### RF65000

Register File

The register file contains 32 general purpose registers.

The register file is a *unified* register file. Registers may contain integer or floating-point data.

|  |  |  |  |
| --- | --- | --- | --- |
| Regno | ABI | ABI Usage |  |
| 0 | 0 | Always zero, for indexed addressing modes, refers to the program counter |  |
| 1 | A0 | First argument / return value register |  |
| 2 | A1 | Second argument / return value register |  |
| 3 | T0 | Temporary register, caller save |  |
| 4 | T1 | Temporary register |  |
| 5 | T2 | Temporary register |  |
| 6 | T3 | Temporary register |  |
| 7 | T4 | Temporary register |  |
| 8 | T5 | Temporary register |  |
| 9 | T6 | Temporary register |  |
| 10 | T7 | Temporary register |  |
| 11 | S0 | Saved register, register variables |  |
| 12 | S1 | Saved register |  |
| 13 | S2 | Saved register |  |
| 14 | S3 | Saved register |  |
| 15 | S4 | Saved register |  |
| 16 | S5 | Saved register |  |
| 17 | S6 | Saved register |  |
| 18 | S7 | Saved register |  |
| 19 | S8 | Saved register |  |
| 20 | A2 | Third argument register |  |
| 21 | A3 | Argument register |  |
| 22 | A4 | Argument register |  |
| 23 | A5 | Argument register |  |
| 24 | A6 | Argument register |  |
| 25 | A7 | Argument register |  |
| 26 |  |  |  |
| 27 | GP2 | Global pointer – rodata section |  |
| 28 | GP1 | Global pointer – bss section |  |
| 29 | GP0 | Global pointer – data section |  |
| 30 | FP | Frame Pointer |  |
| 31 | SP | Stack pointer |  |
| 31 | USP | User stack pointer |  |
| 31 | SSP | System stack pointer |  |
|  | LC | Loop counter |  |
|  | PC | Program counter |  |
|  | SR | Status Register |  |

### Major Opcode

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0x | 0  BRK | 1  {R1} | 2  {CSR} | 3  CMP | 4  ADD | 5  SUB | 6  MUL | 7  DIV |
|  | 8  AND | 9  OR | 10  XOR | 11 | 12 | 13 | 14 | 15 |
| 1x | 16  MOVE | 17  MOVEM | 18 | 19 | 20  FMA | 21  FMS | 22  FNMA | 23  FNMS |
|  | 24  CALL abs | 25 | 26 | 27 | 28  Bcc | 29  DBcc | 30 | 31 |

Operand Swapping

Most instructions allow first and second source operands to be swapped. This is indicated by the swap ‘S’ bit in the instruction.

**Address Modes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Address Mode** | **Format** | **M3** | **Reg5** |
| Register direct | Rn | 0 | Rn |
| Register indirect with displacement | Disp(Rn) | 1 | Rn |
| Post Update Register indirect with displacement | Disp(Rn)+ | 2 | Rn |
| Pre Update Register indirect with displacement | +Disp(Rn) | 3 | Rn |
| Indexed1 | (Rn,Rn) | 4 | Rn |
| Indexed with 32-bit displacement1 | Disp(Rn,Rn) | 5 | Rn |
| Indexed with 64-bit displacement1 | Disp(Rn,Rn) | 6 | Rn |
| Immediate Nine | Imm9 | 7 | 0nnnn |
| Immediate Thirty-Two | Imm32 | 7 | 10000 |
| Immediate Sixty-Four | Imm64 | 7 | 10001 |

1For indexed modes, R0 refers to the program counter

Three-bit displacement is in terms of operation size units.

**Operation Size**

|  |  |  |
| --- | --- | --- |
| **Operation Size** | **Suffix** | **Sz2** |
| Byte | .b | 00 |
| Wyde | .w | 01 |
| Tetra | .t | 10 |
| Octa | .o | 11 |

Operand Swap

|  |  |
| --- | --- |
| **Operand Order** | **S** |
| Mem is 2nd source operand | 0 |
| Mem is 1st source operand | 1 |

Status Register

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 24 | 23 21 | 20 16 | 15 | 14 | 13 | 12 | 11 | 10 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CPL8 | ~3 | Rr5 | tf | ~ | sf | rf | ~ | Im3 | ~ | ~ | df | xf | ~ | ~ | ~ | ~ |

df: decimal arithmetic flag

tf: trace mode

### ABS – Absolute Value

**Description:**

This instruction computes the absolute value of the contents of the source operand and places the result in Rt.

**Integer Instruction Format: R1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | 1 | Rb5 | 03 | 65 | Rt5 | Sz2 | 16 |

**Supported Address Modes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \* | \* | \* | \* | \* | \* |  |

**Operation:**

If <ea> < 0

Rt = -<ea>

else

Rt = <ea>

**Execution Units:** Integer ALU #0

**Clock Cycles: 1**

**Exceptions:** none

**Notes:**

### ADD - Addition

**Description:**

Add two source operands and place the sum in the target register. All registers are integer registers. Arithmetic is signed twos-complement values.

**Supported Address Modes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \* | \* | \* | \* | \* | \* | \* |

**Operation:**

Rt = Ra + Rb

**Clock Cycles:**

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

**Instruction Formats:**

**ADD Rt, Ra, Rb, Rc – Register direct**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | ~ | Rb5 | 03 | Ra5 | Rt5 | Sz2 | 46 |

**ADD Rt,Ra,disp3(Rb) – Register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 46 |

**ADD Rt,Ra,disp32(Rb) – Register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |

**ADD Rt,Ra,disp64(Rb) – Register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**ADD Rt,Ra,disp3(Rb)+ – Post update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 46 |

**ADD Rt,Ra,disp32(Rb)+ – Post update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |

**ADD Rt,Ra,disp64(Rb)+ – Post update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**ADD Rt,Ra,+disp3(Rb) – Pre update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 46 |

**ADD Rt,Ra,+disp32(Rb) – Pre update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |

**ADD Rt,Ra,+disp64(Rb) – Pre update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**ADD Rt,Ra,(Rb, Rc) - Indexed**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 43 | Ra5 | Rt5 | Sz2 | 46 |

**ADD Rt,Ra,disp32(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 53 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement32 | | | | | | | |

**ADD Rt,Ra,disp64(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 63 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**ADD Rt,Ra,Imm9**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Imm8..4 | S | 0 | Imm3..0 | 73 | Ra5 | Rt5 | Sz2 | 46 |

**ADD Rt,Ra,Imm32**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| ~5 | S | 1 | 0000 | 73 | Ra5 | Rt5 | Sz2 | 46 |
| Immediate32 | | | | | | | | |

**ADD Rt,Ra,Imm64 or ADD Rt,Imm64,Ra**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| ~5 | S | 1 | 0001 | 73 | Ra5 | Rt5 | Sz2 | 46 |
| Immediate31..0 | | | | | | | | |
| Immediate63..32 | | | | | | | | |

**Operation:**

Rt = Ra + Imm

**Clock Cycles:** 1

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

### AND – Bitwise And

**Description:**

Bitwise ‘and’ two source operands and place the sum in the target register. All registers are integer registers. Arithmetic is signed twos-complement values.

**Supported Address Modes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \* | \* | \* | \* | \* | \* | \* |

**Flag Updates:**

Condition Register #0 is always updated.

If the result is zero the equals flag is set otherwise it is cleared.

If the result is negative the signed less than flag is set, otherwise it is cleared.

**Operation:**

Rt = Ra & <ea>

**Clock Cycles:**

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

**Instruction Formats:**

**AND Rt, Ra, Rb, Rc – Register direct**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | ~ | Rb5 | 03 | Ra5 | Rt5 | Sz2 | 86 |

**AND Rt,Ra,disp3(Rb) – Register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 86 |

**AND Rt,Ra,disp32(Rb) – Register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement31..0 | | | | | | | |

**AND Rt,Ra,disp64(Rb) – Register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**AND Rt,Ra,disp3(Rb)+ – Post update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 86 |

**AND Rt,Ra,disp32(Rb)+ – Post update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement31..0 | | | | | | | |

**AND Rt,Ra,disp64(Rb)+ – Post update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**AND Rt,Ra,+disp3(Rb) – Pre update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 86 |

**AND Rt,Ra,+disp32(Rb) – Pre update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement31..0 | | | | | | | |

**AND Rt,Ra,+disp64(Rb) – Pre update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**AND Rt,Ra,(Rb, Rc) - Indexed**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 43 | Ra5 | Rt5 | Sz2 | 86 |

**AND Rt,Ra,disp32(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 53 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement32 | | | | | | | |

**AND Rt,Ra,disp64(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 63 | Ra5 | Rt5 | Sz2 | 86 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**AND Rt,Ra,Imm9**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Imm8..4 | S | 0 | Imm3..0 | 73 | Ra5 | Rt5 | Sz2 | 86 |

**AND Rt,Ra,Imm32**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| ~5 | S | 1 | 0000 | 73 | Ra5 | Rt5 | Sz2 | 86 |
| Immediate32 | | | | | | | | |

**AND Rt,Ra,Imm64 or AND Rt,Imm64,Ra**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| ~5 | S | 1 | 0001 | 73 | Ra5 | Rt5 | Sz2 | 86 |
| Immediate31..0 | | | | | | | | |
| Immediate63..32 | | | | | | | | |

**Clock Cycles:** 1

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

### CMP - Comparison

**Description:**

Compare two source operands and place the result in the target condition register. The result is a vector identifying the relationship between the two source operands as signed and unsigned integers and as floating-point values.

**Result Vector:**

|  |  |  |
| --- | --- | --- |
| **Bit** | **Mnemonic** | **Test** |
| 0 | RA | 1 |
| 1 | SR | 0 |
| 2 | HI | !cf & !zf |
| 3 | LS | cf | zf |
| 4 | CC / HS | !cf |
| 5 | CS / LO | cf |
| 6 | NE | !zf |
| 7 | EQ | zf |
| 8 | VC | !vf |
| 9 | VS | vf |
| A | PL | !nf |
| B | MI | nf |
| C | GE | (nf & vf) | (!nf & !vf) |
| D | LT | (nf & !vf) | (!nf & vf) |
| E | GT | (nf & vf & !zf) | (!nf & !vf & zf) |
| F | LE | zf | (nf & !vf) | (!nf & vf) |
| 10 | FOR |  |
| 11 | FUN |  |
| 12 | FGT |  |
| 13 | FLE |  |
| 14 | FGE |  |
| 15 | FLT |  |
| 16 | FNE |  |
| 17 | FEQ |  |
| 18 | DFOR |  |
| 19 | DFUN |  |
| 1A | DFGT |  |
| 1B | DFLE |  |
| 1C | DFGE |  |
| 1D | DFLT |  |
| 1E | DFNE |  |
| 1F | DFEQ |  |

**Supported Address Modes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \* | \* | \* | \* | \* | \* | \* |

**Operation:**

Rt = Src1 ? Src2

**Clock Cycles:**

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

**Instruction Formats:**

**CMP Rt, Ra, Rb – Register direct**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| ~5 | ~ | Rb5 | 03 | Ra5 |  | Ct3 | Sz2 | 36 |

**CMP Rt,Ra,disp3(Rb) – Register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 13 | Ra5 |  | Ct3 | Sz2 | 36 |

**CMP Rt,Ra,disp32(Rb) – Register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| 165 | S | Rb5 | 13 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement31..0 | | | | | | | | |

**CMP Rt,Ra,disp64(Rb) – Register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| 175 | S | Rb5 | 13 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement31..0 | | | | | | | | |
| Displacement63..32 | | | | | | | | |

**CMP Rt,Ra,disp3(Rb)+ – Post update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 23 | Ra5 |  | Ct3 | Sz2 | 36 |

**CMP Rt,Ra,disp32(Rb)+ – Post update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| 165 | S | Rb5 | 23 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement31..0 | | | | | | | | |

**CMP Rt,Ra,disp64(Rb)+ – Post update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| 175 | S | Rb5 | 23 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement31..0 | | | | | | | | |
| Displacement63..32 | | | | | | | | |

**CMP Rt,Ra,+disp3(Rb) – Pre update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 33 | Ra5 |  | Ct3 | Sz2 | 36 |

**CMP Rt,Ra,+disp32(Rb) – Pre update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| 165 | S | Rb5 | 33 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement31..0 | | | | | | | | |

**CMP Rt,Ra,+disp64(Rb) – Pre update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| 175 | S | Rb5 | 33 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement31..0 | | | | | | | | |
| Displacement63..32 | | | | | | | | |

**CMP Rt,Ra,(Rb, Rc) - Indexed**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 43 | Ra5 |  | Ct3 | Sz2 | 36 |

**CMP Rt,Ra,disp32(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 53 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement32 | | | | | | | | |

**CMP Rt,Ra,disp64(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 63 | Ra5 |  | Ct3 | Sz2 | 36 |
| Displacement31..0 | | | | | | | | |
| Displacement63..32 | | | | | | | | |

**CMP Rt,Ra,Imm9**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| Imm8..4 | S | 0 | Imm3..0 | 73 | Ra5 |  | Ct3 | Sz2 | 36 |

**CMP Rt,Ra,Imm32**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| ~5 | S | 1 | 0000 | 73 | Ra5 |  | Ct3 | Sz2 | 36 |
| Immediate32 | | | | | | | | | |

**CMP Rt,Ra,Imm64 or CMP Rt,Imm64,Ra**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | | 7 6 | 5 0 |
| ~5 | S | 1 | 0001 | 73 | Ra5 |  | Ct3 | Sz2 | 36 |
| Immediate31..0 | | | | | | | | | |
| Immediate63..32 | | | | | | | | | |

**Clock Cycles:**

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

### CSR – Control and Special Registers Operations

**Description:**

Perform an operation on a CSR. A CSR field value of zero indicates to use the next instruction word as the CSR register number.

|  |  |  |
| --- | --- | --- |
| **Operation** | **Op3** |  |
| MOVE from CSR | 0 |  |
| MOVE to CSR | 1 |  |
| Or to CSR (set bits) | 2 |  |
| And complement to CSR (clear bits) | 3 |  |
| Exclusive Or to CSR (flip bits) | 4 |  |

|  |  |  |
| --- | --- | --- |
| **Regno** |  |  |
| 0 | {ext} | Extended CSR number |
| 1 | ccr | Condition codes |
| 2 | sr | Status register (privileged) |
| 120 | Tick | Tick count (read only) |
| 121 | Coreno | Core number ( read only) (privileged) |
| 127 |  |  |

**Instruction Formats:**

**MOVE CSR, <ea>**

**OR CSR,<ea>**

**ANDC CSR,<ea>**

**EOR CSR, <ea>**

**MOVE <ea>,CSR**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 11 | 10 8 | 7 6 | 5 0 |
| Rc5 | ~ | Rb5 | 03 | CSRt7 | Op3 | Sz2 | 26 |

### MOVE <dea>,<sea>

**Description:**

Move data from source to destination.

**Supported Operand Sizes:** .b, .w, .t, .o

**Supported Address Modes:**

Source:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \* | \* | \* | \* | \* | \* | \* |

Destination:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \*1 | \*1 | \*1 | \* | \* | \* |  |

1Register indirect with three-bit displacement is not supported for destination operands.

**Instruction Format:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 10 | 9 8 | 7 6 | 5 0 |
| D5 | C | Rb5 | Ms3 | Ra5 | Md3 | Ds | Sz2 | 166 |
| Opt Src Constant31..0 | | | | | | | | |
| Opt Src Constant63..32 | | | | | | | | |
| Opt Dst Constant31..0 | | | | | | | | |
| Opt Dst Constant63..32 | | | | | | | | |

Notes:

### MOVEM – Move Multiple Registers

LOAD reglist, <sea>

STORE reglist, <dea>

**Description:**

Move data from source to destination.

**Supported Operand Sizes:** .b, .w, .t, .o

**Supported Address Modes:**

Source:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \* | \* | \* | \* | \* | \* | \* |

Destination:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \*1 | \*1 | \*1 | \* | \* | \* |  |

1Register indirect with three-bit displacement is not supported for destination operands.

**Instruction Format:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 10 | 9 8 | 7 6 | 5 0 |
| D5 | C | Rb5 | Ms3 | Cr List8 | | ~ | Sz2 | 166 |
| Reglist31..0 | | | | | | | | |
| Opt Src/Dst Constant31..0 | | | | | | | | |
| Opt Src/Dst Constant63..32 | | | | | | | | |

Notes:

### MUL - Multiplication

**Description:**

Multiply two source operands and place the sum in the target register. All registers are integer registers. Arithmetic is signed twos-complement values. The ‘S’ flag indicates to perform an unsigned multiply.

**Supported Address Modes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rn | D(Rn) | D(Rn)+ | -D(Rn) | (Rn, Rn) | D32(Rn,Rn) | D64(Rn,Rn) | Imm |
| \* | \* | \* | \* | \* | \* | \* | \* |

**Operation:**

Rt = Ra \* Rb

**Clock Cycles:**

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

**Instruction Formats:**

**MUL Rt, Ra, Rb – Register direct**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 03 | Ra5 | Rt5 | Sz2 | 66 |

**MUL Rt,Ra,disp3(Rb) – Register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 66 |

**MUL Rt,Ra,disp32(Rb) – Register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 66 |
| Displacement31..0 | | | | | | | |

**MUL Rt,Ra,disp64(Rb) – Register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 13 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**MUL Rt,Ra,disp3(Rb)+ – Post update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 46 |

**MUL Rt,Ra,disp32(Rb)+ – Post update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |

**MUL Rt,Ra,disp64(Rb)+ – Post update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 23 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**MUL Rt,Ra,+disp3(Rb) – Pre update register indirect with three-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Disp5 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 46 |

**MUL Rt,Ra,+disp32(Rb) – Pre update register indirect with thirty-two-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 165 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |

**MUL Rt,Ra,+disp64(Rb) – Pre update register indirect with sixty-four-bit displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| 175 | S | Rb5 | 33 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**MUL Rt,Ra,(Rb, Rc) - Indexed**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 43 | Ra5 | Rt5 | Sz2 | 46 |

**MUL Rt,Ra,disp32(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 53 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement32 | | | | | | | |

**MUL Rt,Ra,disp64(Rb, Rc) – Indexed with displacement**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Rc5 | S | Rb5 | 63 | Ra5 | Rt5 | Sz2 | 46 |
| Displacement31..0 | | | | | | | |
| Displacement63..32 | | | | | | | |

**MUL Rt,Ra,Imm9**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| Imm8..4 | S | 0 | Imm3..0 | 73 | Ra5 | Rt5 | Sz2 | 46 |

**MUL Rt,Ra,Imm32**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| ~5 | S | 1 | 0000 | 73 | Ra5 | Rt5 | Sz2 | 46 |
| Immediate32 | | | | | | | | |

**MUL Rt,Ra,Imm64 or MUL Rt,Imm64,Ra**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 27 | 26 | 25 | 24 21 | 20 18 | 17 13 | 12 8 | 7 6 | 5 0 |
| ~5 | S | 1 | 0001 | 73 | Ra5 | Rt5 | Sz2 | 46 |
| Immediate31..0 | | | | | | | | |
| Immediate63..32 | | | | | | | | |

**Operation:**

Rt = Ra \* Imm

**Clock Cycles:** 8

**Execution Units:** All Integer ALU’s

**Exceptions:** none

**Notes:**

### Bcc – Conditional Branch

Bcc Cn, label

**Description:**

Branch if the condition is met. The BSR instruction places the address of the next instruction on the stack. The displacement is relative to the address of the branch instruction.

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 31 14 | 13 11 | 10 6 | 5 0 |
| Disp19..2 | Cr3 | Cond5 | 286 |

|  |  |  |
| --- | --- | --- |
| **Cond5** | **Mnemonic** | **Test** |
| 0 | RA | 1 |
| 1 | SR | 0 |
| 2 | HI | !cf & !zf |
| 3 | LS | cf | zf |
| 4 | CC / HS | !cf |
| 5 | CS / LO | cf |
| 6 | NE | !zf |
| 7 | EQ | zf |
| 8 | VC | !vf |
| 9 | VS | vf |
| A | PL | !nf |
| B | MI | nf |
| C | GE | (nf & vf) | (!nf & !vf) |
| D | LT | (nf & !vf) | (!nf & vf) |
| E | GT | (nf & vf & !zf) | (!nf & !vf & zf) |
| F | LE | zf | (nf & !vf) | (!nf & vf) |
| 10 | FOR |  |
| 11 | FUN |  |
| 12 | FGT |  |
| 13 | FLE |  |
| 14 | FGE |  |
| 15 | FLT |  |
| 16 | FNE |  |
| 17 | FEQ |  |
| 18 | DFOR |  |
| 19 | DFUN |  |
| 1A | DFGT |  |
| 1B | DFLE |  |
| 1C | DFGE |  |
| 1D | DFLT |  |
| 1E | DFNE |  |
| 1F | DFEQ |  |

### DBcc – Decrement and Branch

DBcc Cn,label

**Description:**

Decrement the loop counter and branch if the condition is false and the loop counter is not equal to minus one. The displacement is relative to the address of the branch instruction.

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 31 14 | 13 11 | 10 6 | 5 0 |
| Disp19..2 | Cn3 | Cond5 | 286 |