Category	Instruction	Operands	Operation	Opcode	Fields
No-operation	nop	N/A	none	00 00 00	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ
Arithmetic	add	rt,ra,rb	rt <= ra + rb	00 10 00	ΦΦ ΦΦΦΦ ΦΦΦΦ BBBB BAAA AAΦT TTTT
	sub	rt,ra,rb	rt <= ra – rb	00 01 00	ΦΦ ΦΦΦΦ ΦΦΦΦ BBBB BAAA AAΦT TTTT
	addi	rt,ra,imm	rt <= ra + immediate value	00 10 01	II IIII IIII IIII ΦAAA AAΦT TTTT
	subi	rt,ra,imm	rt <= ra – immediate value	00 01 01	II IIII IIII IIII ΦAAA AAΦT TTTT
	inc	rt,ra	rt <= ra + 1	00 10 10	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	dec	rt,ra	rt <= ra – 1	00 01 10	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
Logic	not	rt,ra	rt <= NOT ra	01 11 11	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	and	rt,ra,rb	rt <= ra AND rb	01 01 00	ΦΦ ΦΦΦΦ ΦΦΦΦ BBBB BAAA AAΦT TTTT
	or	rt,ra,rb	rt <= ra OR rb	01 10 00	ΦΦ ΦΦΦΦ ΦΦΦΦ BBBB BAAA AAΦT TTTT
	xor	rt,ra,rb	rt <= ra XOR rb	01 11 00	ΦΦ ΦΦΦΦ ΦΦΦΦ BBBB BAAA AAΦT TTTT
	andi	rt,ra,imm	m rt <= ra AND immediate value	01 01 01	II IIII IIII IIII ΦAAA AAΦT TTTT
	ori	rt,ra,imm	rt <= ra OR immediate value	01 10 01	II IIII IIII IIII ΦAAA AAΦT TTTT
	xori	rt,ra,imm	rt <= ra XOR immediate value	01 11 01	II IIII IIII IIII ΦAAA AAΦT TTTT
	shl	rt,ra,n	rt <= ra shifted left by n bits	01 00 01	ΦΦ NNNN ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	shr	rt,ra,n	rt <= ra shifted right by n bits	01 00 00	ΦΦ NNNN ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	rol	rt,ra,n	n rt <= ra rotated left by n bits	01 00 11	ΦΦ NNNN ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	ror	rt,ra,n	rt <= ra rotated right by n bits	01 00 10	ΦΦ NNNN ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
Transfer	move	rt,ra	rt <= ra	10 00 00	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	loadi	rt,addr	rt <= DMEM[addr] {direct addressing}	10 01 01	AA AAAA AAAA AAAA AAAA AAΦT TTTT
	loadr	rt,ra	rt <= DMEM[ra] {register indirect addressing}	10 01 10	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	loado	rt,ra,off	rt <= DMEM[ra+off] {base plus offset addressing}	10 01 11	ΦΦ ΦΦΦO OOOO OOOO ΦAAA AAΦT TTTT
	stori	rt,addr	DMEM[addr] <= rb {direct addressing}	10 10 01	AA AAAA AAAA AAAA AAAA AAΦT TTTT
	storr	rt,ra	DMEM[ra] <= rb {register indirect addressing}	10 10 10	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
	storo	rt,ra,off	DMEM[ra+off] <= rb {base plus offset addressing}	10 10 11	ΦΦ ΦΦΦO OOOO OOOO ΦAAA AAΦT TTTT
control	jmp	off	Jump to IMEM[PC+off]	11 00 00	ΦΦ ΦΦΦΦ OOOO OOOO ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ
	brc	ra,cond,off	f If condition is true, then jump to IMEM[PC+off], else continue Conditions: ra = 0 ; ra ≠ 0 ; ra = 1; ra < 0; ra > 0; ra ≤ 0; ra ≥ 0	11 10 00	ΦΦ ΦΦΦΦ OOOO OOOO ΦAAA AAΦΦ ΦCCC

Operands	Fields
rt,ra,rb	ΦΦ ΦΦΦΦ ΦΦΦΦ BBBB BAAA AAΦT TTTT
rt,ra,off	ΦΦ ΦΦΦO OOOO OOOO ΦAAA AAΦT TTTT
rt,ra,n	ΦΦ NNNN ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
rt,ra,imm	II IIII IIII IIII ΦAAA AAΦT TTTT
rt,ra	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦAAA AAΦT TTTT
rt,imm	II IIII IIII IIII IIII IIΦT TTTT
ra,cond,off	ΦΦ ΦΦΦΦ OOOO OOOO ΦAAA AAΦΦ ΦCCC
rt,addr	AA AAAA AAAA AAAA AAAA AAΦT TTTT
off	ΦΦ ΦΦΦΦ OOOO OOOO ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ
N/A	ΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ ΦΦΦΦ

Spec	num	bits
Instruction set	4.29E+09	32
opcode	64	6
data size	65536	16
registers	32	5
data offset	512	9
instruction offset	256	8
max shift	16	4
flags	7	3

Command	Steps	RA[2:0]	RB[2:0]	WA[2:0]	MA[15:0]	IMM[15:0]	OEN	S[1:4]	AL[2:0]	SH[5:0]	WEN
SHR R3, R1, 5	S1 - Fetch	ØØØ	ØØØ	ØØØ	0xØØØØ	0xØØØØ	0	Ø1ØØ	111	ØØØØØ	0
	S2 - Reg R	000	001	ØØØ	0xØØØØ	0xØØØØ	0	ØØØØ	ØØØ	ØØØØØ	0
	S3 - ALU	ØØØ	ØØØ	ØØØ	0xØØØØ	0xØØØØ	0	00Ø0	011	10101	0
	S4 - Reg W	ØØØ	ØØØ	011	0xØØØØ	0xØØØØ	0	ØØØ0	ØØØ	ØØØØØ	1
LOADI R5, 0xAF1F	S1 - Fetch	ØØØ	ØØØ	ØØØ	0xØØØØ	0xØØØØ	0	Ø1ØØ	111	ØØØØØ	0
	S4 - Mem RW	ØØØ	ØØØ	ØØØ	0xAF1F	0xØØØØ	0	ØØ1Ø	ØØØ	ØØØØØ	0
	S5 - Reg W	ØØØ	ØØØ	101	0xØØØØ	0xØØØØ	0	ØØØ1	ØØØ	ØØØØØ	1
BRNEQ R3, 0x11A	S1 - Fetch	ØØØ	ØØØ	ØØØ	0xØØØØ	0xØØØØ	0	Ø1ØØ	111	ØØØØØ	0
	S2 - Reg R	011	000	ØØØ	0xØØØØ	0x011A	0	11ØØ	101	ØØ000	0
	S3 - ALU	ØØØ	ØØØ	ØØØ	0xØØØØ	0xØØØØ	0	00ØØ	101	ØØ000	0

Command	Opcode	RA[2:0]	RB[2:0]	WA[2:0]	IMM[15:0]	OEN	S[4:1]	AI[2:0]	SH[5:0]	WEN
move R2, R7	10001	111	000	010	0xΦΦΦΦ	0	0ΦΦ0	101	ΦΦ0000	0
movi R3, 0x0000	10000	000	ΦΦΦ	011	0x0000	0	0ΦΦ1	101	ΦΦ0000	0
move R4, R1	10001	001	000	001	0xΦΦΦΦ	0	0ΦΦ0	101	ΦΦ0000	0
movi R5, 0x0001	10000	000	ΦΦΦ	101	0x0001	0	0ΦΦ1	101	ΦΦ0000	0
br R4, 000, loop_end_label	11000	100	000	ΦΦΦ	0x0005	0	ΦΦΦ1	101	ΦΦ0000	0
and R6, R2, R4	01100	010	100	110	0xΦΦΦΦ	0	0ΦΦ0	001	ΦΦ0000	0
xor R3, R6, R4	01110	110	100	011	0xΦΦΦΦ	0	0ΦΦ0	010	ΦΦ0000	0
shr R2, R2, 1	01001	010	000	010	0xΦΦΦΦ	0	0ΦΦ0	101	100001	0
dec R4, R4	00011	100	ΦΦΦ	100	0xΦΦΦΦ	0	0ΦΦ0	100	ΦΦ0000	0
br R0, 000, loop_start_label	11000	ΦΦΦ	000	ΦΦΦ	0xFFFB	0	ΦΦΦ1	101	ΦΦ0000	0
storo R7, R3, 2	10111	011	ΦΦΦ	111	0x0002	1	00Φ1	101	ΦΦ0000	1