
Arquitectura de Computadores - 22GIIN

Actividad 3 - Portafolio

Gagliardo Miguel Angel

22 de Diciembre de 2023

EJERCICIO 1. CARACTERISTICAS

1. El computador tiene palabras de 32 bits y 8 registros de 32 bits.
2. La memoria es de **64 MB**.
3. El juego de instrucciones se forma con el código de operación, y el campo de cada operando debe llevar asociado su modo de direccionamiento.
4. **Todas las instrucciones deben representarse en 32 bits.**
5. El conjunto de instrucciones son las siguientes:
 - a. 14 instrucciones lógico-aritméticas de 3 operandos: 1 en registro y dos en memoria;
 - b. 286 instrucciones de dos operandos: 1 en registro y otro en memoria;
 - c. 24 instrucciones de un operando, en memoria;
 - d. 98 instrucciones de 0 operando.

Se pide:

Verificar (e indicar) si es factible que puedan representarse todas las instrucciones en 32 bits. Deben incluir todos los elementos (aspectos técnicos y conceptuales) que justifiquen su respuesta.

Características	Codigo de Operacion		Modo de Direcccionamiento
Tamaño de la Palabra	32 bits		32 bits
Memoria Total en Bytes	64 MB = 2^{26} bytes		26 bits
Memoria Total en Palabras	64MB / 4 = 2^{24} palabras		24 bits
Maxima direccion efectiva	$2^{24} - 1$		24 bits
Tamaño de Instruccion	24 bits		24 bits
Cantidad de Registros	8		3 bits (2^3)
Cantidad de Operaciones	422		9 bits (2^9)
Modo de Direcccionamiento	3 Operandos	14	4 bits (2^4)
	2 Operandos	286	9 bits (2^9)
	1 Operandos	24	5 bits (2^5)
	0 Operandos	98	7 bits (2^7)

Tabla 1.1

Como vemos en la Tabla 1.1, se requiere que todas las instrucciones se representen en 32 bits y se proporciona información sobre el número de operandos y el modo de direccionamiento, podemos verificar si es factible cumplir con esta restricción para cada tipo de instrucción.

a. 14 Instrucciones lógico-aritméticas de 3 operandos (1 en registro y dos en memoria):

- El código de operación necesita 4 bits para representar las 14 instrucciones.
- Cada operando en memoria es de 24 bits.
- El registro requiere 3 bits para ser representado.
- Necesitaremos 2 bits para cada modo de direccionamiento.

Como vemos ya de inicio necesitamos 2 operandos en memoria, los cuales **cada uno**

requieren 24 bits = $2 * 24 \text{ bits} = 48 \text{ bits solo para los operandos}$. Por tanto este tipo de instrucciones **no pueden representarse en 32 bits**.

COD-OP	MD1	Registro	MD2	Operando 2	MD3	Operando 3	
(C=4)	(M=2)	(R=3)	(M=2)	(P=24)	(M=2)	(P=24)	(S=10)

$$C + 3M + R + 2P + S \text{ (bits no utilizados)} = 4 + 3*2 + 3 + 2*24 + 10 = 96 \text{ bits (3 palabras)}$$

b. 286 Instrucciones de dos operandos (1 en registro y otro en memoria):

- Cada operando en memoria es de 24 bits.
- El código de operación necesita 9 bits para representar las 286 instrucciones.
- El registro requiere 3 bits para ser representado.
- Necesitaremos 2 bits para cada modo de direccionamiento.

Nuevamente, vemos como este tipo de operaciones **no puede** representarse con 32 bits.

COD-OP	MD1	Registro	MD2	Operando 2	
(C=9)	(M=2)	(R=3)	(M=2)	(P=24)	(S=22)

$$C + 2M + R + P + S \text{ (bits no utilizados)} = 9 + 3 \cdot 2 + 3 + 24 + 22 = 64 \text{ bits (2 palabras)}$$

c. 24 Instrucciones de un operando (en memoria):

- El operando en memoria es de 24 bits.
- El código de operación necesita 5 bits para representar las 24 instrucciones.
- Necesitaremos 2 bits para el modo de direccionamiento.

En este caso **podremos representar este tipo de operaciones en 32 bits.**

COD-OP	MD1	Operando	
(C=5)	(M=2)	(P=24)	(S=1)

$$C + M + P + S \text{ (bits no utilizados)} = 5 + 2 + 24 + 1 = 32 \text{ bits (1 palabra)}$$

d. 98 Instrucciones de 0 operandos:

- El código de operación necesita 7 bits para representar las 98 instrucciones.
- Dado que no hay operandos, no necesitaremos mas.

Por tanto, aqui tambien podremos representar esta instruccion en 32 bits.

COD-OP	
(C=7)	(S=25)

$$C + S \text{ (bits no utilizados)} = 7 + 25 = 32 \text{ bits (1 palabra)}$$

EJERCICIO 2. CARACTERISTICAS

1. El computador tiene palabras de 64 bits y 32 registros de 64 bits.
2. La memoria es de **512 Mpalabras**.
3. Los **modos de direccionamiento** permitidos son: inmediato, directo y por registro.
4. El juego de instrucciones se forma con el código de operación, y el campo de cada operando debe llevar asociado su modo de direccionamiento.
5. En las instrucciones aritméticas cada operando debe tener asociado el tipo de representación, los cuales son: Entero, Punto Flotante y Complemento a 2.
6. En los formatos de instrucciones debe aparecer información sobre el tipo de instrucción, ya sean Aritméticas, Lógicas o de Carga-Almacenamiento.
7. El **juego de instrucciones** de esta máquina se describe en la siguiente tabla:

Tipo de Instrucción	Mnemónico y Operandos	Semántica
Aritméticas	SUM destino, operando1, operando2	destino \leftarrow operando1 + operando2
	SUB destino, operando1, operando2	destino \leftarrow operando1 - operando2
	MUL destino, operando1, operando2	destino \leftarrow operando1 * operando2
	DIV destino, operando1, operando2	destino \leftarrow operando1 / operando2
Lógicas	AND destino, operando1, operando2	destino \leftarrow operando1 AND operando2
	OR destino, operando1, operando2	destino \leftarrow operando1 OR operando2
	XOR destino, operando1, operando2	destino \leftarrow operando1 XOR operando2
Carga - Almacenamiento	LD memoria, registro	registro \leftarrow C[memoria] (*)
	STR registro, memoria	memoria \leftarrow C[registro] (*)

(*) C[memoria] = contenido de esa dirección de memoria

C[registro] = contenido de ese registro

Se pide:

Definir (diseñar) el formato de cada una de las instrucciones, e incluir su descripción en forma gráfica.

Para cada instrucción o tipo de instrucciones debe indicar su tamaño en bits o palabras.

Todas las decisiones de diseño deben estar completamente justificadas y razonadas.

Características	Código de Operación	Modo de Direccionamiento
Tamaño de la Palabra	64 bits	64 bits
Memoria Total	512 Mpalabras = 2^{29}	29 bits
Maxima direccion efectiva	$2^{29} - 1$	29 bits
Tamaño de Instrucción	29 bits	29 bits
Cantidad de Registros	32	5 bits (2^5)
Cantidad de Operaciones	9	4 bits (2^4)
Modo de Direccionamiento	3	2 bits (2^2)

Tabla 2.1

Como se observa en la Tabla 2.1:

- Cada operando puede tener cualquiera de los **3 modos de direccionamiento (inmediato, directo y por registro)**, con lo cual para distinguir los 3 se requeriran **2 bits (2^2)**
- Dado que contamos con **9 instrucciones distintas** en total, bastara con **4 bits (2^4)** para diferenciarlas

- Los **32 registros** necesitaran **5 bits** para ser diferenciados (2^5)
- Para instrucciones aritméticas, para cada operando debemos tener asociado uno de **3** (Entero, Punto Flotante y Complemento a 2) tipo de representacion. Por tanto necesitaremos 2 bits mas (2^2) **para estos tipos de operaciones**
- Finalmente, para poder direccionar **512 Mpalabras** se necesitaran **29 bits** (2^{29})

Por lo tanto, las soluciones propuestas son:

1. Para instrucciones de tipo:

- Arimeticas (SUM, SUB, MUL, DIV)
- Logicas (AND, OR, XOR)

Operacion_Aritmetica destino, operando1, operando2

Operacion_Logica destino, operando1, operando2

(C=4)		(M=2)		(S=25)		(M=2)		(R=2)		(P=29)	
COD-OP	RD1	MD1	Operando 1		RD2	MD2	Operando 2	RD3	MD3	Operando 3	
(R=2)		(P=29)		(R=2)		(P=29)		(M=2)			

$$C + 3R + 3M + 3P + S \text{ (bits no utilizados)} = 4 + 3*2 + 3*2 + 3*29 + 25 = 128 \text{ (2 palabras)}$$

2. Para instrucciones de tipo:

- Carga-Almacenamiento (LD, STR)

Operacion_Carga_Almacenamiento memoria, registro

Operacion_Carga_Almacenamiento registro, memoria

COD-OP	MD1	Operando 1	MD2	Operando 2	
(C=4)	(M=2)	(P=29)	(M=2)	(P=29)	(S=62)

$$C + 2M + 2P + S \text{ (bits no utilizados)} = 4 + 2*2 + 2*29 + 62 = 128 \text{ (2 palabras)}$$

Respecto a los modos de direccionamiento, **tanto inmediato como directo**:

MD-i	Valor
(2)	(29)

Caso de **direccionamiento por registro**:

MD-i	Registro	No utilizados
(2)	(5)	(24)

CONCLUSIONES

EJERCICIO 1. Como puede verse en el , dado que la memoria es de $64\text{MB} = 2^{24}$ palabras, y que **todas las instrucciones** deben presentarse en **32 bits** (y **no en multiples de 32 bits**), solo contando la instrucciones del punto **1a** (instrucciones lógico-aritméticas de 3 operandos) que tienen 2 operadores en memoria, significa que serian **48 bits solo** en operadores ($2 * 24$ bits). Con lo cual **no se podria** representar **todas** las instrucciones en 32 bits, dado que, como punto de partida, las logico-aritmeticas no pueden ser representadas.

Aun asi, podríamos considerar la posibilidad de utilizar una **operación extendida** para implementar instrucciones que requieren más bits de los disponibles en una instrucción estándar. Esto podría hacerse mediante la introducción de una instrucción especial o un modo de operación que permita procesar operandos más grandes o con mayor precisión.

Por ejemplo, se podria diseñar una instrucción especial que se encargue de operar con operandos en memoria utilizando más de una instrucción estándar. Esta instrucción especial podría interpretar múltiples instrucciones consecutivas como una única operación extendida.

Sin embargo, es importante señalar que la implementación de operaciones extendidas a menudo introduce complejidad en la arquitectura y puede afectar el rendimiento y la eficiencia del conjunto de instrucciones. Además, este enfoque podría requerir un manejo especial en el ensamblador y el compilador para generar y optimizar el código correspondiente.

En resumen, es posible abordar el problema mediante operaciones extendidas, pero se deben evaluar cuidadosamente los trade-offs y considerar la complejidad que esto podría agregar a la arquitectura del conjunto de instrucciones.

EJERCICIO 2. Este ejercicio resalta la importancia del diseño preciso del formato de instrucciones para una ejecución eficiente y clara de las operaciones en una máquina. La memoria se especificó como 512 MPalabras, lo que equivale a 2^{29} palabras. Este ajuste es crucial para entender la capacidad y la dirección efectiva de la memoria en el diseño del conjunto de instrucciones.

En resumen, el Ejercicio 2 destaca la importancia de considerar la arquitectura de la máquina al diseñar conjuntos de instrucciones, asegurando que sean eficientes y claros para las operaciones específicas que se llevarán a cabo en la computadora. La atención a detalles como el tamaño de palabra, la memoria disponible y los modos de direccionamiento contribuyó a un diseño coherente y funcional.

Por ultimo, ambos ejercicios ilustran la complejidad y la necesidad de considerar diversos factores, como el tamaño de la palabra, la memoria disponible y los modos de direccionamiento,

al diseñar conjuntos de instrucciones para sistemas informáticos. Estos aspectos son cruciales para garantizar la eficiencia y la efectividad de las operaciones en una arquitectura dada.

BIBLIOGRAFIA UTILIZADA

Figueras, G. E. (2019). ARQUITECTURA DE COMPUTADORES. Manual del curso.

Universidad Internacional de Valencia. **Tema 5**