

**PRÀCTICA 2:**

**DISSENY ALU**

**Sergi Carol Bosch i Enric Lenard Uró**

Grau en Enginyeria de Sistemés TIC

**Arquitectura de Computadors**

Curs 2013-14, Grup 10, G12 de pràctiques

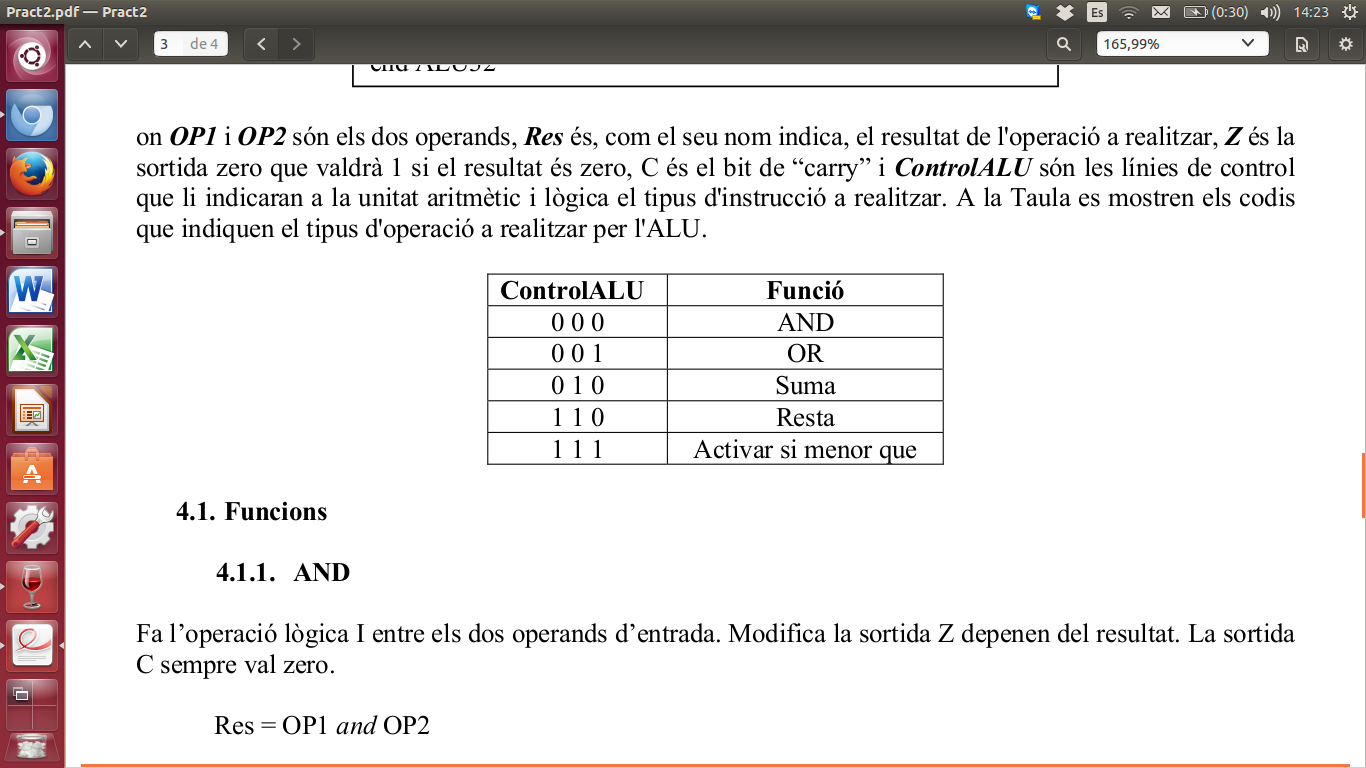
Realització de la pràctica: 24/03/2014

Lliurament del treball: 07/04/2014

Realització pràctica

• L’objectiu d’aquesta practica és repassar els conceptes relacionats amb l’ALU vist a classe, per aquest motiu dissenyarem una ALU de 32 bits per nombres sencers.

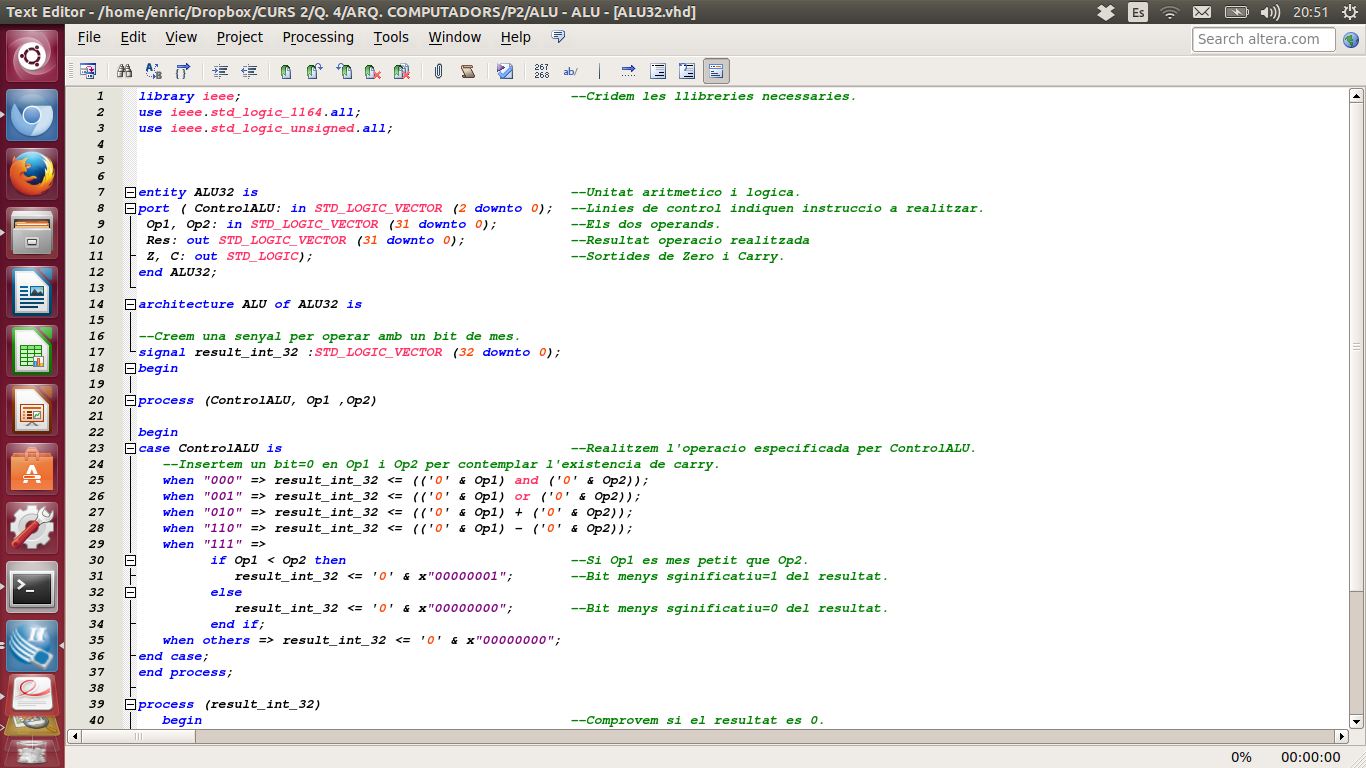
Aquest disseny del processador monocíclic permetrà realitzar sumés i restes de nombres sencers de 32 bits, i les operacions lògiques AND i OR sobre 32 bits. També podrà comparar els dos operands i finalment mostrarà l’existència de Carry i Zero en el resultat.

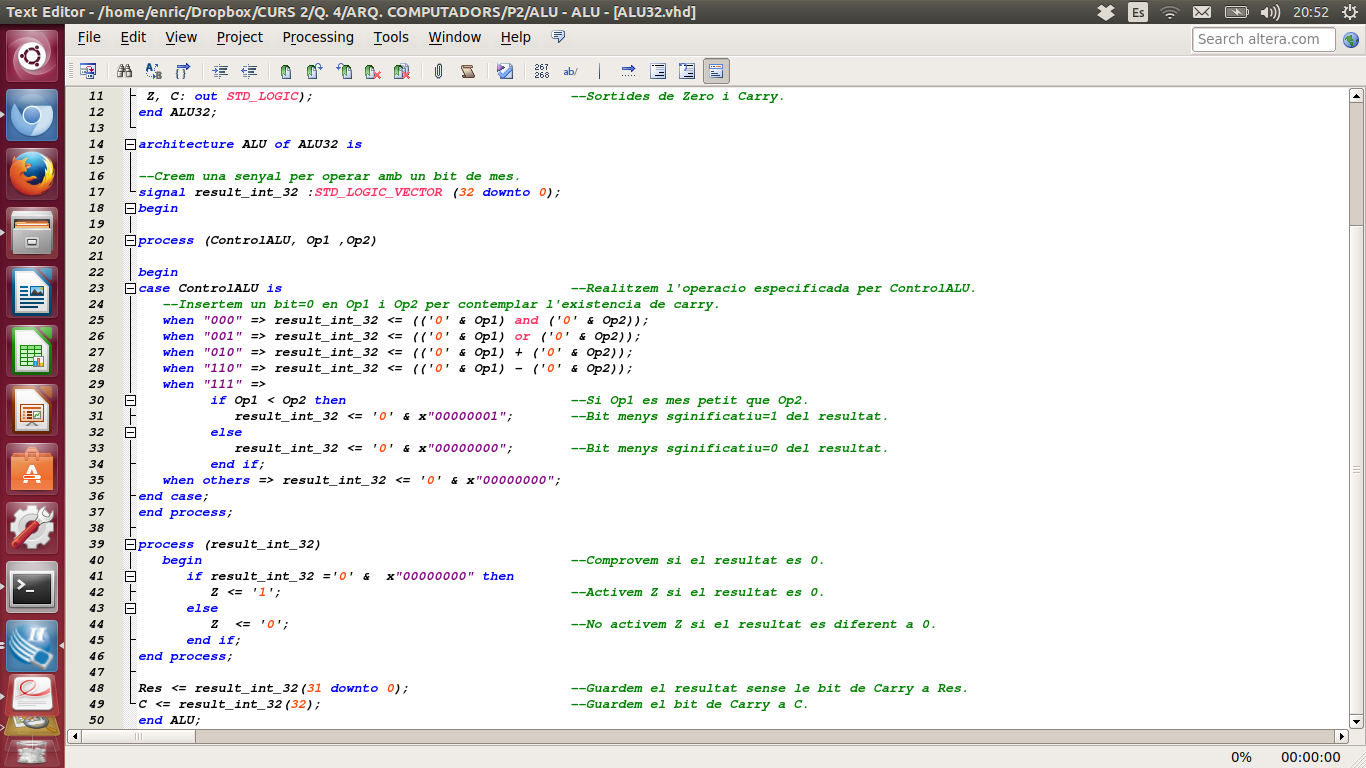


Per aconseguir que l’ALU realitzi les operacions anteriors necessitarem crear una sèrie de variables que ens permetran el correcte funcionament de la unitat aritmètica i lògica.

Aquesta ALU disposarà de ***Op1*** i ***Op2*** que són els operadors, ***Res*** és el resultat de l'operació a realitzar, ***Z*** és la sortida zero que valdrà 1 si el resultat és zero, ***C*** és el bit de “carry” i ***ControlALU*** són les línies de control que li indicaran a la unitat aritmètic i lògica el tipus d'instrucció a realitzar.

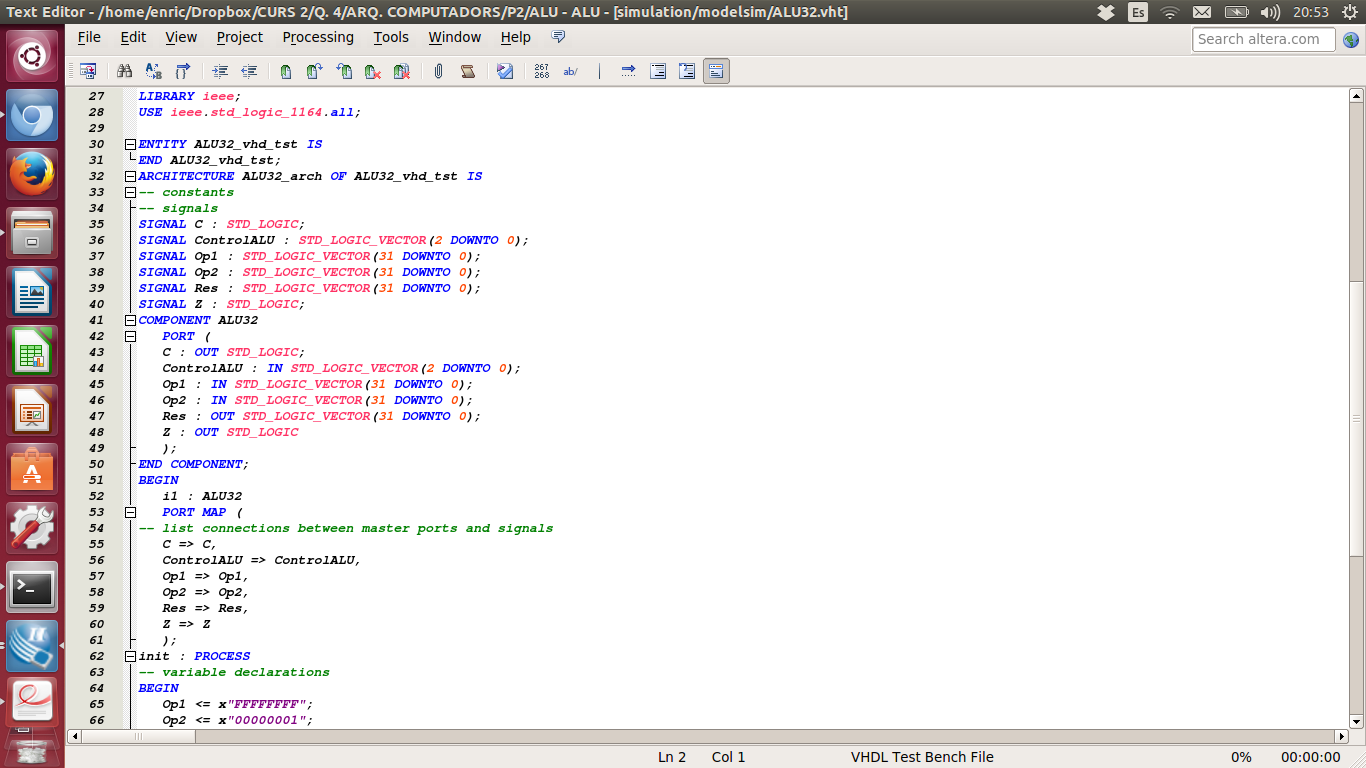
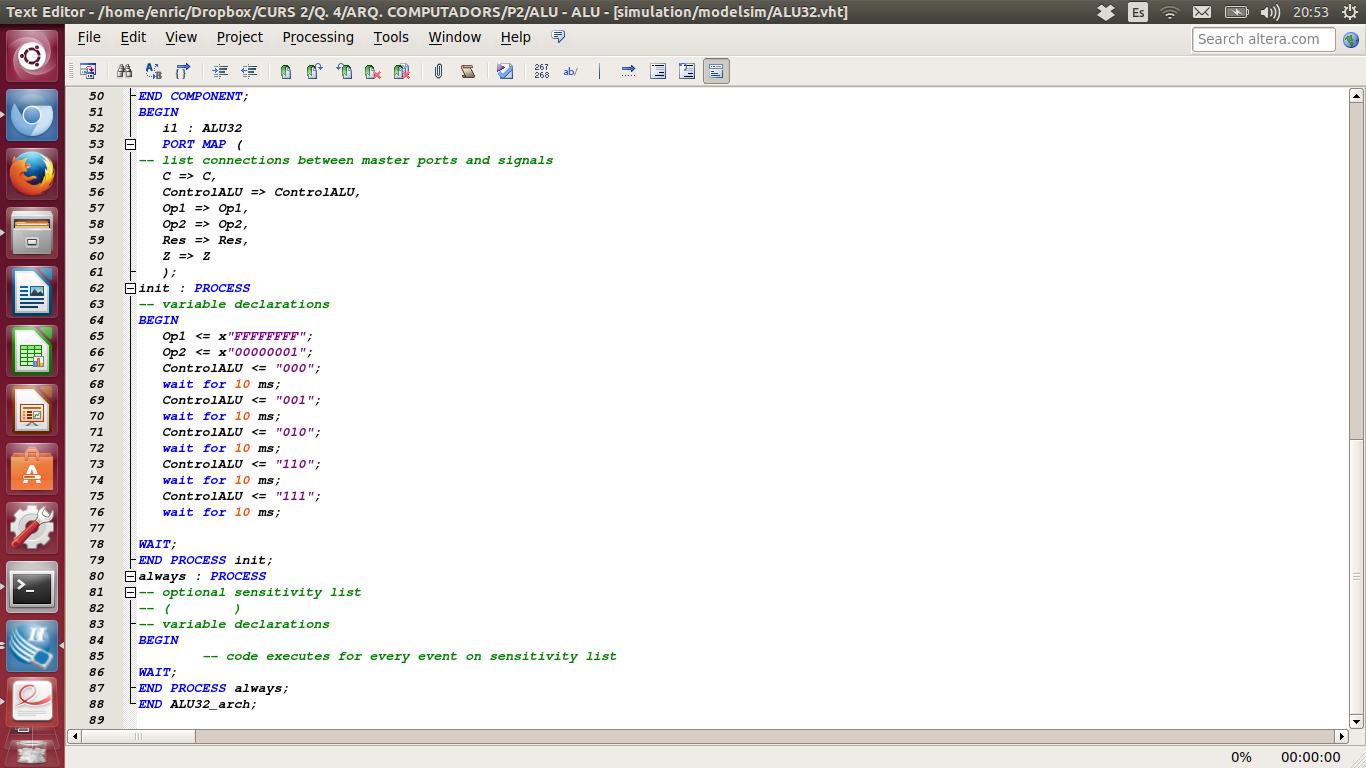
**DISSENY ALU**



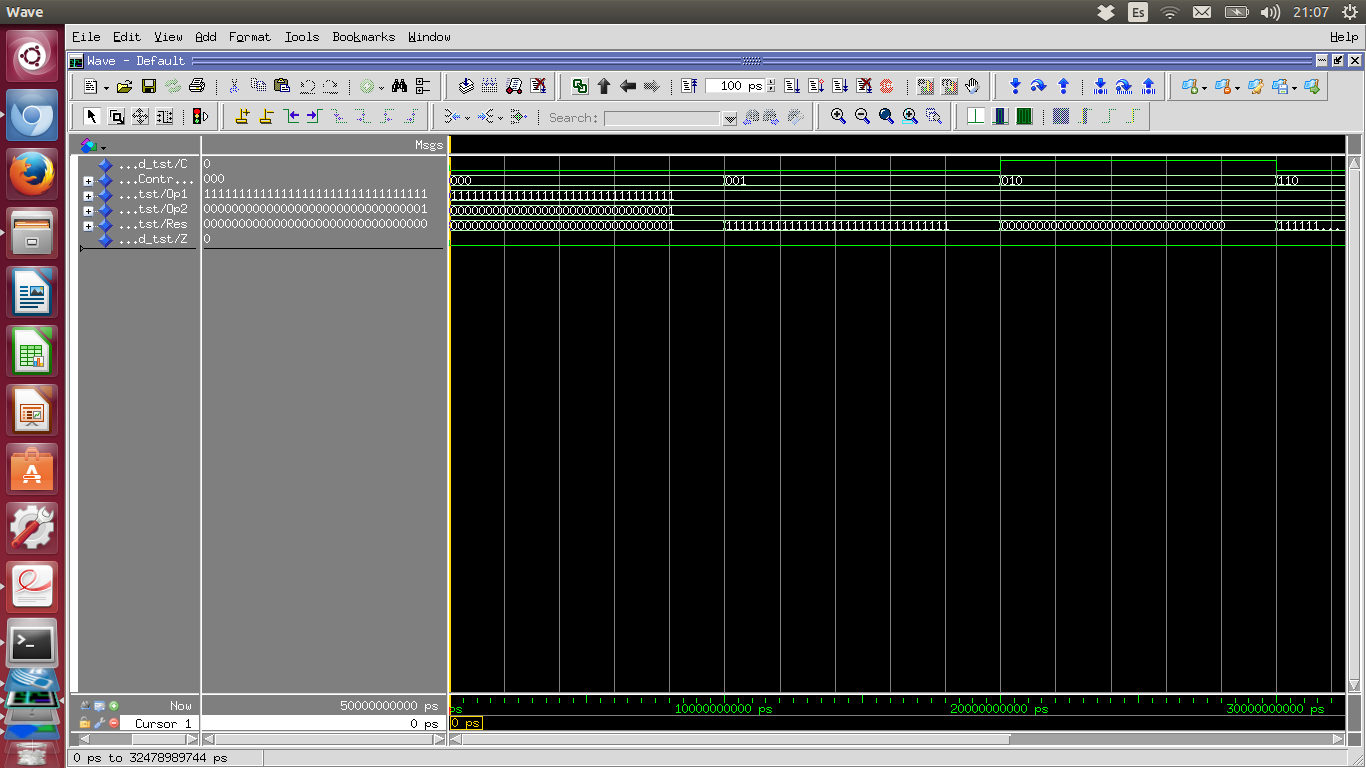


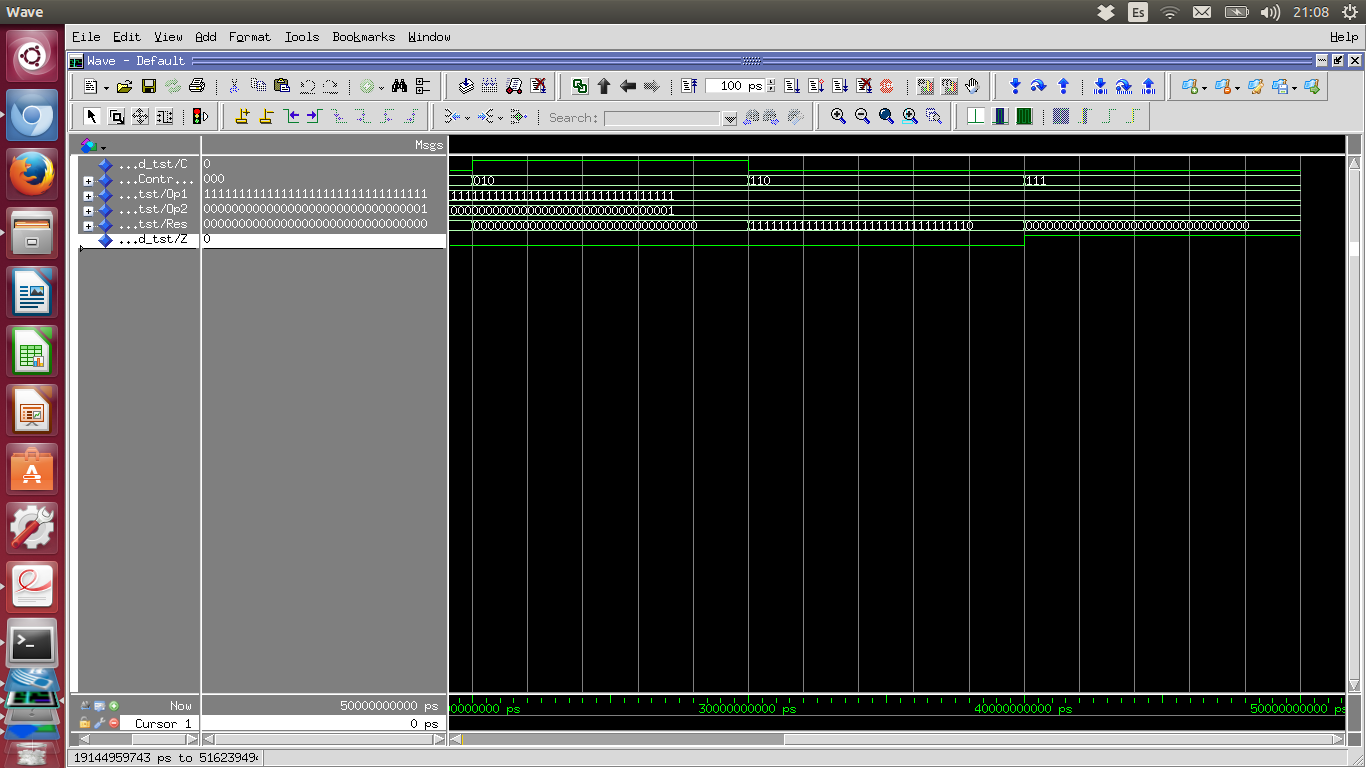
Un cop tenim el disseny de l’ALU acabat realitzem una prova del circuit aplicant uns valors determinats per poder comprovar el seu correcte funcionament.

**TEST BENCH**

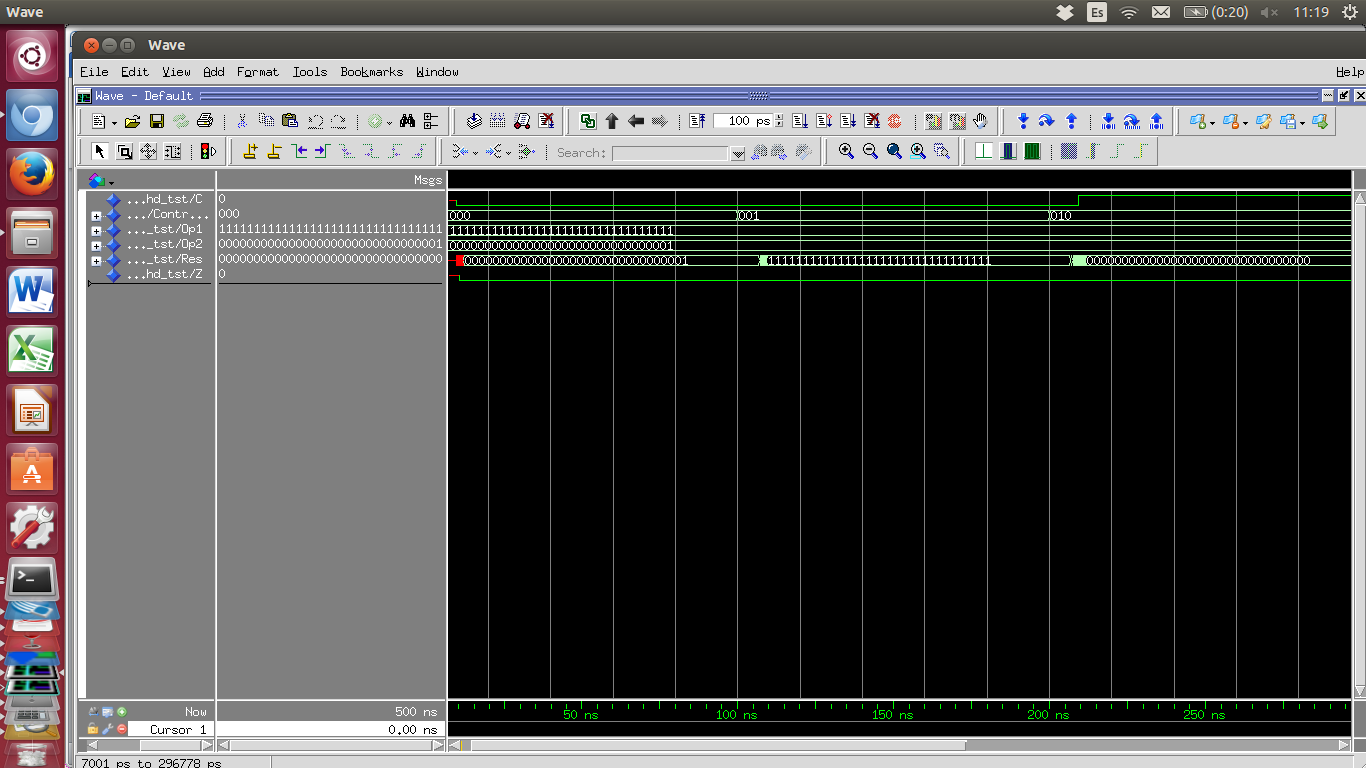
  
Un cop tenim el programa de prova creem una gràfica que ens mostra el resultat visualment.

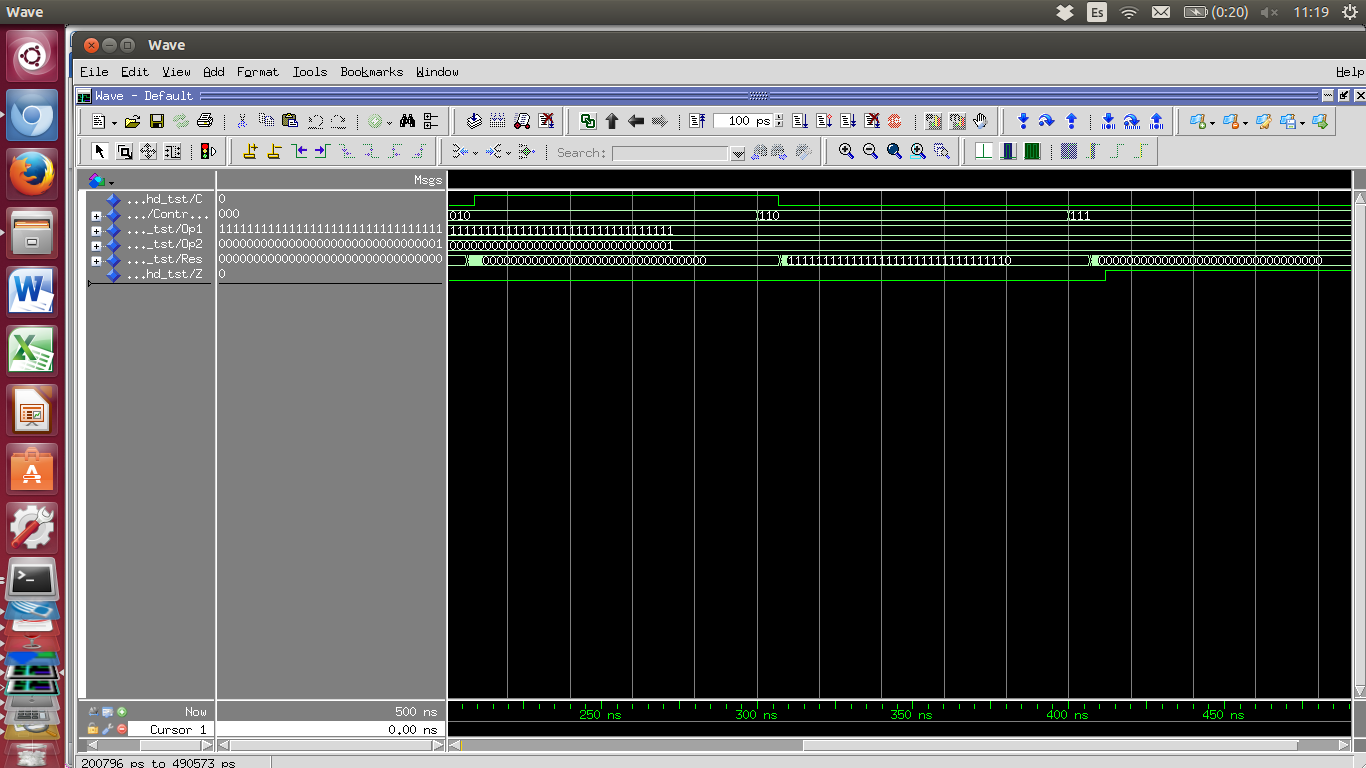
**SIMULACIÓ GRÀFICA IDEAL**





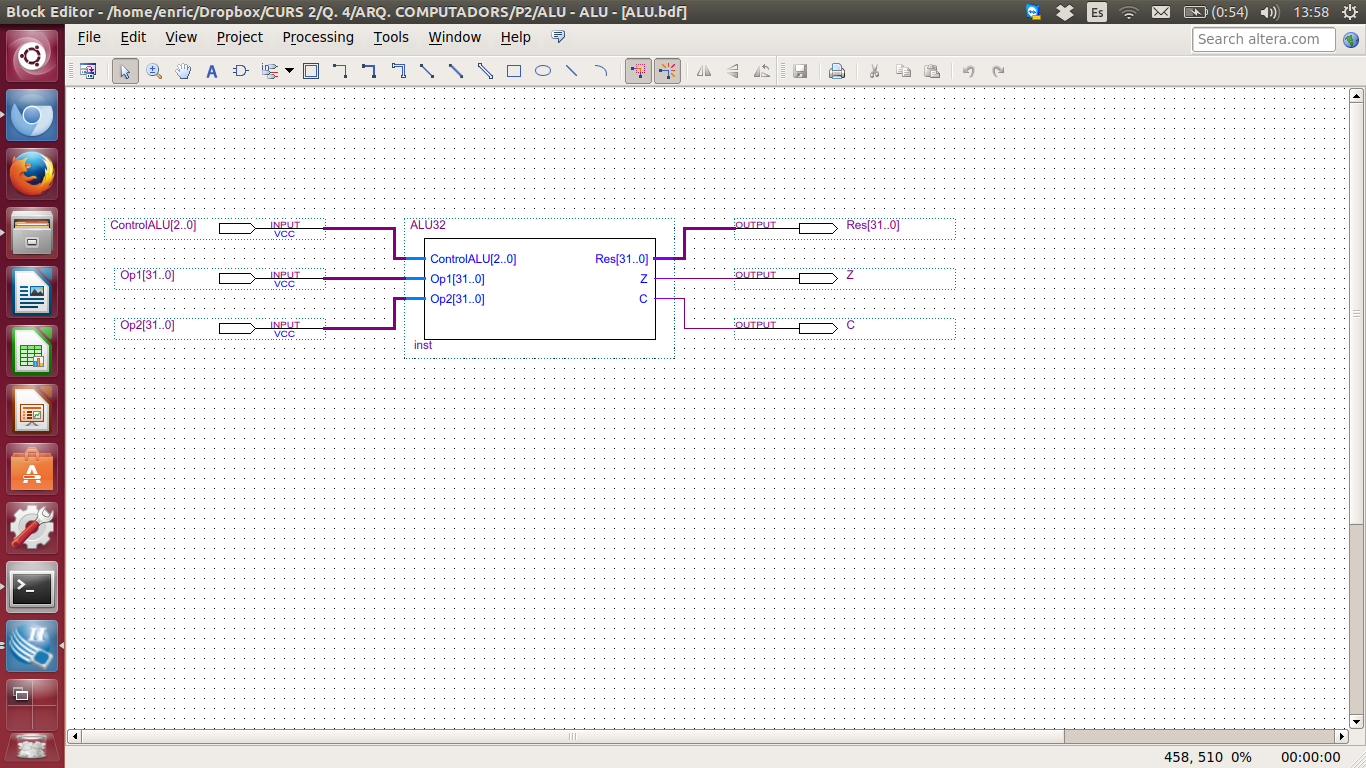
**SIMULACIÓ GRÀFICA REAL**





• La segona part de la pràctica consisteix en crear un símbol de l’ALU de 32 bits utilitzant el disseny anterior.

**SIMBOL ALU**



Preguntes

• Heu pogut fer la simulació dels dos dissenys de la mateixa forma? Quines heu utilitzat i per quin motiu?

Per fer les simulacions hem hagut d’utilitzar dos tipus de simulació. En el primer disseny hem utilitzat ‘RTL Simulation’, en canvi per el segon disseny hem fet servir la ‘Gate Level Simulation’, ja que s’utilitza quan vols comprovar el funcionament d’un símbol creat anteriorment, en cas contrari peta.

• Hi ha un temps mínim per poder fer modificacions als senyals d’entrada?

El temps mínim necessari perquè l’ALU obtingui correctament els valors de les entrades és de 10ns. Si el temps de modificació de senyal és més petit es produeixen errors en la lectura d’aquestes senyals.

• Quin és el temps de resposta dels senyals de sortida?

El temps de resposta dels senyals de sortida és el retard que es produeix al fer les operacions, a comprovar si hi ha Carry i comprovar si el resultat dona Zero.

Generalment el retard que obtenim al modificar la senyal ‘ControlALU’ fins a veure el senyal de sortida es de 9’343 ns. Però, en el cas de la suma, el retard augmenta a 11’454 ns, ja que la suma es l’operació mes complexa de totes.

Per altre banda, la senyal de ‘Zero’ també es veu retardada uns 2’551 ns, degut a que per realitzar la seva comprovació cal recorre els 32 bits a diferencia de la senyal ‘Carry’ que nomes comprova un sol bit.