Table of Contents

[Programming Model 4](#_Toc94526283)

[Branch Instruction 5](#_Toc94526284)

[Target Addresses 5](#_Toc94526285)

[Conditional Branches 5](#_Toc94526286)

[BRA – Branch Always 5](#_Toc94526287)

[BRN – Branch Never 5](#_Toc94526288)

[BHI – Branch Higher 5](#_Toc94526289)

[BLS – Branch Lower or Same 6](#_Toc94526290)

[BHS – Branch Higher or Same 6](#_Toc94526291)

[BLO – Branch Lower 6](#_Toc94526292)

[BNE – Branch Not Equal 7](#_Toc94526293)

[BEQ – Branch Equal 7](#_Toc94526294)

[BVC – Branch Overflow Clear 7](#_Toc94526295)

[BVS – Branch Overflow Set 7](#_Toc94526296)

[BPL – Branch Plus 8](#_Toc94526297)

[BMI – Branch Minus 8](#_Toc94526298)

[BGE – Branch Greater or Equal 8](#_Toc94526299)

[BLT – Branch Less Than 8](#_Toc94526300)

[BGT – Branch Greater Than 9](#_Toc94526301)

[BLE – Branch Less Than or Equal 9](#_Toc94526302)

[LBRA – Long Branch 10](#_Toc94526303)

[LBSR – Long Branch to Subroutine 10](#_Toc94526304)

[RTI – Return from Interrupt 10](#_Toc94526305)

[RTS – Return from Subroutine 10](#_Toc94526306)

[SWI – Software Interrupt 11](#_Toc94526307)

[LEAX – Load Effective Address into X 12](#_Toc94526308)

[LEAY – Load Effective Address into Y 12](#_Toc94526309)

[LEAS – Load Effective Address into S 12](#_Toc94526310)

[LEAU – Load Effective Address into U 12](#_Toc94526311)

[ABX – Add B and X 12](#_Toc94526312)

[CWAI – Wait 12](#_Toc94526313)

[MUL – Multiply 13](#_Toc94526314)

[LD – Load 14](#_Toc94526315)

[ST – Load 14](#_Toc94526316)

[PSHS – Push Registers on Stack 15](#_Toc94526317)

[PULS – Pull Registers from Stack 16](#_Toc94526318)

[PSHU – Push Registers on User Stack 16](#_Toc94526319)

[PULU – Pull Registers from User Stack 16](#_Toc94526320)

[NEG – Negate 16](#_Toc94526321)

[COM – Complement 18](#_Toc94526322)

[LSR – Logical Shift Right 18](#_Toc94526323)

[ROR – Rotate Right 19](#_Toc94526324)

[ASR – Arithmetic Shift Right 20](#_Toc94526325)

[ASL – Arithmetic Shift Left 21](#_Toc94526326)

[ROL – Rotate Left 21](#_Toc94526327)

[DEC – Decrement 22](#_Toc94526328)

[INC – Increment 23](#_Toc94526329)

[TST – Test 24](#_Toc94526330)

[JMP – Jump 24](#_Toc94526331)

[CLR – Clear 25](#_Toc94526332)

[NOP – No Operation 25](#_Toc94526333)

[SYNC – Synchronize 25](#_Toc94526334)

[DAA – Decimal Adjust 25](#_Toc94526335)

[ORCC – OR to Condition Code 25](#_Toc94526336)

[ANDCC – AND with Condition Code 26](#_Toc94526337)

[SEXT – Sign Extend 26](#_Toc94526338)

[EXG – Exchange 26](#_Toc94526339)

[TFR – Transfer 26](#_Toc94526340)

[SUB – Subtract 27](#_Toc94526341)

[CMP – Compare 29](#_Toc94526342)

[SBC – Subtract with Carry 29](#_Toc94526343)

[AND – Bitwise ‘And’ 30](#_Toc94526344)

[BIT – Bitwise Test Bits 31](#_Toc94526345)

[EOR – Bitwise Exclusive ‘Or’ 31](#_Toc94526346)

[ADC – Add with Carry 31](#_Toc94526347)

[OR – Bitwise ‘Or’ 32](#_Toc94526348)

[ADD – Add 32](#_Toc94526349)

# Programming Model

There are seven 32-bit accumulators for which the low order and high order halves may be used independently. There are seven 32-bit indexing registers including the stack pointer.

A condition code register is used to store the status results as operations are performed.

|  |  |
| --- | --- |
| 31 16 | 15 0 |
| AH | AL |
| BH | BL |
| CH | CL |
| DH | DL |
| EH | EL |
| FH | FL |
| GH | GL |
| SI | |
| DI | |
| XR | |
| YR | |
| BP | |
| SP | |
| UP | |
| DB | 0 |
|  | |
| PC | |
|  | |
|  | CCR |

## Instruction Set Description

The constant size for immediate, direct and extended addressing modes is specified with a three-bit field in the instruction Sz3.

|  |  |
| --- | --- |
| Sz3 | Constant Bits |
| 0 | 0 |
| 1 | 16 |
| 2 | 32 |
| 3 | reserved |
| 4 | “ |
| 5 | “ |
| 6 | “ |
| 7 | “ |

For most instructions a five-bit register field specifies the register involved.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reg |  | Reg |  | Reg | nnn |  | Reg | rrr |  |
| 0 | AL | 8 | AH | 16 | 0 | AX | 24 | 0 | XR |
| 1 | BL | 9 | BH | 17 | 1 | BX | 25 | 1 | YR |
| 2 | CL | 10 | CH | 18 | 2 | CX | 26 | 2 | SI |
| 3 | DL | 11 | DH | 19 | 3 | DX | 27 | 3 | DI |
| 4 | EL | 12 | EH | 20 | 4 | EX | 28 | 4 | BP |
| 5 | FL | 13 | FH | 21 | 5 | FX | 29 | 5 | SP |
| 6 | GL | 14 | GH | 22 | 6 | GX | 30 | 6 | UP |
| 7 | one | 15 | zero | 23 | 7 | zero | 31 | 7 | zero |

The indexed addressing mode specification field is eleven bits in size.

Bits rrr specifies one of the index registers, XR, YR, SI, DI, BP, SP, or UP

Bits ddddddd specifies the low order seven bits of a fifteen-bit displacement. The high order eight bits are in the next byte.

The ‘i’ bit indicates one level of indirection is added for the data fetch.

The ‘o’ bit indicates that indexing is applied after indirection.

‘nnn’ specifies one of the 32-bit registers AX, BX, CX, DX, EX, FX, GX, or zero.

|  |  |  |
| --- | --- | --- |
| Ndx Pattern |  |  |
| 0rrrddddddd | ,r + 15 bit offset |  |
| 1rrrio00000 | ,r+ with 8 bit offset | increment by one |
| 1rrrio00001 | ,r+ with 8 bit offset | increment by two |
| 1rrrio00010 | ,r+ with 8 bit offset | increment by four |
| 1rrrio00100 | ,r- with 8 bit offset | decrement by one |
| 1rrrio00101 | ,r- with 8 bit offset | decrement by two |
| 1rrrio00110 | ,r- with 8 bit offset | decrement by four |
| 1rrrio01nnn | r,r with 8 bit offset |  |
| 1rrrio10nnn | r,r with 24 bit offset |  |
| 1rrrio11nnn | r,r with 40 bit offset |  |
|  |  |  |

## Arithmetic Instructions

## ABX – Add B and X

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
|  | 3Ah |

## ADD – Add

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 8Bh |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 9Bh |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | ABh |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | Rt | BBh |

## ADC – Add with Carry

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 89h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 99h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A9h |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | Rt | B9h |

## AND – Bitwise ‘And’

**Instruction Format**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 84h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offs | Sz3 | Reg | 94h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A4h |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | Rt | B4h |

## ASL – Arithmetic Shift Left

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| Am3 | Reg | 48h |

**Instruction Format: DP / EXT**

For this format, the register field specifies a register containing the amount to shift by.

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offs | Sz3 | Reg | 78h |

**Instruction Format: NDX**

For this format, the register field specifies a register containing the amount to shift by.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | 68h |

## ASR – Arithmetic Shift Right

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 07h |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| Am3 | Reg | 47h |

**Instruction Format: DP / EXT**

For this format, the register field specifies a register containing the amount to shift by.

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offs | Sz3 | Reg | 77h |

**Instruction Format: NDX**

For this format, the register field specifies a register containing the amount to shift by.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | 67h |

## BIT – Bitwise Test Bits

**Description:**

Perform a bitwise ‘and’ operation and discard the result but keeping the result status flags in the condition code register.

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 85h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 95h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A5h |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | 31 | B5h |

## CLR – Clear

**Description:**

CLR is an alternate mnemonic for operations that may clear memory or a register.

## CMP – Compare

**Description:**

Perform a subtract operation and discard the result but keeping the result status flags in the condition code register.

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 81h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offs | Sz3 | Reg | 91h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A1h |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | 31 | B1h |

## COM – Complement

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 03h |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| ~3 | Reg | 43h |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 63h |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address32 | 73h |

## DAA – Decimal Adjust

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
|  | 19h |

## DEC – Decrement

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 0Ah |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| Am3 | Reg | 4Ah |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 6Ah |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address32 | 7Ah |

## EOR – Bitwise Exclusive ‘Or’

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 88h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 98h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A8h |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | Rt | B8h |

## EXG – Exchange

**Description:**

EXG allows any pair of 16 or 32 bit register to be exchanged.

**Instruction Format: Reg16**

Exchange a pair of 16-bit registers.

|  |  |  |
| --- | --- | --- |
| 15 12 | 11 8 | 7 0 |
| Reg | Reg | 1Eh |

|  |  |  |  |
| --- | --- | --- | --- |
| Reg |  | Reg |  |
| 0 | AL | 8 | AH |
| 1 | BL | 9 | BH |
| 2 | CL | 10 | CH |
| 3 | DL | 11 | DH |
| 4 | EL | 12 | EH |
| 5 | FL | 13 | FH |
| 6 | GL | 14 | GH |
| 7 | DPR | 15 | zero |

**Instruction Format: Reg32**

Exchange a pair of 32-bit registers.

|  |  |  |
| --- | --- | --- |
| 15 12 | 11 8 | 7 0 |
| Reg | Reg | 18h |

|  |  |  |  |
| --- | --- | --- | --- |
| Reg |  | Reg |  |
| 0 | AX | 8 | XR |
| 1 | BX | 9 | YR |
| 2 | CX | 10 | SI |
| 3 | DX | 11 | DI |
| 4 | EX | 12 | BP |
| 5 | FX | 13 | SP |
| 6 | GX | 14 | UP |
| 7 | zero | 15 | zero |

## INC – Increment

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 0Ch |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| Am3 | Reg | 4Ch |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 6Ch |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address32 | 7Ch |

## LSR – Logical Shift Right

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 04h |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| Am3 | Reg | 44h |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 64h |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 47 8 | 7 0 |
| Address40 | 74h |

## MUL – Multiply

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
|  | 3Dh |

## NEG – Negate

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 00h |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| ~3 | Reg | 40h |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 60h |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address32 | 70h |

## OR – Bitwise ‘Or’

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 8Ah |

**Instruction Format: DP**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 9Ah |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | AAh |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | Rt | BAh |

## ROL – Rotate Left

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 09h |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| Am3 | Reg | 49h |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 69h |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address32 | 79h |

## ROR – Rotate Right

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 06h |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| Am3 | Reg | 46h |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 66h |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address32 | 76h |

## SBC – Subtract with Carry

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 82h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offs | Sz3 | Reg | 92h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A2h |

**Instruction Format: REG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31 23 | 22 18 | 17 13 | 12 8 | 7 0 |
| ~ | Rb | Ra | Rt | B2h |

## SEXT – Sign Extend

**Description:**

The SEXT instruction will sign extend one of the low accumulators (AL, BL, CL, DL, EL, FL, or GL) to the full width of the corresponding accumulator.

**Instruction Format**

|  |  |  |
| --- | --- | --- |
| 15 12 | 11 8 | 7 0 |
| ~ | Reg | 1Dh |

|  |  |  |  |
| --- | --- | --- | --- |
| Reg |  | Reg |  |
| 0 | AL | 8 | AH |
| 1 | BL | 9 | BH |
| 2 | CL | 10 | CH |
| 3 | DL | 11 | DH |
| 4 | EL | 12 | EH |
| 5 | FL | 13 | FH |
| 6 | GL | 14 | GH |
| 7 | DPR | 15 | zero |

## SUB – Subtract

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 80h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Offs | Sz3 | Reg | 90h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A0h |

|  |  |  |
| --- | --- | --- |
| Ndx Pattern |  |  |
| 0rrrddddddd | ,r + 15 bit offset |  |
| 1rrrio00000 | ,r+ with 8 bit offset | increment by one |
| 1rrrio00001 | ,r+ with 8 bit offset | increment by two |
| 1rrrio00010 | ,r+ with 8 bit offset | increment by four |
| 1rrrio00100 | ,r- with 8 bit offset | decrement by one |
| 1rrrio00101 | ,r- with 8 bit offset | decrement by two |
| 1rrrio00110 | ,r- with 8 bit offset | decrement by four |
| 1rrrio01nnn | r,r with 8 bit offset |  |
| 1rrrio10nnn | r,r with 24 bit offset |  |
| 1rrrio11nnn | r,r with 40 bit offset |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| rrr |  | nnn |  |
| 0 | none | 0 | AX |
| 1 | XR | 1 | BX |
| 2 | YR | 2 | CX |
| 3 | SI | 3 | DX |
| 4 | DI | 4 | EX |
| 5 | BP | 5 | FX |
| 6 | SP | 6 | GX |
| 7 | UP | 7 | IP |

## TFR – Transfer

**Description:**

TFR allows any 16-bit register to be transferred to another 16-bit register, or any 32-bit register to another 32-bit register. Bits 12 to 15 identify the source register, bits 8 to 11 identify the destination register. This instruction may be used to clear any register.

**Instruction Format: Reg16**

Transfer a pair of 16-bit registers.

|  |  |  |
| --- | --- | --- |
| 15 12 | 11 8 | 7 0 |
| Reg | Reg | 1Fh |

|  |  |  |  |
| --- | --- | --- | --- |
| Reg |  | Reg |  |
| 0 | AL | 8 | AH |
| 1 | BL | 9 | BH |
| 2 | CL | 10 | CH |
| 3 | DL | 11 | DH |
| 4 | EL | 12 | EH |
| 5 | FL | 13 | FH |
| 6 | GL | 14 | GH |
| 7 | DPR | 15 | zero |

**Instruction Format: Reg32**

Transfer a pair of 32-bit registers.

|  |  |  |
| --- | --- | --- |
| 15 12 | 11 8 | 7 0 |
| Reg | Reg | 15h |

|  |  |  |  |
| --- | --- | --- | --- |
| Reg |  | Reg |  |
| 0 | AX | 8 | XR |
| 1 | BX | 9 | YR |
| 2 | CX | 10 | SI |
| 3 | DX | 11 | DI |
| 4 | EX | 12 | BP |
| 5 | FX | 13 | SP |
| 6 | GX | 14 | UP |
| 7 | zero | 15 | zero |

## TST – Test

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 0Dh |

**Instruction Format: Reg**

|  |  |  |
| --- | --- | --- |
| 1513 | 12 8 | 7 0 |
| ~3 | Reg | 4Dh |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 6Dh |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 47 8 | 7 0 |
| Address40 | 7Dh |

## Branch Instructions

### Target Addresses

The branch target address is relative to the program counter. Branches may be either short using an eight-bit displacement field, or long using a twenty-four-bit displacement field. The displacement field is shifted left once before use as instructions must always be 16-bit aligned. The branch range is ±256B with an eight-bit displacement or ±16777216B using a twenty-four-bit displacement. Branching is relative to the address of the branch instruction.

### Conditional Branches

Branches branch based on status flags in the condition code register. The opcode determines which flag is tested.

## BRA – Branch Always

**Description:**

The branch always instruction always branches regardless of the state of any condition code register bits.

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 20h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 50h |

## BRN – Branch Never

**Description:**

The branch never instruction is treated as a NOP. It does not do anything except occupy bytes in the instruction stream. The displacement field is not used and may contain program data.

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 21h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 51h |

## BHI – Branch Higher

**Description:**

The BHI instruction branches if comparison result indicates one value is higher or greater than the other when values are unsigned numbers.

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 22h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 52h |

## BLS – Branch Lower or Same

**Description:**

The BLS instruction branches if comparison result indicates one value is lower or less than or equal to the other when values are unsigned numbers.

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 23h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 53h |

## BHS – Branch Higher or Same

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 24h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 54h |

## BLO – Branch Lower

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 25h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 55h |

## BNE – Branch Not Equal

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 26h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 56h |

## BEQ – Branch Equal

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 27h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 57h |

## BVC – Branch Overflow Clear

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 28h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 58h |

## BVS – Branch Overflow Set

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 29h |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 59h |

## BPL – Branch Plus

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 2Ah |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 5Ah |

## BMI – Branch Minus

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 2Bh |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 5Bh |

## BGE – Branch Greater or Equal

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 2Ch |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 5Ch |

## BLT – Branch Less Than

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 2Dh |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 5Dh |

## BGT – Branch Greater Than

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 2Eh |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 5Eh |

## BLE – Branch Less Than or Equal

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 2Fh |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 5Fh |

## BSR – Long Branch to Subroutine

**Description:**

The BSR instruction performs a subroutine call by first placing the address of the next instruction on the stack and then performing a program counter relative jump to the target address.

**Instruction Format: D8**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Disp8 | 8Dh |

**Instruction Format: D24**

|  |  |
| --- | --- |
| 31 8 | 7 0 |
| Disp24 | 17h |

## JMP – Jump

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 0Eh |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | 6Eh |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address40 | 7Eh |

## JSR – Jump to Subroutine

**Instruction Format: DP**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Offs | 9Dh |

**Instruction Format: NDX**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Ndx | ADh |

**Instruction Format: EXT**

|  |  |
| --- | --- |
| 39 8 | 7 0 |
| Address32 | BDh |

## NOP – No Operation

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| 12h | 12h |

## RTI – Return from Interrupt

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
|  | 3Bh |

## RTS – Return from Subroutine

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Amt | 39h |

## Memory Operate Instructions

## LD – Load

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 86h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| 31 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 96h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A6h |

## LEA – Load Effective Address

**Instruction Format: IMM**

|  |  |  |  |
| --- | --- | --- | --- |
| n 16 | 1513 | 12 8 | 7 0 |
| Constant | Sz3 | Reg | 86h |

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| 31 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 9Eh |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | AEh |

## PSHS – Push Registers on Stack

**Instruction Format**

|  |  |  |
| --- | --- | --- |
| 31 16 | 15 8 | 7 0 |
| Reglist24 | | 34h |

The stacking order is PC is pushed first, CCR is pushed last. If the bit in the register list is set, then the corresponding register is stacked.

|  |  |  |
| --- | --- | --- |
| Reglist Bitno | Register Stacked |  |
|  | PC hi | higher memory address |
| 22 | PC lo |  |
|  | SP hi |  |
| 21 | SP lo |  |
|  | UP hi |  |
| 20 | UP lo |  |
|  | BP hi |  |
| 19 | BP lo |  |
|  | YR hi |  |
| 18 | YR lo |  |
|  | XR hi |  |
| 17 | XR lo |  |
|  | DI hi |  |
| 16 | DI lo |  |
|  | SI hi |  |
| 15 | SI Lo |  |
| 14 | GH |  |
| 13 | GL |  |
| 12 | FH |  |
| 11 | FL |  |
| 10 | EH |  |
| 9 | EL |  |
| 8 | DH |  |
| 7 | DL |  |
| 6 | CH |  |
| 5 | CL |  |
| 4 | BH |  |
| 3 | BL |  |
| 2 | AH |  |
| 1 | AL |  |
| 0 | CCR | lower memory address |

## PULS – Pull Registers from Stack

**Instruction Format**

|  |  |  |
| --- | --- | --- |
| 31 16 | 15 8 | 7 0 |
| Reglist24 | | 35h |

## PSHU – Push Registers on User Stack

**Instruction Format**

|  |  |  |
| --- | --- | --- |
| 31 16 | 15 8 | 7 0 |
| Reglist24 | | 36h |

## PULU – Pull Registers from User Stack

**Instruction Format**

|  |  |  |
| --- | --- | --- |
| 31 16 | 15 8 | 7 0 |
| Reglist24 | | 37h |

## ST – Store

**Description:**

Store a value to memory. The store instruction may also be used to clear memory by specifying a zero register as the data source.

**Instruction Format: DP / EXT**

|  |  |  |  |
| --- | --- | --- | --- |
| 31 16 | 1513 | 12 8 | 7 0 |
| Offset | Sz3 | Reg | 97h |

**Instruction Format: NDX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n 32 | 31 24 | 23 13 | 12 8 | 7 0 |
| Offs | Constant | Ndx | Reg | A7h |

## ANDCC – AND with Condition Code

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Const | 1Ch |

## CWAI – Wait

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
|  | 3Ch |

## ORCC – OR to Condition Code

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
| Const | 1Ah |

## SWI – Software Interrupt

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
|  | 3Fh |

## SYNC – Synchronize

**Instruction Format**

|  |  |
| --- | --- |
| 15 8 | 7 0 |
|  | 13h |