|  |  |  |
| --- | --- | --- |
| **Fecha de entrega:** 08 de diciembre de 2017 | **Título de la tarea: Instrucciones LOAD, STORE** | **Número de la tarea:** 3.1 |
| **Grupo:** 3CM3 | **Alumno:** Estrada Granados Diego | **Unidad de aprendizaje:** Arquitectura de Computadoras |

**Instrucciones LOAD, STORE.**

**Desarrollo:**

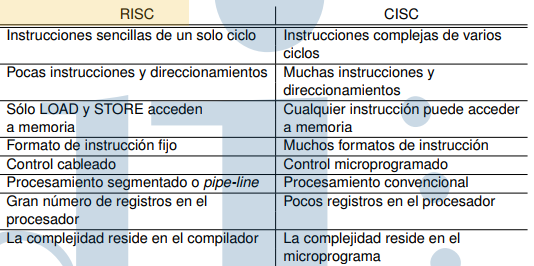
Los conjuntos de instrucciones de las máquinas deben tender a poseer una serie de propiedades, bastante ideales e imprecisas, que pueden resumirse en las siguientes.

* El conjunto de instrucciones de un computador debe ser completo en el sentido de que se pueda construir un programa para evaluar una función computable usando una cantidad de memoria razonable y empleando un tiempo moderado.
* Los juegos de instrucciones también tienen que ser eficientes, esto significa que las funciones más necesarias deben poder realizarse usando pocas instrucciones.

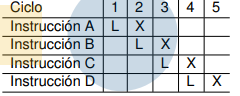
Las instrucciones LOAD, STORE y MOVE, forman parte del conjunto de instrucciones de transferencia de datos. Las instrucciones de transferencia de datos necesitan que se especifiquen el original (fuente u origen) y el lugar donde se desea la copia (destino). Esta especificación variará según sean estos lugares que pueden estar en tres sitios: registros del procesador, memoria o cima de pila. Si el acceso es a una dirección de memoria habrá que especificarla de forma explícita, si se trata de la cima de pila normalmente la especificación será implícita, lo mismo ocurrirá si se trata del acumulador. (Aragón, 2008)



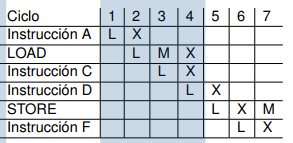
Una vez que se conocen las diferentes arquitecturas (CISC y RISC) de una computadora se realizará una comparación entre estas arquitecturas enfocándonos en las instrucciones LOAD y STORE.



La ejecución de las instrucciones en un solo ciclo de máquina es la propiedad más importante de las máquinas RISC. Esto se consigue por dos razones: en primer lugar, la mayor parte de las instrucciones operan sobre registros, y las que accedan a memoria deben eliminarse del juego de instrucciones, salvo LOAD y STORE; esto evita muchos accesos a memoria que es la operación más lenta; por otra parte, los procesadores RISC utilizan procesamiento segmentado o pipe-line. Con este tipo de procesamiento no se consigue ejecutar una instrucción por ciclo, pero sí, ejecutar n instrucciones en n ciclos, lo que, a efectos prácticos, es equivalente. En la siguiente imagen se observa el proceso de pipeline donde L significa que la instrucción correspondida está en la fase de lectura mientras que X significa que se encuentra en la fase de ejecución. (Valladolid, 2011)



En la imagen siguiente se muestran instrucciones LOAD y STORE en el proceso de pipeline.



Sin embargo, aunque los accesos a memoria estén limitados a las instrucciones LOAD y STORE, éstas no encajan en el esquema anterior ya que necesitan una etapa más en el pipe-line, para el acceso a memoria adicional (M). Esta modificación de los planteamientos supone algunos problemas, porque, como se ve en la figura, la instrucción LOAD que comienza a ejecutarse en el ciclo 2 termina de ejecutarse al menos al mismo tiempo que la instrucción C que comienza en el ciclo 3 (también esto puede suponer problemas por estar los órganos de ejecución ocupados). Mientras la instrucción C no intente usar el registro cargado por LOAD no habrá ningún problema. Depende del compilador asegurarse de que la instrucción siguiente a LOAD no use el registro cargado por ésta (puede hacerlo reordenando las instrucciones). Si aun así el compilador no encuentra la solución, puede insertar, después de la instrucción LOAD, una instrucción de no operación (NOP) que resuelve el problema, aunque alguna vez se pierda un ciclo. (Valladolid, 2011)

# Trabajos citados

Aragón, D. L. (febrero de 2008). *Universidad de Castilla* . Obtenido de https://previa.uclm.es/profesorado/licesio/Docencia/ETC/16\_1\_CBas-RepInstMD\_itis.pdf

Rouse, M. (Septiembre de 2005). *Whalts.com.* Obtenido de http://whatis.techtarget.com/definition/word

Valladolid, U. d. (diciembre de 2011). *Departamento de Informática UV*. Obtenido de https://www.infor.uva.es/~bastida/OC/conjunto.pdf