Instruction	Details	State	IR_enable	RF_enable	MAR_enable	MDR_enable	MDV	PC_enable	nPC_enable	PSR_enable	TBR_enable	WIM_enable	RF_Mux_C_select	RF_Mux_A_select	ALU_Mux_A_select	ALU_Mux_B_select	ALU_Mux_OP_select	MDR_Mux_select	PC_Mux_select	PSR_Mux_select	TBR_Mux_select	RAM_Mux_Op_select	SSE_select	CT_select	OpS	Cr	N2	N1	N0	Sts	Inv	S1	S0	PLS-PUD
Reset	PC <= 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	x	x	x	x	8'b00000001
	nPC <= 4	1	0	0	0	0	0	0	1	0	0	0	0	0	3'b101	3'b011	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	x	x	x	x	x
Fetch	MAR <= PC	3	0	0	1	0	0	0	0	0	0	0	0	0	3'b001	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Enable RAM	4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Enable IR (Waiting for MOC)	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	8'b00001001
	Disable IR; Check conditions if any	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	8'b01010011
Sethi	Enable RF	7	0	1	0	0	0	0	0	0	0	0	2'b00	0	3'b101	3'b001	0	0	0	0	0	0	2'b01	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable RF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	x	x	x	x	8'b01010011
Branch	Prepare for branch	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	8'b00011100
BX True	PC <= nPC & nPC <= PC + 4*disp22	10	0	0	0	0	0	1	1	0	0	0	0	0	3'b001	3'b001	0	0	0	0	0	0	2'b10	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011
BX False	PC <= nPC & nPC = nPC + 4	12	0	0	0	0	0	1	1	0	0	0	0	0	3'b010	3'b011	0	0	0	0	0	0	2'b00	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011
Branch Anulled	Prepare for branch annulled	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0	1	8'b00001010
BX False Annulled	PC <= nPC + 4	15	0	0	0	0	0	1	0	0	0	0	0	0	3'b010	3'b011	0	0	0	0	0	0	2'b01	0	6'b000000	0	0	1	1	x	x	x	x	x
	nPC = PC + 4 (nPC <= nPC + 8)	16	0	0	0	0	0	0	1	0	0	0	0	0	3'b001	3'b011	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011
Call	r[15] <= PC	18	0	1	0	0	0	0	0	0	0	0	2'b01	0	3'b001	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	PC <= nPC & nPC <= PC + 4*disp30	19	0	0	0	0	0	1	1	0	0	0	0	0	3'b001	3'b001	0	0	2'b00	0	0	0	2'b11	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011
Jmpl	r[rd] <= PC	21	0	1	0	0	0	0	0	0	0	0	2'b00	0	3'b001	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	PC <= nPC	22	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2'b00	0	0	0	2'b00	0	0	0	0	1	1	0	1	1	0	8'b00011000
	If r[63] == 0: nPC <= r[rs1] + r[rs2]	23	0	0	0	0	0	0	1	0	0	0	0	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	0	x	x	x	x	8'b00011001
	If r[63] == 1: nPC <= r[rs1] + simm13	24	0	0	0	0	0	0	1	0	0	0	0	2'b00	3'b000	3'b001	0	0	0	0	0	0	0	2'b00	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011
Save	If r[63] == 0: r[rd] <= r[rs1] + r[rs2]	26	0	1	0	0	0	0	0	0	0	0	2'b00	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	0	x	x	x	x	8'b00011100
	If r[63] == 1: r[rd] <= r[rs1] + simm13	27	0	1	0	0	0	0	0	0	0	0	2'b00	2'b00	3'b000	3'b001	0	0	0	0	0	0	0	2'b00	6'b000000	0	0	1	1	x	x	x	x	x
	CWP <= CWP - 1	28	0	0	0	0	0	0	1	0	0	0	0	0	3'b011	3'b100	0	0	0	3'b100	0	0	0	0	6'b000100	0	0	1	1	x	x	x	x	x
	Disable PSR	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
Restore	If r[63] == 0: r[rd] <= r[rs1] + r[rs2]	30	0	1	0	0	0	0	0	0	0	0	2'b00	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	If r[63] == 1: r[rd] <= r[rs1] + simm13	31	0	1	0	0	0	0	0	0	0	0	2'b00	2'b00	3'b000	3'b001	0	0	0	0	0	0	0	2'b00	6'b000000	0	0	1	1	x	x	x	x	x
	CWP <= CWP + 1	32	0	0	0	0	0	0	1	0	0	0	0	0	3'b011	3'b100	0	0	0	3'b100	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable PSR	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
Arithmetical/Logical	If r[63] == 0 & IR[23] == 0: r[rd] <= r[rs1] (operand) r[rs2]	34	0	1	0	0	0	0	0	0	0	0	2'b00	2'b00	3'b000	3'b000	1	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00100110
	If r[63] == 1 & IR[23] == 0: r[rd] <= r[rs1] (operand) simm13	35	0	1	0	0	0	0	0	0	0	0	2'b00	2'b00	3'b000	3'b001	1	0	0	0	0	0	0	2'b00	0	0	0	1	0	x	x	x	x	8'b00100110
	If r[63] == 0 & IR[23] == 1: r[rd] <= r[rs1] (operand) r[rs2]	36	0	1	0	0	0	0	1	0	0	0	2'b00	2'b00	3'b000	3'b000	1	0	0	3'b000	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00100110
	If r[63] == 1 & IR[23] == 1: r[rd] <= r[rs1] (operand) simm13	37	0	1	0	0	0	0	1	0	0	0	2'b00	2'b00	3'b000	3'b001	1	0	0	3'b000	0	0	2'b00	0	0	0	0	1	1	x	x	x	x	x
	Disable RF; Disable PSR	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
Load	If r[63] == 0: MAR <= r[rs1] + r[rs2]	39	0	0	1	0	0	0	0	0	0	0	0	2'b00	3'b000	3'b000	0	0	0	0	0	1	0	0	6'b000000	0	0	1	1	x	x	x	x	8'b00101001
	If r[63] == 1: MAR <= r[rs1] + simm13	40	0	0	1	0	0	0	0	0	0	0	0	2'b00	3'b000	3'b001	0	0	0	0	0	1	0	2'b00	6'b000000	0	0	1	1	x	x	x	x	x
	Enable RAM; MDR <= RAM_Data_Out	41	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	8'b00101001
	r[rd] <= MDR_Out	42	0	1	0	0	0	0	0	0	0	0	2'b00	0	3'b101	3'b010	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable RF	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
Store	If r[63] == 0: MAR <= r[rs1] + r[rs2]	44	0	0	1	0	0	0	0	0	0	0	0	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00101110
	If r[63] == 1: MAR <= r[rs1] + simm13	45	0	0	1	0	0	0	0	0	0	0	0	2'b00	3'b000	3'b001	0	0	0	0	0	0	0	2'b00	6'b000000	0	0	1	1	x	x	x	x	x
	MDR <= r[rd]	46	0	0	0	1	0	0	0	0	0	0	0	2'b01	3'b000	3'b101	0	1	0	0	0	1	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Enable RAM	47	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	8'b00101111
	Disable RAM	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
RDPSR	r[rd] <= PSR	49	0	1	0	0	0	0	0	0	0	0	2'b00	0	3'b011	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable RF	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
RDWIM	r[rd] <= WIM	51	0	1	0	0	0	0	0	0	0	0	2'b00	0	3'b110	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable RF	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
RDTBR	r[rd] <= TBR	53	0	1	0	0	0	0	0	0	0	0	2'b00	0	3'b100	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable RF	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
WRPSR	If r[13] == 0: PSR <= r[rs1] xor r[rs2]	55	0	0	0	0	0	0	1	0	0	0	0	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	6'b000011	0	0	1	1	x	x	x	x	8'b00111001
	If r[13] == 1: PSR <= r[rs1] xor simm13	56	0	0	0	0	0	0	1	0	0	0	0	2'b00	3'b000	3'b001	0	0	0	0	0	0	0	2'b00	6'b000011	0	0	1	1	x	x	x	x	x
	Disable PSR	57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
WRWIM	If r[13] == 0: WIM <= r[rs1] xor r[rs2]	58	0	0	0	0	0	0	0	0	1	0	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	0	6'b000011	0	0	1	1	x	x	x	x	8'b00111100
	If r[13] == 1: WIM <= r[rs1] xor simm13	59	0	0	0	0	0	0	0	0	1	0	2'b00	3'b000	3'b000	3'b001	0	0	0	0	0	0	0	2'b00	6'b000011	0	0	1	1	x	x	x	x	x
	Disable WIM	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
WRTBR	If r[13] == 0: TBR <= r[rs1] xor r[rs2]	61	0	0	0	0	0	0	0	1	0	0	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	0	6'b000011	0	0	1	1	x	x	x	x	8'b00111111
	If r[13] == 1: TBR <= r[rs1] xor simm13	62	0	0	0	0	0	0	0	1	0	0	2'b00	3'b000	3'b001	0	0	0	0	0	0	0	0	2'b00	6'b000011	0	0	1	1	x	x	x	x	x
	Disable TBR	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b01010011
Trap Priority	Check Trap Priority	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	x
Ticc	If r[63] == 0: TBR <= r[rs1] + r[rs2]	65	0	0	0	0	0	0	0	1	0	0	2'b00	3'b000	3'b000	0	0	0	0	0	1	0	0	0	6'b000000	0	0	1	1	x	x	x	x	8'b00100011
	If r[63] == 1: TBR <= r[rs1] + simm13	66	0	0	0	0	0	0	0	1	0	0	2'b00	3'b000	3'b001	0	0	0	0	0	1	0	0	2'b00	6'b000000	0	0	1	1	x	x	x	x	x
	ET <= 0	67	0	0	0	0	0	0	1	0	0	0	0	0	3'b101	3'b101	0	0	0	0	0	0	3'b011	0	6'b000000	0	0	1	1	x	x	x	x	x
	PS <= S	68	0	0	0	0	0	0	1	0	0	0	0	0	3'b011	3'b101	0	0	0	0	0	0	3'b010	0	6'b000000	0	0	1	1	x	x	x	x	x
	S <= 1	69	0	0	0	0	0	0	1	0	0	0	0	0	3'b111	3'b110	0	0	0	0	0	0	3'b001	0	6'b010101	0	0	1	1	x	x	x	x	x
	CWP <= CWP - 1	70	0	0	0	0	0	0	1	0	0	0	0	0	3'b011	3'b100	0	0	0	0	0	0	3'b100	0	6'b000100	0	0	1	1	x	x	x	x	x
	r[17] <= PC	71	0	1	0	0	0	0	0	0	0	0	2'b10	0	3'b001	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	r[18] <= nPC	72	0	1	0	0	0	0	0	0	0	0	2'b11	0	3'b010	3'b101	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	PC <= TBR	73	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2'b10	0	0	0	0	1	1	x	x	x	x	x
	nPC <= TBR + 4	74	0	0	0	0	0	1	0	0	0	0	0	0	3'b011	3'b100	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011
rett	CWP <= CWP + 1	76	0	0	0	0	0	0	1	0	0	0	0	0	3'b011	3'b100	0	0	0	3'b100	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	PC <= nPC	77	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2'b00	0	0	0	0	0	0	0	1	1	0	1	1	1	0	8'b01100100
	If r[63] == 0: nPC <= r[rs1] + r[rs2]	78	0	0	0	0	0	0	1	0	0	0	0	2'b00	3'b000	3'b000	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	0	x	x	x	x	8'b01010000
	If r[63] == 1: nPC <= r[rs1] + simm13	79	0	0	0	0	0	0	1	0	0	0	0	2'b00	3'b000	3'b001	0	0	0	0	0	0	0	2'b00	6'b000000	0	0	1	1	0	x	x	x	x	x
	S <= PS	80	0	0	0	0	0	0	0	1	0	0	0	0	0	3'b011	3'b100	0	0	0	3'b001	0	0	0	0	3'b100101	0	0	1	1	x	x	x	x	x
	ET <= 1	81	0	0	0	0	0	0	1	0	0	0	0	0	3'b101	3'b100	0	0	0	0	3'b011	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable PSR	82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011
Instruction Flow	PC <= nPC	83	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2'b00	0	0	0	0	0	0	0	0	1	1	x	x	x	x	x
	nPC <= nPC + 4	84	0	0	0	0	0	0	1	0	0	0	0	0	3'b010	3'b011	0	0	0	0	0	0	0	0	6'b000000	0	0	1	1	x	x	x	x	x
	Disable nPC	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	8'b00000011

	S1	S0
MOC	0	0
CONS	0	1
IR[13]	1	0
IR[23]	1	1

N2	N1	N0	Sts	M1	M2	
0	0	0	X	0	0	Enconder
0	0	1	X	0	1	?
0	1	0	X	1	0	Control Register
0	1	1	x	1	1	Incrementer
1	0	0	0	0	0	Enconder
1	0	0	1	1	0	Control Register
1	0	1	0	0	0	Enconder
1	0	1	1	1	0	Incrementer
1	1	0	0	1	0	Control Register
1	1	0	1	1	1	Incrementer