

MIPS IT Simulador – Instalación

Laboratorio de arquitectura de Computadoras

ITESM Campus Monterrey

Profesor

Diego Fernando Valencia Martínez

Miguel Morales de la Vega

Javier Mondragón Martín del Campo

A00821541

A01365137

Introducción

Durante el desarrollo de la práctica observaremos las diferentes etapas por las cuales pasa un programa en ensamblador hasta llegar a su ejecución por el equipo de computo. La herramienta principal a utilizar será MIPS Simulator. La cual nos permite tener una visión abstracta de los componentes principales de una arquitectura MIPS, estos componentes son CPU, Memoria, Caché, Consola y I/O.

Al momento de ejecutar un código de ensamblador podremos observar como cada instrucción es procesada por la el CPU, convertida a lenguaje maquina y guardada en los registros internos de CPU para su ejecución. La herramienta nos habilita la configuración sin necesidad de código en ensamblador de los registros internos tanto del CPU como los de Memoria para realizar las simulación y operaciones deseadas,

Procedimiento

Durante esta sección vemos como es usado la herramienta de simulación para el código mostrado en la figura 2.1. El código realiza una carga de valores a los registros del CPU y después realiza la operación de suma de estos valores.

```
#Lab Exercise 1
.set noreorder      #Avoid reordering instructions
.text               #Start generating instructions
.global start       #The label should be globally known
.ent start          #The label marks an entry point
start: li $8, 0x1    #Load the value 1
li $9, 0x2          #Load the value 2
add $10, $8, $9     #sum the values
.end start          #Marks the end of program
```

Figura 2.1
Código ejemplo de ensamblador

Resultados

Los resultados de la simulación son la carga de los valores 1 y 2 a los registros 8 y 9 del CPU, en la figura 3.1 podemos observar cómo estos registros tienen el valor correspondiente y también como el registro PC (Program Counter) obtiene valores correspondiente a la dirección de memoria donde reside la instrucción (Figura 3.1).

Registers			
r0/zero=00000000	r1/at =00000000	r2/v0 =00000000	r3/v1 =00000000
r4/a0 =00000000	r5/a1 =00000000	r6/a2 =00000000	r7/a3 =00000000
r8/t0 =00000001	r9/t1 =00000002	r10/t2 =00000003	r11/t3 =00000000
r12/t4 =00000000	r13/t5 =00000000	r14/t6 =00000000	r15/t7 =00000000
r16/s0 =00000000	r17/s1 =00000000	r18/s2 =00000000	r19/s3 =00000000
r20/s4 =00000000	r21/s5 =00000000	r22/s6 =00000000	r23/s7 =00000000
r24/t8 =00000000	r25/t9 =00000000	r26/k0 =00000000	r27/k1 =00000000
r28/gp =00000000	r29/sp =800bc000	r30/fp =00000000	r31/ra =bfc00088
pc =8002000c	mdhi =00000000	mdlo =00000000	conf =00000000
bad va =00000000	status =00400000	cause =00000000	epc =00000000

Figura 3.1
Ventana de registros del CPU

En la figura 3.2 observamos la memoria interna de la arquitectura MIPS en esta se codifica cada una de las instrucciones que el procesador va a realizar, para el ejemplo de la carga y suma de registros se observan 3 instrucciones cada una con su correspondiente codificación.

Memory				
	Address	Content	Label	
	8001FFD4	00 00 00 00	NOP	
	8001FFD8	00 00 00 00	NOP	
	8001FFDC	00 00 00 00	NOP	
	8001FFE0	00 00 00 00	NOP	
	8001FFE4	00 00 00 00	NOP	
	8001FFE8	00 00 00 00	NOP	
	8001FFEC	00 00 00 00	NOP	
	8001FFF0	00 00 00 00	NOP	
	8001FFF4	00 00 00 00	NOP	
	8001FFF8	00 00 00 00	NOP	
	8001FFFC	00 00 00 00	NOP	
	80020000	24 08 00 01	ADDIU \$08, \$00, 0x1	
	80020004	24 09 00 02	ADDIU \$09, \$00, 0x2	
	80020008	01 09 50 20	ADD \$10, \$08, \$09	
	8002000C	00 00 00 00	NOP	
	80020010	00 00 00 00	NOP	
	80020014	00 00 00 00	NOP	
	80020018	00 00 00 00	NOP	
	8002001C	00 00 00 00	NOP	
	80020020	00 00 00 00	NOP	
	80020024	00 00 00 00	NOP	
	80020028	00 00 00 00	NOP	
	8002002C	00 00 00 00	NOP	
	80020030	00 00 00 00	NOP	
	80020034	00 00 00 00	NOP	
	80020038	00 00 00 00	NOP	
	8002003C	00 00 00 00	NOP	
NOP				
Address mode:	Virtual	View mode:	Assembler	

Figura 3.2
Instrucciones del programa en memoria

Referencias

- https://s3-eu-west-1.amazonaws.com/downloads-mips/documents/MIPS_Architecture_microMIPS64_InstructionSet_AFP_P_MD00594_06.04.pdf