

1 Ladda/Spara

LOAD (LD)

Ladda in från dataminnet till det angivna registret.

Operation: `rD = data(rA + ext(offset))`

OP-kod: 000001

Flaggor: -

6	5	5	16
OP	D	A	offset

STORE (STR)

Skriv det angivna registret till dataminnet.

Operation: `data(rA + offset) = rB`

OP-kod: 000010

Flaggor: -

6	5	5	5	11
OP	-	A	B	offset

Move (MOV)

Flytta data mellan två register.

Operation: `rD = rA`

OP-kod: 000011

Flaggor: -

6	5	5	16
OP	D	A	-

Move High Immediate (MOVHI)

Ladda in immediate till de höga bitarna i det angivna registret.

Operation: `rD[0...15] = Immediate`

OP-kod: 000100

Flaggor: -

6	5	16	5
OP	D	I	-

Move Low Immediate (MOVLI)

Ladda in immediate till de låga bitarna i det angivna registret.

Operation: $rD[16...31] = \text{Immediate}$

OP-kod: 000101

Flaggor: -

6	5	16	5
OP	D	I	-

2 Aritmetik

Add (ADD)

Addera två register och spara det i ett tredje.

Operation: $rD = rA + rB$

OP-kod: 001010

Flaggor: N, Z, V, C

6	5	5	5	11
OP	D	A	B	-

Add Immediate (ADDI)

Addera ett register med en konstant och spara det i ett annat register.

Operation: $rD = rA + \text{ext}(I)$

OP-kod: 001011

Flaggor: N, Z, V, C

6	5	5	16
OP	D	A	I

Subtract (SUB)

Subtrahera ett register från ett annat, och spara det i ett tredje.

Operation: $rD = rA - rB$

OP-kod: 001100

Flaggor: N, Z, V, C

6	5	5	5	11
OP	D	A	B	-

Subtract Immediate (SUBI)

Addera ett register med en konstant och spara det i ett annat register.

Operation: $rD = rA - \text{ext}(I)$

OP-kod: 001101

Flaggor: N, Z, V, C

6	5	5	16
OP	D	A	-I

Increase (INC)

Öka värdet i det givna registret med 1.

Operation: $rA = rA + 1$

OP-kod: Pseudoinstruktion

Flaggor: N, Z, V, C

Decrease (DEC)

Minska värdet i det givna registret med 1.

Operation: $rA = rA - 1$

OP-kod: Pseudoinstruktion

Flaggor: N, Z, V, C

3 Skift

Logic Left Shift (LSL)

Logiskt skifta källregistret vänster ett steg och placera i destinationsregistret.

Operation: $rD = rA \ll 1$

OP-kod: 010100

Flaggor: -

6	5	5	16
OP	D	A	-

Arithmetic Shift Left (ASL)

Aritmetiskt skifta källregistret vänster ett steg och placera i destinationsregistret.

Operation: $rD = rA \ll 1$

OP-kod: 010101

Flaggor: -

6	5	5	6
OP	D	A	-

Logic Shift Right (LSR)

Logiskt skifta källregistret höger ett steg och placera i destinationsregistret.

Operation: $rD = rA \gg 1$

OP-kod: 010110

Flaggor: -

6	5	5	16
OP	D	A	-

Arithmetic Shift Right (ASR)

Aritmetiskt skifta källregistret höger ett steg och placera i destinationsregistret.

Operation: $rD = rA \gg 1$

OP-kod: 010111

Flaggor: -

6	5	5	16
OP	D	A	B

4 Logik

And (AND)

“Ocha” de givna registerna och sparar resultatet i destinationsregistret.

Operation: $rD = rA \& rB$

OP-kod: 011010

Flaggor: -

6	5	5	5	11
OP	D	A	B	-

And Immediate (ANDI)

“Ocha” det givna registret med en omedelbar konstant och sparar resultatet i destinationsregistret.

Operation: $rD = rA \& \text{ext}(I)$

OP-kod: 011011

Flaggor: -

6	5	5	16
OP	D	A	I

Or (OR)

“Ora” de givna registerna och sparar resultatet i destinationsregistret.

Operation: $rD = rA \mid rB$

OP-kod: 011100

Flaggor: -

6	5	5	5	11
OP	D	A	B	-

Or Immediate (ORI)

“Ora” det givna registret med en omedelbar konstant och sparar resultatet i destinationsregistret.

Operation: $rD = rA \mid \text{ext}(I)$

OP-kod: 011101

Flaggor: -

6	5	5	16
OP	D	A	I

Not (NOT)

Inverterar det givna registret och sparar resultatet i destinationsregistret.

Operation: $rD = !rA$

OP-kod: 011110

Flaggor: -

6	5	5	16
OP	D	A	-

Exclusive or (XOR)

“Exklusivt och:ar” de givna registerna och sparar resultatet i destinationsregistret.

Operation: rD

OP-kod: 011111

Flaggor: -

6	5	5	5	11
OP	D	A	B	-

5 Jämföra

Compare (CMP)

Jämför de givna registrerna och uppdatera flaggorna enligt resultatet.

Operation: $rA - rB$

OP-kod: 100011

Flaggor: Z, N, V

6	5	5	5	11
OP	-	A	B	-

Compare Immediate (CMPI)

Jämför det registret och konstanten, uppdatera flaggorna enligt resultatet.

Operation: $rA - \text{ext}(I)$

OP-kod: 100100

Flaggor: Z, N, V

6	5	5	16
OP	-	A	I

6 Hopp

Branch Equal (BEQ)

Hoppa om jämförelsen gav likhet.

Operation: $Z == 1 ? \text{JMP} : \text{NOP}$

OP-kod: 101000

Flaggor: -

6	26
OP	I

Branch Not Equal (BNEQ)

Hoppa om jämförelsen inte gav likhet.

Operation: Z == 0 ? JMP : NOP

OP-kod: 101001

Flaggor: -

6	26
OP	I

Branch Less Than (BLT)

Hoppa om jämförelsen gav att det ena registret var mindre än det andra.

Operation: N != V ? JMP : NOP

OP-kod: 101010

Flaggor: -

6	26
OP	I

Branch Greater Than (BGT)

Hoppa om jämförelsen gav att det ena registret var större än det andra.

Operation: (Z == 0) & (N == V) ? JMP : NOP

OP-kod: 101011

Flaggor: -

6	26
OP	I

Branch Less Than Or Equal (BLTEQ)

Hoppa om jämförelsen gav att det ena registret var mindre än eller lika med det andra.

Operation: (Z == 1) || (N != V) ? JMP : NOP

OP-kod: 101100

Flaggor: -

6	26
OP	I

Branch Greater Than Or Equal (BGTEQ)

Hoppa om jämförelsen gav att det ena registret är större än eller lika med det andra.

Operation: N == V ? JMP : NOP

OP-kod: 101101

Flaggor: -

6	26
OP	I

Jump (JMP)

Ovillkorligt hopp till adressen I (tar med instruktionen efter som alla andra jumps men har 2 hoppnoppar).

Operation: $PC = rA + I$

OP-kod: 101110

Flaggor: -

6	5	5	16
OP	-	A	I

Jump (JMPI)

Ovillkorligt hopp med offseten i det givna registret, och med en frivillig immediate offset.

Operation: $PC = PC + I + 1$

OP-kod: 101111

Flaggor: -

6	26
OP	I

7 Övrigt

No operation (NOP)

Gör ingenting.

Operation: -

OP-kod: 000000

Flaggor: -

6	26
OP	-

Halt processor (HALT)

Stoppa CPU:n.

Operation: -

OP-kod: 110110

Flaggor: -

6	26
OP	-

Henak (HENAK)

Gör en Henak, så att säga.

Operation: $PC = 718$, om man har tur, finns ju en risk att man går vilse.

OP-kod: 111111

Flaggor: -

6	26
OP	-