



Universitat de Lleida

Simulador Von Neumann

Repertori d'instruccions

Estructura de Computadors II

Jordi Vilaplana <jordi.vilaplana@udl.cat>

David Puiggròs <david.puiggros@udl.cat>

Marc Tomàs <marc.tomas@udl.cat>

VN - RI: Paraules reservades

@CODIGO	@DATOS	@PILA	ADD
AND	BRC	BRNC	BRNO
BRNS	BRNZ	BRO	BRS
BRZ	BYTEALTO	BYTEBAJO	CALL
CLI	CODIGO	COMP	DATOS
DEC	DIRECCION	EQU	FIN
FINP	INC	INICIO	INT
IRET	JMP	MOV	MOVH
MOVIL	NEO	NOP	NOT
OR	ORIGEN	PILA	POP
PROCEDIMIENTO	PUSH	R0	R1
R2	R3	R4	R5
R6	R7	RET	STI
SUB	VALOR	VECES	XOR

VN - Repertori d'instruccions

3

Moviment	Aritmètic-Lògiques	Interrupcions	Control
NOP	ADD Rd, Rs1, Rs2	CLI	JMP Inm_8
MOV Rd, Rs	SUB Rd, Rs1, Rs2	STI	JMP Rx
MOV Rd, [Ri]	OR Rd, Rs1, Rs2	INT Inm_8	CALL Inm_8
MOV [ri], Rs	AND Rd, Rs1, Rs2	IRET	CALL Rx
MOVL Rd, Inm_8	XOR Rd, Rs1, Rs2	NOMENCLATURA: - Rs[1/2]/Rd: registre origen i destí de les operacions - Ri: registre índex per adreçament indirecte - Rx: registre amb l'adreça destí en salts indirectes - Inm_8: valor numèric	RET
MOVH Rd, Imn_8	COMP Rs1, Rs2		BR cond Inm_8
Push Rs	NOT Rd/s		
Pop Rd	INC Rd/s		
	DEC Rd/s		
	NEG Rd/s		

VN - RI - Moviment de dades

4

Codi d'instrucció	Mnemònic	Operació	Descripció
00 000 000000000000	NOP	-	Instrucció nul·la
00 001 Rd Rs 00000	MOV Rd, Rs	Rd = Rs	Transferència entre registres
00 010 Rd Ri 00000	MOV Rd, [Ri]	Rd = [Ri]	Copiar el contingut de l'adreça de memòria indicada per Ri a Rd
00 011 Ri Rs 00000	MOV [Ri], Rs	[Ri] = Rs	Copiar el contingut del registre Rs a l'adreça de memòria indicada per Ri
00 100 Rd Inm_8	MOVL Rd, Inm_8	Rd _{baix} = Inm_8	Copia als 8 bits menys significatius de Rd la dada codificada als 8 bits del camp Inm_8
00 101 Rd Inm_8	MOVH Rd, Inm_8	Rd _{alt} = Inm_8	Copia als 8 bits més significatius de Rd la dada codificada als 8 bits del camp Inm_8
00 110 Rs 00000000	PUSH Rs	Pila = Rs	Apila el contingut del registre Rs
00 111 Rd 00000000	POP Rd	Rd = Pila	Desapila el contingut del registre Rd

VN - RI - Instruccions aritmetico-lògiques

5

	Codi d'instrucció	Mnemònic	Operació	Descripció
3 operands	01 000 Rd Rs1 Rs2 00	ADD Rd, Rs1, Rs2	$Rd = Rs1 + Rs2$	Suma el contingut dels registres Rs1 I Rs2 I emmagatzema el resultat en Rd.
	01 001 Rd Rs1 Rs2 00	SUB Rd, Rs1, Rs2	$Rd = Rs1 - Rs2$	Resta el contingut del registre Rs2 al registre Rs1 I emmagatzema el resultat en Rd.
	01 010 Rd Rs1 Rs2 00	OR Rd, Rs1, Rs2	$Rd = Rs1 \text{ OR } Rs2$	Realitza l'operació OR amb el contingut dels registres Rs1 I Rs2 I emmagatzema el resultat en Rd.
	01 011 Rd Rs1 Rs2 00	AND Rd, Rs1, Rs2	$Rd = Rs1 \text{ AND } Rs2$	Realitza l'operació AND amb el contingut dels registres Rs1 I Rs2 I emmagatzema el resultat en Rd.
	01 100 Rd Rs1 Rs2 00	XOR Rd, Rs1, Rs2	$Rd = Rs1 \text{ XOR } Rs2$	Realitza l'operació XOR amb el contingut dels registres Rs1 I Rs2 I emmagatzema el resultat en Rd.

VN - RI - Instruccions aritmetico-lògiques

6

	Codi d'instrucció	Mnemònic	Operació	Descripció
2 operands	01 101Rs1 Rs2 000 00	COMP Rs1, Rs2	Rs1- Rs2	Resta el contingut del registre Rs2 al registre Rs1

VN - RI - Instruccions aritmetico-lògiques

7

	Codi d'instrucció	Mnemònic	Operació	Descripció
1 operand	100 00 Rd/s 00000000	Not Rd/s	$Rd/s = \sim Rd/s$	Realitza l'operació lògica NOT amb els bits del registre Rd/s
	100 01 Rd/s 00000000	INC Rd/s	$Rd/s = Rd/s + 1$	Incrementa el contingut del registre Rd/s en una unitat.
	100 01 Rd/s 00000000	DEC Rd/s	$Rd/s = Rd/s - 1$	Decrementa el contingut del registre Rd/s en una unitat.
	100 11 Rd/s 00000000	NEG Rd/s	$Rd/s = \sim Rd/s + 1$	Canvia de signe (Ca2) el contingut del registre Rd/s

VN - RI - Interrupcions

Codi d'instrucció	Mnemònic	Operació	Descripció
101 00 000000000000	CLI	IF = 0	Col·loca un 0 al flag de màscara d'interrupció
101 01 000000000000	STI	IF = 1	Col·loca un 1 al flag de màscara d'interrupció
101 10 000 Inm_8	INT Inm_8	Pila = SR Pila = PC PC = [Inm_8]	Genera la interrupció software amb vector d'interrupció Inm_8
101 11 000000000000	IRET	PC = Pila SR = Pila	Retorn d'una interrupció

	Codi d'instrucció	Mnemònic	Operació	Descripció
Salts incondicionals	11 00 0 000 Inm_8	JMP Inm_8	$PC = PC + EX_{16}(Inm_8)$	Realitza un salt relatiu.
	11 00 1 Rx	JMP Rx	$PC = Rx$	Realitza un salt indirecte absolut a l'adreça de memòria continguda al registre Rx

	Codi d'instrucció	Mnemònic	Operació	Descripció
Salts condicio nals	11 11 0 cond Inm_8	BR_cond Inm_8	Si condició certa: $PC = PC + Ex_{16}(Inm)$	El desplaçament amb el signe estès es suma a PC sempre i quan la condició codificada al camp Cond sigui certa.

Cond → Tres bits que codifiquen la condició de salt de la següent forma:

000 CF = 1 BRC (bit de Carry)
001 CF = 0 BRNC
010 OF = 1 BRO (bit d'Overflow)
011 OF = 0 BRNO
100 ZF = 1 BRZ (bit de Zero)
101 ZF = 0 BRNZ
110 SF = 1 BRS (bit de Signe)
111 SF = 0 BRNS

	Codi d'instrucció	Mnemònic	Operació	Descripció
Crides a procediments	11 01 0 000 Inm_8	CALL Inm_8	$Pila = PC$ $PC = PC + Ex_{16}(Inm_8)$	Realitza un salt relatiu a un procediment
	11 011 Rx 00000000	JMP Rx	$PC = Rx$	Realitza un salt indirecte absolut a un procediment
	11 10 0 000000000000	RET	$PC = Pila$	Retorna des d'un procediment (desapila PC)