

SPARC V8	Comentarios	Assembler	PC	R[rd]	mem	icc	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	
108	Suma	ADD rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] + r[rs2]cte13			1	0		rd			0	0	0	0	0	0		rs1	i																		
106	Y	AND rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] AND r[rs2]cte13			1	0		rd			0	0	0	0	0	1		rs1	i																		
106	O	OR rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] OR r[rs2]cte13			1	0		rd			0	0	0	0	1	0		rs1	i																		
106	O excluyente	XOR rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] XOR r[rs2]cte13			1	0		rd			0	0	0	0	1	1		rs1	i																		
110	Resta	SUB rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] - r[rs2]cte13			1	0		rd			0	0	0	1	0	0		rs1	i																		
106	Y con rs2cte13 negada	ANDN rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] AND NOT(r[rs2]cte13)			1	0		rd			0	0	0	1	0	1		rs1	i																		
106	O con rs2cte13 negada	ORN rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] OR NOT(r[rs2]cte13)			1	0		rd			0	0	0	1	1	0		rs1	i																		
106	O excluyente negada	XNOR rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] XNOR r[rs2]cte13			1	0		rd			0	0	0	1	1	1		rs1	i																		
108	Suma con acarreo	ADDX rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] + r[rs2]cte13 + c			1	0		rd			0	0	1	0	0	0		rs1	i																		
113	Mult. sin signo	UMUL rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] * r[rs2]cte13			1	0		rd			0	0	1	0	1	0		rs1	i																		
113	Mult. con signo	SMUL rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] * r[rs2]cte13			1	0		rd			0	0	1	0	1	1		rs1	i																		
110	Resta con acarreo	SUBX rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] - r[rs2]cte13 - c			1	0		rd			0	0	1	1	0	0		rs1	i																		
115	Div. sin signo	UDIV rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] / r[rs2]cte13			1	0		rd			0	0	1	1	1	0		rs1	i																		
115	Div. con signo	SDIV rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] / r[rs2]cte13			1	0		rd			0	0	1	1	1	1		rs1	i																		
108	Suma	ADDcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] + r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	0	0	0	0		rs1	i																		
106	Y	ANDcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] AND r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	0	0	0	1		rs1	i																		
106	O	ORcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] OR r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	0	0	1	0		rs1	i																		
106	O excluyente	XORcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] XOR r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	0	0	1	1		rs1	i																		
110	Resta	SUBcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] - r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	0	1	0	0		rs1	i																		
106	Y con rs2cte13 negada	ANDNcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] AND NOT(r[rs2]cte13)	icc<-icc_x		1	0		rd			0	1	0	1	0	1		rs1	i																		
106	O con rs2cte13 negada	ORNcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] OR NOT(r[rs2]cte13)	icc<-icc_x		1	0		rd			0	1	0	1	1	0		rs1	i																		
106	O excluyente negada	XNORcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] XNOR r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	0	1	1	1		rs1	i																		
108	Suma con acarreo	ADDXcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] + r[rs2]cte13 + c	icc<-icc_x		1	0		rd			0	1	1	0	0	0		rs1	i																		
113	Mult. sin signo	UMULcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] * r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	1	0	1	0		rs1	i																		
113	Mult. con signo	SMULcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] * r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	1	0	1	1		rs1	i																		
110	Resta con acarreo	SUBXcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] - r[rs2]cte13 - c	icc<-icc_x		1	0		rd			0	1	1	1	0	0		rs1	i																		
115	Div. sin signo	UDIVcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] / r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	1	1	1	0		rs1	i																		
115	Div. con signo	SDIVcc rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] / r[rs2]cte13	icc<-icc_x		1	0		rd			0	1	1	1	1	1		rs1	i																		
107	Desp. a la izq. (llena con ceros)	SLL rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] << r[rs2]cte13			1	0		rd			1	0	0	1	0	1		rs1	i																		
107	Desp. a la der. (llena con ceros)	SRL rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] >> r[rs2]cte13			1	0		rd			1	0	0	1	1	0		rs1	i																		
107	Desp. a la der. (llena con signo)	SRA rd,rs1,rs2cte13	PC<PC+4	r[rd]<- r[rs1] / (2**r[rs2]cte13)			1	0		rd			1	0	0	1	1	1		rs1	i																		

rs2cte13 es rs2 cuando i=0 sino es cte13

En UMUL y SMUL sólo se usan los primeros 16 bits de rs1 y rs2

SPARC V8	Comentarios	Assembler	PC	R[rd]	mem	acc	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			
90	Carga en reg. dato de MEM	LD rd,[rs1+rs2]	PC<PC+4	R[rd]<m[r[rs1]+r[rs2]]			1	1		rd		0	0	0	0	0	0				rs1		0	0	0	0	1	0	1	0								rs2			
90	Carga en reg. dato de MEM	LD rd,[rs1+cte13]	PC<PC+4	r[rd]<m[r[rs1]+cte13]			1	1		rd		0	0	0	0	0	0				rs1		1															cte13			
95	Almacena el reg. en la MEM	ST [rs1+rs2],rd	PC<PC+4	m[r[rs1]+r[rs2]]<r[rd]			1	1		rd		0	0	0	1	0	0				rs1		0	0	0	0	1	0	1	0								rs2			
95	Almacena el reg. en la MEM	ST [rs1+cte13],rd	PC<PC+4	m[r[rs1]+cte13]<r[rd]			1	1		rd		0	0	0	1	0	0				rs1		1															cte13			
119	No brinca	BN cte22	PC<4*cte22 + PC WHEN '0'='1' ELSE PC+4				0	0	0	0	0	0	0	1	0																							cte22			
119	Brinca si son iguales	BE cte22	PC<4*cte22 + PC WHEN z='1' ELSE PC+4				0	0	0	0	0	0	1	0	1	0																						cte22			
119	Brinca si es menor o igual (S)	BLE cte22	PC<4*cte22 + PC WHEN zOR(nXORv)='1' ELSE PC+4				0	0	0	0	1	0	0	0	1	0																						cte22			
119	Brinca si es menor (S)	BL cte22	PC<4*cte22 + PC WHEN nXORv='1' ELSE PC+4				0	0	0	0	0	1	1	0	1	0																						cte22			
119	Brinca si es menor o igual (U)	BLEU cte22	PC<4*cte22 + PC WHEN cORz='1' ELSE PC+4				0	0	0	0	1	0	0	0	1	0																						cte22			
119	Brinca si c es uno (menor (U))	BCE cte22	PC<4*cte22 + PC WHEN c='1' ELSE PC+4				0	0	0	0	1	0	1	0	1	0																						cte22			
119	Brinca si es negativo	BNEG cte22	PC<4*cte22 + PC WHEN n='1' ELSE PC+4				0	0	0	0	1	1	0	0	1	0																						cte22			
119	Brinca si v es uno	BVS cte22	PC<4*cte22 + PC WHEN v='1' ELSE PC+4				0	0	0	0	1	1	1	0	1	0																						cte22			
119	Brinca siempre	BA cte22	PC<4*cte22 + PC WHEN '1'='1' ELSE PC+4				0	0	0	1	0	0	0	0	1	0																						cte22			
119	Brinca si no son iguales	BNE cte22	PC<4*cte22 + PC WHEN NOTz='1' ELSE PC+4				0	0	0	1	0	0	1	0	1	0																						cte22			
119	Brinca si es mayor (S)	BG cte22	PC<4*cte22 + PC WHEN NOT(zOR(nXORv))='1' ELSE PC+4				0	0	0	1	0	1	0	0	1	0																						cte22			
119	Brinca si es mayor o igual (S)	BGE cte22	PC<4*cte22 + PC WHEN NOT(nXORv)='1' ELSE PC+4				0	0	0	1	0	1	1	0	1	0																						cte22			
119	Brinca si es mayor (U)	BGU cte22	PC<4*cte22 + PC WHEN NOT(cORz)='1' ELSE PC+4				0	0	0	1	0	1	0	0	1																							cte22			
119	Brinca si c es cero (may,Igu (U))	BCC cte22	PC<4*cte22 + PC WHEN NOTc='1' ELSE PC+4				0	0	0	1	0	1	0	1	0																							cte22			
119	Brinca si es positivo	BPOS cte22	PC<4*cte22 + PC WHEN NOTn='1' ELSE PC+4				0	0	0	1	1	1	0	0	1	0																						cte22			
119	Brinca si v es cero	BVC cte22	PC<4*cte22 + PC WHEN NOTv='1' ELSE PC+4				0	0	0	1	1	1	1	0	1	0																						cte22			
125	Llamado a rutina	CALL cte30	PC<PC+4*cte30	%o7<PC			0	1																													cte30				
85	Retorno de rutina	RETL	PC<%o7+8				1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0		
126	Salto con reg.	JMPL rd,rs1+rs2	PC<r[rs1]+r[rs2]	r[rd]<PC			1	0		rd		1	1	1	0	0	0				rs1		0	0	0	0	0	0	0	0	0						rs2				
126	Salto con cte.	JMPL rd,rs1+cte13	PC<r[rs1]+cte13	r[rd]<PC			1	0		rd		1	1	1	0	0	0				rs1		1														cte13				
rs2cte13 es rs2 cuando i=0 sino es cte13																																									
Las instrucciones con * requieren multiplicar por 4 la dirección para mantener la compatibilidad con el SPARC V8																																									
Carga cte en parte alta																																									
104	SETHI rd,cte22		PC<PC+4	R[rd]<cte<<10				0	0		rd	1			0	0																									cte22
Para mantener compatibilidad del programa hay que colocar un NOP después de cada salto o brinco																																									

Sinónimos

No hace operaciones	NOP	ADD R0,R0,R0
Mueve	MOV rd,rs2cte13	ADD rd,R0,rs2cte13
Compara mayor o menor	CMP rs1,rs2cte13	SUBcc R0,rs1,rs2cte13
Compara con cero	TST rs2	ORcc R0,R0,rs2
Invertir los bits	NOT rd,rs1	XNOR rd,rs1,R0
Complemento a 2	NEG rd,rs2	SUBB rd,R0,rs2
Incrementa en uno	INC rd,rs1	ADD rd,rs1,1
Decrementa en 1	DEC rd,rs1	SUBB rd,rs1,1
Borra un registro	CLR rd	OR rd,R0,R0
Brinca si está n	BnS cte22	BNEG cte22
Brinca si no está n	BnC cte22	BPOS cte22
Brinca si está z	BzS cte22	BE cte22
Brinca si no está z	BzC cte22	BNE cte22
Brinca si está c	BcS cte22	BCS cte22
Brinca si no está c	BcC cte22	BCC cte22
Brinca si está v	BvS cte22	BVS cte22
Brinca si no está v	BvC cte22	BVC cte22

