu correcte funcionament.

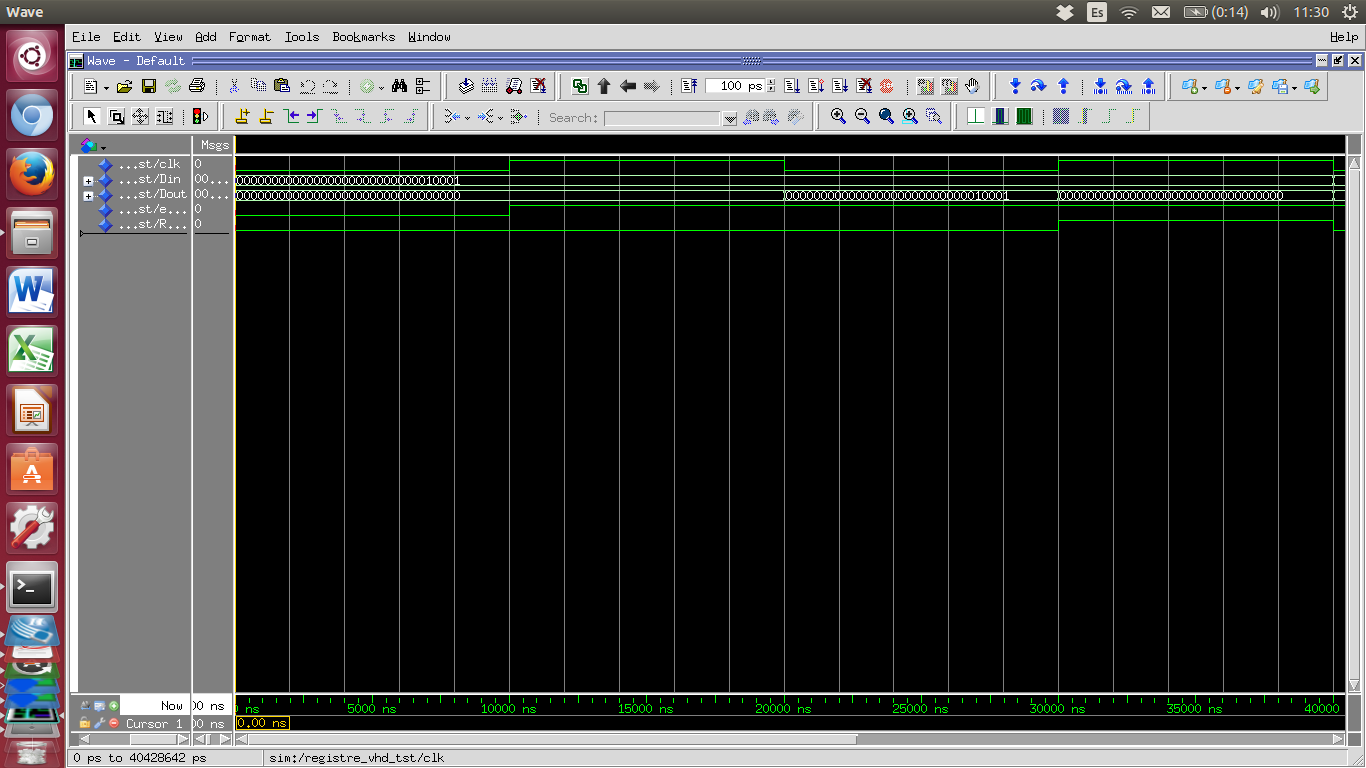
**TEST BENCH REGISTRE**





Un cop tenim el programa de prova creem una gràfica que ens mostra el resultat visualment.

**SIMULACIÓ GRÀFICA REGISTRE**



A continuació, per connectar els registres de 32 bits amb els busos d’entrada i sortida creem el mòdul RegSortida3Estats.

Per realitzar el codi de la entitat RegSortida3Estats tenim dues opcions per triar:

1. Connectar totes les sortides dels registres a un gran multiplexor perquè deixi arribar als busos del processador (connectats a Sor1 i Sor2) el contingut dels registres seleccionats per les entrades rf1 i rf2. En aquest cas serien dos multiplexors que tindrien 32 entrades de 32 bits i una sortida de 32 bits,  
controlats per rf1 i rf2 cadascun.

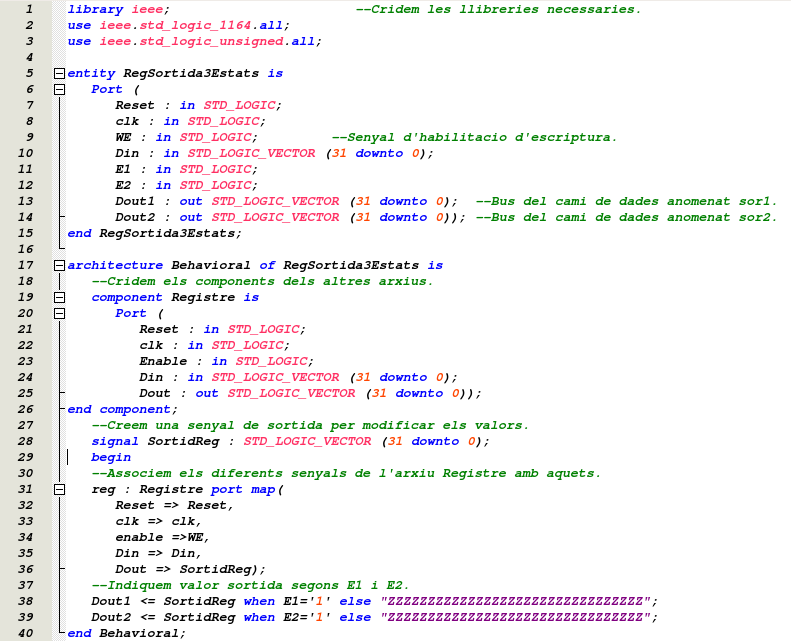
2. Dotar als registres que s’acaben de definir de dues sortides de tres estats fent que en funció de si es vol llegir un registre determinat per algun dels dos busos s'activi la sortida corresponent. D'aquesta manera es pot connectar directament les sortides dels 32 registres als busos de sortida Sor1 i Sor2.

Nosaltres utilitzarem la segona opció , ja que es la que trobem més senzilla.

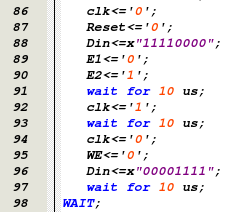
En aquet cas utilitzarem la entitat Registre explicada anteriorment per tal de controlar el clock i el reset. Així el RegSortida3Estats l'únic que farà serà crear un component de la entitat Registre ,si el senyals de habilitació de escriptura està WE està activat es realitzarà una operació de escriptura al registre.

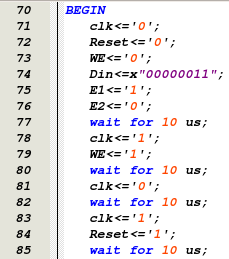
A continuació es comprovarà si els valors de lectura del registre E1 i E2 estan activats. Si un o els dos valors està activat el valor de Din passarà a ser el de Dout, en cas contrari les sortides es trobaran en alta impedància (Z) .

**RegSortida3Estats**

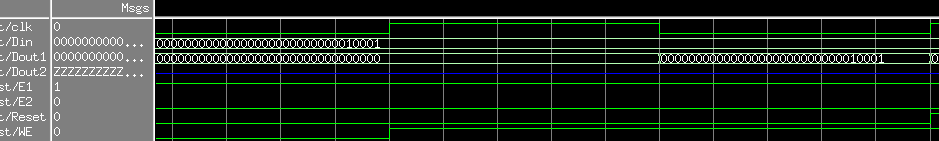
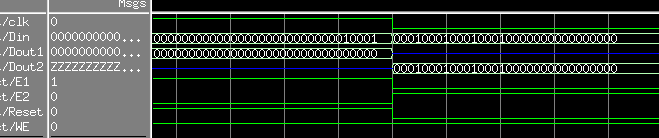


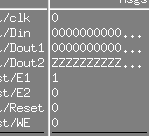
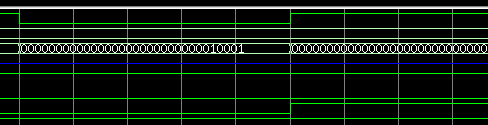
A continuació creem un test en el que anirem fent toggle al enable per veure com el valor canvia.

**TEST BENCH RegSortida3Estats**



**SIMULACIÓ GRÀFICA RegSortida3Estats**

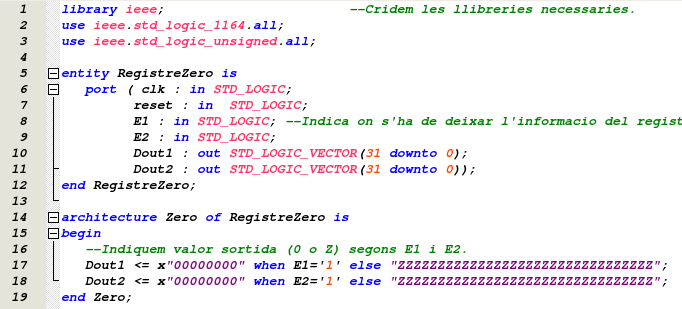




A continuació vam realitzar la entitat RegZero.

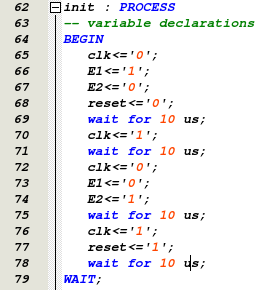
El registre 0 és un registre especial en el que el seu valor no és pot canviar I sempre val 0. Aquet valor és independent del valor del clock, encara que estigui declarat, ja que no és pot sobreescriure. Com podem veure sempre té el valor de 0 i en el cas de que no es seleccioni la activació el valor serà de alta impedancia (Z).

**REGISTRE ZERO**



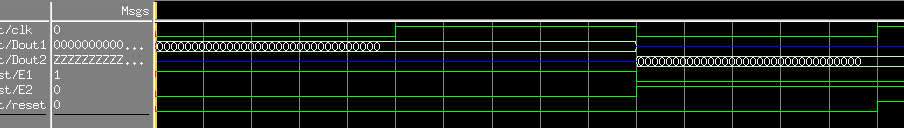
El registre 0 té un Test bench relativament senzill ja que no s’hi pot escriure i sempre té el mateix valor.

**TEST BENCH REGISTRE ZERO**



La gràfica resultant és la següent:

**SIMULACIÓ GRÀFICA REGISTRE ZERO**

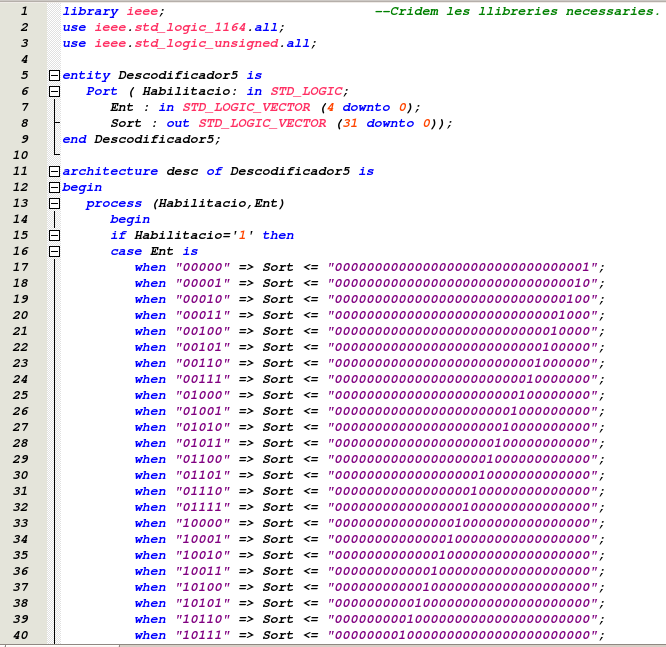
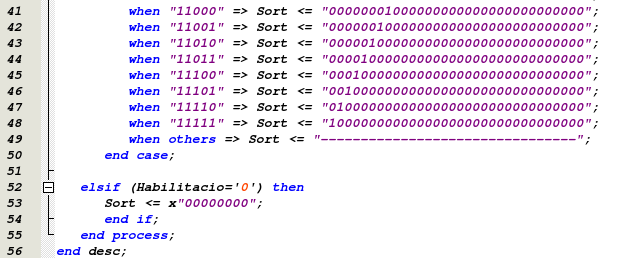


Com podem veure el valor resultant sempre és 0.

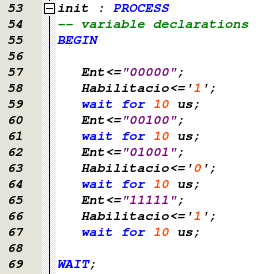
Abans de parlar del Banc de Registres tenim que fer un descodificador.

El descodificador és l'encarregat de descodificar les entrades de 5 bits cap a sortides de 32 bits per tal de triar un registre i els enables de les sortides que es troben a la entitat RegSortida3Estats. Això es fa desplaçant un bit cap a l'esquerra cada cop que el numero de entrada s'incrementa en 1 fins que el bit mencionat arriba a la posició 32.

**DESCODIFICADOR**

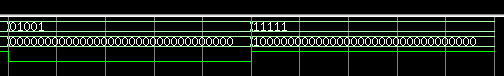


**TEST BENCH DESCODIFICADOR**



**SIMULACIÓ GRÀFICA DESCODIFICADOR**

****

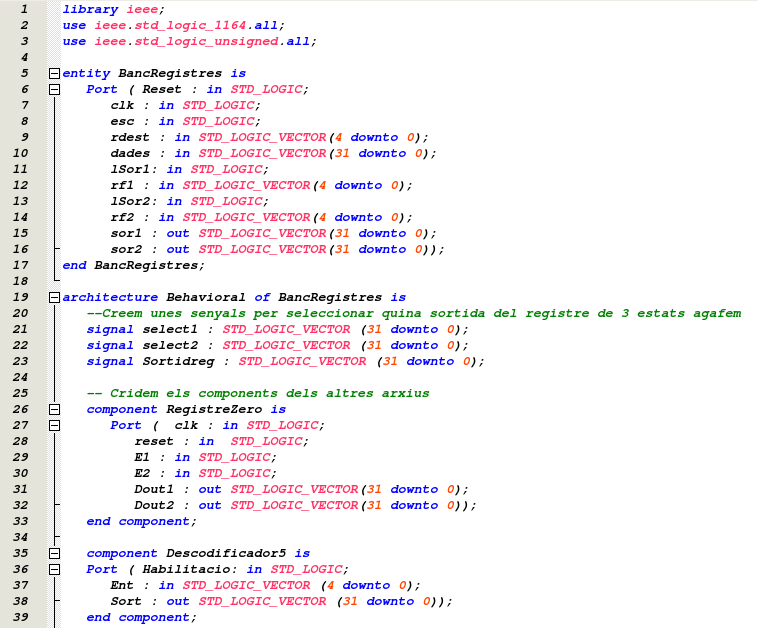


Finalment creem la entitat Banc de Registres.

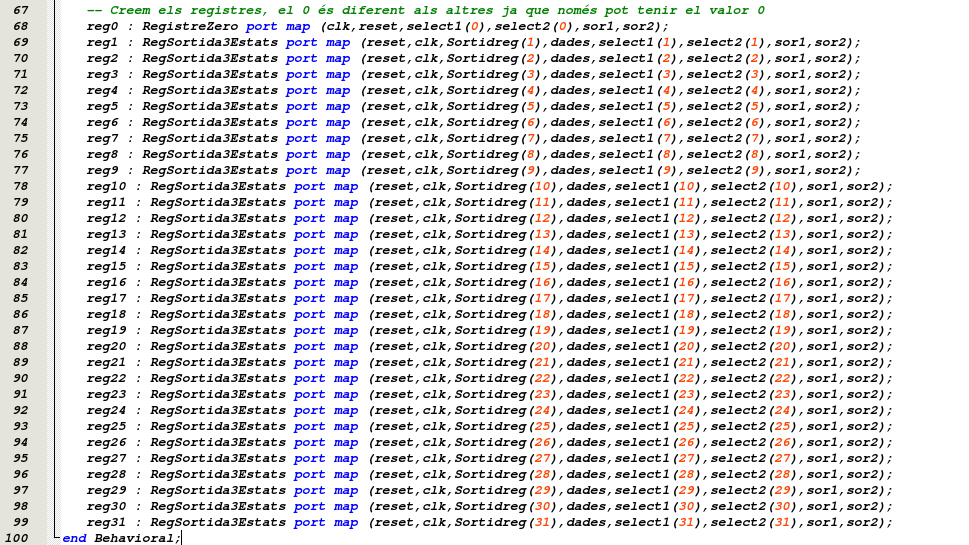
El banc de registres és el component que engloba totes les altres entitats. En si el banc de registre el que fa es descodificar les entrades rf1,rf2 i rdest, deixant les entrades lSor1 ,lSor2 i esc com a enables dels descodificadors anteriors respectivament.

Un cop descodificats els valors procedim a crear 32 registres utilitzant les entitats de RegistreZero i RegSortida3Estats i passant com a paràmetres els valors descodificats.

**BANC DE REGISTRES**

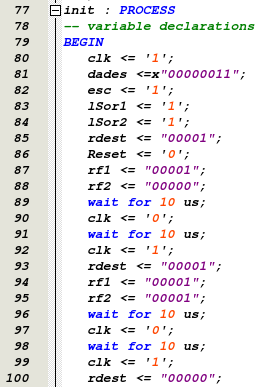
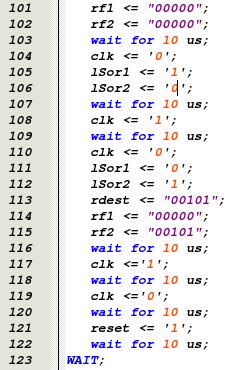
****

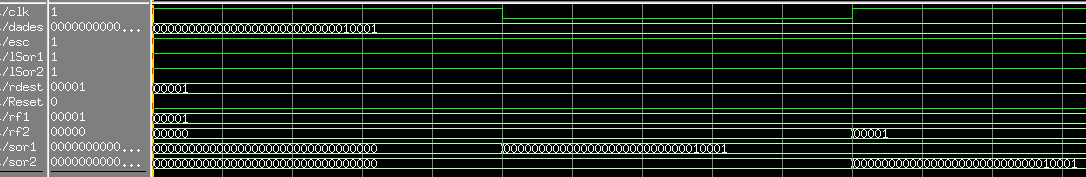


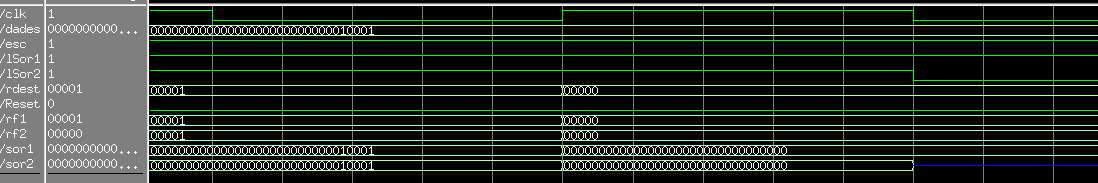
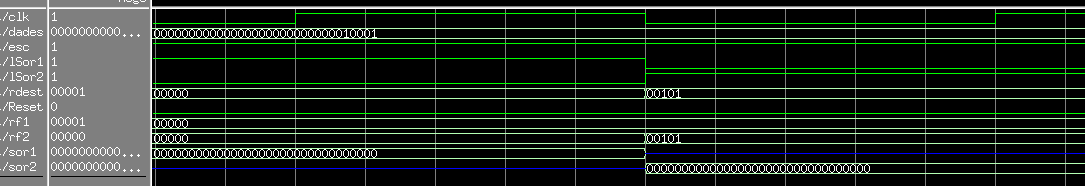


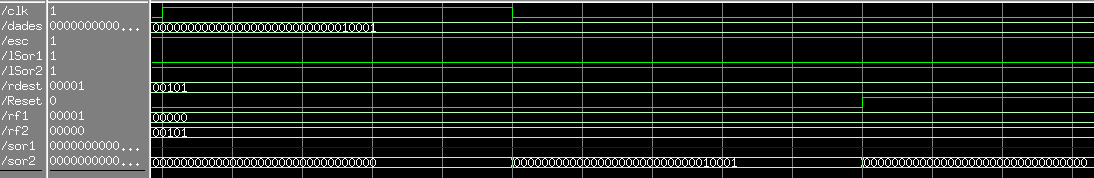
Un cop hem creat els 32 registres creem un test .

**TEST BENCH BANC DE REGISTRES**

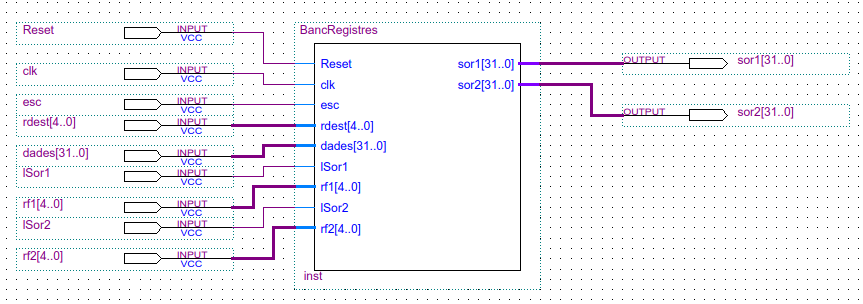


**GRAFICA BANC DE REGISTRES**

****

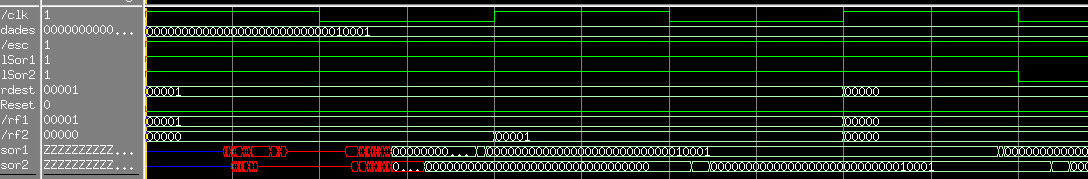
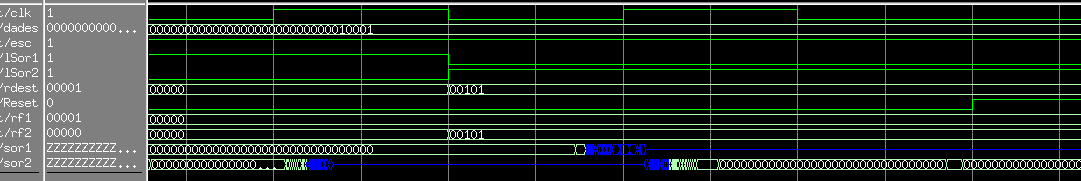
****

**SIMBOL BANC REGISTRE**



Finalment simulem en banc de registres a nivell de porta lògica, o sigui en cas real.

**SIMULACIÓ GRÀFICA REAL BANC DE REGISTRES**

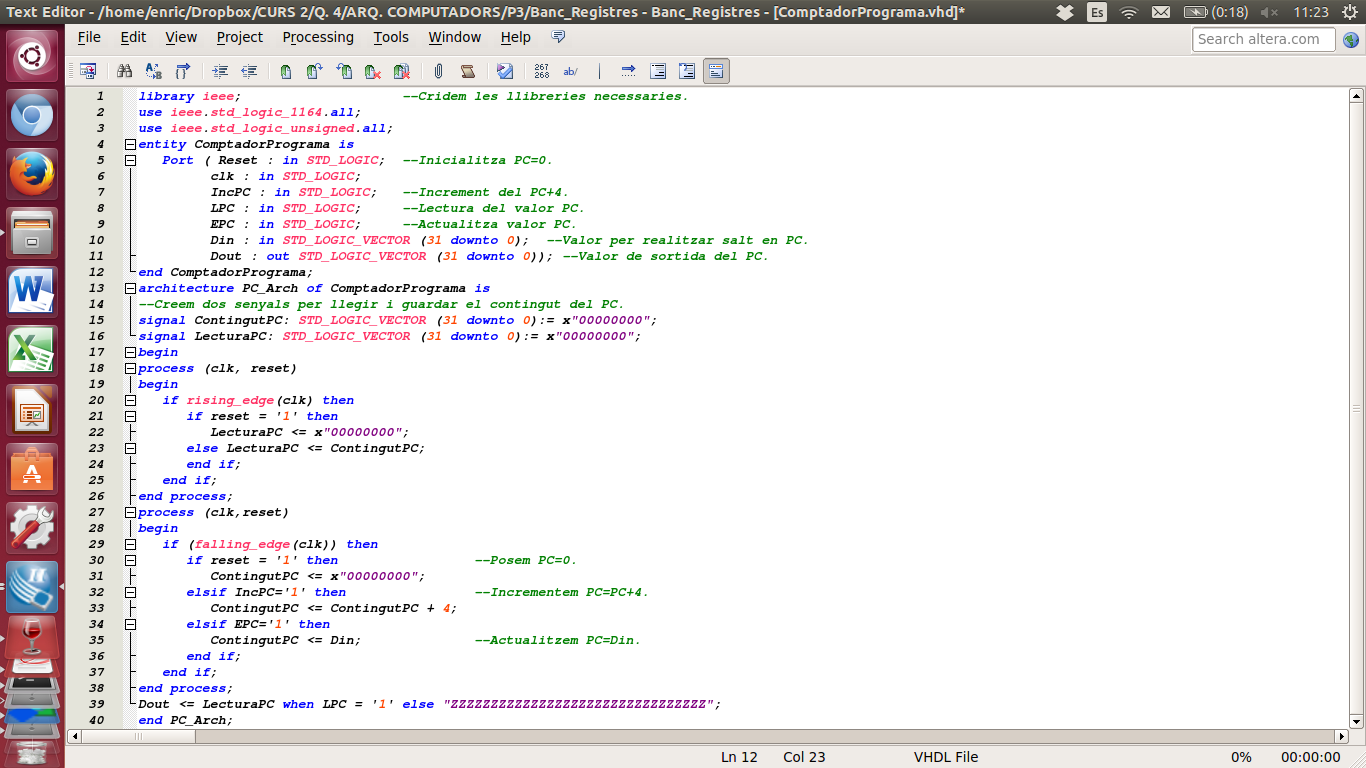


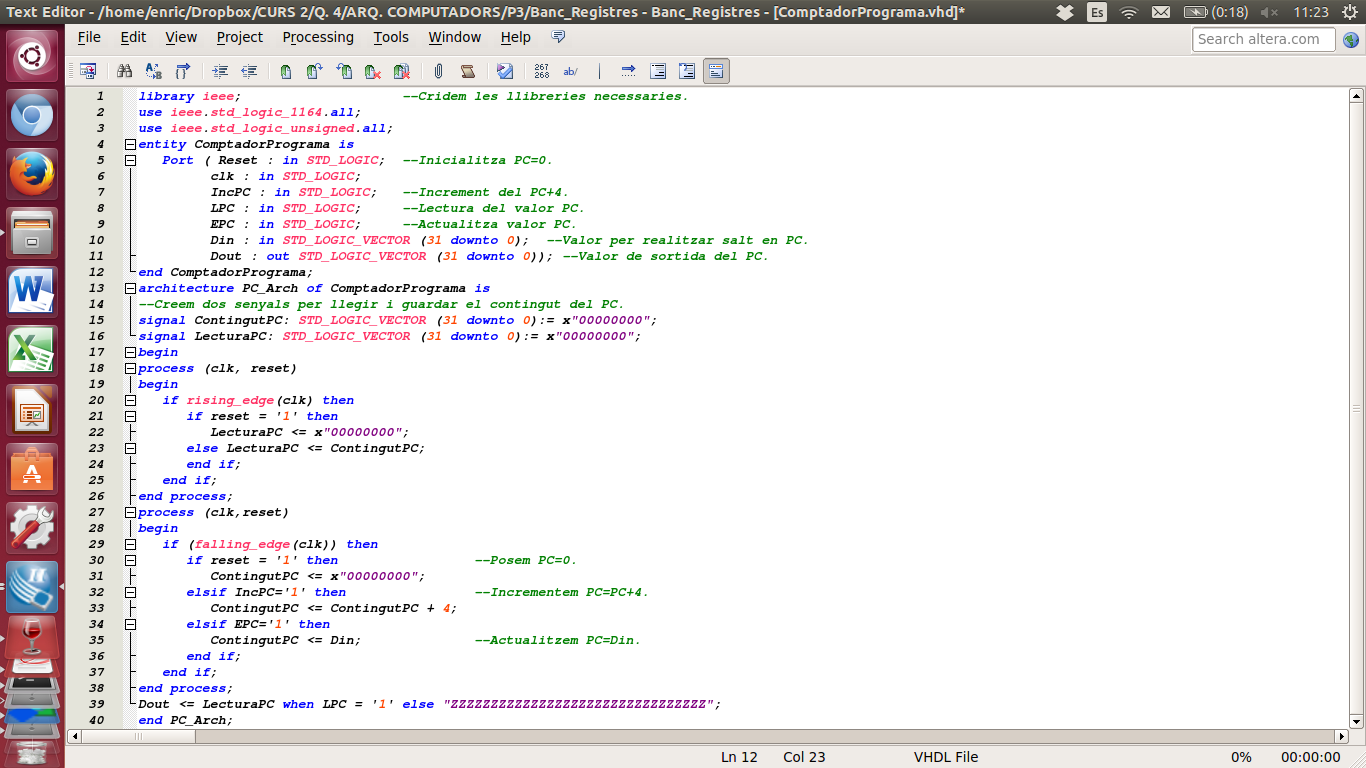
Podem veure com es produeixen retards i com les portes lògiques introdueixen alguns problemes. Aixó és degut al baix temps que hem ficat en el test bench entre clock i clock.

Finalment dissenyem el PC (Program Counter), que ens permetrà incrementar en quatre unitats el seu contingut a cada cicle d’instrucció i actualitzar el seu valor amb una dada nova en el cas de les instruccions de salt.

Per tant, en el flanc de pujada del rellotge es captura el valor actual del PC que guardarem en la senyal LecturaPC, que sera la que posarem al bus quan ho ordeni un senyal de lectura. En canvi, en el flanc de baixada es quan realitzarem les funcions d’actualització i modificació del PC utilitzant la senyal ContingutPC.

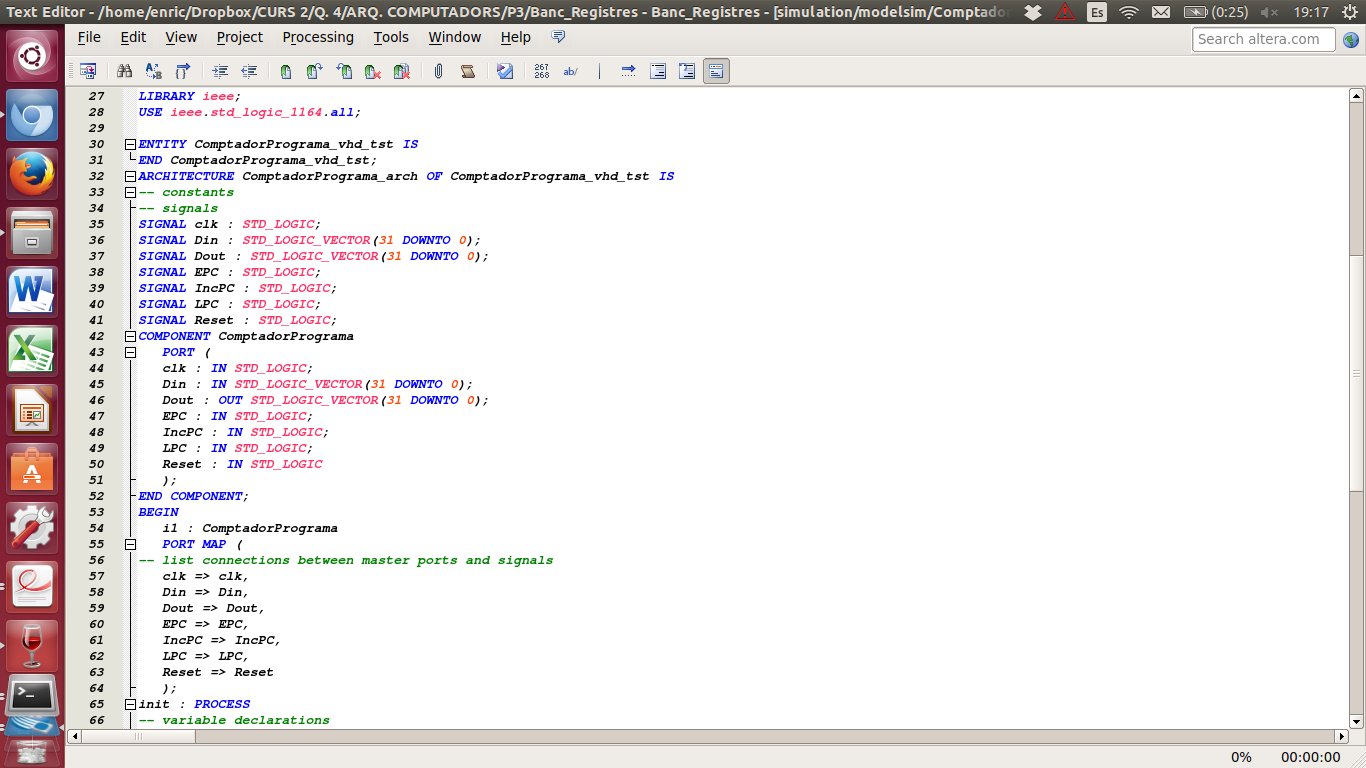
**PROGRAM COUNTER**

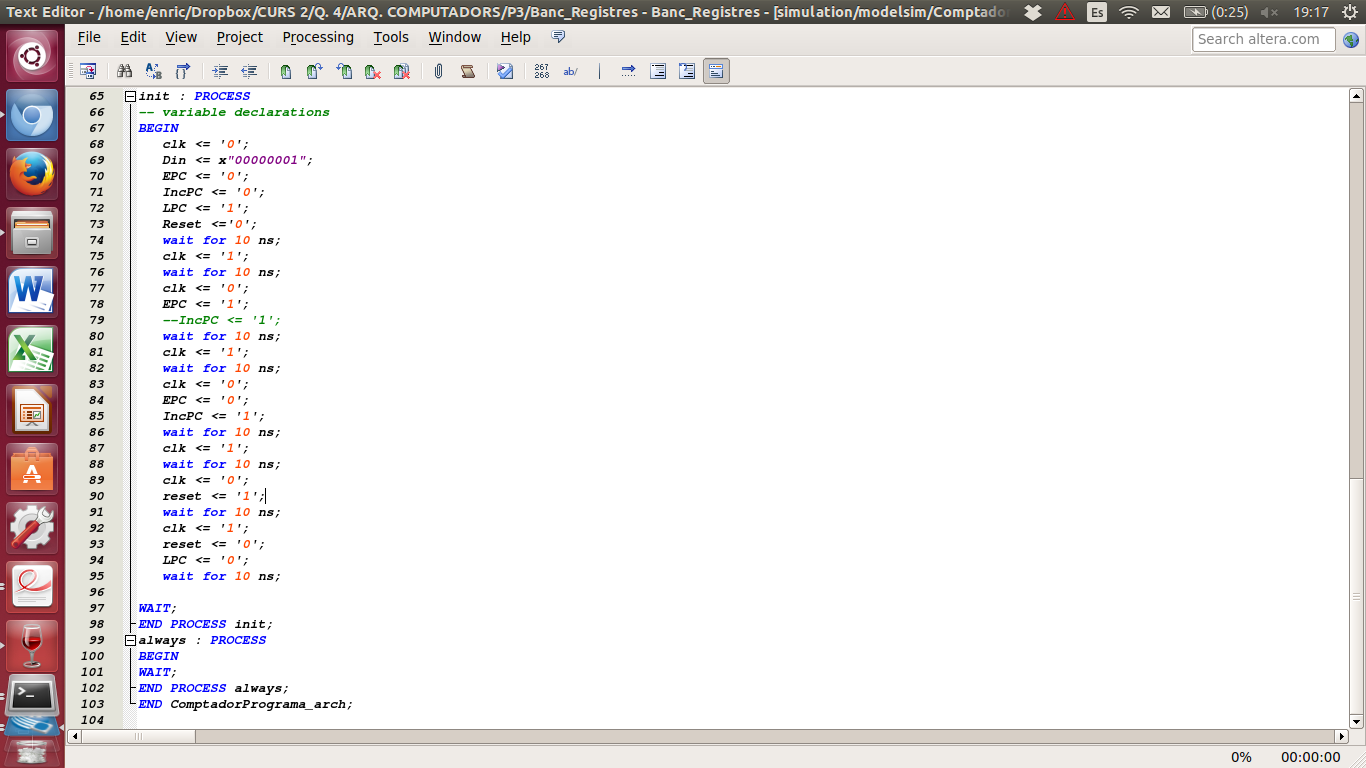




Un cop tenim el disseny del PC acabat realitzem una prova del circuit aplicant uns valors determinats per poder comprovar el seu correcte funcionament.

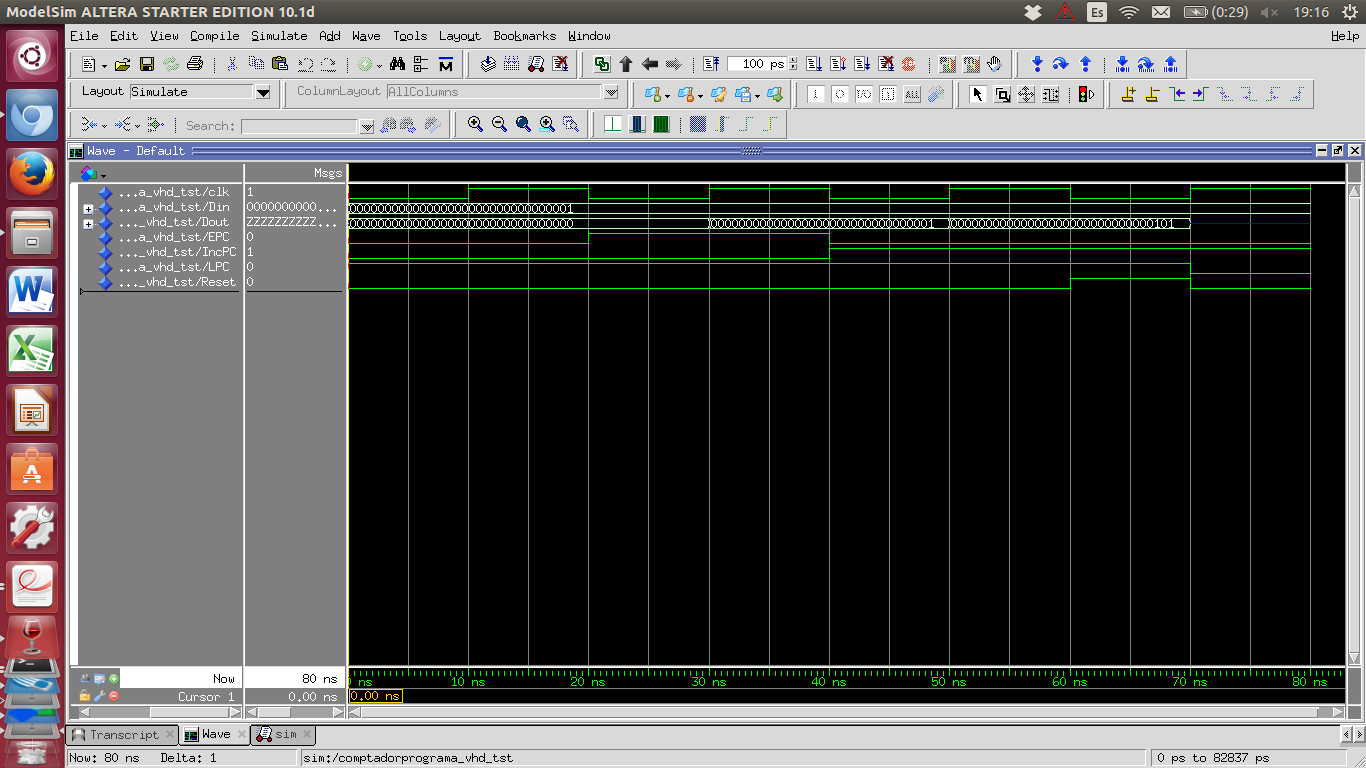
**TEST BENCH PC**



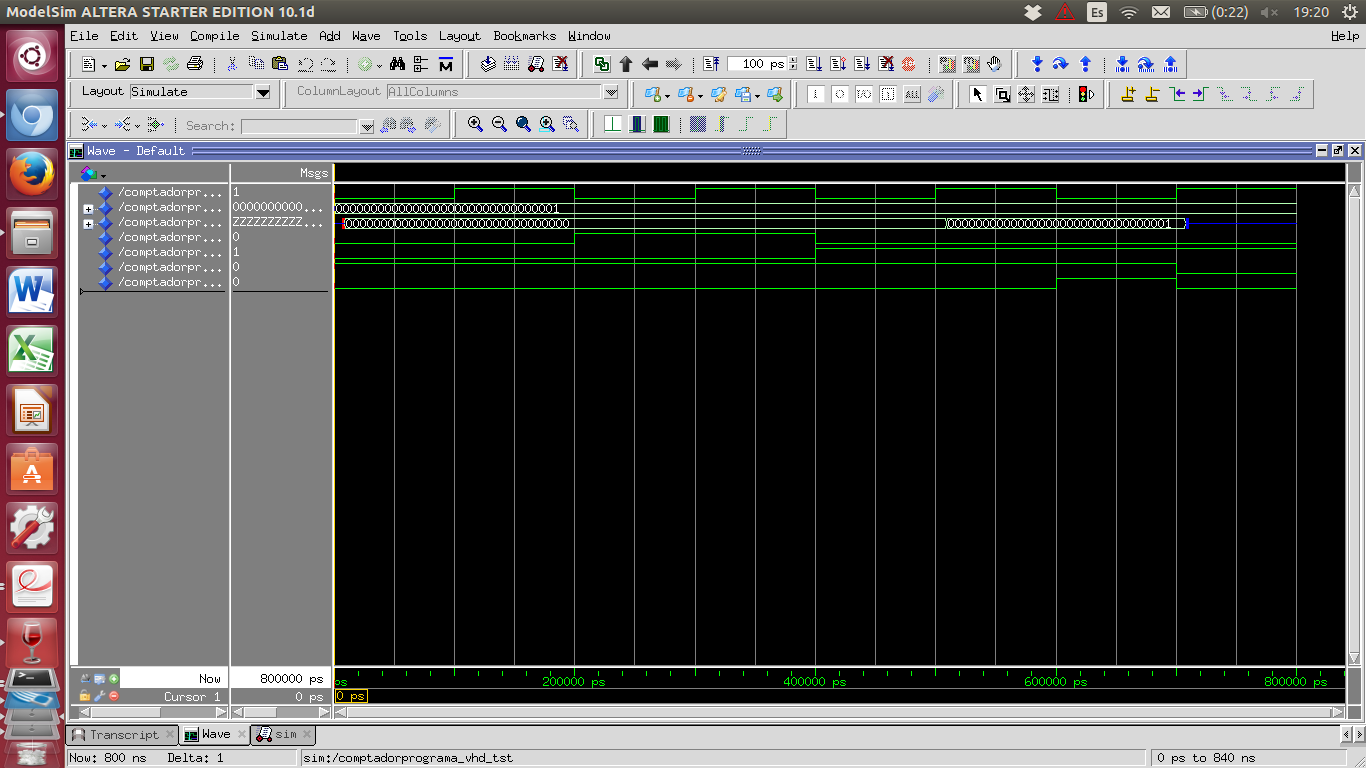


Un cop tenim el programa de prova creem les gràfiques que ens mostrant els resultats visualment observant la diferencia entre elles si es ten en compte els retards.

**SIMULACIÓ GRÀFICA IDEAL PC**



**SIMULACIÓ GRÀFICA REAL PC**



Per ultim realitzarem el símbol del PC utilitzant el disseny anterior.

**SIMBOL PC**

