

1. Introducció

El projecte que s'exposa a continuació té com a objectiu la visualització del camí de dades, així com el contingut dels registres i de la memòria de dades, que segueix una instrucció, codificada en binari i introduïda per l'usuari, en l'arquitectura MIPS de tipus unicycle.

En definitiva l'usuari introduirà des de teclat la instrucció a executar i el programa s'encarregarà de descompondre-la, segons el tipus de instrucció que es tracte, en les parts necessàries per executar-la i després de realitzada la descomposició, activar les línies, tant de control com les línies que uneixen parts de l'arquitectura, que siguin necessàries per a dur a terme la instrucció. Així com també mostrar el contingut de les línies activades en cada moment.

Cal fer constar que existeixen dos tipus d'arquitectura MIPS:

- Multicle, com s'imagina la instrucció es realitza en diversos cicles. El comportament de la màquina ve regit pel rellotge, açò suposa que cada pas en que s'ha de realitzar la instrucció ve determinat per la freqüència del rellotge abans esmentat. Els passos, com veurem més endavant, on els explicarem més detingudament, depenen del tipus d'instrucció que introduïm.
- Unicycle, és el model emprat per a l'execució del present projecte, es tracta, com el mateix nom ho indica, de realitzar la búsqueda, descodificació i posterior execució de la instrucció (depenent del tipus d'instrucció, és clar) en un mateix instant de temps. Açò indica que no es parteix en cicles la instrucció, sinó que la instrucció es descompon i directament passa per les parts necessàries per a la seua execució, com s'imagina el nombre de parts per fer possible aquesta feina és major que el de la màquina multicicle. Un exemple pel que ocorre açò és que si la instrucció a realitzar és una suma, la ALU (Unitat Aritmètico-Lògica) hauria de realitzar tant la suma dels operands com l'operació d'incrementar el PC (Comptador de Programa), per tant per a realitzar ambdues operacions en tan sols un cicle ens faran falta dues ALU. Mentre que per a l'arquitectura multicicle, s'empra només una ALU. Encara que la realització de l'arquitectura de tipus multicicle és un tant més difícil de desenvolupar.

Encara que com he dit la opció escollida ha estat la màquina unicycle, per a una millor visualització del comportament real de l'arquitectura, la instrucció es realitza per passos, encara que açò sembla estrany, és més semblant més apropiat parlar d'un model multicicle pel fet que la instrucció es descompon en passos, el

principal motiu de la tria d'aquesta manera de fer-ho, ha sigut que el programa del present projecte, va a ser utilitzat acadèmicament, per tant la visualització pas a pas de les línies activades, dona una millor idea del funcionament de l'arquitectura. Encara que, com es veurà més tard al capítol següent, la diferència entre ambdues arquitectures és fàcil d'observar gràficament.

S'ha parlat fins ara de diversos tipus d'instrucció, aquests poden ser tres diferents, com ara:

- Tipus R o registre, la instrucció a executar consta de tres registres (rs,rt i rd). L'execució de la instrucció ve determinada per un camp, (una part de la instrucció), anomenat camp de funció, que conte la informació de l'operació a realitzar entre dos registres (rs i rt) i la posterior guarda del resultat al registre destí (rd).
- Tipus I o immediat, ara l'operació es realitza entre dos registres i un nombre introduït com a immediat.
- Tipus J o de bot, (de l'anglès *jump*) com indica, el cos de la instrucció conté informació de l'adreça on s'ha de botar, que no és més el .

Feta aquesta valoració de la màquina que es tractarà d'ací a endavant, seguidament es presenta una breu explicació dels capítols que componen aquest projecte:

Capítol 1: Introducció.

Es descriu l'objectiu del projecte així com un breu comentari de l'arquitectura MIPS.

Capítol 2: Arquitectura MIPS.

Valoració completa de l'arquitectura, explicant cada component que la forma i el seu funcionament i finalitat.

Capítol 3: Introducció al Borland C++ Builder.

Una breu explicació d'aquest llenguatge de programació.

Capítol 4: Disseny principal del programa.

Parts més interessants del codi del programa.

Capítol 5: Resultats.

Execució del programa amb una instrucció codificada i posterior visualització dels passos depenent de la instrucció introduïda.

Capítol 6: Conclusions.

Un resum de la utilitat del simulador.

Capítol 7: Treballs futurs.

Enumeració de les millores que es poden aportar per enfortir aquest simulador.