ROC24_3sz operator list **James C Brakefield © 2023**

register zero used to hold return address		five bit immediate field is value of -4..7, length - 1 of postfixed immediate or first 4 bits of 12 bit signed value					
register zero as index register reads as zero		last four registers are residue, frame pointer, stack pointer & PC					
Y scale factor is data size times (1 to 4)		each register has a 4-bit type code: unsigned, signed, float & Posit (28-bits total)					
format name	**eventual layout (little endian)**	**usage**	**address arithmetic**		**instruction fields**		
fmt24drsi	sssss rrrrr i ddddd 00xxxxxx	basic 24-bit instruction		**a**	type code		
fmt24drs	sssss rrrrr 0 ddddd 00xxxxxx	R op S ->D		**b**	base address register #		
fmt24dri	nnnnn rrrrr 1 ddddd 00xxxxxx	R op immediate ->D		**c**	condition code predicate (execute instruction if condition true)		
fmt24dbx	jjjjj bbbbb 0 ddddd 00xxxxxx	load/store base + scaled index	B + J * ds	**d**	result register #		
fmt24dbn	nnnnn bbbbb 1 ddddd 00xxxxxx	load/store base + offset	B + N	**ds**	data size in bytes		
fmt24dr	xxxxx rrrrr z ddddd 00xxxxxx	op R -> D; 32 single operand instructions		**e**	2nd result register #		
fmt24dn	nnnnn nnnnn n ddddd 00xxxxxx	eleven bit signed offset or displacement	PC + N	**i**	s is immediate field if i = 1		
fmt24dan	nnnnn nnnaa x ddddd 00xxxxxx	eight bit immediate with type code		**j**	register # for scaled index		
				m	small constant or length of following immediate for store immediate value		
fmt32drsc	ttttt zzu sssss rrrrr i ddddd 01xxxxxx	basic 32-bit instruction		**n**	small constant or length of following immediate		
fmt32drsc	ccccc zzu sssss rrrrr i ddddd 01xxxxxx	R op S ->D; execute on predicate match		**r**	operand register #		
fmt32drnc	ccccc zzu nnnnn rrrrr 1 ddddd 01xxxxxx	R op imm ->D; execute on predicate match		**s**	2nd operand register #		
fmt32dbjs	jjjjj yyu sssss bbbbb i ddddd 01xxxxxx	load from base + index * scaling + offset	B + J * Y + S	**t**	3rd operand register #		
fmt32dbjn	jjjjj yyu nnnnn bbbbb 1 ddddd 01xxxxxx	load from base + index * scaling + offset	B + J * Y + N	**u**	update CC enable, for ST instructions is store immediate enable		
fmt32dbjs	jjjjj yy0 sssss bbbbb 0 ddddd 01xxxxxx	store to base + index * scaling + offset	B + J * Y + S	**v**	swap operands enable		
fmt32dbjn	jjjjj yy0 nnnnn bbbbb 1 ddddd 01xxxxxx	store to base + scaled index + offset	B + J * Y + N	**w**	4th operand register #		
fmt32mbjs	jjjjj yy1 sssss bbbbb 0 ddddd 01xxxxxx	store immediate to base + index * scaling + offset	B + J * Y + S	**x**	op-code bits		
fmt32mbjn	jjjjj yy1 nnnnn bbbbb 1 ddddd 01xxxxxx	store immediate to base + index * scaling + offset	B + J * Y + N	**y**	2-bit code for: ds * (1..4)		
fmt32dr	ccccc xxu xxxxx rrrrr z ddddd 01xxxxxx	op R -> D; 128 single operand instructions		**z**	negate/invert enables for one or two operands		
fmt32drst	ttttt zzu sssss rrrrr i ddddd 01xxxxxx	op(R, S, T) -> D; triple operand instructions		**k**	alternate immediate format: -4..4 or 12-bit or 1..3 byte immediate, useful for 12-bit ir		
fmt32dn	nnnnn nnn nnnnn nnnnn n ddddd 01xxxxxx	nineteen bit signed offset or immediate	PC + N				
fmt32dan	nnnnn nnn nnnnn nnnaa x ddddd 01xxxxxx	sixteen bit immediate with type code					
fmt40drsct	ttttt xxx ccccc xxu sssss rrrrr i ddddd 10xxxxxx		basic 40-bit instruction with three source registers & predication				
fmt48drscte	ddddd i rrrrr sssss zzu ccccc xxx ttttt xxx eeeee 11xxxxxx		basic 48-bit instruction with 3 source, 2 destination & predication				
fmt48drsctwe	ddddd i rrrrr sssss zzu ccccc xww ttttt www eeeee 11xxxxxx		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						