		OP	**rrmov/cmovXX**	**irmov**	**rmmov**	**mrmov**	**jXX**	**call**	**push**	**ret**	**pop**	**iadd**	**leave**
		rA, rB	**rA, rB**	**V, rB**	**rA, D(rB)**	**D(rB), rA**	**Dest**	**Dest**	**rA**		**rA**	**V, rB**	
Fetch	icode:ifun	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]	M_1[PC]
	rA:rB	M_1[PC+1]	M_1[PC+1]	M_1[PC+1]	M_1[PC+1]	M_1[PC+1]			M_1[PC+1]		M_1[PC+1]	M_1[PC+1]	
	valC			M_8[PC+2]	M_8[PC+2]	M_8[PC+2]	M_8[PC+2]	M_8[PC+2]				M_8[PC+2]	
	valP	PC+2	PC+2	PC+10	PC+10	PC+10	PC+9	PC+9	PC+2	PC+1	PC+2	PC+10	PC+1
Decode	valA	R[rA]	R[rA]		R[rA]	R[rA]			R[rA]	R[%rsp]	R[%rsp]		R[%rbp]
	valB	R[rB]			R[rB]	R[rB]		R[%rsp]	R[%rsp]	R[%rsp]	R[%rsp]	R[rB]	R[%rbp]
Execute	valE	valB OP valA		0+valC	valB+valC	valB+valC		valB+(–8)	valB+(–8)	valB+8	valB+8	valB+valC	valB+8
	cnd		Cond(CC, ifun)				Cond(CC, ifun)						
Memory	M_8[valE]				valA			valP	valA				
	valM					M_8[valE]				M_8[valA]	M_8[valA]		M_8[valA]
Writeback	R[%rsp]							valE	valE	valE	valE		valE
	R[rA]					valM					valM		
	R[rB]	ValE	valE if cnd	valE								valE	R[%rbp] = valM
PC Updt.	PC	valP	valP	valP	valP	valP	cnd?valC:valP	valC	valP	valM	valP	valP	valP