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 postfixed immediate in bytes
last four registers are condition code, frame pointer, stack pointer & PC

format name	**eventual layout**	**usage**	**address arithmetic**
fmt24drsi	00xxxxxx ddddd i rrrrr sssss	basic 24-bit instruction	
fmt24drs	00xxxxxx ddddd 0 rrrrr sssss	R op S ->D	
fmt24dri	00xxxxxx ddddd 1 rrrrr nnnnn	R op immediate ->D	
fmt24dbx	00xxxxxx ddddd 0 bbbbb jjjjj	load/store base + scaled index	B + J * ds
fmt24dbn	00xxxxxx ddddd 1 bbbbb nnnnn	load/store base + offset	B + N
fmt24dr	00xxxxxx ddddd z rrrrr xxxxx	op R -> D; single operand instructions	
fmt24dn	00xxxxxx ddddd nnnnnnnnnnn	eleven bit signed offset or immediate	PC + N
fmt32drsc	01xxxxxx ddddd i rrrrr sssss zz u ccccc	basic 32-bit instruction	
fmt32drsc	01xxxxxx ddddd 0 rrrrr sssss zz u ccccc	R op S ->D; execute on predicate match	
fmt32drnc	01xxxxxx ddddd 1 rrrrr nnnnn z v u ccccc	R op imm ->D; execute on predicate match	
fmt32dbjs	01xxxxxx ddddd 0 bbbbb sssss yy u jjjjj	load from base + index * scaling + offset	B + J * Y + S
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 rrrrr xxxxx - - u ccccc	op R -> D; single operand instructions	
fmt32dn	01xxxxxx ddddd nnnnnnnnnnnnnnnnnnn	nineteen bit signed offset or immediate	PC + N

instruction fields	
b	base address register #
c	condition code predicate (execute instruction if condition true)
d	result register #
ds	data size in bytes
i	s is immediate field if i = 1
j	register # for scaled index
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
v	swap operands enable
x	op-code bits
y	2-bit code for: ds * (1..4)
z	negate/invert enables for one or two operands

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