

ROC32_8 operator list

James C Brakefield © 2020

register zero used to hold return address		five bit immediate field is value of -8..15 or length - 1 of postfixed immediate in bytes			
register zero as index register reads as zero		last four registers are condition code, frame pointer, stack pointer & PC			
Y scale factor is data size times (1 to 4)		combined interger & floating-point register file			
format name	eventual layout	usage	address arithmetic	instruction fields	
fmt24drsi	00xxxxxx dddd i rrrrr sssss	basic 24-bit instruction		b	base address register #
fmt24drs	00xxxxxx dddd 0 rrrrr sssss	R op S ->D		c	condition code predicate (execute instruction if condition true)
fmt24dri	00xxxxxx dddd 1 rrrrr nnnnn	R op immediate ->D		d	result register #
fmt24dbx	00xxxxxx dddd 0 bbbbb jjjjj	load/store base + scaled index	B + J * ds	ds	data size in bytes
fmt24dbn	00xxxxxx dddd 1 bbbbb nnnnn	load/store base + offset	B + N	e	2nd result register #
fmt24dr	00xxxxxx dddd z rrrrr xxxxx	op R -> D; 32 single operand instructions		i	s is immediate field if i = 1
fmt24dn	00xxxxxx dddd n nnnnn nnnnn	eleven bit signed offset or immediate	PC + N	j	register # for scaled index
				m	small constant or length of following immediate for store immediate value
fmt32drsc	01xxxxxx dddd i rrrrr sssss zzu ccccc	basic 32-bit instruction		n	small constant or length of following immediate
fmt32drsc	01xxxxxx dddd 0 rrrrr sssss zzu ccccc	R op S ->D; execute on predicate match		r	operand register #
fmt32drnc	01xxxxxx dddd 1 rrrrr nnnnn zvu ccccc	R op imm ->D; execute on predicate match		s	2nd operand register #
fmt32dbjs	01xxxxxx dddd 0 bbbbb sssss yyu jjjjj	load from base + index * scaling + offset	B + J * Y + S	t	3rd operand register #
fmt32dbjn	01xxxxxx dddd 1 bbbbb nnnnn yyu jjjjj	load from base + index * scaling + offset	B + J * Y + N	u	update CC enable, for ST instructions is store immediate enable
fmt32dbjs	01xxxxxx dddd 0 bbbbb sssss yy0 jjjjj	store to base + index * scaling + offset	B + J * Y + S	v	swap operands enable
fmt32dbjn	01xxxxxx dddd 1 bbbbb nnnnn yy0 jjjjj	store to base + scaled index + offset	B + J * Y + N	w	4th operand register #
fmt32mbjs	01xxxxxx mmmmm 0 bbbbb sssss yy1 jjjjj	store immediate to base + index * scaling + offset	B + J * Y + S	x	op-code bits
fmt32mbjn	01xxxxxx mmmmm 1 bbbbb nnnnn yy1 jjjjj	store immediate to base + index * scaling + offset	B + J * Y + N	y	2-bit code for: ds * (1..4)
fmt32dr	00xxxxxx dddd z rrrrr xxxxx xzu ccccc	op R -> D; 128 single operand instructions		z	negate/invert enables for one or two operands
fmt32drst	01xxxxxx dddd i rrrrr sssss zzu ttttt	op(R, S, T) -> D; triple operand instructions			
fmt32dn	01xxxxxx dddd n nnnnn nnnnn nnn nnnnn	nineteen bit signed offset or immediate	PC + N		
fmt40drsc	10xxxxxx dddd i rrrrr sssss zzu ccccc xxx ttttt	basic 40-bit instruction with three source registers & predication			
fmt48drsc	11xxxxxx dddd i rrrrr sssss zzu ccccc xxx ttttt xxx eeeee	basic 48-bit instruction with 3 source, 2 destination & predication			
fmt48drsc	11xxxxxx dddd i rrrrr sssss zzu ccccc xww ttttt www eeeee	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				