

Arquitectura de Computadores - 22GIIN

Actividad 3 - Portafolio

Alumno: Jack Nathaniel Ross

Fecha de entrega: 22 de diciembre 2023

Universidad Internacional de Valencia

Grado de ingeniería informática

Arquitectura de Computadores

ACTIVIDAD 2 ARQUITECTURA DE COMPUTADORES

ALUMNO: JACK ROSS

En la Unidad Competencial 3 se trata el tema de Repertorio de Instrucciones. Durante el diseño del conjunto de instrucciones para una máquina dada, una de las actividades a realizar es la definición (diseño) del formato de cada instrucción. Esto es la descripción en bits de una instrucción en términos de las distintas partes o campos que la componen. Debe incluir un código de operación (codop) e, implícita o explícitamente, cero o más operandos. Cada operando explícito se referencia utilizando uno de los modos de direccionamiento que provee la máquina, es decir el formato sugerido debe indicar el modo de direccionamiento para cada operando, implícita o explícitamente. En la mayoría de los repertorios de instrucciones se emplean más de un formato de instrucción, y pueden tener tamaño variable. En principio, es deseable que el tamaño de las instrucciones sean múltiplos del tamaño de la palabra de memoria. En el caso se requiera un diseño del repertorio con la limitación a un número específico de bits, aparecen algunas estrategias para lograrlo.

A continuación, se definen algunos aspectos de dos máquinas.

EJERCICIO 1. CARACTERÍSTICAS.

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.

Todas las instrucciones deben representarse en 32 bits			
Características	Valores		Representación en bits
Memoria total en (palabras)	32 bits		32 bits
Memoria total (Bytes)	64 MBytes = $2^6 * 2^{20} = 2^{26}$		26 bits
Memoria total (palabras) - La memoria se divide en bloques/palabras de 32 bit	$64 \text{ MB} / 4 \text{ B} = 2^{26} / 2^2 = 2^{24}$		24 bits
Máxima dirección (longitud)	$2^{24} - 1$		24 bits (P)
Tamaño/longitud de las instrucciones	Palabras de 24 bits		24 bits
Cantidad de registros	8 registros		3 bits (R)
Cantidad de instrucciones de:	3 operando	14	4 bits (C)
	2 operando	286	9 bits (C)
	1 operando	24	5 bits (C)
	0 operando	98	7 bits (C)
Modos de direccionamiento	Asumo que son implícitos		

a). Formato de instrucción de 14 instrucciones lógico-aritméticas, 1 registro y dos en memoria:

3 operandos:

COD-OPER	Operando 1 / Registro 3	Operando 2	Operando 3	Sin utilizar
(C = 4)	(R = 3)	(P = 24)	(P = 24)	S = 96 - 55
$C + R + 2P + S$ (bits sin utilizar) no es un múltiplo de la palabra de memoria. $4 + 3 + 2 \cdot 24 + S = 64 \text{ bits} \rightarrow 2 \text{ palabras}$				

La longitud del formato de instrucciones de 14 instrucciones y operandos no puede representarse en 32 bits, no sería un conjunto de instrucciones valido.

b). Formato de instrucción de 286 instrucciones de dos operandos, 1 registro y otro en memoria:

2 operandos:

COD-OPER	Operando 1 / Registro 3 bits	Operando 2	Sin utilizar
(C = 9)	(P = 3)	(P = 24)	S = 64 - 36
$C + R + P + S$ (bits sin utilizar) no es un múltiplo de la palabra de memoria. $9 + 3 + 24 + S = 64 \text{ bits} \rightarrow 2 \text{ palabras}$			

La longitud del formato de instrucciones de 286 instrucciones y 2 operandos no puede representarse en 32 bits, no sería un conjunto de instrucciones valido.

c). Formato de instrucción de 24 instrucciones de un operando en memoria:

1 operando:

COD-OPER	Operando 1	Sin utilizar
(C = 5)	(P = 24)	S = 32 - 29
$C + P + S$ (bits sin utilizar) no es un múltiplo de la palabra de memoria. $5 + 24 + S = 32 \text{ bits} \rightarrow 1 \text{ palabras}$		

La longitud del formato de instrucciones de 24 instrucciones y 1 operando permite representarse 32 bits, sería un conjunto de instrucciones valido.

d). Formato de instrucción de 98 instrucciones de 0 operando:

0 operando:

COD-OPER	Sin utilizar
(C = 7)	S = 32-7
$C + S$ (bits sin utilizar) no es un múltiplo de la palabra de memoria. $7 + 25 = 32 \text{ bits} \rightarrow 1 \text{ palabra}$	

La longitud del formato de instrucciones de 98 instrucciones y sin operandos, se puede representar en 32 bits, sería un conjunto de instrucciones valido para nuestra arquitectura.

Códigos de expansión

Para implementar códigos de operación, debemos hallar los bits que tenemos disponibles. Empezando por el operando 1 hasta el operando 0:

24 instrucciones de 1 operando, en memoria

24 instrucciones serán abarcados por 5 bits. 5 bits pueden representar hasta 32 instrucciones.

Instrucciones	bits				
0	0	0	0	0	0
...
23	1	0	1	1	1

COD-OPER (C = 5)	Operando 1 (P = 24)	Sin utilizar (S = 1)
0 0 0 0 0	1 dirección	1
...		
1 1 1 1 0		

Desde 0 0 0 0 0 hasta 1 1 1 1 0 estarán representadas las 24 instrucciones de 1 operando.

Para que el procesador pueda identificar las instrucciones de 0 operando, emplearé la última dirección binaria disponible: 11 1111. Este valor binario es un valor sobrante a las instrucciones previas de 24 instrucciones, de un operando. Concretamente podríamos haber empleado desde 25 a 31, o en valores binarios de 01 0111 a 01 1111. Por lo tanto, a partir del valor 11 1111, los valores binarios serán identificados como instrucciones de 98 instrucciones de 0 operando.

Para representar 98 instrucciones, sin operando

COD-EXPANSIÓN (CE = 5)	COD-OPER (C = 7)	Sin utilizar (S = 20 bits)
1 1111	000 0000	20 bits
...	...	
1 1111	111 1111	

EJERCICIO 2. CARACTERÍSTICAS.

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:

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

Características	Valores	Representación en bits
Memoria total en (palabras)	64 bits	64 bits
Memoria total (Bytes)	512 MBytes = $2^9 * 2^{20} = 2^{29}$	29 bits
Memoria total (Palabras) - La memoria se divide en bloques/palabras de 64 bits	512 MBytes / 8 bytes $2^{29} / 2^3 = 2^{26}$	26 bits
Máxima dirección (longitud)	$2^{26} - 1$	26 bits (P)
Tamaño/longitud de las instrucciones	Palabras de 23 bits	26 bits
Cantidad de registros	32 registros	5 bits (R)
Cantidad de instrucciones	9	4 bits (C)
Modos de direccionamiento	3	2 bits (M)
Tipo de representación	3	2 bits (RD)

Modos de direccionamiento

Inmediato

MD-1	Valor
(M = 2)	(P = 26)

Directo

MD-1	Dirección
(M = 2)	(P = 26)

Por registro

MD-1	Operando
	Registro 5 bits / 21 bits quedan libres
(M = 2)	(R = 5)

2 operandos (Tipos de instrucciones: carga-almacenamiento de memoria a registro y viceversa)

COD-OPER	RD	MD-1	Operando 1 Registro 5 bits	MD-2	Operando 2	Sin utilizar
(C = 9)	(RD = 2)	(M = 2)	(R = 5)	(M=2)	(P = 26)	S = 64-41
$C + R + 2M + P + S$ (bits sin utilizar) no es un múltiplo de la palabra de memoria. $9 + 2 + 2*2 + 26 + S = 64 \text{ bits} \rightarrow 1 \text{ palabra}$						

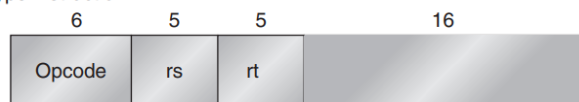
3 operandos (Tipos de instrucciones: aritméticas y lógicas)

COD-OPER	MD-1	Operando 1	MD-2	Operando 2	MD-3	Operando 3	Sin utilizar
(C = 9)	(M = 2)	(P = 26)	(M = 2)	(P = 23)	(M = 2)	(P = 26)	S = 128-93
$C + 3M + 3P + S$ (bits sin utilizar) no es un múltiplo de la palabra de memoria. $9 + 3*2 + 3*26 + S = 128 \text{ bits} \rightarrow 2 \text{ palabras}$							

Conclusión:

En los diversos ejercicios realizados podemos ver formatos de instrucción de 0 a 3 operandos, cada una de ellas con sus respectivos apartados. Las variaciones se deben a diferentes instrucciones que se están mandando al procesador. En cada ejercicio se ha adaptado el código de operación, los operandos (valores y/o direcciones), modos de dirección (cuando había que especificarlos explícitamente), campo de registro (cuando había operaciones en el registro), y el campo final donde se implementa la longitud del formato de instrucción, cuando la instrucción no es un múltiplo de los bits del procesador. Gracias a esta estos ejercicios tengo más experiencia con repertorios de instrucciones, me gustaría poder plasmar algunos ejemplos de formatos de instrucciones en el mundo real. En la figura 1 podemos observar formatos de instrucción, de arquitectura RISC de 32 bits. Los códigos de operación son de 6 bits, carecen de modos de direccionamiento al trabajar exclusivamente en el registro y sus operandos son de 5 bits.

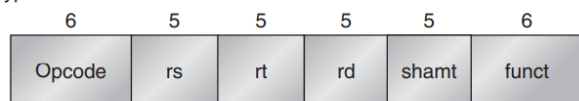
I-type instruction



Encodes: Loads and stores of bytes, half words, words, double words. All immediates ($rt \leftarrow rs \text{ op immediate}$)

Conditional branch instructions (rs is register, rd unused)
Jump register, jump and link register
($rd=0$, rs =destination, immediate=0)

R-type instruction



Register-register ALU operations: $rd \leftarrow rs \text{ funct } rt$

Function encodes the data path operation: Add, Sub, . . .

Read/write special registers and moves

J-type instruction



Jump and jump and link

Trap and return from exception

Figura 1: Instrucciones MIP de tipo inmediato, registro y de salto. (Hennessy et al., 2012)

En la figura 2 podemos ver instrucciones CISC de 32 bits, al tratarse de un procesador con otro repertorio de instrucciones, tendremos diferentes formatos. Tendremos condicionales, códigos de operación de 4 bits. Al tener capacidad de trabajar con memoria, se implementan más modos de direccionamiento.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data processing immediate shift	Cond	0	0	0	Opcode				S	Rn				Rd				Shift amount				Shift	0	Rm								
Data processing register shift	Cond	0	0	0	Opcode				S	Rn				Rd				Rs		0	Shift	1	Rm									
Data processing immediate	Cond	0	0	1	Opcode				S	Rn				Rd				Rotate		Immediate												
Load/store immediate offset	Cond	0	1	0	P	U	B	W	L	Rn				Rd				Immediate														
Load/store register offset	Cond	0	1	1	P	U	B	W	L	Rn				Rd				Shift amount		shift	0	Rm										
Load/store multiple	Cond	1	0	0	P	U	S	W	L	Rn				Register list																		
Branch/branch with link	Cond	1	0	1	L	24-bit offset																										

Figura 2: Representación simbólica del formato de instrucciones CISC. (Stallings, 2010)

Bibliografía:

Hennessy, J. L., Patterson, D. A., & Asanović, K. (2012). Computer Architecture: A Quantitative Approach. Elsevier.

Stallings, W. (2010). Computer Organization and Architecture: Designing for Performance. Prentice Hall.