(I) op = 21

BMN

RegDst = X
Jump = 0
Branch = 1
MemRead = 1
ALUop = 00
MemWrite = 0
ALUSrc = 1
RegWrite = 0
B_new = 1
Flag_sig = 10
Link = X
addr_from_mem = 1

(I) op = 19

JALM

RegDst = 0
Jump = 0
Branch = 1
MemRead = 1
MemtoReg = 1
ALUop = 00
MemWrite = 0
ALUSrc = 1
RegWrite = 1
B_new = 1
Flag_sig = 01
Link = 1
addr_from_mem = 1

(J) op = 24

B#

RegDst = X
Jump = 1
Branch = 0
MemRead = X
MemtoReg = X
ALUop = X
MemWrite = 0
ALUSrc = X
RegWrite = 0
B_new = 1
flag = 00
Link = X
addr_from_mem = X

(R) funct = 20

BRZ

RegDst = X
Jump = 0
Branch = X (0)
MemRead = X
MemtoReg = 0
ALUop = 10 → AND
MemWrite = 0
ALUSrc = X
RegWrite = X
B_new = 1
Flag_sig = 00
Link = X
addr_from_mem = 1

funct = 23

BALRN (R)

RegDst = 1
Jump = 0
Branch = X (0)
MemRead = 0
MemtoReg = 0
ALUop = 10 → 101 (basta)
MemWrite = 0
ALUSrc = X
RegWrite = 1
B_new = 1
Flag_sig = 01
Link = X
addr_from_mem = 1

SALV (R) funct = 6

RegDst = 1
Jump = 0
Branch = 0
MemRead = X
MemtoReg = 0
ALUop = 10 → 100 (basta)
MemWrite = 0
RegWrite = 1
B_new = X
Flag_sig = X
Link = 0
addr_from_mem = X

Sim_1

0 srlv $2, $1, $0 → 0000 0000 0010 0000 0001 0000 0000 0110

4 sw $2, 0($3) → 1010 1100 0110 0010 0000 0000 0000 0000

8 jalm $9, 0($3) → 0100 1100 0110 0000 0000 0000 0000 0000

12 add $4, $0, $1 → 0000 0000 0000 0001 0010 0000 0010 0000

16 sub $4, $0, $1 → 0000 0000 0000 0001 0010 0000 0010 0010

initially

$0 → 64 = 0x40 = 0100 0000

$1 → 2

$3 → 4

0 → $2 = 16 = 0x10

4 → mem[4] = 0x10

8 → pc = 0x10 $9 = 0x0C

16 → $4 = 3e

S.m - 2

0	sub	$2, $1, $0	→ 0000 0000 0010 0000 0001 0000 0010 0010
4	bmn	0($3)	→ 0101 0100 0110 0000 0000 0000 0000 0000
8	or	$2, $2, $0	→ 0100 0000 0100 0000 0001 0000 0010 0101
12	add	$2, $1, $0	→ 0000 0000 0010 0000 0001 0000 0000 0000
16	bmn	4 ($3)	→ 0101 0100 0110 0000 0000 0000 0000 0000
20	add	$2, $1, $3	→ 0000 0000 0010 0011 0100 0000 0010 0000
24	add	$2, $2, $0	→ 0000 0000 0100 0000 0001 0000 0010 0000

initially

$1 → 7

$0 → 12

$3 → 8

mem[8] = 12

mem[12] = 24

---

0 → $2 = -9

4 → branch to 12

8 → X

12 → $2 = 0x13

16 → Branch X

20 → $2 = 0xF

24 → $2 = 0x16

Sim-3

0 add $2, $0, $1 → 00 01 10 20

4 blt $3 → 0001 0000 0110 0000 0000 0000 0001 0100

8 sw $3, 0($4) → 1011 1100 1000 0011 0000 0000 0000 0000

12 add $2, $2, $1 → 00 41 10 20

16 blt $5 → 0001 0000 1010 0000 0000 0000 0001 0100

20 lw $0, 0($4) → 1000 1100 1000 0000 0000 0000 0000 0000

24 add $2, $0, $1 → 00 01 10 20

initial.

$0 → -1

$2 = 4

$1 → 1

$3 → 12 (0xc)

$4 → 4

$5 → 24 (0x18)

mem [4] = 0x34

---

0 → $2 = 0

4 → Branch to 12

8 → X

12 → $2 = 1

16 → Branch X

20 → $0 ← 0x34

24 → $2 = 0x35

Sim_4

0 add $2, $0, $1 → 00 10 10 20

4 bz (3*4) → 0110 0000 0000 0000 0000 0000 0000 0011

8 sw $3, 0($4) → 9C 83 00 00

12 add $2, $2, $1 → 00 41 10 20

16 bz (6*4) → 0110 0000 0000 0000 0000 0000 0000 0110

20 lw $0, 0($4) → 8C 80 00 00

24 add $2, $0, $1 → 00 01 10 20

initial

$0 → -1

$1 → 1

[] → 20 (0x14)

$4 → 4

mem[4] = 0x26

---

0 → $2 = 0

4 → Branch to 12

8 → X

12 → $2 = 1

16 → Branch X

20 → $0 = 0x26

24 → $2 = 0x27

Sim-9

0 sub $2, $1, $0 => 00 22 10 22

4 beqz $3, $6 -> 0000 0000 0110 0000 0011 0000 0001 0011

8 or $2, $2, $0 -> 00 40 10 25

12 add $2, $1, $0 -> 00 20 10 20

16 beqz $4, $6 -> 0000 0000 1000 0000 0011 0000 0001 0011

20 add $2, $1, $3 -> 00 23 10 20

24 add $2, $2, $0 -> 00 40 10 20

Initially

$0 = 12 (0xC)

$1 = 7

$3 = 12 (0xC)

$4 = 24 (0x18)

---

0 -> $2 = -9

4 -> branch to 12

8 -> X

12 -> $2 = 19 (0x13)

16 -> branch X

20 -> $2 = 0x13

24 -> $2 = 0x13