

ROC32_8 operator list

James C Brakefield © 2020

register zero used to hold return address
register zero as index register reads as zero

five bit immediate field is value of -8..15 or length - 1 of postfixd immediate in bytes
last four registers are condition code, frame pointer, stack pointer & PC

format name	eventual layout	usage	address arithmetic	instruction fields
fmt24drsi	00xxxxxx ddddd i rrrr sssss	basic 24-bit instruction		b base address register #
fmt24drs	00xxxxxx ddddd 0 rrrr sssss	R op S ->D		c condition code predicate (execute instruction if condition true)
fmt24dri	00xxxxxx ddddd 1 rrrr nnnnn	R op immediate ->D		d result register #
fmt24dbx	00xxxxxx ddddd 0 bbbbb jjjjj	load/store base + scaled index	B + J * ds	ds data size in bytes
fmt24dbn	00xxxxxx ddddd 1 bbbbb nnnnn	load/store base + offset	B + N	i s is immediate field if i = 1
fmt24dr	00xxxxxx ddddd z rrrr xxxxx	op R -> D; single operand instructions		j register # for scaled index
fmt24dn	00xxxxxx ddddd nnnnnnnnnnn	eleven bit signed offset or immediate	PC + N	m small constant or length of following immediate for store immediate value
				n small constant or length of following immediate
				r operand register #
				s operand register #
				u update CC enable, for ST instructions is store immediate enable
fmt32drsc	01xxxxxx ddddd i rrrr sssss zz u ccccc	basic 32-bit instruction		v swap operands enable
fmt32drs	01xxxxxx ddddd 0 rrrr sssss zz u ccccc	R op S ->D; execute on predicate match		x op-code bits
fmt32drnc	01xxxxxx ddddd 1 rrrr nnnnn z v u ccccc	R op imm ->D; execute on predicate match	B + J * Y + S	y 2-bit code for: ds * (1..4)
fmt32dbjs	01xxxxxx ddddd 0 bbbbb sssss yy u jjjjj	load from base + index * scaling + offset	B + J * Y + N	z negate/invert enables for one or two operands
fmt32dbjn	01xxxxxx ddddd 1 bbbbb nnnnn yy u jjjjj	load from base + index * scaling + offset	B + J * Y + N	
fmt32dbjs	01xxxxxx ddddd 0 bbbbb sssss yy 0 jjjjj	store to base + index * scaling + offset	B + J * Y + S	
fmt32dbjn	01xxxxxx ddddd 1 bbbbb nnnnn yy 0 jjjjj	store to base + scaled index + offset	B + J * Y + N	
fmt32mbjs	01xxxxxx mmmmm 0 bbbbb sssss yy 1 jjjjj	store immediate to base + index * scaling + offset	B + J * Y + S	
fmt32mbjn	01xxxxxx mmmmm 1 bbbbb nnnnn yy 1 jjjjj	store immediate to base + index * scaling + offset	B + J * Y + N	
fmt32dr	00xxxxxx ddddd z rrrr xxxxx xx u ccccc	op R -> D; single operand instructions		
fmt32dn	01xxxxxx ddddd nnnnnnnnnnnnnnnnn	nineteen bit signed offset or immediate	PC + N	
fmt40drsc	10xxxxxx ddddd i rrrr sssss zzuccccc xxxtttt		basic 40-bit instruction with three source registers	
fmt48drsc	11xxxxxx ddddd i rrrr sssss zzuccccc xxxtttt xxeeeee		basic 48-bit instruction with 3 source, 2 destination & predication	
fmt48drscw	11xxxxxx ddddd i rrrr sssss zzuccccc xwwttttt wwweeeee		basic 48-bit instruction with 4 source, 2 destination & predication	

Some 32-bit instructions support three operand operations
40-bit load/store instructions support predication
40-bit instructions support three operands and predication
40 and 48-bit instructions support CARRY mechanism: register field(s) for extended operand and extended result