Guía para el Trabajo Práctico Número 4

Arquitectura de Computadoras 15/04/2026

Problemas Propuestos

2) *[Modificado]* Comparar las máquinas de 0, 1, 2 y 3 operandos, escribiendo un programa para calcular la siguiente expresión:

$$R = \frac{A - B}{B * C}$$

Las instrucciones disponibles para cada uno de los tipos de máquinas son las siguientes:

0 Operandos	1 Operando	2 Operandos	3 Operandos
PUSH M	LOAD M	MOVE X, Y	LOAD R1, M
POP M	STORE M	SUM X, Y	STORE R1, M
SUM	SUM M	RES X, Y	SUM R1, R2, R3
RES	RES M	MUL X, Y	RES R1, R2, R3
MUL	MUL M	DIV X, Y	MUL R1, R2, R3
DIV	DIV M		DIV R1, R2, R3

Donde M es una dirección de memoria de 16 bits, X e Y son direcciones de 16 bits o números de registro de 4 bits y R1, R2 y R3 son números de registro de 4 bits. La máquina de cero operandos usa una pila (Stack), la de un operando usa un acumulador, la de dos operandos tiene 16 registros, e instrucciones en las que cada operando puede estar en un registro o en memoria, mientras que la de tres operandos tiene 16 registros, e instrucciones para cargar los registros y para operar los mismos. En todos los casos, se pueden utilizar las posiciones de memoria auxiliares que se desee.

a) Suponiendo códigos de operación de 4 bits y longitudes de instrucción múltiplos de 8 bits, ¿cuántos bits necesita cada computadora para almacenar el programa propuesto?
b) Si en todos los casos se trata de máquinas de 8 bits, ¿cuántos accesos a memoria requiere cada computadora para completar el programa propuesto?

6) Se desea diseñar el set de instrucciones de una máquina que posee un banco de 8 registros de propósito general de 16 bits. Debe usted decidir el formato de las 13 instrucciones que operan con constantes inmediatas, buscando maximizar la velocidad de ejecución de las mismas, sabiendo que los accesos a memoria demoran 1T tanto para datos como para instrucciones, y que la decodificación y ejecución de las mismas demoran 2T adicionales a la búsqueda de la instrucción. Para ayudarlo en la decisión usted dispone de la tabla en la que se muestra una estadística con el porcentaje de operaciones y el tamaño de constante que requieren las mismas.

Tamaño de Cte	Cantidad
2	17 %
4	15 %
6	16 %
8	9 %

Tamaño de Cte.	Cantidad
10	4 %
12	10 %
14	7 %
16	22 %

Soluciones Completas

2)

- *Maquina de 0 Operandos*: Un ejemplo de este tipo de máquina es la JVM (Maquina Virtual de Java) que es una maquina de stack, es decir de 0 operandos. Para esta máquina se distinguen claramente dos tipos de instrucciones, aquellas que agregan o sacan un dato al stack y las que operan con el mismo.

  Recordando que las longitud instrucciones deben ser múltiplo de 8 bits, el formato de las instrucciones PUSH y POP sería el siguiente:

OpCode 4 Bits	Sin uso 4 Bits	Dirección de Memoria 16 Bits

El otro tipo de instrucción, no tiene acceso a memoria, simplemente operan accediendo al stack, por lo que el formato es:

OpCode 4 Bits	Sin uso 4 Bits

Una vez definido el formato de las instrucciones, hay que ver cuantos accesos a memoria requiere cada una:

- PUSH M: 3 accesos para leer la instrucción, 1 acceso para traer el dato de la memoria y 1 acceso para guardar el dato en el stack.
- POP M: 3 accesos para leer la instrucción, 1 acceso al stack para sacar el dato y 1 acceso a memoria para guardar el dato.
- Aritméticas: 1 acceso para leer la instrucción, 2 accesos al stack para sacar dos datos y operarlos y 1 acceso al stack para guardar el resultado

El programa para la máquina de 0 operandos es el siguiente:

0 Operandos	**Operacion**	**Tamaño**	**Accesos**
PUSH B		3	5
PUSH C		3	5
MUL	(B * C)	1	4
PUSH B		3	5
PUSH A		3	5
RES	(A - B)	1	4
DIV	(A - B) / (B * C)	1	4
POP R		3	5
8 Instrucciones		**18 Bytes**	**37 Accesos**

- *Maquina de 1 Operandos*: Un ejemplo de este tipo de máquina es la Familia 8080 que es una maquina de acumulador, el cual actúa como fuente y destino de todas las operaciones de procesamiento. En estos set de instrucciones el acumulador es un operando implícito y solo se especifica un único operando que siempre esta en memoria.

  Para esta maquina no hay distinción de operaciones, todas acceden a memoria. Nuevamente recordando que la longitud de las instrucciones debe ser múltiplo de 8 bits, el formato de todas las instrucciones de esta máquina sería:

OpCode 4 Bits	Sin uso 4 Bits	Dirección de Memoria 16 Bits

Los accesos a memoria para todas las instrucciones también van a ser iguales: 3 accesos para leer la instrucción y 1 acceso para buscar el dato o guardarlo en el caso de un STORE. El programa para la máquina de 1 operandos es el siguiente:

1 Operando	**Operacion**	**Tamaño**	**Accesos**
LOAD B		3	4
MUL C	(B * C)	3	4
STORE R		3	4
LOAD A		3	4
RES B	(A - B)	3	4
DIV R	(A - B) / (A * C)	3	4
STORE R		3	4
7 Instrucciones		**21 Bytes**	**28 Accesos**

- *Maquina de 2 Operandos*: Un ejemplo de este tipo de máquina es el Pentium que es un procesador de dos operandos. Esta máquina presenta múltiples formatos para una misma instrucción, dependiendo de los operandos que use.

  Cuando ambos son direcciones de memoria el formato es:

OpCode 4 Bits	Sin uso 4 Bits	Dirección de Memoria Uno 16 Bits	Dirección de Memoria Dos 16 Bits

  La cantidad de accesos a memoria para todas las instrucciones de este formato, excepto las instrucciones MOVE, será de 5 accesos para leer la instrucción, 2 accesos para obtener los operandos de las direcciones de memoria especificadas y 1 acceso para guardar el dato operado. En el caso de las instrucciones MOVE hay un acceso menos, porque hay que leer solo un operando desde memoria.

  Cuando ambos son direcciones de memoria el formato es:

OpCode 4 Bits	Registro 4 Bits	Dirección de Memoria 16 Bits

  La cantidad de accesos a memoria para este tipo de instrucciones va a variar dependiendo de si el primer operando es una dirección de memoria o un registro y viceversa. Como regla general el primer paso es determinar cuantos accesos son necesarios para leer entera la instrucción, después determinar los accesos a memoria para la lectura de los operandos que no sean registros, y finalmente considerar el accesos a memoria para almacenar el resultado si corresponde.

  Cuando ambos son registros el formato es:

OpCode 4 Bits	Sin uso 4 Bits	R1 4 Bits	R2 4 Bits

  La cantidad de accesos a memoria para todas las instrucciones de este formato será de 2 accesos para leer la instrucción.

  El programa para la máquina de 2 operandos es el siguiente:

2 Operandos	**Operacion**	**Tamaño**	**Accesos**
MUL C, B	C = (C * B)	5	8
RES A, B	A = A - B	5	8
DIV A, C	A = (A - B) / (B * C)	5	8
MOV R, A	R = (A - B) / (B * C)	5	7
4 Instrucciones		**20 Bytes**	**31 Accesos**

- *Maquina de 3 Operandos*: Un ejemplo de este tipo de máquina es el MIPS que es un procesador de tres operandos. Este tipo de máquina es del tipo LOAD/STORE, esto quiere decir que solo esas operaciones acceden a memoria, mientras que el resto opera solo con los registros.

  El formato de las instrucciones LOAD/STORE será:

OpCode = 4 Bits	R = 4 Bits	M = 16 Bits

  La cantidad de accesos a memoria para las instrucciones de este formato será de 3 accesos para leer la instrucción y 1 acceso para leer el dato de la memoria o guardarlo en la misma.

  El formato del resto de las instrucciones será:

OpCode = 4 Bits	R1 = 4 Bits	R2 = 4 Bits	R3 = 4 Bits

  La cantidad de accesos a memoria para las instrucciones de este formato será de 2 accesos para leer la instrucción.

El programa para la máquina de 3 operandos es el siguiente:

3 Operandos	**Operacion**	**Tamaño**	**Accesos**
LOAD R1,A		3	4
LOAD R2,B		3	4
LOAD R3,C		3	4
MUL R4,R2,R3	(B * C)	2	2
RES R5,R1,R2	(A - B)	2	2
DIV R6,R4,R6	(A - B) / (B * E)	2	2
STORE R6,R		3	4
7 Instrucciones		**18 Bytes**	**22 Acceso**

6) Cuando se desea diseñar un set de instrucciones uno de los pasos principales es dimensionar el opcode. Para este caso en particular se requieren 13 instrucciones que trabajen con constantes inmediatas, por lo tanto son 4 bits para el opcode. Lo siguiente es saber cuantos bits destinarle a los operandos. En este caso uno de ellos es un registro, como tengo que elegir uno entre ocho registros necesito 3 bits. Hasta ahora hemos utilizado 7 bits, por lo que nos quedan 9 bits libres, debido a que las palabras son de 16 bits.
Hay dos opciones, si utilizamos una sola instrucción tenemos 9 bits para la constante y deberíamos desdoblar en dos instrucciones a las operaciones con constantes inmediatas de más de 8 bits, ya que no tenemos información de las constantes de 9 bits en la tabla suministrada. Esto quiere decir que cuando la constante sea de hasta 8 bits solo vamos a ejecutar una instrucción, mientras que si es mayor se van a ejecutar dos instrucciones, en donde en la segunda se va a completar el valor de la constante. Teniendo en cuenta que: 57 % Constantes es menor o igual a 8 Bits y que el 43 % restante corresponde a Constantes mayores 8 Bits

$$CPI = 1T(traer\ 1\ palabra) + 2T(decodificacion\ y\ ejecucin) = 3T$$

$$\overline{CPI} = (57\,\% * 3T) + (43\,\% * 2(Instrucciones) * 3T) = 4,29T$$

Si en cambio utilizamos 2 palabras para cada instrucción con una constante alojando la misma en la segunda palabra. Obtenemos:

$$\overline{CPI} = 2T(traer\ 2\ palabras) + 2T(decodificacion\ y\ ejecucion) = 4T$$

Con estos resultados se ve claramente que en este caso es conveniente utilizar 2 palabras por cada operación con constantes inmediatas. Esto se debe a que al usar una sola instrucción de dos palabras, me estoy ahorrando la decodificación extra que ocasiona el primer caso.