|  |  |  |
| --- | --- | --- |
| **ALDAVERA GALLAGA IVÁN** | INSTITUTO POLITÉCNICO NACIONAL | ESCUELA SUPERIOR DE CÓMPUTO |
| **TAREA 05 (SEGUNDO PARCIAL)** | **DEFINICIÓN DE CONCEPTOS** | FECHA DE ENTREGA:  **15 DE ABRIL DEL 2019** |
| GRUPO (**3CM2)** | ARQUITECTURA DE COMPUTADORAS |  |

|  |  |  |
| --- | --- | --- |
| **INCISO** | **CONCEPTO** | **DEFINICIÓN** |
| **A** | ***Index register*** | Es el registro en las instrucciones de transferencia de datos se inventó originalmente para mantener un índice de una matriz con el desplazamiento utilizado para la dirección de inicio de una matriz. |
| **B** | ***Load Instruction*** | El formato de la instrucción de carga es el nombre de la operación seguida por el registro a cargar, luego una constante y registro se usa para memoria de acceso. |
| **C** | ***store instruction*** | Es una arquitectura del conjunto de instrucciones que divide las instrucciones en dos categorías: acceso a la memoria (carga y almacenamiento entre la memoria y los registros) y operaciones ALU (que solo ocurren entre los registros). |
| **D** | ***alignment restriction*** | Un requisito en el que los datos están alineados en la memoria en los límites naturales. |
| **E** | ***Explicar la instrucción [ lw $t0,8($s3) ]*** | Supongamos que A es una matriz de 100 palabras y que el compilador ha asociado las variables **g** y **h** con los registros **$s1** y **$s2** como antes. Supongamos también que la dirección de inicio, o la dirección base, de la matriz está en **$s3**. Compila esta sentencia de asignación de C: **g = h + A [8];**  Aunque hay una sola operación en esta declaración de asignación, uno de los operandos está en la memoria, por lo que primero debemos transferir **A[8**] a un registro. La dirección de este elemento de la matriz es la suma de la base de la matriz A, que se encuentra en el registro **$s3**, más el número para seleccionar el elemento 8. Los datos deben colocarse en un registro temporal para su uso en la siguiente instrucción: ***lw $t0,8($s3) # Temporary reg $t0 gets A[8]***  (Haremos un pequeño ajuste a esta instrucción, pero usaremos esta versión simplificada por ahora.) La siguiente instrucción puede operar en el valor en **$t0** (que es igual a **A[8]**) ya que está en un registro. La instrucción debe agregar h (contenida en **$s2**) a **A[8]** (contenida en **$t0**) y poner la suma en el registro correspondiente a g (asociada con **$s1**)  add **$s1**,**$s2**,**$t0 # g** = **h + A[8]** |
| **F** | ***Starting address, base register, and index register*** | “rs” es el registro base que es sumado a “address” para formar la dirección de memoria. Para una operación “store” “rt” es el registro fuente cuyo valor será almacenado en memoria. |
| **G** | ***Convensión para los registros en el procesador del modelo MIPS*** | La convención del modelo MIPS es usar nombres de dos caracteres después de un signo de dólar  para representar un registro.  Por ahora, usaremos $s0, $s1,... para registros que corresponden a variables en programas C y Java y $t0, $t1,... para los registros temporales necesarios para compilar el programa en instrucciones MIPS. |
| **H** | ***machine language*** | Una representación binaria de instrucciones de la máquina. |
| ***machine code*** | Una secuencia de instrucciones de lenguaje máquina. |
| **I** | ***Big-Endian*** | Aquellas computadoras que usan la dirección del byte más a la izquierda o el "extremo más grande" como la word address |
| ***Little-Endian*** | Aquellas computadoras que usan la dirección del byte más a la derecha o el "extremo más pequeño" como la word address |
| **J** | ***Conditional Branch*** | Instrucciones de decisión tipo if-equal e if-not equal. Instrucciones que diferencian a una computadora de una calculadora, las cuales se encargan de tomar decisiones. |
| ***beq*** | branch on equal |
| ***bnq*** | branch on not equal |
| ***blt*** | branch on less than |
| ***ble*** | branch less than  or equal |
| ***bgt*** | branch greater than |
| ***bge*** | branch greater than  or equal |